Politecnico di Milano

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING

Master of Science – Energy Engineering

Integration of a data-driven classifier trained by adaptive sampling with a neural network for the online prediction of the cool-down time in a subsea pipeline after an unplanned shutdown

Supervisor

Prof. Enrico Zio

Co-Supervisor

Dr. Ahmed Shokry

Industrial Supervisor

Dr. Marco Montini

Candidate

Alberto Gerri – 913330

Academic Year 2019 – 2020

Ringraziamenti

Questa tesi nasce all'interno di un programma di tirocinio in Eni S.p.A..

Un sentito ringraziamento va a Marco e ai ragazzi di produzione che in questi nove mesi si sono mostrati disponibili, dei punti di riferimento e validi dispensatori di consigli nonostante la pandemia e lo smart-working.

Un ringraziamento va a Ahmed per il suo generoso e fondamentale aiuto nella realizzazione di questo lavoro.

Un sincero grazie va poi alla mia famiglia che da ventiquattro anni sempre mi supporta (e sopporta).

Vorrei ringraziare in modo speciale gli amici e gli affetti che sempre ci sono e che sempre ci saranno, per i loro incoraggiamenti di questi anni e il supporto degli ultimi mesi.

E poi i compagni del poli, di avventure norvegesi e di treno. Ringrazio Enrico, instancabile e inseparabile collega.

Acknowledgements

This thesis was born during a program of internship in Eni S.p.A..

Heartfelt thanks to Marco and to the production guys who, in these nine months, have always been available and valid advisors despite the pandemic and the smart-working.

Special thanks go to Ahmed for his kind and primary help in the realization of this work.

Then, sincere thanks go to my family: they have been supporting me and bearing with me for the last twenty-four years.

I'd like to specially thank friends and affections that are always there for encouraging me and supporting me.

And then, Politecnico's, Norwegian life's and train's fellows. I thank Enrico, tireless and inseparable colleague.

Sommario

Nell'industria dell'Oil e Gas, il crescente interesse nella produzione da riserve off-shore che si trovano in acque profonde porta a importanti sfide connesse alla sicurezza e all'affidabilità dei complessi assets sottomarini che operano in condizioni estreme. La formazione di idrati nei componenti sottomarini e nelle pipeline è una delle maggiori preoccupazioni per le squadre di ingegneri che si occupano di Flow Assurance. Un design efficace degli assets sottomarini evita la formazione di idrati in condizioni stazionarie. Durante shut-downs non programmati, invece, i fluidi si depositano e iniziano a raffreddarsi a causa dello scambio termico con l'acqua del mare, aumentando il rischio di formazione di idrati. In questi casi alcuni interventi devono essere eseguiti prima di una soglia temporale per prevenire la formazione di idrati e per mantenere l'asset in sicurezza. Questa soglia temporale è chiamata cool-down time (CDT) che è il periodo tra l'evento di shut-down e il raggiungimento delle condizioni fisiche di pressione e temperatura favorevoli alla formazione di idrati nell'asset. In questo contesto, una valutazione rapida e affidabile del CDT è di grande importanza. Quests stima è effettuata basandosi su complessi modelli fisici (basati su flussi multifase, fluidodinamica, scambio termico e di massa, ecc.) che simulano le condizioni dell'asset e del flusso. Solitamente, tali modelli richiedono tempi di calcolo lunghi che sono molto rischiosi nelle applicazioni online. Questa tesi presenta una nuova metodologia orientata all'affidabilità per lo sviluppo di modelli surrogati che sono capaci di predire in modo veloce e accurato il CDT nelle pipeline sottomarine dopo shut-downs non programmati. La metodologia è basata su due principali fasi:

1. la costruzione di una coppia di modelli surrogati, basata su Reti Neurali Artificiali, ognuna responsabile della previsione dei CDT delle operazioni di alto rischio (bassi CDT) o basso rischio (alto CDT), e un classificatore orientato ai dati che assegna il livello di rischio alle condizioni operative dell'asset.

2. considerando il fatto che il dati per l'allenamento dei modelli surrogati includono rarissime informazioni riguardo alle condizioni operative di alto rischio e ai corrispondenti CDT, una procedura ibrida di campionamento è adottata per originare dati dal sotto-dominio delle operazioni di alto rischio e, conseguentemente, per migliorare la performance del classificatore e dei modelli surrogati.

L'efficacia della metodologia proposta è validata dalla sua applicazione a due esempi matematici presi da letteratura e a un case study riguardante un modello fisico di una pipeline di un asset offshore dell'Africa Occidentale sviluppato con il software OLGA.

Parole chiave

Cool-down time, shut-down, idrati, machine learning, campionamento adattivo, affidabilità.

Abstract

In the Oil and Gas industry, the growing interest in deep water fields' production is associated with the significant challenge of the safe and reliable operation of such complex and exorbitant subsea assets, which are operated under extreme conditions and uncertainties. Hydrates formation in subsea equipment and pipelines is one of the main reliability concerns for flow assurance teams and engineers. An efficient design of the subsea assets guarantees the avoidance of hydrate formation in regular steady-state operating conditions. But, during unplanned shut-downs, the fluids settle and cool down due to the heat transfer with the sea water, which increases the hydrate formations chances. In these cases, some interventions must to be performed before a specific threshold of time, in order to prevent the hydrate formation and keep the asset safe. This time threshold is called the cool-down time (CDT), which is the period between the shut-down event and the achievement of the pressure and temperature conditions favourable for hydrate formation in the pipeline. In this context, a fast and reliable assessment of CDT is of significant importance. This task is performed relying on complex physical models (based on multi-phase flow, fluid dynamics, heat and mass transfer, etc.) that simulate the asset and flow conditions. Such models usually demand long computational time, which could be very risky in online production environments. This thesis presents a novel reliability-directed methodology for the development of surrogate models that are able to predict, in a fact and accurate way, the CDT in subsea pipelines after unplanned shutdowns. The methodology is based on two main stages:

1. building a couple of surrogate models, based on Artificial Neural Networks (ANN), each responsible of predicting the CDT at high risk (low CDT) or low risk (high CDT) operating conditions, and a data-driven classifier that assign the risk level (high or low) to the operating condition of the asset.

2. driven by the fact that the training data include very rare information about high risk operating conditions and the corresponding CDT, a hydrate sequential sampling procedure is adapted to collect training data from the high-risk operating condition subdomain and consequently to improve the performance of the classifier and the surrogate models.

The effectiveness of the proposed methodology is validated by its application to two mathematical examples from the engineering reliability literature and to a case study involving a physics-based black box model developed by the OLGA software of a pipeline of an offshore Western African Asset.

Keywords

Cool-down time, unplanned shut-down, hydrates, machine learning, adaptive sampling, reliability.

Extended Abstract

Introduction

The oil and gas sector is becoming more and more reliant on offshore industry, in particular on the production from deep water fields [1]. These fields, located in offshore areas with water depth higher than 200 meters, present new challenges related to the complexity of the subsea assets (the one totally under the surface and usually located at the bottom of the ocean) and to the adverse natural environment.

Hydrates can be considered one of the worst flow assurance concerns. Hydrates are crystalline solids that can form at some pressure and temperature (P-T) conditions if water and light gas molecules are present. Hydrates formation is something really unwanted because it can lead to pipeline blockages with strong consequences concerning safety, damages, difficult remediation operations, incomes and increasing operative expenditure [2] [3].

For steady state operations, the pipelines are designed to avoid hydrates formation by prevention techniques. Usually for oil fields pipeline, thermal insulation is enough, but water removal and inhibitor injection (chemical substances which usually shift the P-T hydrates formation conditions) can also take place. However, during shutdowns the fluid phases separate, settle and cool down. If no prior inhibitor injection is accomplished, as in the case of unplanned shutdowns, this leads to hydrates formation whether no posterior actions is taken [4].

The operator during shut-downs accomplish a series of procedures described by the operating philosophy for such an asset. Usually, three main time frameworks are followed: the No-Touch Time (NTT), the Light Touch Time (LTT) and the Circulation (CIR). It's clear that the time spent for carrying out the operations must be lower than the cool-down time (CDT) which is the time for the fluid to achieve favourable pressure and temperature conditions that lead to hydrates formations. Usually the assets are designed in order to allow to perform all the necessary operations to preserve the line before the CDT if any unplanned shut-down happens [5].

Even if the operating philosophy guarantees a certain CDT, an accurate computation of CDT in real time can be very beneficial. In fact, the operator can extend the time

frames for securing the asset if the predicted CDT is higher enough or, on the contrary, he can inject inhibitors during online operations in order to preserve the asset in case of shutdowns if CDT predictions are lower than a certain value. Therefore, the CDT can be considered a reliability index indicating a threshold value before which the operations have to be carried out for securing the asset. Given that, low CDT values can be seen as high-risk (i.e., the operator may have insufficient time to perform the asset preservation sequence), while high CDT are indicated as low-risk (i.e., the operator will have insufficient time to perform the sequence). The development of a reliable, accurate and fast online tool for CDT's prediction is seen as of great importance.

The industry standard tool for transient simulation of multi-phase petroleum production is OLGA software. OLGA is a one dimensional code simulator [6] developed to simulate multi-phase flow in pipelines and pipelines networks, with processing equipment included. In particular, OLGA is a three-fluid (oil, gas and water) model, based on seven conservation equations and one equation of state to be solved using the finite volume method and semi-implicit time integration [7]. A reliable CDT measure can be computed by launching simulations with OLGA and by post-processing the outputs. However, since the high computational times and the usually few licenses owned by the companies, OLGA is not suitable for online predictions.

The development of proxy models based on classical Machine Learning (ML) algorithms is proven to be not sufficient for the reliable prediction of CDT for high risk values. Moreover, the classical static techniques of Design of Experiment, such as Latin Hypercube Sampling, fill the input space homogeneously without focusing on the region of the domain which leads to high-risk values of CDT.

The work presents a novel reliability-directed methodology for the development of fast and accurate surrogate models for CDT prediction after unplanned shut-downs. Dividing the total domain of the asset operating conditions into high risk and lo risk sub-domains, the proposed methodology is based on two main stages:

- building a composite model based on a Support Vector Machine (SVM) risk classifier and two Artificial Neural Network (ANN) regression sub-models where the classifier labels the points based on their level of risk (high or low), while the regressors are used for the predictions of the CDT values, given their level of risk.

- enhancing the composite model by a novel adaptive sampling methodology based on coupling two different adaptive sampling techniques: the SVM-based and the GPR-based. This is done in order to improve the performance of the SVM classifier and to add points in the high-risk region, hence enhancing the performance of the high-risk regressor.

The proposed methodology is then proven and applied to two mathematical functions

and to a case study involving the physics-based model for a pipeline of an asset located offshore a West African country and developed by a joint venture with Eni S.p.A as leading operator.

Problem statement

The OLGA physics-based model is considered as a black box model f which correlates the m input variables $\mathbf{x} \in \mathbb{R}^m$ to an output $y \in \mathbb{R}$. The main drawback of a physics-based model like f is related to the high computational time. The objective is the creation of fast and reliable surrogates model that can be substitute to f for real time prediction of the output y starting from the same m input variables $\mathbf{x}$.

In literature the classical surrogate models usually do not regard the levels of change of the output variable y and they give the same importance to all the value y. Moreover they are usually based on classical Design of Computer Experiment (DOCE) techniques which are aimed at the homogeneously filling of the space.

In this work the output y (e.g. the CDT for physics-based model) is considered a reliability index. Its value can be related to a value of risk: high-risk or low-risk. Hence, two different risk-regions are drawn: the high-risk region Θ_x^{HR} and the low-risk region Θ_x^{LR}. Considering a threshold value $\bar{y}$, if $y \leq \bar{y}$ hence $y \in \Theta_x^{HR}$. On the contrary, if $y > \bar{y}$ hence $y \in \Theta_x^{LR}$.

Even if the importance of a good prediction over the global domain is seen of great importance, an accurate prediction for high-risk y has a major importance.

The surrogate system searched should care about three main concerns:

- it should implement an alarm system $RL = C(x)$ that, based on given input conditions x, it should be able to assign the level of risk (RL)
- it should reliably predict the CDT. This should be done with two different predictors based on the region of belonging of the given input conditions: $\hat{y}^{LR} = P^{-}(x)\,,\ x \in \Theta_x^{LR}$ and $\hat{y}^{HR} = P^{+}(x)\,,\ x \in \Theta_x^{HR}$. An ensemble of this type should provide better accuracy than classic global surrogate models.
- it should embrace an adaptive sampling technique, since the classical static DOCE techniques doesn't allow a good focus on the high-risk region leading to the collection of the majority of the data in the low-risk area. Indeed, the adaptive sampling technique should collect data for enhancing both the alarm system and the ensemble of the regressors.

Methodology

The methodology proposed follows two main stages. The first one consists in the development of an ensemble of Machine Learning algorithms driven by the reliability nature of CDT. The second one deals with the necessity to improve the information that the considered domain carries. This is accomplished by a novel hybrid adaptive sampling methodology.

The composite model

The first stage of the methodology deals with the coupling of two different Machine Learning algorithms: SVM and ANN. The former is used for the development of a risk classifier, the latter for the development of two regression sub-models for CDT prediction.

First of all, an input domain is generated by Latin Hypercube Sampling (LHS) technique. After that, the output for each point is collected and a dataset is generated. From this, a 10% of the data is saved as testing set, while the 90% is used for the training session. Moreover, a k-folds cross-validation is adopted and a 15% of the points for the training session are used as validation set.

The SVM classifier is needed for labelling each point with its predicted risk level (high-risk/low-risk) and it is trained with the totality of the training set. Given $C(\mathbf{x})$ the classifier and given a new point $\mathbf{x_i}$, if $C(\mathbf{x_i}) = 1$ the point is predicted to be a high-risk point, hence to belong to the high risk region Θ_x^{HR}. On the contrary, $C(\mathbf{x_i}) = 0$ labels $\mathbf{x_i}$ as low-risk point, so as a point belonging to the safe region Θ_x^{LR}.

The two ANN regression sub-models are instead trained by tailored training sets: points belonging to Θ^+ train the ANN, here indicated as $P^+(\mathbf{x})$, for low-risk region while on the contrary, points belonging to Θ^- train the ANN, $P^-(\mathbf{x})$, for the high risk region.

For a reliable prediction of the CDT for a given condition, the composite model first uses the classifier for assessing the risk-level, then it runs the corresponding regressor (for high or low risk). Finally, the CDT value is predicted.

Hybrid adaptive sampling methodology

The double goal of improving the performance of the risk classifier and of getting more information about the risk region is addressed by a novel Hybrid adaptive sampling methodology which couples both a proposed SVM-based and a proposed Gaussian Process Regression (GPR)-based adaptive sampling methods. The adaptive sampling method iteratively add points in the domain in order to enhance the surrogate models' performance.

The SVM-based adaptive sampling is based on the exploitation of the "score" values. These values represent the distance of a given point, to the high-low risk boundary

detected by the SVM model. The methodology has the main scope to improve the classifier performance. A detailed algorithm follows.

1. Train the SVM classifiers by k-folds cross validation.
2. For each SVM model:
   (a) Test 5k points given by the LHS method.
   (b) Assess the score of each point.
   (c) Select the points with $S_{min} < \ score \ < S_{max}$
   (d) Apply K-means clustering selecting K_1 clusters.
   (e) Choose as candidate points the closest K_1 points to the K_1 centroids (in order to not get points close each other).
3. Merge the candidate points in only one set.
4. Apply for the second time K-means clustering selecting K_2 clusters.
5. Choose as final candidate points the K_2 points closest to the K_2 centroids.
6. Simulate the final K_2 candidate points by physics model or by mathematic function.
7. Add the final candidates points with their output values in the training set.

The GPR-based adaptive sampling is based on the exploitation of both the mean prediction and the standard deviation assessed by a GPR model trained on the same training set of the composite model. In particular a U function is used in the selection of the candidate points:

$$U = \frac{\hat{y} - h}{\hat{\sigma}} \tag{1}$$

where $\hat{y}$ is the predicted output value, $\hat{\sigma}$ is the predicted standard deviation while h is a threshold parameter used to guide the exploration of the high risk region. The following algorithm shows the details of the GPR-based adaptive sampling methodology.

1. Train the k GPR models by k-folds cross validation.
2. Choose a set of positive h values by Monte Carlo Sampling based on a normal distribution which covers all the risk subdomain.
3. For each GPR model:
   (a) Test 5k points given by the LHS method.
   (b) Assess the U of each point.

(c) Select the points with $U_{min} < U < U_{max}$

(d) Apply K-means clustering selecting K_3 clusters.

(e) Choose as candidate points the closest K_3 points to the K_3 centroids.

4. Merge the candidate points in only one set.
5. Apply for the second time K-means clustering selecting K_4 clusters.
6. Choose as final candidate points the K_4 points closest to the K_4 centroids.
7. Simulate the final K_4 candidate points by physics model or by mathematic function
8. Add the final candidates points with their output values in the training set.

Application

The proposed methodology is validated first of all by its application to two mathematical functions taken from reliability analysis literature. Then the methodology is applied to a case study which considers a pipeline of an offshore Western African production asset. A comparison among both a global ANN and a composite model trained by both LHS and hybrid adaptive sampling methodology is carried out.

Mathematical functions

The mathematical functions are both defined in a domain given by two variables, $x_1, x_2 \in \mathbb{R}$ while the high-risk area is considered the one whose outcomes y are lower than 0. For some boundary ranges, $x_1, x_2 \in [-5; 5]$, it follows that the high-risk area is small compared to the one of low-risk leading to an imbalance distribution of outcomes y.

Function 1 is defined by

$$g\left(x\right) = q - \frac{1}{20}\left(x_1^2 + 4\right)\left(x_2 - 1\right) + sin\left(\frac{5}{2}x_1\right) \tag{2}$$

With $q = 2$.

Function 2 is given by a translated series system. The series system is given by:

$$p(x) = \min \begin{cases} p_1(x) = k + 0.1\left(x_1 - x_2\right)^2 - \frac{x_1 + x_2}{\sqrt{2}} \\ p_2(x) = k + 0.1\left(x_1 - x_2\right)^2 + \frac{x_1 + x_2}{\sqrt{2}} \\ p_3(x) = \left(x_1 - x_2\right) + \frac{m}{\sqrt{2}} \\ p_4(x) = \left(x_2 - x_1\right) + \frac{m}{\sqrt{2}} \end{cases} \tag{3}$$

where the parameters are the same as in [8]: $m = 6$, $k = 3$ with $x_1, x_2 \in [-5, 5]$. The considered function $g(x)$ is a translation of the series system:

$$g\left(x\right) = p\left(x\right) + 3 \tag{4}$$

Table 1. Function 1. Performance indices for both LHS and Hybrid Adaptive Sampling Methodology for both global ANN and Composite model.

	LHS		Hybrid adap. Sampling	
	Global ANN	Composite	Global ANN	Composite
	NRMSE, [%]			
Global performance	4.59	4.70	4.86	4.90
Low-risk performance	4.31	4.37	5.00	5.11
High-risk performance	6.45	6.82	3.51	2.32
	z, [-]			
classifier	-	0.87	-	0.90

Table 2. Function 2. Performance indices for both LHS and Hybrid Adaptive Sampling Methodology for both global ANN and Composite model.

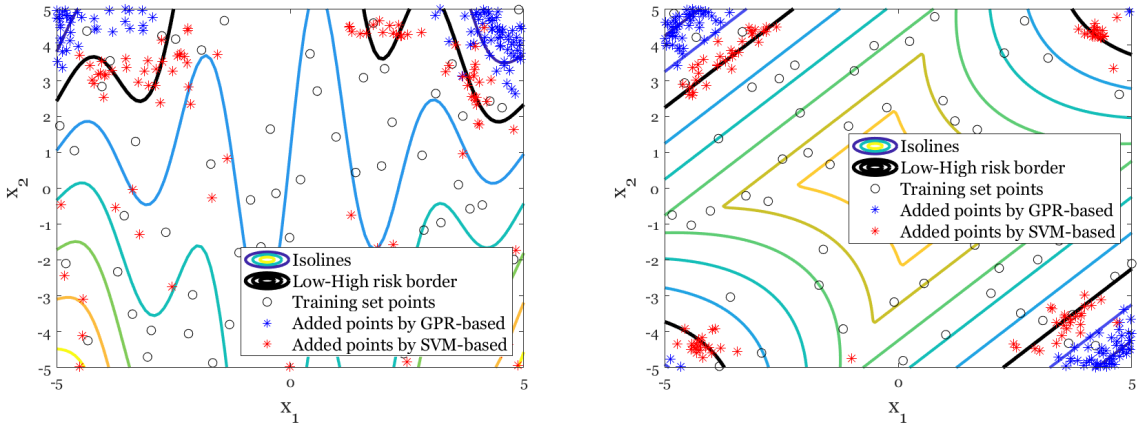
	LHS		Hybrid adap. Sampling	
	Global ANN	Composite	Global ANN	Composite
	NRMSE, [%]			
Global performance	1.82	2.44	2.80	3.82
Low-risk performance	1.83	2.32	2.93	4.00
High-risk performance	1.69	3.38	0.79	0.75
	z, [-]			
classifier	-	0.85	-	0.94

Applying the hybrid adaptive sampling methodology, the main outcomes are shown in tables 1 and 2 while the distribution of the added points, highlighting both the contributions of SVM-based approach and GPR-based approach, are shown in figures 1a and 1b.

It's possible to state that the proposed methodology works good both in adding points in the target areas (figures 1a and 1b) and in enhancing the performance both for the classifier and for the high-risk regressor (tables 1 and 2): the two mathematical functions validate the hybrid adaptive sampling methodology.

Case study

The reference asset is part of the producing field block located offshore a West African country and developed by a joint venture with Eni S.p.A as leading operator. For the purpose of the study, a part of this asset is modelled in OLGA. In particular, the pipeline carrying fluids from a subsea manifold located at a depth of about 1200 meters towards a Floating Production Storage and Offloading unit (FPSO) located about 15 km far from



(a) (b)

Figure 1. Function 1 (left) & 2 (right). Distribution of initial and final training set applying the hybrid adaptive sampling methodology: contribution of SVM-based (left) and GPR-based (right) methodologies.

the manifold is considered.

Five variables - separator pressure (Pout), oil flow rate at standard condition (Qoil), Gas-Oil ratio at standard condition (GOR), water cut (WC), and temperature of the fluid at the manifold (Tin) - and their operational ranges are chosen to be the input variables for the surrogate model.

The initial dataset is given by 400 input points created by LHS method which are then simulated by OLGA and post-processed in order to get the CDT values. Both a composite and a global ANN are trained by both LHS method and hybrid adaptive sampling. Table 3 shows the main outcomes. Figure 2 compares the distributions of the added points by LHS method and by hybrid methodology (for the adaptive sampling methodology the points are mainly added in the high-risk area and in the nearness of the risk boundary as expected).

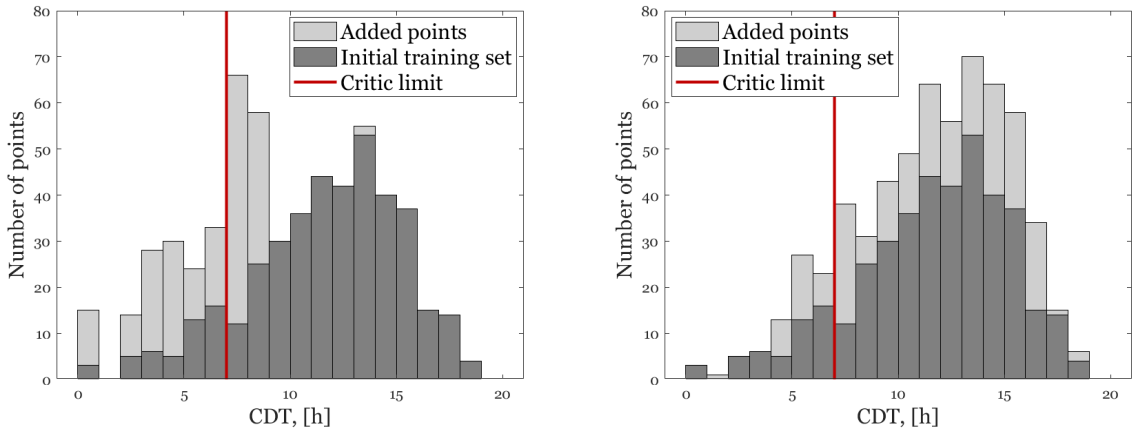
It's possible to infer that, also for the case study, the best option for high-risk performance is given by the composite model trained by the hybrid adaptive sampling methodology.

Further Analysis

Further analysis are conducted on the physics-based model. In particular, it is shown the goodness of the composite model against a global ANN or GPR for different training set sizes and for different critic CDT. Moreover, an analysis about the computational time for training sessions and predictions is carried out. The main outcome shows the goodness of the surrogate models for their online application.

Table 3. Physics model. Performance indices for both LHS and Hybrid Adaptive Sampling Methodology for both global ANN and Composite model.

	LHS		Hybrid adap. Sampling	
	Global ANN	Composite	Global ANN	Composite
	NRMSE, [%]			
Global performance	3.50	4.14	3.56	3.64
Low-risk performance	3.05	4.04	3.31	3.59
High-risk performance	5.83	4.79	5.00	4.01
	z, [-]			
classifier	-	0.86	-	0.92



(a) (b)

Figure 2. Physics model. Distribution of initial and final training set applying the hybrid adaptive sampling methodology (on the left) and applying the LHS (on the right)

Conclusion

The current work proposes a novel methodology for the development of surrogate models capable of the online prediction, in a fast and accurate way, of the CDT for subsea pipeline after an unplanned shutdown. In particular a composite model based on a risk-oriented SVM classifier that classifies the asset conditions after the shutdown into low or high risk levels, and two ANNs, each of which is responsible for the CDT prediction at low risk or high-risk subdomain of the asset conditions that are previously assigned by the SVM classifier.

In order to overcome the problem of the rarity of the available input-output (asset conditions-CDT) training data from the high-risk region, an adaptive sampling procedure is proposed to collect more training samples from this region. This novel procedure combines two adaptive sampling techniques, which are a SVM-based and a GPR-based. The hybrid procedure leads to an exploration of space target to the high-risk region and to the high-low risk boundary in order to improve both the high-risk regressor and classifier performance.

The methodology is validated by two mathematical functions and is applied to a case study which considers a pipeline of an Offshore Western African production Asset. The outcomes in applying the hybrid adaptive sampling methodology show an improvement in the performance for high risk region and for the classifier in all the applications. Moreover, the benefits concerning high-risk region of using the composite model instead of global ANN models is proven for the physics-based model. Finally the main outcome from the evaluation of the computational time expenses shows that online prediction of CDT by the composite model is hugely less computational demanding than the prediction made by OLGA simulator, hence validating the possibility to use the surrogate models as online tools.

Future work lines include the uncertainty quantification of the surrogate model predictions, the sensitivity analyses in order to quantify the relative importance of the input variables with respect to the CDT and the investigation of effect of considering a multiclass risk classifier (e.g., high, medium, low risk) in the hybrid surrogate model structure.

Contents

Ringraziamenti **iii**

Acknowledgements **v**

Sommario **vii**

Abstract **ix**

Extended Abstract **xi**

Contents **xxiv**

List of Figures **xxvi**

List of Tables **xxvii**

1 Introduction **1**

2 Problem Statement **5**

3 Methodology **9**

3.1 The composite model 9

3.1.1 Dataset Generation 9

3.1.2 SVM classifier 9

3.1.3 ANN regressors 10

3.1.4 Composite model's prediction 11

3.2 Hybrid adaptive sampling methodology 11

3.2.1 SVM-based adaptive sampling methodology 12

3.2.2 GPR-based adaptive sampling methodology 13

3.2.3 Hybrid adaptive sampling methodology and stopping criteria . . . 14

4 Application **17**

4.1 Mathematical functions 17

4.1.1 Function 1 18

4.1.2 Function 2 . . . 22
4.2 Case Study . . . 26

5 Further Analysis **35**
5.1 Training set size effect . . . 35
5.2 Imbalance ratio effect . . . 37
5.3 GPR model for regression prediction. Comparison with ANN . . . 39
5.4 Computational time expenditure . . . 41
5.4.1 Computational time expenditure for Online prediction . . . 41
5.4.2 Computational time expenditure for offline development . . . 41

6 Conclusion **43**

A Oil & Gas Production Assets **45**
A.1 Reservoir, fluids and wellbores . . . 45
A.2 Production platforms . . . 47

B Hydrates **49**
B.1 Flow assurance . . . 49
B.2 Hydrates in nature . . . 50
B.3 Hydrates Formation . . . 51
B.4 Working principle of Hydrates inhibitors . . . 52
B.5 Hydrate prevention . . . 53
B.6 Hydrate Remediation . . . 54
B.7 Hydrate formation during shut-down's situations . . . 54

C OLGA simulator **57**
C.1 Fluid models . . . 59

D OLGA model specifications **61**
D.1 Modelling of the contextual pipeline-riser geometry . . . 61
D.2 Setting the fluid model . . . 62
D.3 Boundary conditions . . . 63
D.4 Simulation options . . . 65

E Artificial Intelligence & Machine Learning **67**
E.1 Artificial intelligence . . . 67
E.2 Machine Learning . . . 68

F Artificial Neural Networks **71**
F.1 Similarities with biological neural network . . . 72

F.1.1 Single-input neuron . . . 72
F.2 Transfer function . . . 73
F.3 Multiple-input neuron . . . 73
F.4 Classic neural network architecture . . . 74
F.4.1 Single layer of neurons . . . 74
F.4.2 Multiple layers of neurons . . . 74
F.5 Learning algorithms . . . 75
F.5.1 Optimization Algorithms . . . 75
F.5.2 First application in artificial neural networks . . . 77
F.5.3 Backpropagation . . . 78
F.5.4 Variation on backpropagation . . . 81
F.6 Generalization of an artificial neural network . . . 82
F.6.1 Early Stopping . . . 82

G Gaussian Process Regression **83**
G.1 Multivariate Gaussians . . . 83
G.2 Bayesian linear regression . . . 85
G.3 Gaussian process . . . 86
G.4 Gaussian process regression . . . 87
G.5 Parameters' estimation . . . 89

H Support Vector Machines **91**
H.1 Classification algorithms . . . 91
H.2 Support Vector Machines . . . 92
H.2.1 Theory . . . 92
H.2.2 Class imbalance . . . 95
H.2.3 Cost-sensitive SVM . . . 95
H.2.4 Performance Parameters . . . 96

I K-means Clustering **99**
I.1 Cluster Analysis . . . 99
I.2 K-means Clustering . . . 100
I.2.1 K-means Clustering Algorithm . . . 101
I.2.2 Proximity measure and objective functions . . . 101
I.2.3 Choosing initial Centroids . . . 102
I.2.4 Strengths and weaknesses . . . 103

J Design of Experiment & Cross-Validation **105**
J.1 Design of Experiment . . . 105
J.2 Cross Validation . . . 106

Acronyms **109**

Bibliography **111**

List of Figures

Figure 1 Function 1 (left) & 2 (right). Distribution of initial and final training set applying the hybrid adaptive sampling methodology: contribution of SVM-based (left) and GPR-based (right) methodologies. xviii
Figure 2 Physics model. Distribution of initial and final training set applying the hybrid adaptive sampling methodology (on the left) and applying the LHS (on the right) xix

Figure 2.1 Schematics representation of the input variables domain, where the red points are training data. 6

Figure 3.1 Flow chart of the composite model 11
Figure 3.2 Flowchart for Hybrid Adaptive Sampling Methodology. The left side refers to the SVM-based, the right side refers to GPR-based adaptive sampling technique. 15

Figure 4.1 Function 1. Hybrid adaptive sampling methodology: NRMSE, $F_{measure}$, G_{mean} and z improvement. 19
Figure 4.2 Function 1. Distribution of initial and final training set applying the hybrid adaptive sampling methodology (left) and LHS method (right). 20
Figure 4.3 Function 1. Distribution of initial and final training set applying the hybrid adaptive sampling methodology: contribution of SVM-based (left) and GPR-based (right) methodologies. 20
Figure 4.4 Function 1. Spatial distribution of points. 21
Figure 4.5 Function 2. Hybrid adaptive sampling methodology: NRMSE, $F_{measure}$, G_{mean} and z improvement. 23
Figure 4.6 Function 2. Distribution of initial and final training set applying the hybrid adaptive sampling methodology (left) and LHS method (right). 24
Figure 4.7 Function 2. Distribution of initial and final training set applying the hybrid adaptive sampling methodology: contribution of SVM-based (left) and GPR-based (right) methodologies. 24
Figure 4.8 Function 2. Spatial distribution of points. 25
Figure 4.9 The hydrate curve used in this work. 28

Figure 4.10 Training set for physics-based model: scatter plots for the input variables. Blu points are low-risk points. Red points are high-risk points. 28
Figure 4.11 Distribution of h threshold parameter for physics model, given by the sampling of 1000 points. 29
Figure 4.12 Physics model. Hybrid adaptive sampling methodology: NRMSE, $F_{measure}$, G_{mean} and z improvement. 30
Figure 4.13 Physics model. Distribution of initial and final training set applying the hybrid adaptive sampling methodology (on the left) and applying the LHS (on the right) 31
Figure 4.14 Physics model. Distribution of initial and final training set applying the hybrid adaptive sampling methodology: contribution of SVM-based (left) and GPR-based (right) methodologies. 31
Figure 4.15 Physics model. Spatial distribution of points. 32

Figure 5.1 NRMSE of ANN model vs training set size. 36
Figure 5.2 NRMSE of both composite and ANN model for global (left) and high-risk (right) region vs training size 36
Figure 5.3 Testing performance indices for classifier vs training set size. . . . 37
Figure 5.4 Global NRMSE of both composite and ANN model vs critic CDT (left) and imbalance ratio (right). 38
Figure 5.5 Testing performance indices for classifier vs critic CDT (left) and imbalance ratio (right). 38
Figure 5.6 NRMSE of both composite and ANN model for high-risk region vs critic CDT (left) and imbalance ratio (right). 39
Figure 5.7 ANN and GPR models performance for global and high-risk region vs trainingset size. 40
Figure 5.8 ANN, GPR and composite models performance for global and high-risk region during hybrid adaptive sampling vs trainingset size. . . . 40

Figure D.1 Pipeline Geometry 62
Figure D.2 OLGA model. In red the main variables to be specified for closing the equations' system and run the simulator. 63
Figure D.3 Water temperature vs water depth. 64

Figure F.1 Single input neuron [49] 73

List of Tables

Table 1 Function 1. Performance indices for both LHS and Hybrid Adaptive Sampling Methodology for both global ANN and Composite model. . . . xvii
Table 2 Function 2. Performance indices for both LHS and Hybrid Adaptive Sampling Methodology for both global ANN and Composite model. . . . xvii
Table 3 Physics model. Performance indices for both LHS and Hybrid Adaptive Sampling Methodology for both global ANN and Composite model. xix

Table 4.1 Function 1. Performance indices for both LHS and Hybrid Adaptive Sampling Methodology for both global ANN and Composite model. . . . 19
Table 4.2 Function 2. Performance indices for both LHS and Hybrid Adaptive Sampling Methodology for both global ANN and Composite model. . . . 23
Table 4.3 27
Table 4.4 Physics model. Performance indices for both LHS and Hybrid Adaptive Sampling Methodology for both global ANN and Composite model. 30

Table 5.1 Computational time expenditure for offline development and online prediction. 42

Chapter 1

Introduction

In oil and gas industry, the offshore sector is becoming increasingly reliant on the production from deep water fields (fields located in areas with a water depth greater than 200 metres) [1]. The reliable and safe operation of such deep-water production facilities represents a huge challenge, both due to the extreme and uncertain natural environment and to the high complexity of the production assets and equipment themselves (see appendix A).

Among many risk and safety concerns that must be considered and handled during the operation of offshore production facilities, hydrates formation problem is on the top of the list [9]. Hydrates are solid substance that usually forms in multi-phase flowlines and, in specially, in subsea assets (the ones fully submerged and placed at the bottom of the sea). Gas hydrates are solid crystalline with physical properties similar to those of ice and form at the presence of gas and water under combinations of certain pressure and temperature (P-T) values (usually high pressure and low temperature). Once the hydrates are formed in the pipeline, this could lead to major unwanted consequences on the flow transmission process. A small amount of hydrates can reduce the flow area, increasing the pressure difference and hence accelerating further hydrate formation [2]. High quantity of hydrates in the pipeline can lead to the plugging of valves and/or equipment, which can result in more serious damage on the entire pipeline and asset, bringing revenue losses and OPerative EXpenditure (OPEX) up to millions of dollars [3] (see Appendix B).

Usually, subsea assets are originally designed to minimize the chances of hydrate formation during regular steady-state operating conditions, for example, by the insulation of the pipelines to reduce the heat transfer with the subsea environment. However, this is not guaranteed during transient situations, such as pipeline shutdowns. When shutdowns happen, the gas and liquid phases inside the pipeline separate and start to cool down due to the colder subsea environment, which exposes the pipeline to the risk of hydrate formation [4]. A very common Flow Assurance practice for avoiding hydrates formation when thermal insulation is not enough, consists in injecting chemical substances, called inhibitors, in the flow. The most common are the thermodynamic inhibitors which act

by shifting the physical conditions (pressure and temperature) necessary for hydrates formation. Hence, if no inhibitor injection is performed before the shutdown event, as in the case of emergency or unplanned shutdowns, the operator must act against time in order to prevent hydrate formation and secure the asset. This is accomplished by following the instructions given by what is called the "operating philosophy" [5].

The operating philosophy is a series of instructions to be performed with an adequate time frame in order to preserve the line from hydrate formation. The total time frame is composed by three subsequent intervals, including the No-Touch Time (NTT), the Light Touch Time (LTT) and the CIRculation (CIR). In the same time, this time frame is limited by an upper bound called the CDT, which is, generally, defined as the period between an unplanned shutdown and the achievement of the pressure and temperature conditions favourable for hydrate formation in the pipeline. Usually the NTT is the first interval of time after an emergency shutdown event during which the operator should performs the tasks related to the recovery of the subsea controls by re-pressurization of the umbilical lines - if needed. Moreover, during the NTT, the operator should realize how much time is needed for asset re-start-up. Based on this, he decides whether to go through the entire preservation sequence (i.e., the entire instructions of the operational policy) for a long-term shutdown or to restart the production of the field with the adequate procedure. Such a decision should be taken as fast as possible. After the NTT, the following temporal sequence is the LTT, during which the operator should secure some critic elements of the subsea asset (such as Christmas trees subsea manifolds, well jumpers, etc.)) by performing flushing with an inhibitor. After the LTT, an operation, known as CIRculation (CIR) is carried, in which all the fluids inside the pipeline have to be displaced. Then, the so called dead oil, which is a production oil already stabilized by water removal, is injected inside the pipeline through the service lines. This aims at the total displacement of risky fluids from the pipeline. At this point, the safety of the whole asset is guaranteed.

In this context, a fast and reliable identification of the CDT is of significant importance. Since depending on the CDT value, the operating philosophy must be adapted. For example, if the CDT is long, the NTT could be relaxed, LTT could be postponed or, on the contrary, if the CDT is short, inhibitor injection can be anticipate avoiding pipeline blockages and worthless operative expenditure. Hence, the development of a reliable, accurate and fast online tool for CDT's prediction is seen as of great importance.

Therefore, the CDT can be considered a reliability index indicating a threshold value before which the operations have to be carried out for securing the asset. Given that, low CDT values can be seen as high-risk (i.e., the operator may have insufficient time to perform the asset preservation sequence) while high CDT are indicated as low-risk (i.e., the operator will have insufficient time to perform the sequence).

In the oil and gas industry literature, most of the CDT studies are usually accomplished

relying on complex physics-based models that simulate the asset thermal behaviour and flow conditions [10] [11] [12]. For example, the OLGA software is one of the most common tool for building simulation models for transient situations involving multiphase flow in the petroleum industry (see appendix C). Such complex physics-based models demand very long computational time to converge, because they involve very intricate and highly nonlinear principles (e.g., multiphase flow, fluid dynamics, and heat and mass transfer) and are solved by time consuming numerical tools such as finite element and finite volume methods. This could be very risky in such online production environment, where information must be obtained and decisions must be set in fractions of second.

Recently, in almost all the engineering fields, surrogate models (also called proxy models or metamodels) techniques are used for the complexity reduction of high-fidelity physics-based models. The main idea is to use input-output data, generated from the complex physics-based model simulation, to build data-driven model able to predict the outputs in accurate but much faster way. Generally, in the oil and gas industry literature, the applications and development of surrogate modelling techniques are concentrated in offline studies/area, e.g., product/process/system design optimization[13][14], reliability and risk analysis[15], sensitivity analysis[16], etc. While their applications and development in real-time or online studies (e.g., process operation, monitoring) are few, especially in the area of flow assurance in deep water assets.

This work presents a novel reliability-directed methodology for the development of fast and accurate surrogate models for CDT prediction after unplanned shut-downs. The main novelties of the work are: 1) mainly, given that the output of the targeted surrogate model can be considered as a reliability/risk index of the asset, the proposed surrogate modelling methodology is designed on reliability and safety basis, and 2) secondary, the application itself.

The main idea is that the proposed method divides the total domain of the asset operating conditions (i.e., the proxy model inputs) into high risk subdomain (leading to low CDT) and another low risk subdomain (resulting in high CDT). Hence, a proposed methodology is based on two main stages:

- building a hybrid model composed by a data-driven Support Vector Machine (SVM) classifier, which is trained to classify the asset conditions into low or high-risk levels, coupled to two Artificial Neural Networks (ANNs), where each ANN is responsible for the CDT predictions at low risk or high risk subdomain of the asset conditions.

- enhancing the hybrid model (SVM and two ANNs) performance, through a novel hybrid adaptive sampling procedure, which is based on the combined use of two different adaptive sampling techniques: the SVM-based and the Gaussian Process Regression (GPR)-based. While the first acts on improving the performance of the SVM classifier by collecting training points from the nearness of the high risk-low

risk classification hyperplane or boundary, the GPR-based adaptive sampling works by collecting training points from the high-risk subdomains.

The goodness of the proposed methodology is assessed by its application to two mathematical examples adapted from the engineering reliability literature, and to a case study involving a physics-based black box model developed by OLGA software of a pipeline of an offshore Western African asset.

The rest of the work is organized as follows. Chapter 2 formulates the considered problem of reliability-driven surrogate modelling. Chapter 3 describes the proposed methodology. Chapter 4 validates the methodology through its applications to two mathematical functions from reliability analysis literature and to a case study involving a complex model for subsea asset developed by OLGA. Chapter 5 deals with further analysis. Finally, chapter 6 draws the final conclusions.

Chapter 2

Problem Statement

We consider a black box, complex physics-based model f that correlates the input variables $\mathbf{x} \in \mathbb{R}^m$ (e.g., the subsea asset conditions: temperature at the manifold T_{in}, pressure at the Floating Production Storage and Offloading unit (FPSO) P_{out}, oil flow rate Q_{oil}, water cut WC, gas-oil ratio GOR) and the output variable $y \in \mathbb{R}$ (e.g., the CDT), such that:

$$y = f(\mathbf{x}) \tag{2.1}$$

The model f is computationally demanding and this hinders its usage for the online prediction of the output variables y whose real-time value is essential for supporting the operational decisions required to keep the system operating in a safe a reliable way. Therefore, the main objective of this work is the development of a surrogate model(s) that approximate(s) the mapping between $\mathbf{x}$ and y relying on input-output data generated from the simulation of the complex model f. The surrogate model(s) is/are aimed at providing very accurate estimates of $\hat{y}$ and, more importantly, in a much faster way with respect to the physics-based model, so it can be used for real-time prediction, thus supporting the reliable and safe operation of the system.

In the literature, classical surrogate modelling techniques typically:

- do not regard the levels of change of the output variable y (e.g., high, medium, low, etc.),
- gives the same importance to all the values of y, therefore, they fit one global surrogate model $\hat{y} = F(x)$ that approximates the behaviour of y over the total domain Θ_x of the input variables $\mathbf{x}$, and
- use classical Design of Computer Experiment (DOCE) techniques for the training data generation, which sample over the entire input domain Θ_x with the same importance, in order to generate a very uniform sampling plan of input data see Figure 2.1.

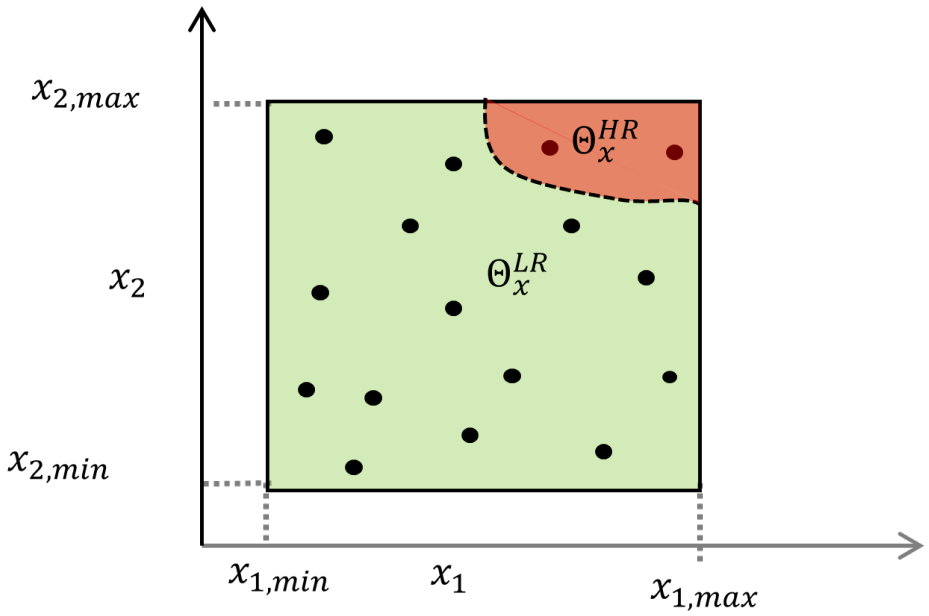


Figure 2.1. Schematics representation of the input variables domain, where the red points are training data.

In our work, we consider the output y as an operational/risk index for the entire system, with a threshold value $\bar{y}$ that classifies the risk level, such that $y^{HR}: \ y \leq \bar{y}$ are the output values at high risk operational situations, while $y^{LR}: \ y > \bar{y}$ are the output values at low risk situations. Consequently, the total domain Θ_x of the input variables x can be divided into two subdomains, where Θ_x^{HR} is the input variables subdomain leading to y^{HR} and Θ_x^{LR} is the other subdomain leading to y^{LR}, see Figure 2.1. Although we are interested in the accurate prediction of y over the total domain Θ_x, we give much more importance to the accurate prediction of the output high risk values y^{HR} over the high-risk subdomain Θ_x^{HR}.

In order that the targeted surrogate system to become supportive for the reliable online operational of the system, it should provide the following requirements, or, in other words, it should solve the following problems, including the need for:

1. An alarm system $RL = C(x)$ that, given the measurements of the operating condition x, should be able to recognize the Risk Level (RL) (and also the subdomain) in which these conditions lies.

$$RL = \begin{cases} 0 & x \in \Theta_x^{LR} \quad , \ if \quad y > \bar{y} \\ 1 & x \in \Theta_x^{HR} \quad , \ if \quad y \leq \bar{y} \end{cases} \tag{2.2}$$

   where 0 corresponds to low risk and 1 to high risk.

2. Two predictors (surrogate models) $\hat{y}^{LR} = P^{-}(x), \ x \in \Theta_x^{LR}$ and $\hat{y}^{HR} = P^{+}(x), \ x \in$

Θ_x^{HR} each of which predicts the output variable at each of input subdomains, Θ_x^{LR} and Θ_x^{HR}, respectively. In this setting, we assume that the use of such ensemble to two local metamodels ($P^-(x)$ and $P^+(x)$) is supposed to provide better accuracy for the prediction of the values of the output variable y^{HR} at high risk subdomain Θ_x^{HR} than that of one global surrogate model $\hat{y} = F(x)$, $x \in \Theta_x$.

3. An efficient sampling procedure which enables the collection of input-output training data from the high-risk areas Θ_x^{HR}. Since the use classical DOCE will result in highly imbalanced dataset, i.e., the majority of the data will be collected from the low risks subdomain Θ_x^{LR}, while very few data points will be collected from the high risk subdomain Θ_x^{HR}.

Chapter 3

Methodology

The methodology proposed follows two main stages. The first one consists in the development of an ensemble of Machine Learning algorithms driven by the reliability nature of CDT. The second one deals with the necessity to improve the information that the considered domain carries. This is accomplished by a novel hybrid adaptive sampling methodology.

3.1 The composite model

The first stage of the methodology deals with the coupling of two different Machine Learning algorithms: SVM and ANN. The former is used for the development of a risk classifier, the latter for the development of two regression sub-models for CDT prediction.

3.1.1 Dataset Generation

At the beginning there's the necessity to generate an input domain by Latin Hypercube Sampling (LHS) technique (see appendix J). The output for each point is collected (e.g by OLGA simulations). Each point is then labelled with 1 if $y \leq \bar{y}$ or 0 if $y > \bar{y}$. The ensemble of these pieces of information gives the initial dataset. From this, a 10% of the data is saved as testing set and it's used for assessing the goodness of generalization of the surrogate models (i.e the performance of the models). The remaining 90% is used for the training session. A k-folds cross-validation is adopted for training each considered surrogate model while a 15% of the points for the training is used as validation set.

3.1.2 SVM classifier

The SVM classifier is needed for labelling each point with its predicted risk level (high-risk/low-risk) and it is trained with the totality of the training set. Given $C(\mathbf{x})$ the

classifier and given a new point $\mathbf{x_i}$, if $C(\mathbf{x_i}) = 1$ the point is predicted to be a high-risk point, hence to belong to the high risk region Θ_x^{HR}. On the contrary, $C(\mathbf{x_i}) = 0$ labels $\mathbf{x_i}$ as low-risk point, so as a point belonging to the low-risk region Θ_x^{LR}.

The SVM model is based on a gaussian kernel function. Moreover, it's worth to mention that, since the problem presents two clearly imbalanced classes (high-risk and low-risk), a cost matrix is applied:

$$\text{cost matrix } = \begin{bmatrix} 0 & 1/\text{num0} \\ 1/\text{num1} & 0 \end{bmatrix} \tag{3.1}$$

where $num1$ is the number of points in the high-risk region while $num0$ is the number of points of the low-risk region.

For assessing the performance of the classifier with respect to a testing set, the z index is used:

$$z = \frac{F + G_{mean}}{2} \tag{3.2}$$

where

$$F = 2\frac{PPV * TPR}{PPV + TPR} = 2\frac{TP}{2TP + FP + FN} \tag{3.3}$$

and

$$G_{mean} = \sqrt{PPV * TPR} \tag{3.4}$$

The closer z index is to 1, the better the classifier is.

3.1.3 ANN regressors

The two ANN sub-models are needed for the regression task, hence for predicting the values of CDT. These regressors are trained through tailored training sets: points belonging to Θ_x^{LR} train the ANN, $P^+(\mathbf{x})$, for low-risk region while on the contrary, points belonging to Θ_x^{HR} train the ANN, $P^-(\mathbf{x})$, for the high risk region.

The regression models are feed-forward networks while the learning algorithm is the Levenberg-Marquardt (see appendix F). In this work, for regression surrogate models, the chosen performance parameter is the Normalized Root Mean Square Error NRMSE. It is given by normalizing the Root Mean Square Error RMSE.

$$\text{RMSE [h]} = \sqrt{\frac{\sum_{\text{t=1}}^{\text{T}} \left(\hat{\text{y}}_\text{t} - \text{y}_\text{t}\right)^2}{\text{T}}} \tag{3.5}$$

Where $\hat{y}_t$ is the predicted value from the model, y_t is the target value and T is the number of samples considered. Normalizing:

$$\text{NRMSE [\%]} = \frac{\text{RMSE}}{y_{\text{max}} - y_{\text{min}}} * 100 \tag{3.6}$$

Where y_{max} and y_{min} are the maximum and minimum CDT values of the dataset. The lower the NRMSE, the better the performance.

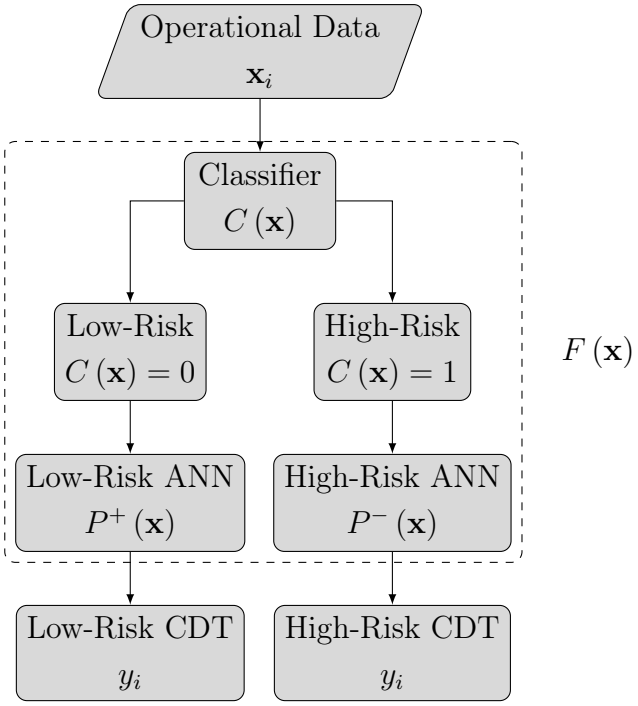


Figure 3.1. Flow chart of the composite model

3.1.4 Composite model's prediction

The structure of the developed composite model $F(x)$ is shown in figure 3.1. Given a new point $\mathbf{x_i}$ the predicted CDT y_i is given by the simultaneously combination of the risk classifier $C(\mathbf{x})$ and of the two regression models $P^+(\mathbf{x})$ and $P^-(\mathbf{x})$. The described ensemble is the composite model $F(\mathbf{x})$. When the composite model $F(\mathbf{x})$ has to predict the CDT for a given new point $\mathbf{x_i}$, first $C(\mathbf{x})$ classifies the point labelling it with low-risk $C(\mathbf{x_i}) = 0$ or high-risk $C(\mathbf{x_i}) = +1$, then, based on the labelling, the customized regression model $P^+(\mathbf{x})$ or $P^-(\mathbf{x})$ runs providing the predicted CDT, y_i.

3.2 Hybrid adaptive sampling methodology

The prediction's quality of a surrogate model is highly dependent on the size and distribution of the given training points. Generally, the aim of adopting an adaptive sampling is about the improvement of the performance of the surrogate model in a smart way. In particular, here there's the double goal of improving the performance of the risk classifier and of getting more information about the risk region. The proposed hybrid adaptive sampling methodology is made by coupling two adaptive novel methodologies based on the SVM classifier and on a GPR model.

The topic of adaptive sampling is well discussed in literature. Adaptive sampling is applied to SVM classifiers for example by balancing sub-datasets [17], by changing the

constraints inside the optimization problem of the SVM algorithm [18] or by improving SVM boundaries [19]. For what concerns GPR surrogate models, the state of art of adaptive sampling for Gaussian processes has been provided in [20]. In reliability analysis problems, adapting sampling strategies based on GPR have been applied by Subset Simulation in [21] , [22] and [23]. Other approaches that exploit the nature of the GPR of providing a posterior distribution have been shown in [24], [25] and [8].

Due to the novelty of the addressed problem, and due to the necessity to develop of an adaptive sampling approach with double goals, no directly suitable adaptive sampling technique has been found in literature. The hybrid adaptive sampling methodology exploits the "score" measure of the SVM algorithm together with a U function given by the outputs of a GPR model.

3.2.1 SVM-based adaptive sampling methodology

SVM defines for each point a measure called "score" which is the distance of that point to the high-low risk boundary. The magnitude of this measure is a guess of the reliability of the prediction. By intuition, if the score value falls in the boundary region, the probability to have a misclassification becomes higher. The proposed SVM-based adaptive sampling methodology tries to explore the boundary region defined by the optimization problem during the training session (see appendix H).

The SVM-based methodology aims at improving the performance of the classifier by adaptively choosing points close to the boundary region. In order to not choose too close points, the choice of the candidate points is accomplished by K-means Clustering (KMC). KMC is a Machine Learning (ML) algorithm for unsupervised clustering which aims to divide a set of points in a prior decided number of clusters, K (see appendix I). The nearest point to each of the K centroids is selected as candidate point. The algorithm of the methodology is shown here.

1. Train the SVM classifiers by k-folds cross validation.
2. For each SVM model:
   (a) Test 5k points given by the LHS method.
   (b) Assess the score of each point.
   (c) Select the points with $s_{min} < \ score \ < s_{max}$
   (d) Apply K-means clustering selecting K_1 clusters.
   (e) Choose as candidate points the closest K_1 points to the K_1 centroids.
3. Merge the candidate points in only one set.

4. Apply for the second time K-means clustering selecting K_2 clusters.

5. Choose as final candidate points the K_2 points closest to the K_2 centroids.

6. Simulate the final K_2 candidate points by physics model or by mathematic function.

7. Add the final candidates points with their output values in the training set.

3.2.2 GPR-based adaptive sampling methodology

In order to add information in the high-risk subdomain, and hence improve the performance of the regression risk surrogate model $P^{-}(\mathbf{x})$, a GPR model is exploited. Differently from ANN models, for a new X_* and given a training set of points X and outcomes $\vec{y}$, GPR model' outcome is a posterior distribution:

$$\overrightarrow{y_*} \mid \vec{y}, X, X_* \sim \mathcal{N}(\mu^*, \Sigma^*) \tag{3.7}$$

with μ^* and Σ^* that are computed algebraically and which represent the expected value and the expected standard deviation (see appendix G).

The proposed GPR-based adaptive sampling methodology exploits the output information of the GPR model by exploring a space created through LHS's method. It then chooses the candidate points by K-means clustering among the ones selected by a learning function U. This is given by

$$U = \frac{\hat{y} - h}{\hat{\sigma}} \tag{3.8}$$

where $\hat{y}$ is the predicted output value for the given point, $\hat{\sigma}$ is the predicted standard deviation for the given point while h is a threshold parameter used to guide the exploration of the high risk region. The following algorithm gives the detailed workflow for this methodology:

1. Train the k GPR models by k-folds cross validation.

2. Choose a set of positive h values by Monte Carlo Sampling based on a normal distribution which covers all the risk subdomain.

3. For each GPR model:

   (a) Test 5k points given by the LHS method.

   (b) Assess the U of each point.

   (c) Select the points with $U_{min} < U < U_{max}$

   (d) Apply K-means clustering selecting K_3 clusters.

   (e) Choose as candidate points the closest K_3 points to the K_3 centroids.

4. Merge the candidate points in only one set.
5. Apply for the second time K-means clustering selecting K_4 clusters.
6. Choose as final candidate points the K_4 points closest to the K_4 centroids.
7. Simulate the final K_4 candidate points by physics model or by mathematic function
8. Add the final candidates points with their output values in the training set.

3.2.3 Hybrid adaptive sampling methodology and stopping criteria

The hybrid adaptive sampling methodology, which ensembles both SVM and GPR, is represented by the flow chart in figure 3.2. Moreover, a stopping criterion is needed. This can be simply a maximum number of iterations or it can be given by number of iterations in a row that don't show improvements in a chosen performance parameter.

Since here the work is dealing with complex ML structures and since the most straightforward stopping criterion is the maximum number of iterations, this is used in this work.

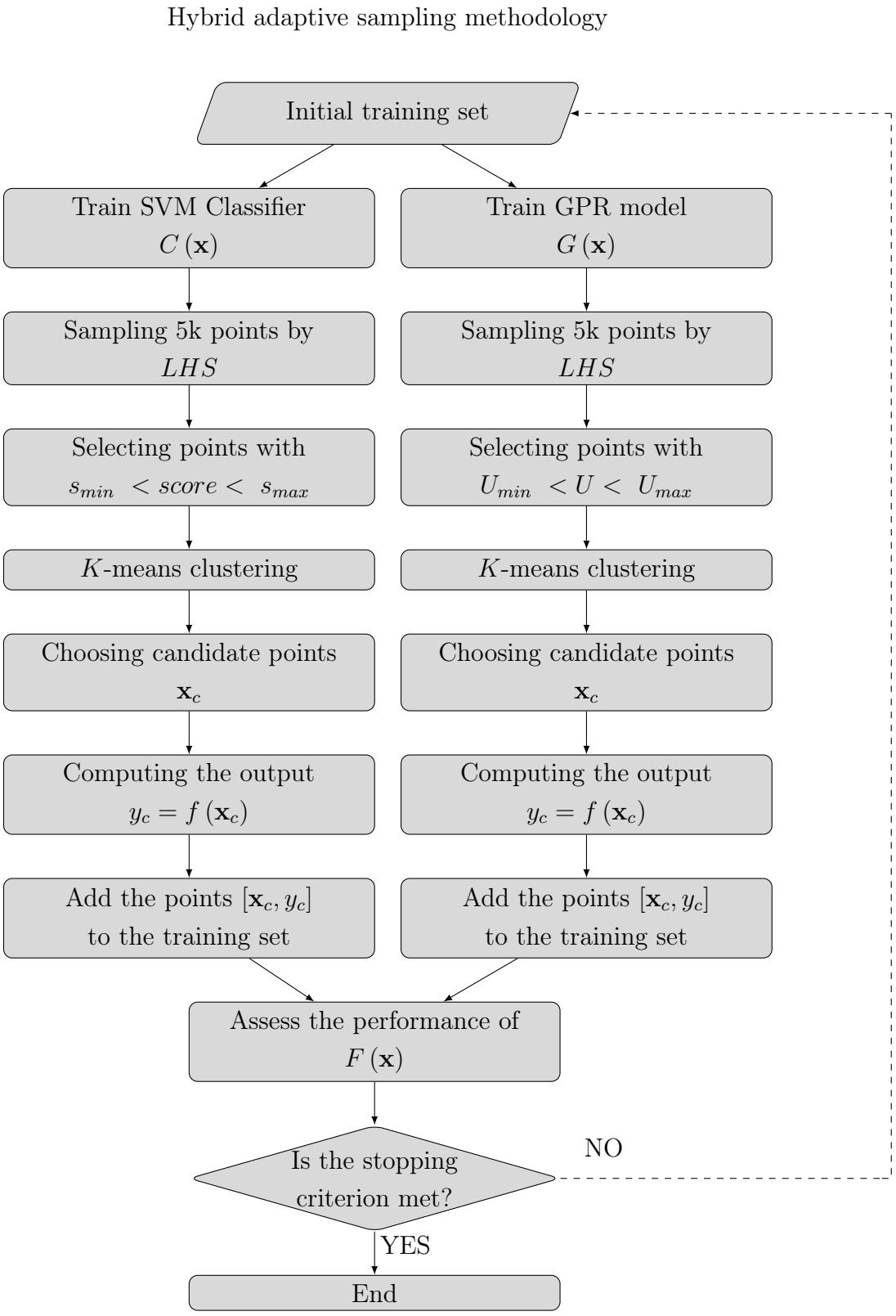


Figure 3.2. Flowchart for Hybrid Adaptive Sampling Methodology. The left side refers to the SVM-based, the right side refers to GPR-based adaptive sampling technique.

Chapter 4

Application

Three examples of application are considered. The first two examples use complex mathematical functions in order to validate the proposed methodology. Afterwards, the methodology is applied to the case study.

For all the applications, the hybrid adaptive sampling approach is applied to both an ANN model and a composite model. Then, their performance is compared also with the performance of both a ANN and a composite model trained by a set populated by LHS with the same number of points.

4.1 Mathematical functions

The two mathematical functions share some common features and for both, common methodologies' parameters are applied.

First of all, both the equation are defined by $x_1, x_2 \in \mathbb{R}$ and by the output $g(x) \in \mathbb{R}$. Moreover, for both the functions the high-risk region is considered for values of $g(x) < 0$. Considering $x_1, x_2 \in [-5; 5]$, the high-risk region is much less extended than the low-risk region leading to a high imbalanced $g(x)$ distribution.

For what concerns the methodology's parameters:

- Training set: the initial training set size is populated by 60 points which are created by LHS. Five folds for cross validation are considered.
- Testing set: the testing set is made by 300 points created by LHS.
- Iterations: the number of iterations to be performed is set to 20, adding 10 points to the training set each iteration.
- Parameters for SVM-based adaptive sampling methodology: for both the functions the picked points have $-1 < score < 0.5$. $K_1 = 10$ and $K_2 = 5$ (considering the same notation introduced in chapter 3 are chosen as the number of clusters

respectively for each trained model by cross validation and for each iteration. Hence 5 final candidate points are chosen.

- Parameters for GPR-based adaptive sampling methodology: for function 1 the minimum value shown by the outputs of the points is $min = -3$, while for function 2 $min = -2.42$. Hence h values are chosen by Monte Carlo Sampling considering the distributions:

$$h \sim \mathcal{N}\left(-3, \quad 1\right)$$

for the first function and

$$h \sim \mathcal{N}\left(-2.42, \quad 0.8\right)$$

for the second function. The mean of the distributions is the minimum value while the standard deviation is computed by $\sigma = (\bar{y} - min)/3$ where $\bar{y} = 0$ is the critic value. Afterwards, only the absolute values of h are taken. The U function picks the points for $0 < U < +2$. For clustering, $K_3 = 10$ and $K_4 = 5$ are set. Hence 5 final candidate points are selected.

- Hybrid adaptive sampling methodology application: the hybrid adaptive sampling is applied starting from a training set of 60 points, adding 10 points each iteration for 20 iterations. Eventually, 260 points are present in the training set. In order to assess the goodness of the results, the performance of both an ANN and a composite model trained by the adaptive sampling methodology is compared with the performance of both an ANN and a composite model trained by a training set of the same size populated by LHS.

4.1.1 Function 1

The first function is a non-linear bi-variate performance function used in [8]. It is given by

$$g\left(x\right) = q - \frac{1}{20}\left(x_1^2 + 4\right)\left(x_2 - 1\right) + sin\left(\frac{5}{2}x_1\right) \tag{4.1}$$

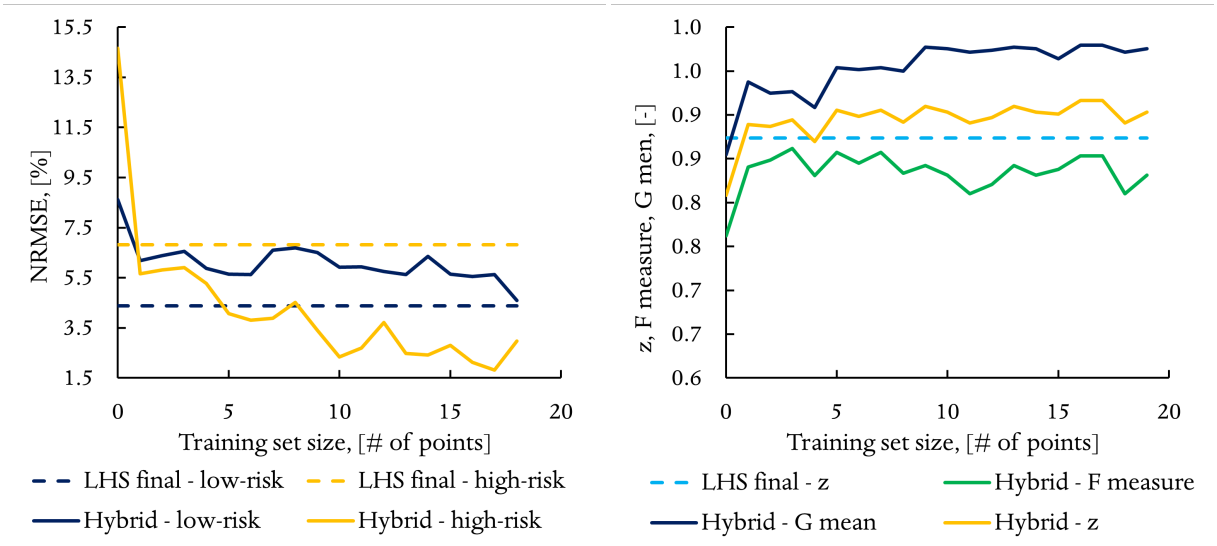
With $q = 2$ and $x_1, x_2 \in \mathbb{R}$. Moreover the domain for x_1 and x_2 is given by the range $[-5; 5]$. Results are shown in table 4.1.

Added points distribution

The goodness of the hybrid methodology can be assessed by looking at graphical distribution of the added points. In figure 4.2a the final and initial datasets for the hybrid adaptive approach are represented by an histogram. The added points are highlighted with different colours. Looking at figure 4.4, it's possible to evaluate the spatial distribution of points. The great majority of the added points is in the high-risk region which is a small area of

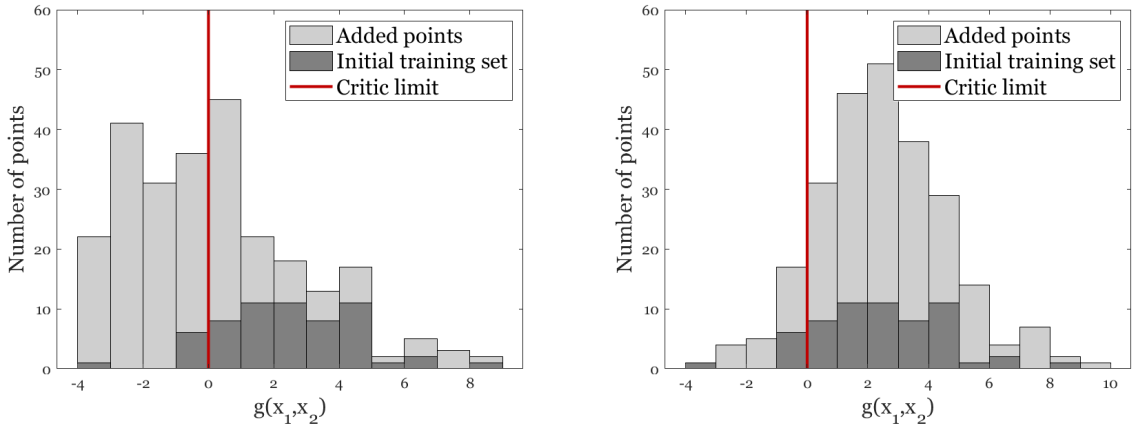
Table 4.1. Function 1. Performance indices for both LHS and Hybrid Adaptive Sampling Methodology for both global ANN and Composite model.

	LHS		Hybrid adap. Sampling	
	Global ANN	Composite	Global ANN	Composite
	NRMSE, [%]			
Global performance	4.59	4.70	4.86	4.90
Low-risk performance	4.31	4.37	5.00	5.11
High-risk performance	6.45	6.82	3.51	2.32
	z, [-]			
classifier	-	0.87	-	0.90



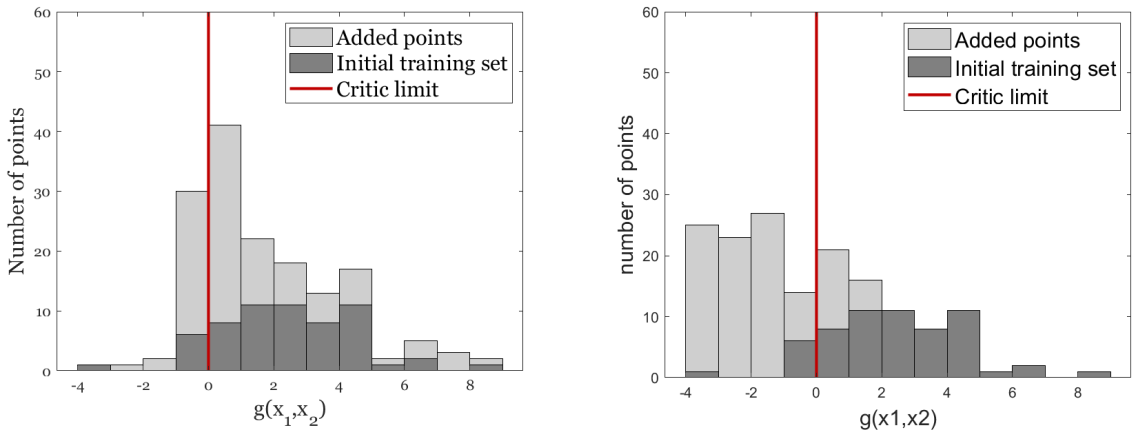
(a) (b)

Figure 4.1. Function 1. Hybrid adaptive sampling methodology: NRMSE, $F_{measure}$, G_{mean} and z improvement.



(a) (b)

Figure 4.2. Function 1. Distribution of initial and final training set applying the hybrid adaptive sampling methodology (left) and LHS method (right).



(a) (b)

Figure 4.3. Function 1. Distribution of initial and final training set applying the hybrid adaptive sampling methodology: contribution of SVM-based (left) and GPR-based (right) methodologies.

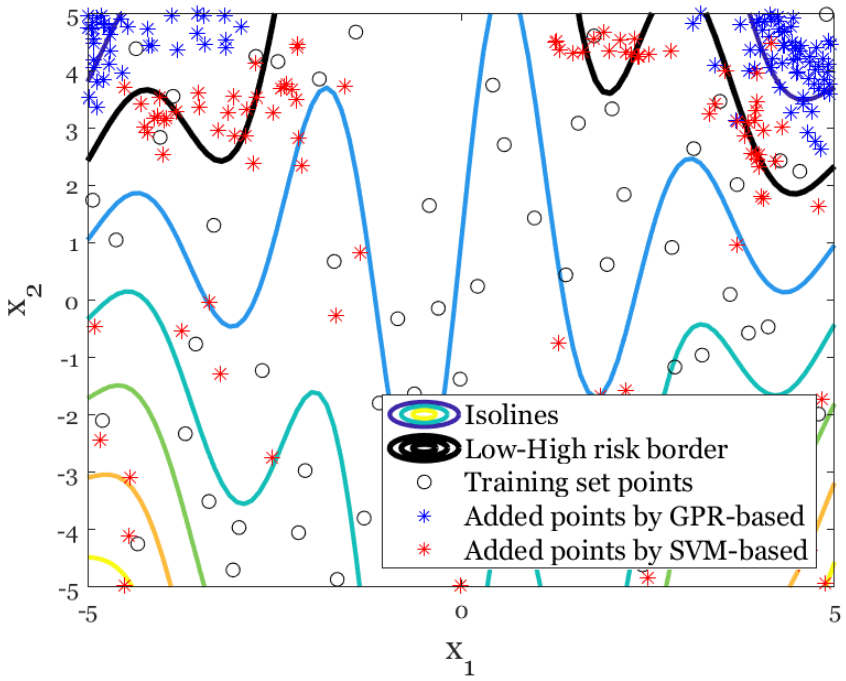


Figure 4.4. Function 1. Spatial distribution of points.

the domain. In order to evaluate the contribution from each single adaptive methodology applied, one can refer to the different colours of points on the map. Furthermore, it's possible to refer to the histograms 4.3b and 4.3a. As expected, while the SVM-based adaptive sampling methodology adds points mainly in the nearness of the critic boundary region, the GPR-based adaptive sampling methodology works by adding points in the high-risk region. Finally, a comparison about the distribution of points for hybrid adaptive sampling and LHS is given by figure 4.2a and 4.2b.

Comparison between Hybrid approach and LHS approach

Referring to table 4.1, a comparison between LHS technique and Hybrid sampling methodology is held. The main outcome is that the hybrid adaptive technique improves the performance of both ANN and composite models for high-risk region and for the classifier. On the other hand, for low-risk region, the LHS gives better performance for both ANN and composite models.

Considering the composite model, the Figure 4.1a shows the computed NRMSE for each iteration during the adaptive process: low-risk, high-risk and global indicators are drawn. Threshold lines, indicating the NRMSE for high-risk and low-risk regions achieved by the composite model with LHS technique, highlight the goodness of the hybrid adaptive methodology. In the same way figure 4.1b shows the improvement of the z, $F_{measure}$ and G_{mean} indices. Also in this case a threshold given by the final z value for LHS sampling is

drawn.

Comparison between ANN models and composite models

Always referring to table 4.1 and considering the case of LHS, the global ANN performs better in all the performance indices. On the contrary considering the adaptive sampling, the composite model shows a worse performance for low-risk and global values but it shows a great improvement in the high-risk region.

The best model for high-risk performance is the composite model trained by adaptive sampling technique. This is in accordance with the expected goal: training a composite model with the hybrid adaptive sampling methodology brings to good results in the high-risk area.

4.1.2 Function 2

The second function analysed in this work is a translated series system already used in literature for reliability calculations. The series system is given by

$$p(x) = \min \begin{cases} p_1(x) = k + 0.1\,(x_1 - x_2)^2 - \frac{x_1+x_2}{\sqrt{2}} \\ p_2(x) = k + 0.1\,(x_1 - x_2)^2 + \frac{x_1+x_2}{\sqrt{2}} \\ p_3(x) = (x_1 - x_2) + \frac{m}{\sqrt{2}} \\ p_4(x) = (x_2 - x_1) + \frac{m}{\sqrt{2}} \end{cases} \tag{4.2}$$

where the parameters are the same as in [8]: $m = 6$, $k = 3$ with $x_1, x_2 \in [-5, 5]$. The considered function $g(x)$ is a translation of the series system:

$$g\,(x) = p\,(x) + 3 \tag{4.3}$$

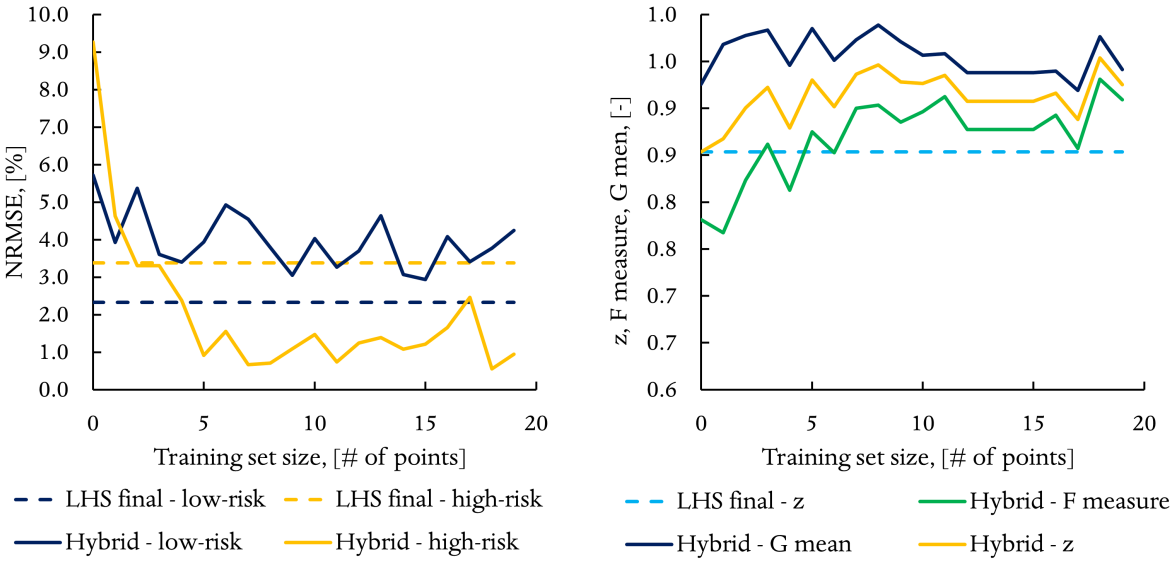
As for function 1, the high-risk region is considered for values $g(x) < 0$. Also in this case only small areas of the domain are considered high-risk, leading to an high imbalanced $g(x)$ distribution considering a DoE made by LHS. Table 4.2 shows the results.

Added points distribution

The high-risk area is well explored by the hybrid adaptive method. In fact, looking at figure 4.8, one can see the disposition of added points in the domain: high-risk regions and the region in the nearness of the risk-boundary are well filled. The different colours of the points indicate the contributions of both SVM-based and GPR-based approached. The same contribution are outlined by assessing the added points' distribution through histograms (figures 4.7a and 4.7b). The last comparison is about the distribution of added points by hybrid adaptive sampling methodology and the points added by LHS method (figures 4.6a and 4.6b).

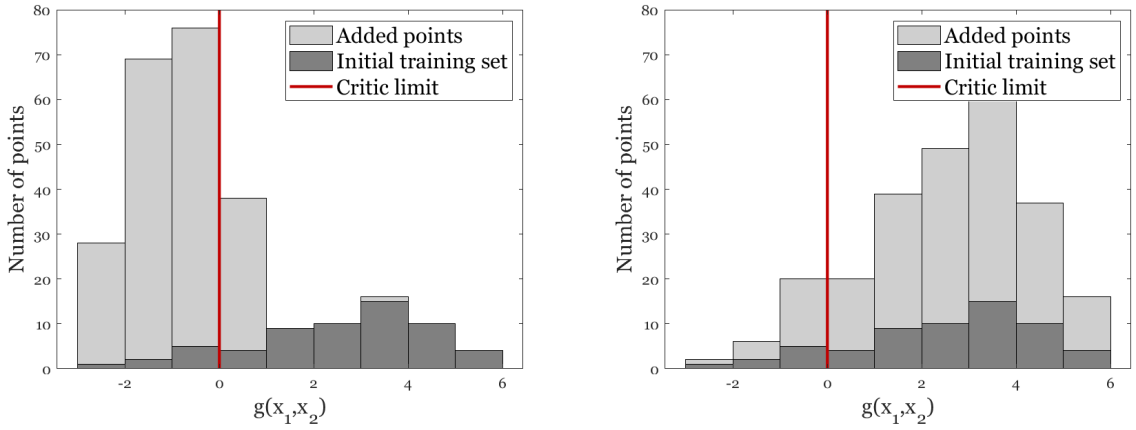
Table 4.2. Function 2. Performance indices for both LHS and Hybrid Adaptive Sampling Methodology for both global ANN and Composite model.

	LHS		Hybrid adap. Sampling	
	Global ANN	Composite	Global ANN	Composite
	NRMSE, [%]			
Global performance	1.82	2.44	2.80	3.82
Low-risk performance	1.83	2.32	2.93	4.00
High-risk performance	1.69	3.38	0.79	0.75
	z, [-]			
classifier	-	0.85	-	0.94



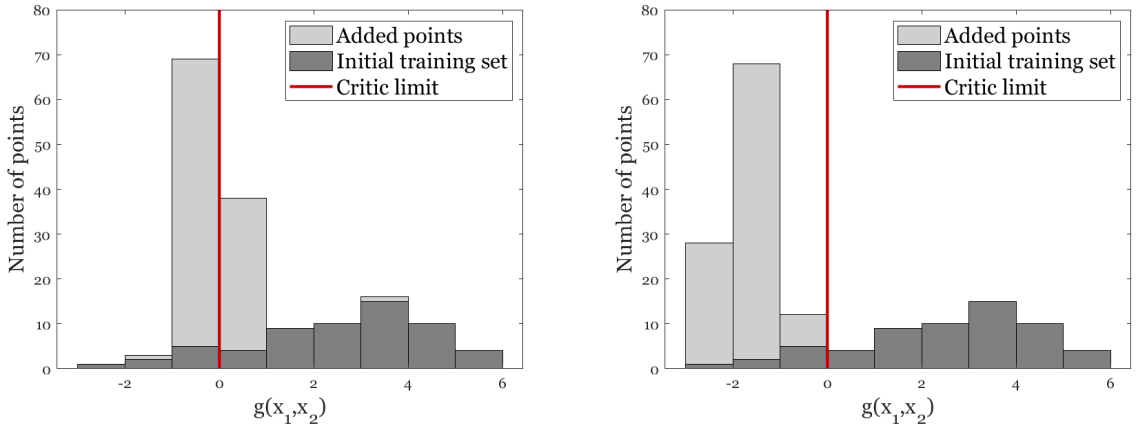
(a) (b)

Figure 4.5. Function 2. Hybrid adaptive sampling methodology: NRMSE, $F_{measure}$, G_{mean} and z improvement.



(a) (b)

Figure 4.6. Function 2. Distribution of initial and final training set applying the hybrid adaptive sampling methodology (left) and LHS method (right).



(a) (b)

Figure 4.7. Function 2. Distribution of initial and final training set applying the hybrid adaptive sampling methodology: contribution of SVM-based (left) and GPR-based (right) methodologies.

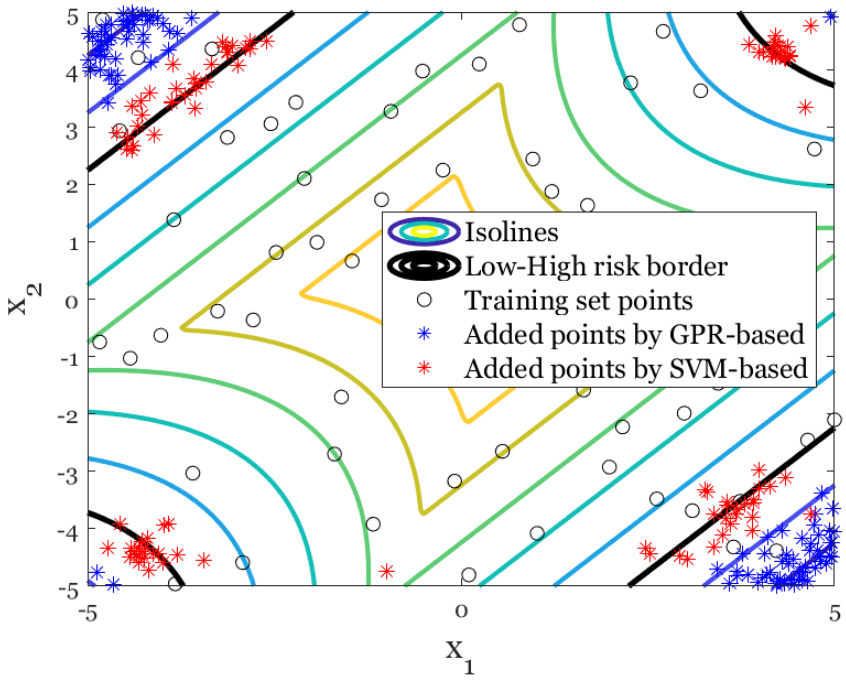


Figure 4.8. Function 2. Spatial distribution of points.

Comparison between Hybrid approach and LHS approach

As for function 1, a comparison between LHS technique and Hybrid sampling methodology is held. As one can notice from the table 4.2, also in this case, for both global ANN and composite model, the hybrid adaptive sampling works well in improving the performance in the high-risk region and for the classifier.

The same kind of plots used for function 1 are proposed in figures 4.5a and 4.5b which show the performance improvement along with the hybrid adaptive sampling methodology.

Comparison between ANN models and composite models

As in the case of function 1, the comparison between the global ANN and the composite model adopting the LHS shows better performance for the global ANN. On the contrary, adopting the hybrid sampling methodology, the composite model shows better performance for high-risk region.

In this case the global ANN presents performance with really low values which can make the efforts of applying the methodology useless. This can be interpreted by considering that the function 2 is not as non-linear as function 1, and moreover as the physics-based model. This also suggests that a combined used of LHS and ANN can be sufficient to train these kind of functions which present only few "difficulties".

However, since also in this case the best model for high-risk performance is the

composite model trained by adaptive sampling, the methodology is proven to well work.

4.2 Case Study

The main scope of this work is the development of a surrogate model whose aim is to predict in fast and reliable way the CDT of a subsea pipeline. The reference asset considered is part of the producing field block located offshore a West African country and developed by a joint venture with Eni S.p.A as leading operator.

The reference field has different producing wells that convey the oil to subsea cluster manifolds before sending it with one single line to the FPSO by means of a flexible flowline. Here the produced oil is stored and off-loaded to tankers which are headed to the coast where further treatments are applied. On the other hand, the produced gas is compressed and used as fuel and for gas lift which is one of the main artificial lift techniques.

For the purpose of the study, a part of this asset is modelled in OLGA. In particular the asset under study is made by a subsea manifold which is located at a water depth of about 1200 meters. In the subsea manifold a mixture of oil, gas and water phases from different wellheads is collected and is canalized in a flexible pipeline towards the FPSO. The pipeline can be seen as built up by two components: a subsea line which is lying on the seabed and which is long about 15 km, and a riser which is the part of the pipe which "rises" the fluid from the bottom of the ocean up to the floating platform. The floating platform is another crucial component of the asset and it's located about 15 km far from the subsea manifold collecting the flows coming from different locations of the field. Eventually, two valves, the manifold choke valve and the surface choke valve are considered. In order to emulate the real asset, OLGA provides many components that allow to build up and simulate very complex system in steady and, above all, in transient state. A description of the elements, variables, parameters and simulation values considered in this project can be found in appendix D.

Five variables are chosen as input variables for the surrogate model. Fixing all the design parameters in OLGA, the five input variables are the ones needed for closing the system of equations OLGA simulator is based on. First of all, the separator pressure (P_{out}) is chosen. From an operational point of view this parameter is really important since its calibration can control the flow and hence the production for a given manifold pressure. The flow rates are then taken as input variables by specifying the oil flow rate at standard condition (Q_{oil}) , Gas-Oil ratio at standard condition (GOR) and the water cut (WC). GOR is defined as the ratio between the produced gas and the produced oil. WC is the ratio of water produced and the volume of total liquids produced. The last chosen parameter is the temperature of the fluid at the manifold (T_{in}). This is actually the most important parameter for the issue this work is facing.

The operative ranges of these variables are then evaluated and chosen. The domain to explore during the methodology application is hence given by table 4.3.

Table 4.3

Input Variables	Ranges	Units of measure
Q_{oil}	5000-20000	$[bopd]$
GOR	500-4000	$[scf/bbl]$
WC	0-0.6	$[-]$
T_{in}	40-130	°C
P_{out}	5-40	bar

OLGA simulation doesn't compute the CDT automatically and a post-process work is performed. The algorithm developed for computing the CDT is based on the main concept that for hydrate formation to be possible, four conditions should be met in a specific point of the pipeline. First of all the fluid should be in the hydrate thermodynamic stable region (left part considering the hydrate equilibrium curve) 4.9. In OLGA this is checked by the setting the keyword DTHYD>0. DTHYD indicates the difference between the temperature for the hydrate formation at that Pressure and the actual temperature of the fluid. The second condition to be met is presence of gas which is checked in OLGA by the keyword AL>0. The last two conditions are related to the necessity to have water. First, one should check the presence of liquid given by OLGA keyword HOL>0 and then check for the presence of water inside the liquid. This is checked by OLGA keyword WC>0 (here WC indicates the OLGA keyword for the local Water Cut). The algorithms computes CDT by assessing at every point along the length of the pipe and over the entire simulation time. When the four condition are met, it gives back the time at which this happens. This time is the CDT.

The CDT computed through the post-processing of simulation outputs varies over a range of 20 hours. The critic CDT is set to 7 as a reasonable value for accomplishing the operating philosophy for this asset. The setup of the methodology is set by the following parameters.

- Training set: as usual, an initial training set is needed. In this case, a domain of 400 points is created by LHS and CDT are computed by means of simulations. Unphysical points (the one with too high pressure values at the inlet of the pipe) and the points whose $CDT < 0$ are deleted from the training set. Points with $CDT > 20\ h$ are set to be 20. Ten-folds cross validation is used. Figure 4.10

- Testing set: the testing set is composed by 300 points created by LHS.

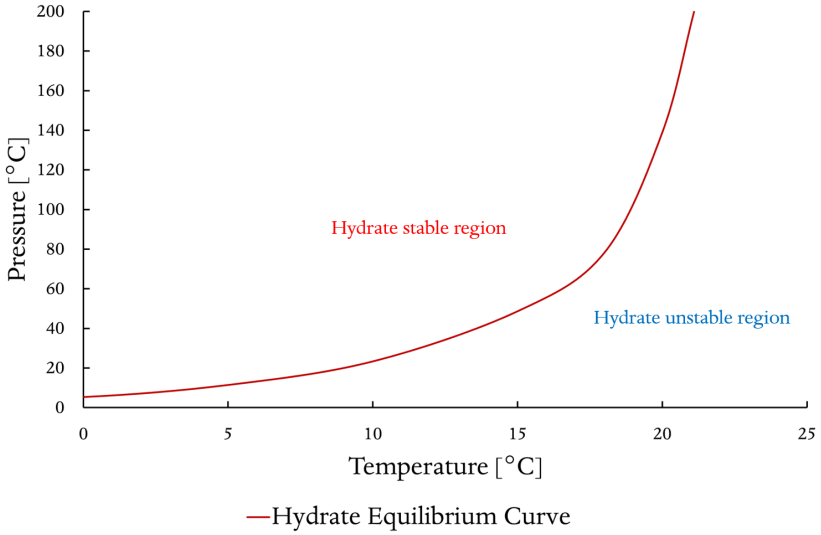


Figure 4.9. The hydrate curve used in this work.

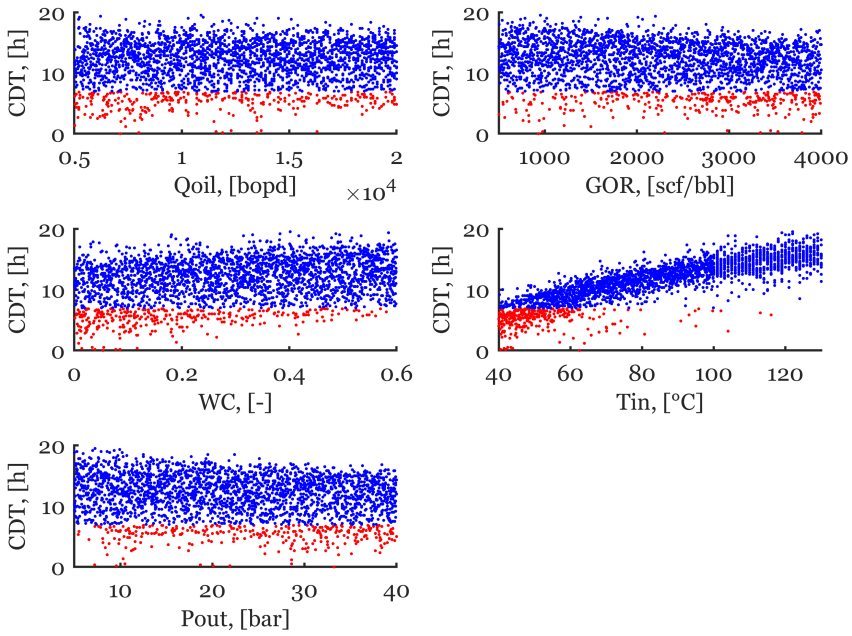


Figure 4.10. Training set for physics-based model: scatter plots for the input variables. Blu points are low-risk points. Red points are high-risk points.

- Iterations: The number of iterations to be performed is set to 20, adding 10 points to the training set each iteration. Totally 200 points are added.

- Parameters for SVM-based adaptive sampling methodology: a domain of 5k points is investigated by both SVM-based and GPR-based adaptive sampling methodologies. The points searched are the ones located in the boundary region, hence $-1 < score < 1$. While $K_1 = 10$ and $K_2 = 5$ are set. Each iteration adds 5 points.

- Parameters for GPR-based adaptive sampling methodology: in physics based model, the minimum CDT is $min = 0\ h$. Differently from the mathematical models, the normal distribution for the choice of the threshold h is not set with a mean equals to the minimum value. Due to the nature of physics system, such a configuration would lead to the selection as candidate points of too many points very close or less than zero. Hence, in order to explore the high-risk domain in a good way, h values are chosen by Monte Carlo Sampling considering the right part of a normal distribution with mean equal to 1.5 h and with $\sigma = 0.09$ (figure 4.11). Then, the points are selected by assessing the U function in each point and picking up the points with $0 < U < +2$. For clustering, $K_3 = 10$ and $K_4 = 5$ are set. Hence 5 final candidate points are added to the training set.

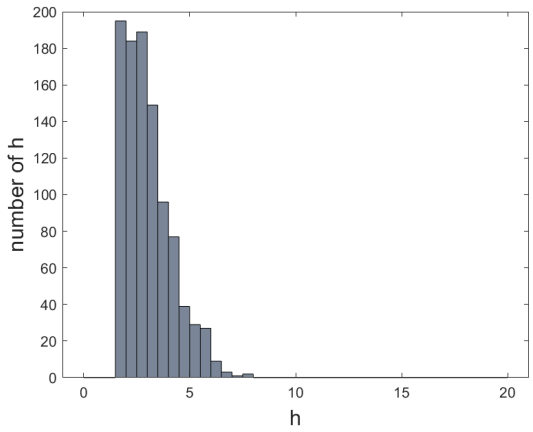


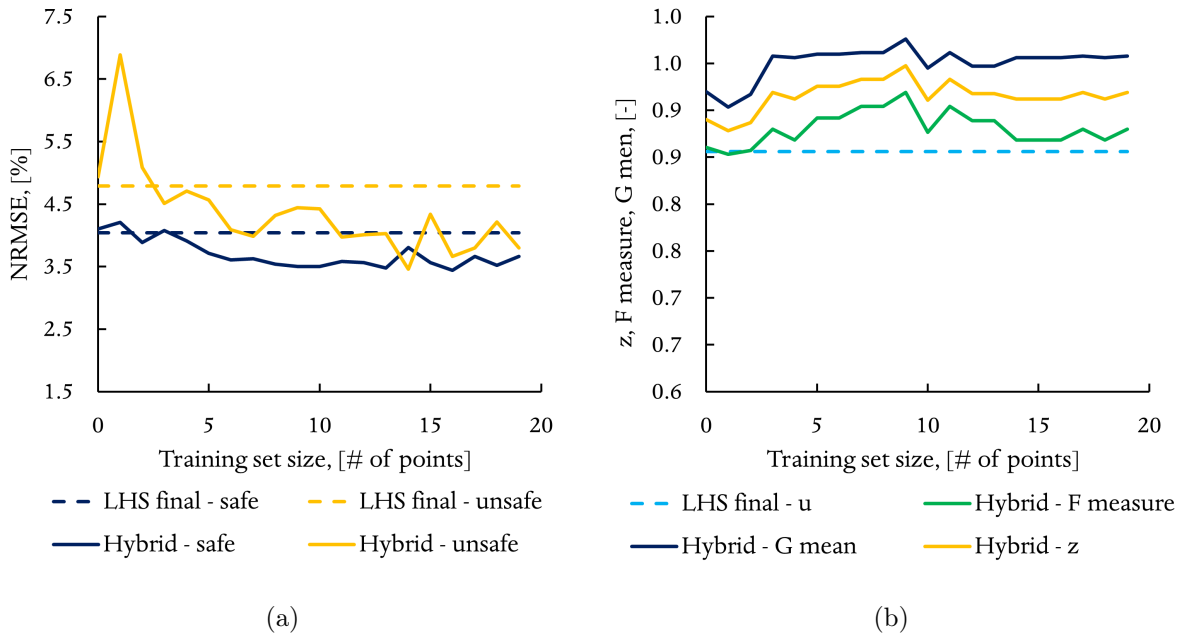
Figure 4.11. Distribution of h threshold parameter for physics model, given by the sampling of 1000 points.

- Hybrid adaptive sampling methodology application: after having applied the hybrid adaptive methodology to the composite model, the training set is made by 600 points. In order to assess the goodness of the results, the performance of both an ANN and a composite model trained by the adaptive sampling methodology is compared with the performance of both an ANN and a composite model trained by a training set of the same size populated by LHS.

Table 4.4. Physics model. Performance indices for both LHS and Hybrid Adaptive Sampling Methodology for both global ANN and Composite model.

	LHS		Hybrid adap. Sampling	
	Global ANN	Composite	Global ANN	Composite
	NRMSE, [%]			
Global performance	3.50	4.14	3.56	3.64
Low-risk performance	3.05	4.04	3.31	3.59
High-risk performance	5.83	4.79	5.00	4.01
	z, [-]			
classifier	-	0.86	-	0.92

The main outcomes are shown in table 4.4.

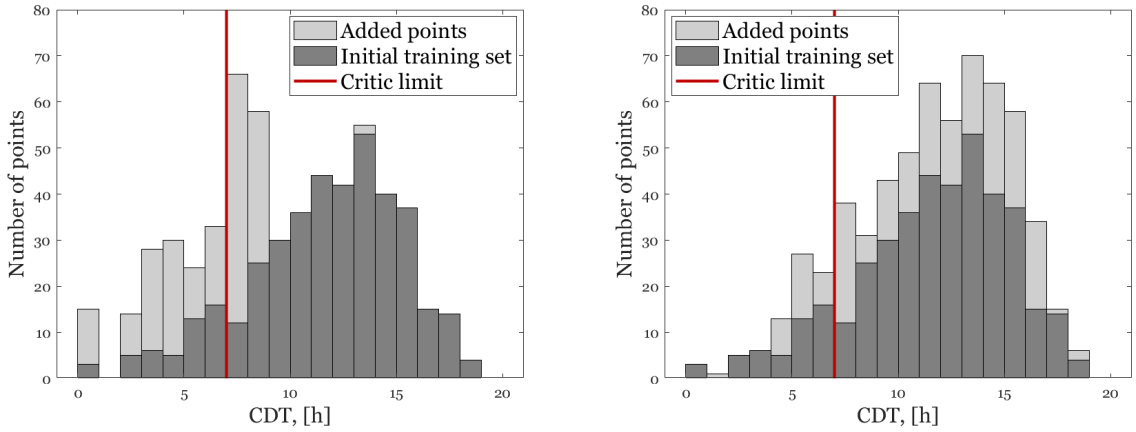


(a) (b)

Figure 4.12. Physics model. Hybrid adaptive sampling methodology: NRMSE, $F_{measure}$, G_{mean} and z improvement.

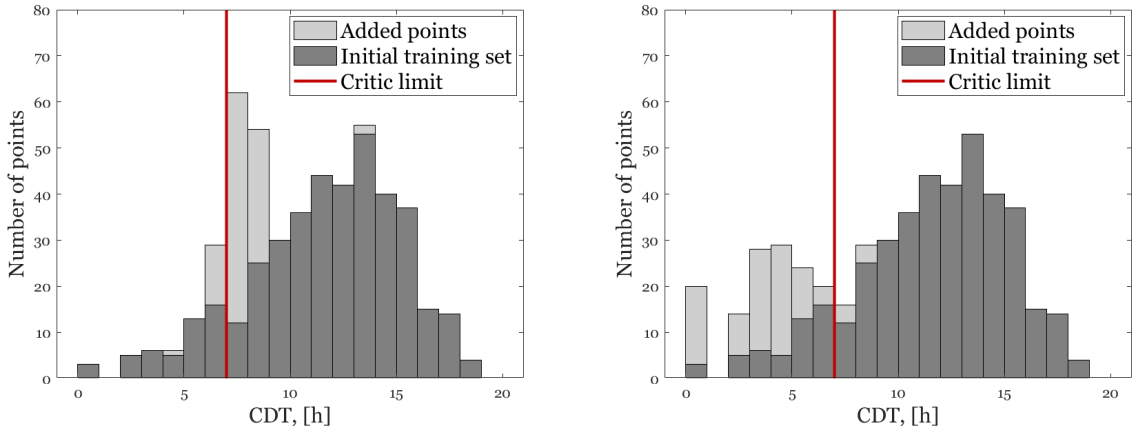
Added points distribution

Figure 4.13a shows the distribution of the final and initial training sets. Moreover figures 4.14a and 4.14b show the single contributions from both SVM-based and GPR-based methodologies. On the contrary, figure 4.13b shows the distribution of the final training set built by LHS method. The distribution given by the hybrid approach seems to be more homogeneous and to cover the entire domain of CDT. On the contrary LHS technique adds points which lead to higher CDT, populating a region of the domain already well



(a) (b)

Figure 4.13. Physics model. Distribution of initial and final training set applying the hybrid adaptive sampling methodology (on the left) and applying the LHS (on the right)



(a) (b)

Figure 4.14. Physics model. Distribution of initial and final training set applying the hybrid adaptive sampling methodology: contribution of SVM-based (left) and GPR-based (right) methodologies.

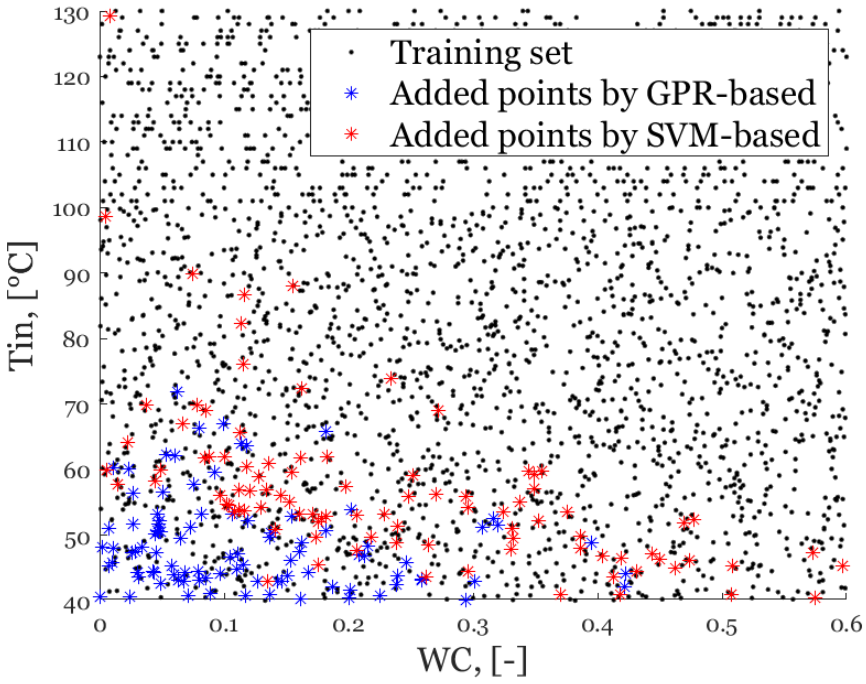


Figure 4.15. Physics model. Spatial distribution of points.

filled. From the comparison of the results, it's possible to infer that the proposed hybrid adaptive sampling methodology brings to smarter choices of the points to add in the domain, and, consequently, higher performance. Finally, figure 4.15 shows the distribution of added points in the plane given by two of the five input varibales, T_{in} and WC (which should be the most important ones considering the points distributions in figure 4.10). The contributions of SVM-based and GPR-based in figure 4.15 are highlighted with different colours.

Comparison between Hybrid approach and LHS approach

As for the mathematical functions, a comparison between LHS technique and Hybrid sampling methodology is carried out. As one can notice in table 4.4, also in this case, for both global ANN and composite model, the hybrid adaptive sampling works well in improving the performance in the high-risk region and for the classifier.

Moreover, in this case, the hybrid adaptive sampling approach is shown to enhance the performance for the composite model both in the high-risk and in the low-risk regions. This is probably given by a favourable distribution of points added in the nearness of the critic CDT. Even if not searched, this outcome shows another benefit in the use of the hybrid adaptive approach.

In the same way of mathematical functions, the performance improvement along with the hybrid adaptive sampling methodolog are shown in 4.5a and 4.5b.

Comparison between ANN models and composite models

For the physics-based model considered, as shown in chapter 5, building a composite model is usually beneficial in terms of high-risk performance. This can be seen also in table 4.4 where both in the case of the LHS method and in the case of hybrid adaptive sampling, the performance related at the the high-risk region are improved.

As usual, the best high-risk performance is given the composite model trained by the adaptive sampling. It's possible to infer that applying the proposed methodology to the case study improves the performance of the high-risk region endorsing the goodness of the approach.

Chapter 5

Further Analysis

For the physics-based model the benefits of training a composite model instead of a global regressor (such as an ANN or GPR model) are evident. In this chapter, that is investigated. Moreover, an analysis about the computational time is carried out.

5.1 Training set size effect

In this section an analysis about the impact of the training set size on the performance of both ANN and composite model of the physics based model is carried out. To be noticed that the adaptive sampling is not run in this case. The analysis has the goal to show the goodness and robustness of the application of the composite model when the training size changes.

Using as reference case the one with CDT critic equal to 7 hours, the ANN model and the composite model are trained with different training set size and using the same testing set for assessing their performance. The smallest size considered is about 130 points, while the biggest training set is made by about 2650 points.

First of all, it's important to notice the trend for a single trained ANN model shown in Figure 5.1. One main outcome should be highlighted: increasing the training set size, in particular moving from about 130 points to about 2650 points leads to a very small improvement: the global NRMSE passes from about 4.8% to 3.35%.

A similar behaviour can be seen also in figure 5.2a. Here the comparison between the global performance of ANN and composite model is carried out. It's possible to notice that, also for the composite model, the NRMSE struggles to decrease. Moreover it is seen that the global performance of the composite are usually worse than the global performance of the ANN.

Nevertheless, the main comparison is shown in figure 5.2b. Here, the performance for the high-risk region are shown both for the ANN and for the composite model. It's possible to infer that, for the physics based model, applying the composite model architecture

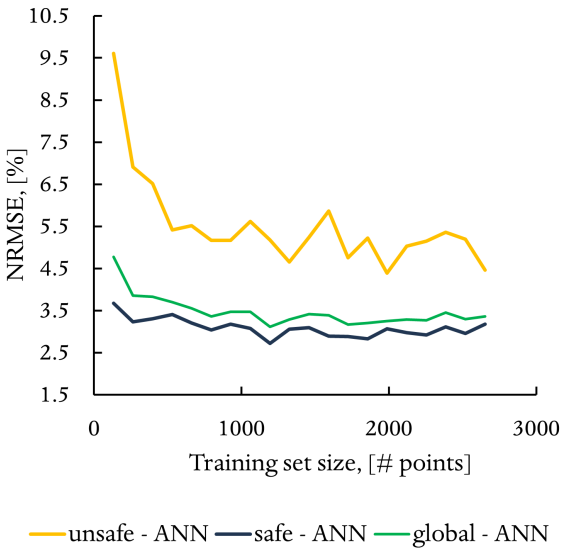
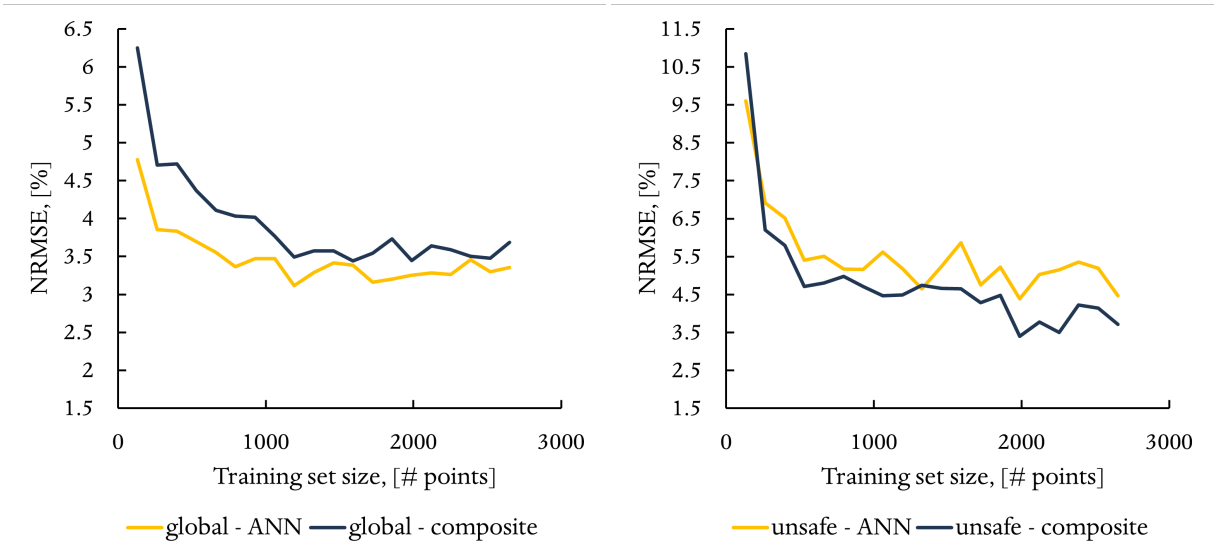


Figure 5.1. NRMSE of ANN model vs training set size.



(a) (b)

Figure 5.2. NRMSE of both composite and ANN model for global (left) and high-risk (right) region vs training size

leads to an enhanced performance for the critical region.

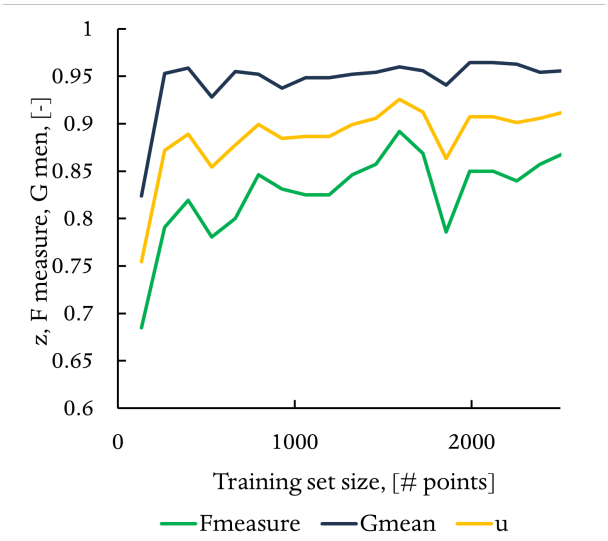


Figure 5.3. Testing performance indices for classifier vs training set size.

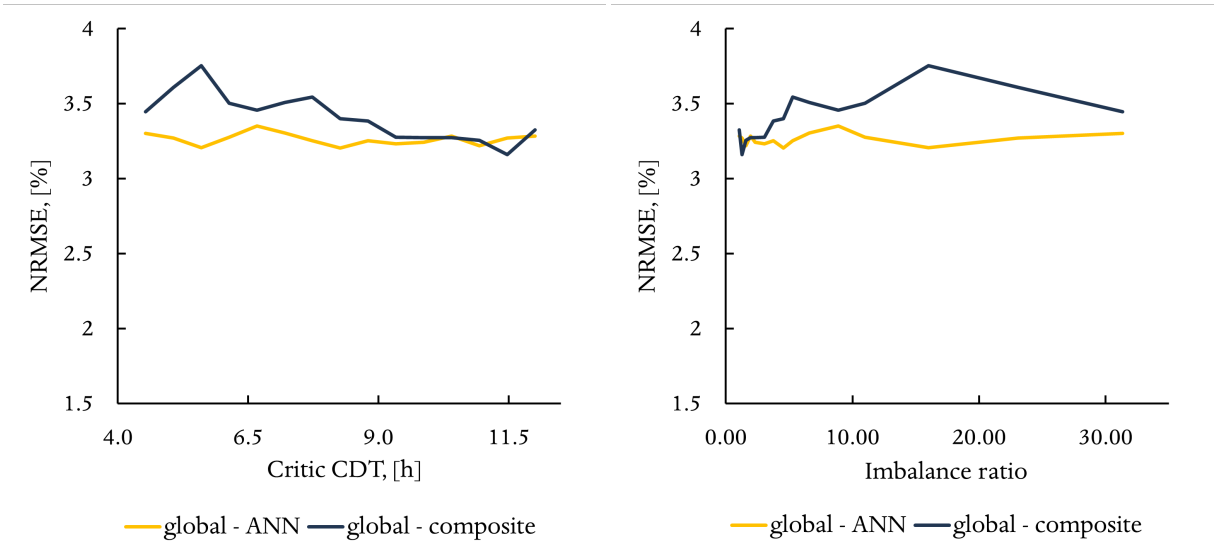
Eventually, in figure 5.3, the performance indices of the classifier are plotted against the size of the training set. It's possible to state that no huge improvement is obtained for training set sizes higher than about 1000 points.

5.2 Imbalance ratio effect

In section 5.1 the analysis is carried out using a constant critic CDT, and, consequently, a constant imbalance ratio. The scope of this section is to analyse the effect of imbalance ratio on the performance indices both for ANN and adaptive model.

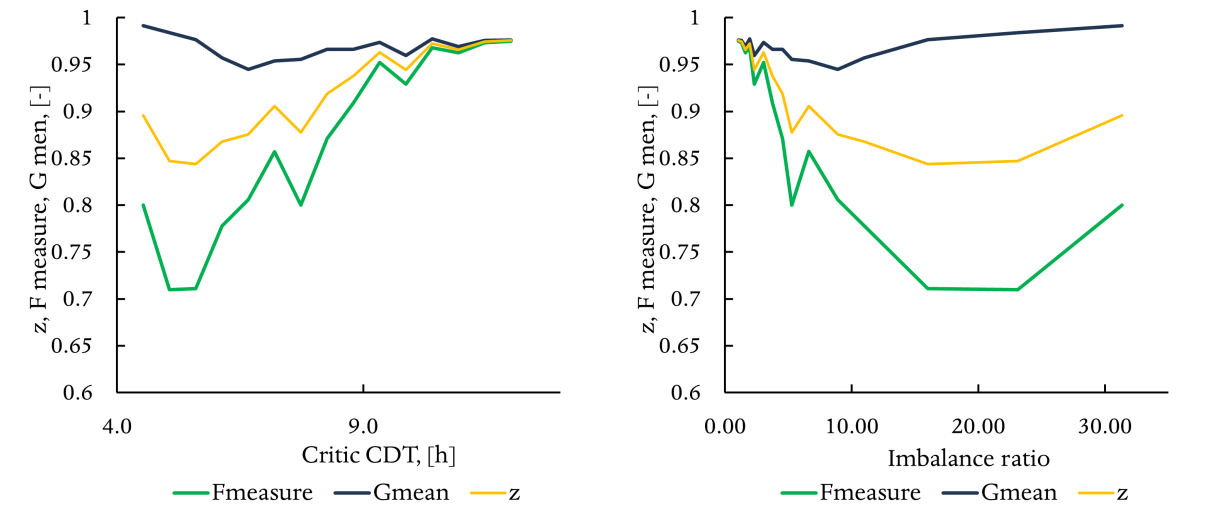
First of all, let's consider the global performance index NRMSE (figure 5.4a and 5.4b). As expected, the global performance of the composite model is usually worse than the performance of the ANN model. This has to be related to the False Positive (FP) misclassified points. In fact the SVM classifier is set with a cost matrix that make the classifier more sensitive with respect to high-risk points. Hence, the misclassified points are usually FP which worsen the performance for low-risk region and, due to the imbalance nature of the problem, the global performance. It's interesting to notice that for low imbalance ratio, or for high enough critic CDT, the performance of the composite and of the ANN tend to be the same. This can be related to the better performance of the classifier for higher critic CDT (figures 5.5a and 5.5b).

On the other hand, figures 5.6a and 5.6b show the performance for the high-risk region comparing ANN model and composite model. It's clear that, for higher enough imbalance



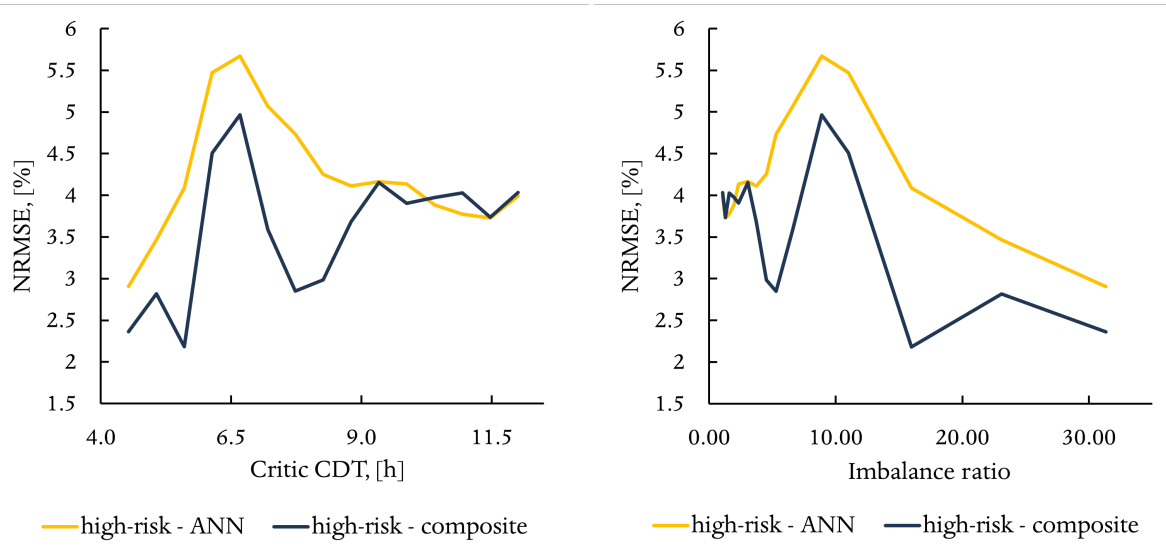
(a) (b)

Figure 5.4. Global NRMSE of both composite and ANN model vs critic CDT (left) and imbalance ratio (right).



(a) (b)

Figure 5.5. Testing performance indices for classifier vs critic CDT (left) and imbalance ratio (right).



(a) (b)

Figure 5.6. NRMSE of both composite and ANN model for high-risk region vs critic CDT (left) and imbalance ratio (right).

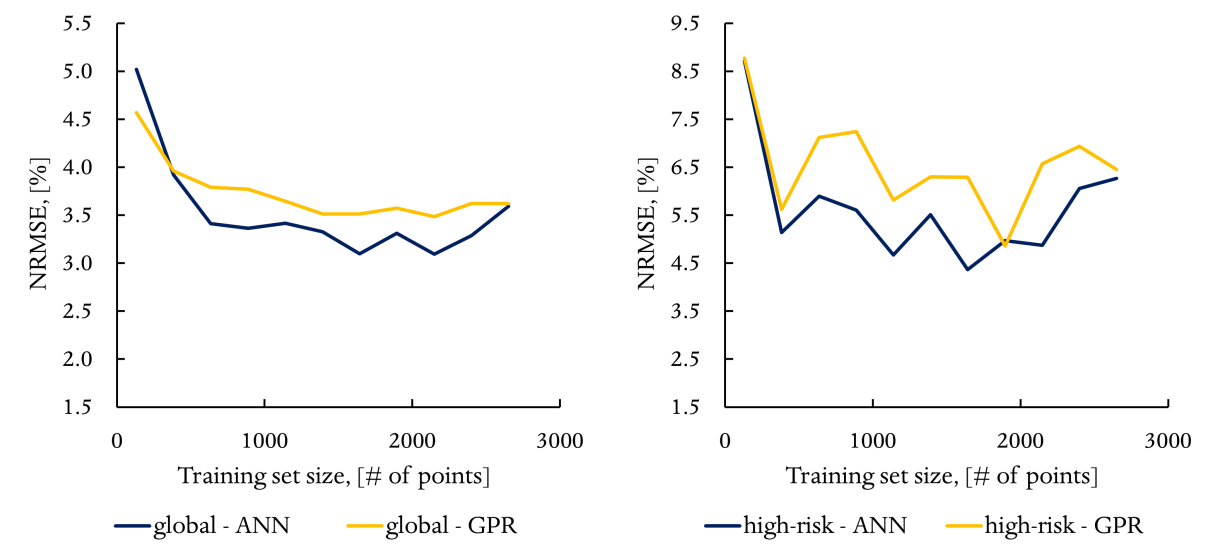
ratio, the composite model is better than the ANN model. Moreover, for low imbalance ratios, applying the composite model does not worsen the overall performance.

5.3 GPR model for regression prediction. Comparison with ANN

This section investigates the advantages to train an ANN instead of a GPR to be used as a single model or as sub-models for the composite model. Here, the performance advantages are investigated. In section 5.4 the advantages related to the computational time are described.

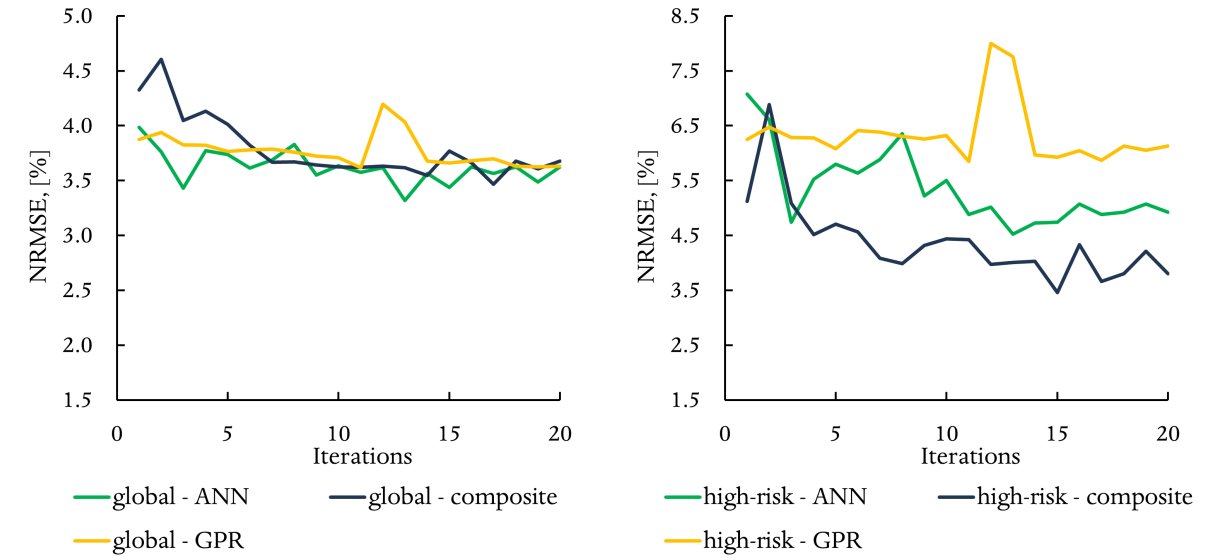
First of all a performance comparison between ANN model and GPR model is carried out by training both the models with training sets of different sizes. The main outcomes are shown in figures 5.7a and 5.7b. The former one shows the global NRMSE with respect to different training set sizes for both ANN and GPR. The latter shows the NRMSE evaluated for the high-risk region for both ANN and GPR models. The two plots clearly evidence that the ANN model can give higher performance with respect to GPR model.

Another analysis should evaluate the performance improvement of the GPR model for high-risk region after the application of the hybride adaptive sampling apporach. Figures 5.8a and 5.8b compare the performance improvement during the same hybrid adaptive sampling procedure applied in chapter 4. Considering the global performance, GPR model



(a) (b)

Figure 5.7. ANN and GPR models performance for global and high-risk region vs trainingset size.



(a) (b)

Figure 5.8. ANN, GPR and composite models performance for global and high-risk region during hybrid adaptive sampling vs trainingset size.

has generally similar performance to the ANN. As shown from in 5.8b, for high-risk region, GPR is not beneficial and its performance are worse than the one of ANN, and, for sure, than the one of composite model.

The main outcome is that, at least for the physics-based model, using ANN is more indicated than GPR due to its higher performance.

5.4 Computational time expenditure

In this section, an analysis related to the computational time both for offline development and for online prediction is given.

5.4.1 Computational time expenditure for Online prediction

One very important parameter for assessing the goodness of the proposed methodology is the computational time for the online prediction. The developed surrogate models, in fact, should be fast, assuring a reliable online prediction in the case of emergency shut-downs. Referring to the table 5.1, it's possible to see the computational times for the online prediction application for a global ANN model, for a composite model and for OLGA simulator. The computational expense for OLGA simulation is about 25 seconds. The composite model proposed takes for a prediction about 0.49 seconds while the faster ANN model takes less than a tenth of second. Hence, one can state that the composite surrogate model is about 50 times less computational demanding than the OLGA simulator, showing a very good behaviour as expected. It should be considered that in this application, the considered model of OLGA is very simple and in industrial applications OLGA models are based on more complex assets, hence procuring very higher computation expenses during simulations. This confirms the necessity of a surrogate model for online applications.

5.4.2 Computational time expenditure for offline development

Also the computational time expenditure for offline development should be considered as an important parameter for industrial application.

In table 5.1, a comparison between the average computational time for the creation of the training set of 400 and 600 points by LHS and by hybrid adaptive sampling approach is held. Of course, the hybrid approach, since the necessity to train SVM and GPR models every time 10 points are added, is the most computational time expensive as expected.

Table 5.1. Computational time expenditure for offline development and online prediction.

	Creation of the dataset (offline development)		Testing(online prediction)	
	Initial training data generation	Adaptive sampling		
Global ANN (400)	10000	N.A.	0.09	[s]
Global ANN (600_LHS)	15000	N.A.	0.09	[s]
Global ANN (600_adaptive)	10000	11500	0.09	[s]
Composite (400)	10000	N.A.	0.14	[s]
Composite (600_LHS)	15000	N.A.	0.49	[s]
Composite (600_hybrid)	10000	11500	0.49	[s]
OLGA	N.A.	N.A.	25.00	[s]

Chapter 6

Conclusion

In deep-water oil and gas industry, hydrates formation in subsea pipelines after unplanned shutdowns is one of the main risk sources that seriously threaten the safety and productivity of the facility. The operational policy required to be performed after unplanned shutdowns in order to secure the asset against the formation of the hydrates is significantly dependent on (and limited by) the CDT, which is the period between the onset of the shutdown and the time at which pipeline pressure and temperature reach to favourable conditions for hydrate formation. Existing methods for the CDT analysis are highly dependent on the use of very complex physics-based models that demand large computational time, which hinders their usage in an online environment, where information and decisions are required in fraction of seconds.

The current work proposes a novel methodology for the development of surrogate models capable of the online prediction, in a fast and accurate way, of the CDT for subsea pipeline after an unplanned shutdown. The surrogate modelling methodology is, innovatively, tailored on the basis of reliability perspective, by treating the CDT (the surrogate model output) as a risk index whose low values indicate high risk (operator may have not sufficient time to perform the operational policy to secure the asset against the hydrate formation) and high values indicating low risk (operator will have sufficient time).

Therefore, the method is based on the development of a hybrid model, which consists in a SVM technique that classifies the asset conditions after the shutdown into low or high risk levels, and two ANNs, each of which is responsible for the CDT prediction at low risk or high-risk subdomain of the asset conditions that are previously assigned by the SVM classifier. In order to overcome the problem of the rareness of the available input-output (asset conditions-CDT) training data from the high-risk region, an adaptive sampling procedure is proposed to collect more training samples from this region. This novel adaptive procedure combines two adaptive sampling techniques, which are a SVM-based and a GPR-based techniques. The first collects training data from the regions very close to the classification boundary between high and low risk subdomains in order to

improve classifier performance, while the second collects training points from the high-risk subdomain exploiting the characteristics of GPR model.

The methodology is validated by its application to two mathematical functions and to a case study involving a complex simulation model of a subsea pipeline of an Offshore Western African Asset (chapter 4). The outcomes show an improvement in the performance for high risk region in all the applications.

The results show:

- first, the very good capabilities of the hybrid surrogate modelling approach (i.e., SVM+2 ANNs) for enhancing the prediction of the CDT at high risk regions in front of the classical one (i.e., modelling the entire system with one global ANN). These capabilities are further confirmed when trying different sizes of the training datasets and different imbalance ratios (sections 5.1 and 5.2). Moreover, a further comparison between the use of global ANN and global GPR models is carried out (section 5.3). This showed a general better performance of ANN than of GPR models justifying the use of ANN in this this work.

- second, the adaptive sampling procedure shows very good ability to enhance the performance of the hybrid surrogate model, especially the SVM and the ANN model that predicts the CDT at high risk regions, which is of significant importance.

- third, the analysis of the computational time for the development of the model is carried out, mainly showing the huge computational time saving for online CDT predictions given by the surrogate models compared with OLGA simulator 5.4.

The proven capabilities of the proposed methodology confirm that it efficiently provides the requirements (rapidity and accuracy) for online prediction of the CDT in complex subsea asset, overcoming the limitations of physics-based models simulations which are computational demanding.

Future work lines include the uncertainty quantification of the surrogate model predictions, the sensitivity analyses in order to quantify the relative importance of the input variables (T_{in}, P_{out}, WC, Q_{oil}, GOR) with respect to the CDT and the investigation of effect of considering a multiclass risk classifier (e.g., high, medium, low risk) in the hybrid surrogate model structure.

Appendix A

Oil & Gas Production Assets

In this chapter the main features of a typical oil production system situated in deep water environment is described. Deep water oil or gas field is referred to that fields with a water depth between 200 and 2000 meters. First of all a brief physical description of reservoir and fluids is pointed out. After that, the most common production systems for offshore and deep water environment are illustrated.

A.1 Reservoir, fluids and wellbores

A reservoir is defined as a subsurface body of rock having sufficient porosity and permeability to store and transmit fluids [26]. When talking about reservoir's fluids is common to refer to oil, gas and water. Usually, the reservoir fluids is several kilometres below ground, while the depth of the column of oil itself is often less than 100 m. The areal extent is generally several square kilometres. The gas and oil are held in the pore spaces of the rock at high temperatures and pressures which can be estimated or measured by gauges at the bottom hole of wells (if any) [27].

Reservoir can be broadly classified depending on the composition of hydrocarbon mixture of the reservoir, initial reservoir pressure and temperature, pressure and temperature of the surface production. All these conditions are expressed in different types of diagrams called phase diagrams, usually pressure-temperature diagrams. In general, reservoirs are conveniently classified into basically two types, oil and gas, depending on the initial temperature T_i with respect to the critical point ($T < T_c$ oil reservoir, $T > T_c$ gas reservoir). Depending upon the initial pressure P_i, an oil reservoir can be classified as undersaturated oil reservoir ($P_i > P_b$, bubble point pressure of the reservoir fluid), saturated oil reservoir ($P_i = P_b$) and gas-cap or two-phase reservoir ($P_i < P_b$). Typically oil reservoirs can be also classified depending on physical and chemical properties: ordinary black oil, low-shrinkage crude oil, high-shrinkage (volatile) crude oil, near-critical crude oil.

Concerning gas reservoirs, they can be classified on the basis of their phase diagrams and the prevailing reservoir conditions. Retrograde gas-condensate, near-critical gas-condensate, wet gas and dry gas are the four common categories.

The hydrocarbon mixture may exist in either the gaseous or liquid state, depending on the reservoir and operating conditions to which they are subjected. In order to predict the correct behaviour of the reservoir fluids, empirical equations of state are commonly used as a quantitative tool. These equation must be calibrated through detailed compositional analyses of the hydrocarbon system and complete descriptions of the physical and critical properties of the mixture individual components. The former analysis are carried out by accurate laboratory studies of PVT and phase-equilibria behaviour. For the latter, many characteristic properties of these individual components (in other words, pure substances) have been measured and compiled over the years [28].

Usually the natural forces in the reservoir that displace hydrocarbons out of the reservoir into the wellbore and up to surface are called reservoir-drive mechanisms. The most effective is the water drive mechanism, the second most important is the gas drive mechanism both gas-cap drive and dissolved gas. Less important is gravity drainage mechanism. If the force provided by these natural mechanism is not enough one can use artificial lifting technologies or inject water, gas or more complex mixture in the reservoir in order to boost the production [26].

For mathematically controlling and predicting the production flow rate, a system of equations representing all the asset must be solved. The main equations are called Inflow Performance Relationship (IPR), the vertical flow performance (VFP) and the flowline performance.

Inflow performance relationship (IPR) relates the bottom hole flowing pressure P_{wf} to the production rate Q_o or Q_{gas}. Different kind of IPR are available in literature. The most common ones are the straight line inflow performance, the Vogel one and the Fetkovich one. When building the IPR, the reservoir pressure or static pressure is considered constant. Actually when producing, the static pressure decreases since the reservoir is depleted. As already stated, one can usually inject water (into a lower water-bearing section of the reservoir) or gas (into the upper gas section of the reservoir) in order to maintain the static pressure, to avoid pressure depletion and to sustain the production.

It's possible to displace the fluids inside a reservoir by means of a production wellbore, a drilled hole completed with cemented casings of different diameter (larger at the top, narrower at each successive section) which assure hydraulic connection between the reservoir and the producing tubing (a narrow tube inside the inner casing). The surface termination of a wellbore that incorporates facilities for installing casing hangers is called wellhead. The wellhead also incorporates a mean of hanging the production tubing and installing the Christmas tree and surface flow-control facilities in preparation for the

production phase of the well [26]. The Christmas tree is a an assembly of valves, spools, pressure gauges and chokes fitted to the wellhead of a completed well to control production. Usually in the offshore context, one refers to wet tree or dry tree if the surface termination of the wellbore is at the mud line or if it is collocated on the deck of the production platform. The mathematical equation that represents the flow inside the well is called vertical flow performance in which bottom flowing pressure is given by the summation of wellhead pressure, hydrostatic pressure and friction losses contribution.

Among the valves on the Christmas tree, the one used for controlling the production is called choke valve. If fully open, the valve assures the maximum possible flowing rate otherwise can limit the oil or gas production.

In the case of wet tree, the flow coming from the well needs to be collected by a gathering system and be sent to the production platform. In this case, flowline performance equations have to be used. Otherwise, if the wellhead is close to surface facilities as in the case of dry tree, the use of flowline performance equations is not necessary. In the case the choke valve is fully open, the separator pressure is a fundamental parameter since it can be adjusted in order to control the production.

In the offshore context, subsea production assets are very important since they allow to exploit very large fields whose production is collected by means of pipelines and conveyed at the floating production platform.

Pipelines are continuous and reliable means of transport which can adapt to a wide variety of environments, hence they well suit the challenge related to offshore production.

A.2 Production platforms

Once at the surface, production from the well is sent to a separator to be divided into its base components — oil, gas and water. The oil is dehydrated in a bulk oil treater before being sent to storage. It is then exported via a crude oil pipeline or a shuttle tanker to a refinery. The gas is also dehydrated before it is compressed and exported by pipeline. In some cases, injection wells are drilled to store gas safely in a reservoir for potential production in the future. The produced water is cleaned to required levels and then, depending on the location, may be discharged overboard, pumped into a disposal well or injected into the reservoir as a pressurizing system for further oil recovery [29].

There are different types of offshore production platforms. The type of facility depends on the location, water depth, climate and the facility's size and capabilities. Usually if water depth is less than 400 m, a fixed structure is used. This kind of facility consists of a jacket made up by pipe legs and tubular steel cross supported by piles driven into the seafloor and a deck for process facilities placed on the top of the jacket. For deeper scenarios, four floating systems are common:

- TLP – Tensioned Leg Platform which can support a drilling rig and production facilities. They are anchored to the seafloor by a mooring system made of tension legs, the "tendons" which limit vertical movements. TLPs can be used in up to 6,000 feet (about 1800 m) of water.

- Semi-submersible production platform which consists of a deck supported by four columns and connected underwater by four pontoons. Unlike TLPs, their floating hull uses a conventional lateral mooring system of steel cables to keep the platform in position and are connected to subsea wells via flowlines. They do not support a drilling rig.

- SPARS, Singol Point Reservoir Mooring which are moored to the seafloor and supported by a floating, hollow cylinder containing extra weight in the bottom, similar to a huge buoy. About 90 percent of the structure is underwater, so it has great stability in very deep waters — as much as 10,000 feet (about 3000 meters).

- FPSOs - Floating production storage and offloading units, which can operate in water depths up to 10,000 feet (about 3000 meters) and are best suited for milder climates or where there is limited pipeline systems to transport oil to shore. These ship-like vessels can usually process all of the oil or gas produced from a reservoir and store the oil until it can be offloaded to tankers for transportation. Subsea wells send production to the FPSO through lines called "risers," which are flexible enough to resist the heaving motion of the vessel above. Most vessels use mooring systems connected to a "turret." The turret is mounted to the hull and allows the vessel to rotate freely allowing the vessel's bow always to point into the winds and currents, minimizing the impact of those forces. FPSOs are either modified existing tankers or can be newly constructed. [29]

While the TPL and SPAR systems has direct access to the well as a dry tree is present, semi-sub platforms and FPSOs have not direct access to the well due to the presence of subsea assets [28].

Appendix B

Hydrates

The main topics of this chapter are related to flow assurance and to the physics of hydrates.

B.1 Flow assurance

Flow assurance refers to "the practice of identifying, quantifying, and mitigating of all the flow risks associated with offshore pipelines and subsea systems" [30] or again "flow assurance is an engineering analysis process that is used to ensure that hydrocarbon fluids are transmitted economically from the reservoir to the end user over the life of a project in any environment" [4]. When water, oil and gas are flowing together inside the pipeline some problems can arise: hydrate, wax and asphaltene can form and deposit, corrosion can happen, scales can form and severe slugging can be induced.

The prevention and the control of solids and deposit that can form in the asset due to certain pressures and temperatures is the main focus of flow assurance challenges. Moreover system reliability, thermal behaviour, solids and chemistry inhibitors are the main flow assurance concerns. The strategies adopted can be subdivided into:

- Thermodynamic control, when the main goal is to keep pressure and temperature of the entire system out of ranges favourable for solids formation and deposition.
- Kinetic control, when the strategy is to avoid the deposition of the solid.
- Mechanical control, when the solids are allowed to deposit but are periodically removed by pigging.

Flow assurance has become a key practice since fields development are involving long-distance tie-back and deepwater in which low temperature and high hydrostatic pressures are common. Obtaining reliable, manageable and profitable flow of hydrocarbon fluids from reservoir to the end user is the target of flow assurance.

For flow assurance problems project-specific strategies are required but usually the main issues associated with the flow assurance process are:

- Fluid characterization and flow property assessments, dependent on a careful analysis of samples from the wellbores. The key analyses are PVT properties such as phase composition, GOR, bubble point and wax properties. It's not possible to cover all application ranges needed through measures made on fluid samples. Thus, fluid models (equation o state) that can predict the fluid PVT behaviour are needed (SKR, PR, modified PR). Moreover the composition of the brine is an important factor in the hydrate prediction and scaling tendency. Hydrate stability curve is given, inhibitor dosing requirements are calculated and thermal-hydraulic models for the wells are developed.

- Steady-state hydraulic and thermal performance analyses, usually made by software like PIPESIM or HYSYS. Main objectives are the determination of relationship between flow rate and pressure drop along the flowline, temperature and pressure distribution, insulation, diameter.

- Transient flow hydraulic and thermal performance analyses, usually performed by OLGA and ProFES in order to simulate some scenarios as start-up and shutdown, emergent interruptions, blowdown and warm-up, ramp up/down, oil displacement, pigging/slugging.

- System design and operating philosophy for flow assurance issues.

B.2 Hydrates in nature

Natural gas hydrates, also known as clathrate hydrates, are crystalline compounds formed by the physical combination of water molecules and certain small molecules in hydrocarbons fluid such as methane, ethane, propane, nitrogen, carbon dioxide, and hydrogen sulphide [31]. Before talking about flow assurance problem given by the formation of gas hydrates along the asset, it's worth to mention that natural gas hydrates are a very huge energy resource for methane. This kind of unconventional natural gas resources have remained stable for millions of years in marine and permafrost environments and the estimation for hydrate resource is enormous compared to the conventional gas resources and shale gas [32]. Nowadays experimental programs have been performed and have shown that gas hydrates can be produced in the short term using conventional hydrocarbon recovery methods. However only in future large-scale methane production from gas hydrates would be seen. Moreover important economic considerations related to developing the infrastructure to collect and distribute the gas must be addressed because gas hydrates occur in remote

frontier marine and permafrost settings [33]. Finally it must be stressed that, since gas hydrates are one of the largest sources of carbon on earth and a potential source of clean carbon-based energy for humanity in the near future and since there's a huge tendency for the transition of the economy from oil based to natural gas based, the inclusion of natural gas hydrates as reserves will increase a lot in the next decades.

On the other hand the increasing demand for energy has moved the oil and gas industry to the extremes by increasing explorations in deep water and the Arctic. As already mentioned, this has significantly increased the risk of flow assurance problems. Pipelines, processing facilities, and transportation system can be blocked by hydrate thus the blockage cause reduce and stop the fluid flow. It means hydrate blockage can cause loss production and operation shutdown. [34]. This is why, when seen from a flow assurance perspective, hydrates represent a problem –often the largest problem to be dealt with in multi-phase flow-lines. [9].

B.3 Hydrates Formation

As already stated, natural gas hydrates are crystalline water structure with properties similar to those of ice and with low molecular weight guest molecules which give stability and allow hydrates to exist at much higher temperature than ice. Four components are needed to form gas hydrates: water, light hydrocarbon gases, low temperature and usually high pressure. One typical situation that can lead to hydrate formation, as it's investigated in this work, is the shut-down event when the temperature of the working fluid reaches the temperature of surrounding.

As already mentioned, hydrates will form for specific pressure and temperature values. These values are collected and shown in the hydrate formation and dissociation curves (also called equilibrium curve). The curves may be generated by a series of laboratory experiments or predicted using thermodynamic software such as MultiFlash or PVTSIM based on the composition of the hydrocarbon and aqueous phases in the system [31]. As one can see in fig 4.9, the P-T diagram divides the space into two regions:

- The stable hydrate region in which hydrates are thermodynamically stable and have the potential to form
- The hydrate free region in which hydrates do not form

To be underlined that for an asset, being in the stable hydrate region doesn't mean that there is for sure hydrate formation or that formed hydrates will cause operational difficulties. [31] In fact during steady state condition it may happen that small particles of hydrate form and merge into a mixture of oil and hydrate particles called hydrate slurry. [9]. Even though both kinetics of hydrate formation and hydrate dissociation are poorly understood

and field predictions result difficult, it's known that hydrates formation usually needs a delay time to form which is related to a subcooling temperature. Subcooling temperature is the difference between the stability temperature and the operating temperature at which hydrate starts to form. Hence a certain amount of subcooling is needed in order to have a sufficient hydrate formation rate and in general, a subcooling of 2.8 °C is sufficient to lead to hydrate formation. When hydrates starts to form, one can distinguish two mechanisms of plug formation: one in which hydrates slowly build up on the bottom of the pipe, the second in which hydrates form in the bulk fluid. As reported later, in order to control the hydrates, one way can be to keep the system outside the region in which hydrates are stable. During operation, for an oil production asset, the temperature are high enough for being outside the hydrate stable region, while during shut-down the subsea equipment temperature drops due to ambient sea temperature. [35] On the other hand shutdowns and start up operations are critical phases that can lead to hydrate formation. Blockages are unwanted because they lead to safety risks and to big revenue losses.

B.4 Working principle of Hydrates inhibitors

As already stated in section B.1, when one talks about control strategy, he can refer to thermodynamic and kinetic kind. Inhibitors in the same way are of two types: thermodynamic inhibitors (THIs) or low-dosage hydrate inhibitors (LDHIs). The former are chemical substances which act shifting towards lower temperatures for a pressure value the hydrate formation curve. Among thermodynamic inhibitors one can mention methanol, ethanol, glycols, sodium chloride and calcium chloride. In order to roughly estimate the temperature shift of the hydrate formation curve, Hammerschmidt [36] proposed the following formula:

$$\Delta T = \frac{KW}{M\,(100 - W)} \tag{B.1}$$

where:

- ΔT is temperature shift, hydrate depression, °C
- K is constant, defined in literature
- W is concentration of the inhibitor in weight percent in the aqueous phase;
- M is the molecular weight of the inhibitor divided by the molecular weight of water

Considering various inhibitors with the same weight fraction, salt has the most dramatic impact in hydrate stability curve. Accounting correctly for the produced brine salinity is important in designing a hydrate treatment plan. The smaller the molecular components, the lower hydrate formation temperature at the same pressure. Moreover, the higher the

weight fraction of the inhibitor, the greater the temperature shift is shown. The most common thermodynamic inhibitor is MEG, even though methanol, ethanol, other glycols (DEG and TEG) and salts are used. [31].

Hence, while thermodynamic hydrates inhibitors act changing the hydrate curve reducing temperature at which hydrate form, LDHIs does not change the thermodynamic properties but they prevent hydrate blockages at significantly lower concentrations than THIs (even if they are expensive and not recoverable). LDHIs include kinetic inhibitors and anti-agglomerates. The former work delaying hydrate crystal nucleation and/or growth typically 24-48 hours. They usually don't work if the temperature falls more than 10 °C below the subcooling temperature. Anti-agglomerates allow hydrate crystals to form but keep the particles small and well dispersed in the hydrocarbon fluid flow. [9]

B.5 Hydrate prevention

Hydrate prevention techniques for subsea systems include:

- Thermodynamic inhibitors;
- Low-dosage hydrate inhibitors (LDHIs);
- Low-pressure operation;
- Water removal;
- Insulation;
- Active heating.

The first two have been already mentioned and their selection is often based on economics, downstream process specifications, environmental issues and/or operator preferences [31].

Low-pressure operation refers to the process of maintaining a system pressure that is lower than the pressure corresponding to the ambient temperature based on the hydrate dissociation curve. Usually it's not possible maintain low pressure values in wellbores and flowlines. When possible and generally for export lines, dehydration process removes water to such a content that hydrate formation will not occur. Insulation helps in maintaining temperatures above hydrate formation conditions. Insulation however is not applied to gas production systems because of the low heat capacity of gas and due to Joule-Thompson effect. Another way to maintain the temperature is providing thermal energy to the flow by electrical heating and hot fluid circulation in a pipe bundle [31].

B.6 Hydrate Remediation

When hydrate form and blockage of the line occurs, hydrate plug are very difficult to remove. It takes a large amount of energy to dissociate the hydrate and heat transfer through the hydrate phase is slow. Moreover safety is a concern due to gas release from hydrate plugs. Hydrate dissociation is highly endothermic and Joule-Thomson cooling effect due to expanding gas is also possible. Hence additional hydrates and/or ice can form during dissociation process. [31]

Although the asset is designed in order not to have any blockage due to hydrate , a hydrate blockage remediation plan should be developed for a subsea system. Hydrate remediation techniques can be summarized in the following list [31]:

- Depressurization from two sides or one side, the most common technique, by reducing pressure below the hydrate formation pressure at ambient temperature. This will cause the hydrate to become thermodynamically unstable. The process of depressurization of the flowlines is known as blowdown. Due to its high complexity and high operational time it can lead to big revenue losses.
- Thermodynamic inhibitors by direct contact can melt blockages.
- Active heating providing heat and increasing the temperature can dissociate a blockage.
- Mechanical methods such as drilling, pigging, and scraping but generally not recommended.
- Replace the pipeline segment.

B.7 Hydrate formation during shut-down's situations

In flow assurance process, in which the engineering analysis process of developing a design and operating guidelines for the control of solids deposition in subsea systems must be carried out, one of the main issues addressed is the performance analysis during a shut-down. When one talks about shut-downs from a steady-state condition, he can distinguish between planned shutdowns and unplanned shutdowns. In general the two are the same but for a planned one, hydrate inhibitor can be injected into the system prior to shut-down. In this way the product fluid is in safe condition and no other operation needs to be done before the start-up. When a shut-down happens, the flowline temperature will decrease because of the heat transfer from the system surrounding water.

As described in 2 when an emergency shutdown happens, the operator should follow a series of instruction in order to preserve the line from hydrates formation. It's to be

underlined that during the shutdown, the extent to which the gas, oil, and water partition limits the growth of hydrates. In fact an oil layer between water and gas slows down transport of the hydrate-forming molecules. Moreover hydrates usually form in a thin layer between oil and water layers which inhibits further contact between the water and gas molecules. Hence during shut-downs blockages are not probable but when the well is restarted or dead oil is circulated, agitation leads to a good mixing of the subcooled water and gas hence blockage can form. [31]

Appendix C

OLGA simulator

OLGA is the industry standard tool for transient simulation of multiphase petroleum production [6].OLGA is a one dimensional code simulator developed to simulate multiphase flow in pipelines and pipelines networks, with processing equipment included. [7]. OLGA which has been gone through continuous research since 1980 when it was firstly developed by the Institute for Energy Research (IFE) – Norway. Though years SINTEF and IFE, supported by Statoil developed and validate the multiphase flow correlations of OLGA by data collected from large scale flow loops. Since 1990 OLGA software has been commercially available and it has become a production engineering tool exploited by operators throughout field life. OLGA in fact is used for networks of wells, flowlines, pipelines and process equipment. Since its transient capabilities, OLGA dramatically increases the range of applicability compared with steady state simulators. OLGA in particular is a three-fluid (oil, gas and water) model [7].

The model deals with five mass transport equation, (eq. C.1) for five mass fields which are the mass of gas phase, mass of oil in the liquid layers, mass of water in the liquid layers, mass of oil droplets in gas, and mass of water droplets in gas; three equations for momentum conservation (eq. C.2), one for energy conservation (eq. C.4) and one equation for volume conservation (eq. C.3). The equations are solved using the finite volume method and semi-implicit time integration.

Mass transport equation:

$$\partial_t m_i + \partial_z \left(m_i U_i \right) = \sum_j \Psi_{ji} + G_i \tag{C.1}$$

Where m_i is the mass field traveling at velocity U_i, ∂_t and ∂_z denote differentiation in time and space, Φ_{ji} denotes the rate of mass transfer between the j_{th} and i_{th} mass field, and G_i denotes any mass source/sink.

Momentum balance Equations:

$$\partial_t \left(m_i U_i\right) + \partial_z \left(m_i U_i^2\right) = m_i g \cos\varphi + P_i + G_i U_i + \sum_j \left(\Psi_{ji}{}^{+} U_i - \Psi_{ji}{}^{-} U_i\right) + \\ + \sum_j F_{ji}^i \left(U_j - U_i\right) - F_i^w U_i \tag{C.2}$$

Where $m_i U_i$ is the momentum field, g is the acceleration of gravity, ϕ is the pipe angle relative to the gravitational vector, P_i is the pressure force, $G_i U_i$ is the momentum contribution corresponding to the mass source/sink G_i, Φ_{ji} are friction forces between the i_{th} and j_{th} mass field, and F_w denotes the wall friction. F_{ji}^i denotes momentum contributions corresponding to the mass transfer between the j_{th} and i_{th} mass field. In the equation C.2 Φ_{ji}^{+} accounts for a net contribution from mass field i to j while Φ_{ji}^{-} accounts for a net contribution from mass field j to i.

Conservation of volume:

$$\Sigma_L \left(\frac{m_L}{\rho_L^2} \frac{d'\rho_L}{\partial' P}\right) \partial_t P + \sum_L \left(\frac{m_L}{\rho_L^2} \frac{\partial'\rho_L}{d'T}\right) \partial_t T + \sum_L \frac{1}{\rho_L} \left(\partial_Z \left(m_L U_L\right) + G_L\right) = 0 \tag{C.3}$$

Where P is the pressure, ρ denotes the density while L denotes the existing phases.

Energy Balance:

$$\partial_t \left(m_i E_i\right) + \partial_z \left(m_i U_i H_i\right) = \int_i + Q_i + \sum_j T_{ij} E_j \tag{C.4}$$

Where E_i denotes the field energy, H_i denotes the field enthalpy, s denotes enthalpy source/link, Q is the heat flux through the pipe wall, and T_{ij} models the energy transfer between fields.

These equations, which are the foundations of the simulator are linearized while pressure and temperatures are de-coupled that is pressure is computed based on the previous temperature. In order to close the system of equations, fluid properties, boundary and initials conditions are required.

Flow regimes play a fundamental role in OLGA since they determine some quantities essential for the equation given before. The flow regimes that OLGA recognize are stratified flow, annular flow, dispersed bubble flow and slug flow.

OLGA is used from conceptual studies to the support of operations, for simulating operating procedure and training the operators. Moreover OLGA can be embedded in on-line systems and exploited for monitoring, forecasting and planning of operations. It can interface with all major dynamic process simulators allowing the making of integrated engineering simulators in an high fidelity model. Hence design and engineering, management of the risk, safety analysis and control of operational procedures and limitations like the study of emergency procedures and contingency plans are some of the challenges OLGA can tackle. Eventually among all the operational situations OLGA can deal with

(change in production, process equipment, pipeline pigging etc.), the most important for the scope of this work are the ones related to flow assurance. In particular one can say that OLGA is well suited for dealing with pipeline shut-down events and hydrates control [37].

C.1 Fluid models

One of the most important part when dealing with oil and gas simulators is of course the fluid model to be used. OLGA proposes four different way to describe fluid properties (PVT). The first one is the so called "lookup table" where the fluid properties tabulated at given pressures and temperature are read from a PVT-file. This is the least computational demanding method and it's suitable when the fluid composition does not change much along the flowpaths or over time. The second one is the "compositional method" in which thermodynamic equilibrium calculations are calculated using a proper PVT package called Multiflash. Mass equations are solved for each component in each phase, enabling simulation of scenarios with a high level of detail and accuracy. The main drawback is that the compositional tracking is the most computational demanding method. Compositional method should be considered when the fluid composition does not change much along the flowpaths or over time. Than the "blackoil" method in which the fluid properties are computed based on blackoil correlations which is useful when limited information about production fluid is available. The last method is called 'Single component' which enables tracking of a single component, for example H2O or CO2, that crosses the saturation line in time or space in a pipeline. It's possible to infer that OLGA simulations, using fluid tables, will in many cases give satisfactorily accurate results [6].

Usually production engineers use software like PVTsim for simulating fluid properties. PVTsim is developed for reservoir engineers, flow assurance specialists, PVT lab engineers and process engineers. It's based on an extensive data collected over a period of more 25 years and it enables to build reliable fluid models. Thanks to the robust regression algorithms PVTsim incorporates, matching fluid properties and experimental data is possible. For this work '3phase.tab' file provided by OLGA for the example cases is chosen.

Appendix D

OLGA model specifications

In this chapter a detailed description about the specification adopted for this work for the OLGA model is carried out.

D.1 Modelling of the contextual pipeline-riser geometry

A FLOWPATH component has been used in order to simulate the pipeline. The geometry has been chosen in order to replicate a possible seabed line and the riser path (figure D.1). At the bottom a MASS SOURCE component just after the CLOSE NODE component is placed with the function of simulate the manifold from which the fluids depart at a depth of 1000 m. The flowline is 15 km long with a slight downward slope. The riser sections start from the deepest point which is 1200 m deep. The riser allows the fluids to move from 1200 m under the sea level towards the FPSO whose choke valve is situated 30 meters above the sea level. Hence the total pipeline length, which has been divided into 138 numerical sections of different length for computational reasons, is of 16.3 km. Diameter of the flowline has been chosen of 8 inches while the roughness has been set at $5\ 10^{-5}$ m which are quite realistic values. The pipe wall has been built using five layers of materials having different thickness and properties chosen in order to emulate the thermal behaviour of the wall of a flexible riser [38] [39] [30] [40]. No distinctions between riser and flowline pipe walls were made. Only two main valves have been added to the flowline. The first valve is positioned just after the mass source component, at the beginning of the flowline and simulates a subsea choke valve for the manifold. The second valve is placed at the FPSO , at the ending of the flowline. Both the valves has the same diameter of 8 inches and they are fully open before the shutdown. When the shutdown happens they are simultaneously closed within one minute. Eventually, a PRESSURE NODE which represents the FPSO is linked to the flowpath at the end of the riser and placed at 30 m

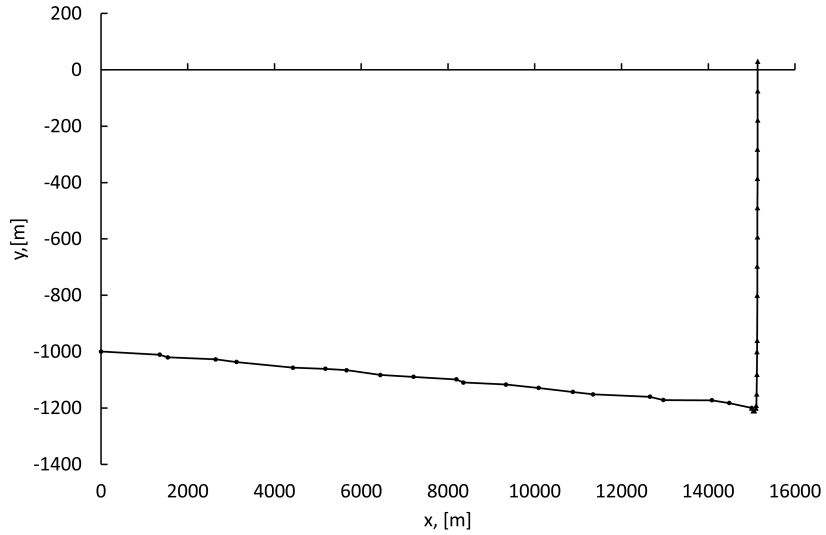


Figure D.1. Pipeline Geometry

above the sea level.

D.2 Setting the fluid model

One of the most important part when dealing with oil and gas simulators is of course the fluid model to be used. OLGA proposes four different way to describe fluid properties (PVT). The first one is the so called "lookup table" where the fluid properties tabulated at given pressures and temperature are read from a PVT-file. This is the least computational demanding method and it's suitable when the fluid composition does not change much along the flowpaths or over time. The second one is the "compositional method" in which thermodynamic equilibrium calculations are calculated using a proper PVT package called Multiflash. Mass equations are solved for each component in each phase, enabling simulation of scenarios with a high level of detail and accuracy. The main drawback is that the compositional tracking is the most computational demanding method. Compositional method should be considered when the fluid composition does not change much along the flowpaths or over time. Than the "blackoil" method in which the fluid properties are computed based on blackoil correlations which is useful when limited information about production fluid is available. The last method is called 'Single component' which enables tracking of a single component, for example H_2O or CO_2, that crosses the saturation line in time or space in a pipeline. It's possible to infer that OLGA simulations, using fluid tables, will in many cases give satisfactorily accurate results [6].

In this work a .tab file has been used for the simulations. Usually production engineers use software like PVTsim for simulating fluid properties. PVTsim has been developed for reservoir engineers, flow assurance specialists, PVT lab engineers and process engineers.

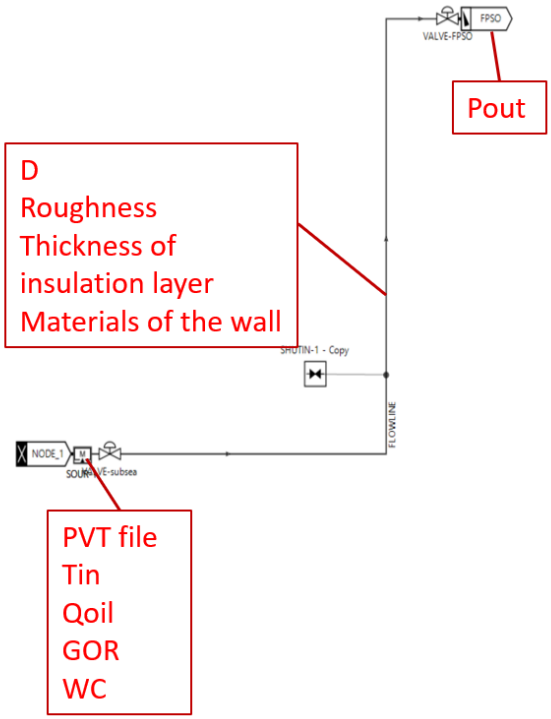


Figure D.2. OLGA model. In red the main variables to be specified for closing the equations' system and run the simulator.

It's based on an extensive data collected over a period of more 25 years and it enables to build reliable fluid models. Thanks to the robust regression algorithms PVTsim incorporates, matching fluid properties and experimental data is possible. For this work '3phase.tab' file provided by OLGA for the example cases has been chosen. The .tab file provides the properties of 900 points which are the binary combinations of pressure and temperature values inside two ranges: 1 – 200 bar for pressure and -10 – 100 °C for temperature. Starting from these points OLGA performs regression and interpolation for the desired value also if the point is outside the ranges. 3phase.tab models a fictious fluid which is fully determined inside the region of three phase flow (oil, gas and water are considered).

D.3 Boundary conditions

In order to simulate the heat transfer between the environment and the fluid flowing in the pipe, heat boundary conditions must be applied. This was done by adding the HEATTRANSFER keyword. For the heat transfer between the ocean water and the pipe, a table for describing the variation of temperature vs ocean depth has been provided (figure D.3 [41]. Moreover the ambient heat transfer coefficient of sea water is computed

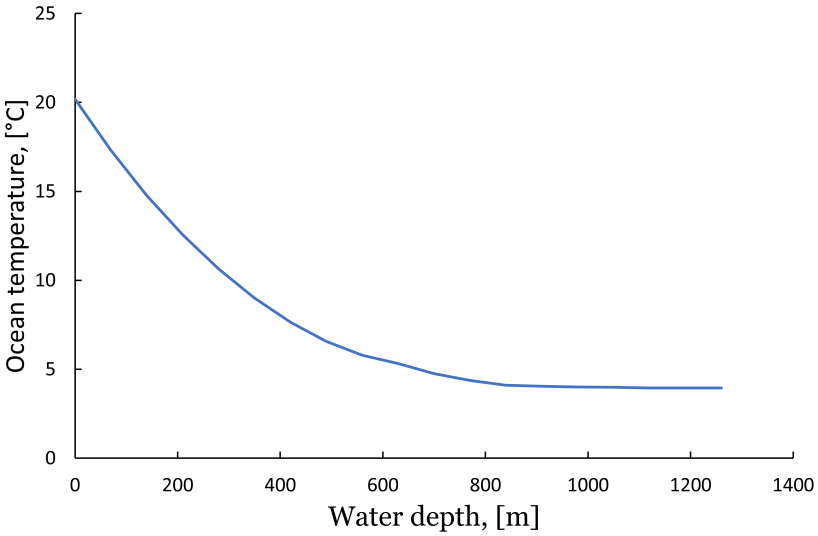


Figure D.3. Water temperature vs water depth.

by OLGA. For the last part of the riser, the one exposed to air, a fixed air temperature has been fixed to 35°C and, as for water, the heat transfer coefficient for this zone is computed by OLGA. Default properties for air and water given by OLGA are given in [6].

Afterwards the missing boundary conditions must be specified. In particular MASS SOURCE element, which is placed just after the initial CLOSE NODE represents the starting point for the flow and it asks for figures to characterize the flow. In order to describe the flow three parameter are specified: the standard flow rate of oil phase STDFLOWRATE expressed in STB/d, the GOR which is the gas-oil ratio at standard condition in scf/STB and the water cut WC defined as the water volume fraction in oil/water mixture. Moreover a value for pressure in the PRESSURE NODE is needed in order to fully constrain the model.

The key points of the project is to assess to possibility to form hydrates. In order to check if conditions for hydrate formation are met, the variable DTHYD is added as output variable:

$$DTHYD = T_{hydrate\ curve\ at\ operative\ pressure} - T_{\ operative}$$

DTHYD gives back the distance between the hydrate formation curve temperature and the operative temperature for the operative pressure. If this value is greater than zero, we are inside the thermodynamically stable region for hydrate formation. In order to activate and use the DTHYD variable, the HYDRATECHECK keyword is recalled under FA-model for the FLOWPATH keyword and a pressure-temperature table representing hydrate curve formation diagram is added in OLGA project. PVTsim can be used for precisely creating the curve given the fluid characteristics. For this work a curve taken

from literature has been applied to OLGA model (figure 4.9).

D.4 Simulation options

For this work a 30 hours simulation has been chosen. In the first 10 hours the model is kept steady state simulating the normal operating condition of the pipeline. After ten-hours the shut-down event is simulated: the two valves implemented in the model are closed in one minute and the asset becomes isolated. The following 20 hours are used to assess the CDT of the asset after a shut-down. The other possibility was the use of the steady-state pre-processor which is incorporated in OLGA and could avoid the first 10 hours of simulation. Nevertheless it has been chosen to include in the simulation also the first ten hours with the purpose to check if the boundary conditions are physically possible or give strange values. In particular the inlet pressure is monitored in the steady state part of the simulation. If Pin values higher than 150 bar are achieved, then the boundary values are believed to bring to unphysical conditions.

Eventually the SHUT-IN keyword has been added to the FLOWPATH FA-models and activated during shut-down event. The purpose is to enable more robust simulation when the pressure and flow in the pipeline is dominated by hydrostatic head, phase transition and thermal effects [6] as in this case.

Appendix E

Artificial Intelligence & Machine Learning

Machine learning is a subject that can be arranged under the much more bigger subject of artificial intelligence. An introduction about these two terms is given.

E.1 Artificial intelligence

Artificial intelligence as a science was born in the middle of twentieth century and it relies on many past scientific achievements in computer science. In the 1930's Alan Turing, British mathematician and logician, a pioneer in the fields of informatics and artificial intelligence science [42], Kurt Gödel, Austrian mathematician and Alonzo Church, American mathematician and logician, laid important foundations for logic and theoretical computer science. In the 1940s, based on results from neuroscience, the first mathematical model for neural networks was designed. However, computers at that time lacked sufficient power to simulate even the simplest brains. In 1950 Turing defined the Turing Test, a tool for defining a machine as intelligent. Even if this test was very interesting philosophically, for practical AI which deals with problem solving, this wasn't a very relevant test. In 1956, John McCarthy, organized a conference in Dartmouth College and the name "Artificial Intelligence" was firstly introduced. That is considered the formal birth of AI. In that occasion, thanks to programmable computers, Logic Theorist, a theorem prover, and the new language LISP were firstly introduced and proved the possibility to process symbols. In 1958 Rosemblatt, American psycologist, introduced the perceptron which is considered the ancestor of the modern artificial neural network. [37]

During 60s, 70s and 80s, a lot of improvements were achieved in symbols processing. AI programs able to play chess r algorithms to prove theorems and to translate text were devised. Moreover in the middle of 80s, back-propagation learning algorithm which was first discovered by Bryson and Ho in 1969 was reinvented [43]. Only later, when

computers were becoming sufficiently powerful, mathematically modelled neural networks gained importance among computer scientists, physicists and cognitive scientists. Bayesian networks, fuzzy logic, hybrid systems, data mining, distributed artificial intelligence and deep learning are example of development of artificial intelligence over the years. [37]

But what is actually Artificial intelligence? McCarthy, already cited before as one of the pioneers of AI, in the 1955 stated: "The goal of AI is to develop machines that behaves as though they were intelligent". This definition showed some weaknesses and over the years the definition of AI was updated. Elaine Rich, American computer scientists, gave the following definition of artificial intelligence in 1983: "Artificial intelligence is the study of how to make computers do things at which, at the moment, people are better". Or again, as reported by encyclopaedia Britannica: " Artificial intelligence (AI) is the ability of a digital computer to perform tasks commonly associated with intelligent beings." The processes characteristics of humans can be referred to the ability to reason, discover meaning, generalize, or learn from past experience [44]. Nowadays AI is based on interesting discoveries from such diverse fields as logic, operations research, statistics, control engineering, image processing, linguistics, philosophy, psychology and neurobiology. AI science offers a huge range of tools to be used in a manageable way for accomplishing very different tasks. It's possible to list some mature applications of AI: Robotic Vehicles, Speech Recognition,, Autonomous Planning and Scheduling, Game Playing, Spam-Fighting, Logistic Planning, Robotics, Machine Translation. Most of the AI tools are well developed and are available as finished software libraries. For any AI project, as for this thesis work, the selection of the right tool is a mandatory and exciting task for the AI user. [37]

E.2 Machine Learning

Machine learning (ML) is a subfield of artificial intelligence and cognitive science [45]. ML represents a category of techniques, developed with the same tasks of artificial intelligence such as recognition, diagnosis, robot control, planning, prediction etc. The machine learns and adapts itself changing its structure based on the data given as inputs or in response to external information in order to improve its future performance. This gives computers the ability to learn without being explicitly programmed. Many techniques in machine learning derive form the efforts of scientists to make more precise their theories of animal and human learning through computational models. There are several reasons for building machines in such a way:

- They can help in understanding how animals and humans learn.
- Some tasks cannot be defined well except by example; machines can adapt their

internal structure to produce correct output.

- Possibility to find important relationships and correlations among large piles of data. Too large knowledge for humans could be captured better by machines.
- Machine learning methods can help improving existing machine designs on the field, based on the characteristics of the working environment.
- If the environment changes over time, machine can adapt themselves without the need of constant redesign [46].

Machine Learning techniques can be subdivide in three main branches:

- Supervised learning which refers to a class of algorithms that determines a predictive model using points with known outcomes. It's possible to distinguish between two categories: classification whose aim is to find the classes where to put the inputs and regression where the output is a real value.
- Unsupervised learning which refers to a category of algorithms that, given a dataset, learns useful properties and patterns of the dataset which is not labelled and with minimal human supervision.
- Reinforcement learning which uses a no-labelled dataset and whose model learns over time interacting with its environment.

Among all the techniques of machine learning the focus of this project will be mainly on Artificial Neural Networks, Gaussian Process regression, Support Vector machines and K-means clustering.

Appendix F

Artificial Neural Networks

Artificial Neural Network (ANN) models are part of the broader field of machine learning which is a portion of the artificial intelligence subject [43]. ANNs provide a general, practical robust approach to approximate real-valued, discrete-valued and vector-valued functions from examples. Algorithms such as backpropagation use gradient descent to tune network parameters to best fit a training set of input-output pairs. ANN learning is robust to errors in the training data and has been successfully applied to problems such as interpreting visual scenes, speech recognition and learning robot control strategies [47].

These models consist of non-linear elements interconnected through adjustable weights and their architecture reminds the one of a human brain (which learns from the sensitive world, elaborates the information through an internal net of neurons and gives an interpretation of the outside). Since fifties scientists tried to model algorithms whose architecture was similar to the one of the human brain. The first main achievement in this sense was the Rosenblatt's perceptron algorithm presented in 1958. However only the availability and increasing computational power achieved at the beginning of 21^{th} century led to huge improvements in the practical applications of theories. Of course having computer models which can learn and act as humans is a very difficult task even for today's computers. It's to be underlined that, however, some tasks are well accomplished by machines based on ANN. Image recognition, self-driving cars, and game playing are typical examples of fields in which ANN helps in the creation of models that can act and accomplish tasks even better than humans. The greatness of ANN is that they can theoretically learn any mathematical function with a sufficient training set. With the increasing amount of data available and with the increasing power of computers, no one can actually know which goals and tasks will be accomplished by ANN (and more generally by ML). [48]

F.1 Similarities with biological neural network

As already stated, ANN models are remotely related to biological neural networks. First of all even if ANN does not approach the complexity of the brain, there are two keys similarities between biological and artificial neural networks: first, the single elements which constitute both the systems are simple computational devices which are deeply interconnected (even if biological neurons are much more complex). Second, the interconnections between these elements characterize the function of the network. Moreover it has to be noticed that even if biological neurons are very slow when compared to electrical circuits ($10^{-3}s$ compared to $10^{-3}s$), biological brain can be much faster in performing many tasks thanks to the massively parallel structure of biological neural networks that ANN can't implement as well as human brain.

A brief description of the biological brain and neural network is here given for sake of clarity. First of all, the brain is built by about 10^{11} elements and each element is highly connected to other elements through about 10^4 connections. These elements are called neurons and it's possible to consider, for the scope of this project, each neuron simply built three principal components:

- The dendrites, tree-like receptive networks of nerve fiber that carry electrical signals into the cell body.
- The cell body, sums and treats the incoming signals.
- The axon, fiber which brings signal from the cell body out to other neurons

A synapse is called the point of contact between an axon of one cell and a dendrite of another cell. The function of the neural network is given by the arrangement of neurons and the strength of individual synapsis which are determined by a complex chemical process. The brain structure is not fixed but it changes over time: the neural networks already present at the birth are later modified thanks to, for example, learning process. Moreover they continue to change throughout life increasing or decreasing the importance of various synapsis [49]. In the follow a description of artificial neurons together with the function related to is given starting from the single-input neuron.

F.1.1 Single-input neuron

A general neuron can be seen as a mathematical model in which given an input, an output is generated. Figure F.1 shows the schematic of the neuron model: p is a scalar input which is multiplied by the scalar weight w, 1 is the second input which is multiplied by the scalar bias b (the bias is much like a weight). n is the sum of wp and b and it's passed to the transfer function f which produces the scalar neuron output as result. If

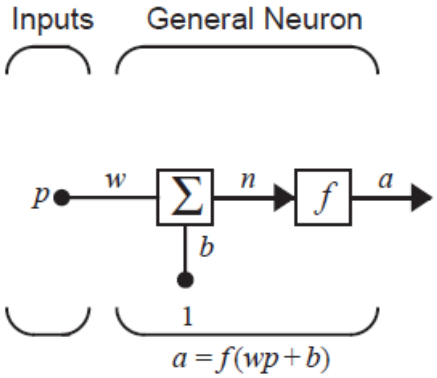


Figure F.1. Single input neuron [49]

the parallelism between biology and mathematics holds, the weight corresponds to the strength of a synapse, the cell body is represented by the summation and by the transfer function (called also activation function), while the signal on the axon is represented by the neuron output [49].

F.2 Transfer function

The transfer function may be linear or non-linear function of n. The most common are:

1. 1. The hard limit transfer function sets the output of the neuron to 0 if the function argument is less than 0, or 1 if its argument is greater than or equal to 0.
2. 2. The linear transfer function has an output equal to its input: $a = n$.
3. The log-sigmoid transfer function squashes the output into the range 0 to 1 according to the expression:
$$a = \frac{1}{1 + e^{-n}}$$
4. The hyperbolic tangent sigmoid transfer function has the output in the range -1 to +1. The mathematical function is:
$$a = \frac{e^{n} - e^{-n}}{e^{n} + e^{-n}}$$

F.3 Multiple-input neuron

Typically, a neuron has more than one input. The individual inputs $p_1, p_2, \ldots, p_R$ are each weighted by corresponding elements $w_{1,1}, w_{1,2}, w_{1,3}, \ldots, w_R$ of the weight matrix $\mathbf{W}$.

The neuron has a bias which is summed to the weighted inputs form for the net input n:

$$n = w_{1,1}p_1 + w_{1,2}p_2 + w_{1,3}p_3 + \ldots + w_R p_R + b \tag{F.1}$$

The expression can be written in matrix form:

$$n = \mathbf{W}\mathbf{p} + b \tag{F.2}$$

where the matrix $\mathbf{W}$ for the single neuron case has only one row. In the same matrix form, the neuron output can be written as [49]:

$$a = f\left(\mathbf{W}\mathbf{p} + b\right) \tag{F.3}$$

In the weight matrix, the first index indicates the particular neuron destination for that weight. The second index indicates the source of the signal fed to the neuron. For example, the indices in $w_{1,2}$ say that this weight represents the connection to the first neuron from the second source. A typical weight matrix is like:

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

F.4 Classic neural network architecture

F.4.1 Single layer of neurons

Let's consider a single-layer network of S neurons. Each element of the R input vectors $\mathbf{p}$ is connected to each neuron through the weight matrix $\mathbf{W}$. Each neuron has a bias b, a sum, a transfer function f and an output a_i. Taken together, the outputs form the output vector. It is common for the numbers of input to a layer to be different from the number of neurons [49].

F.4.2 Multiple layers of neurons

Considering a network with several layers, each layer has its own weight matrix, its own bias vector, a net input vector and an output vector. In order to identify the layers, a superscript is appended. Usually the last layer is called output layer because it is the network output while the first one is the input layer. The other layers are called hidden layers.

Multilayer networks are more powerful than single-layer networks. For instance, a two-layer network having a sigmoid first layer and a linear second layer can be trained

to approximate most functions arbitrarily well while single-layer networks cannot do this. The number of inputs to the network and the number of outputs from the network are usually defined by external problem specifications. Hence the number of the input neurons to the network is equal to the number of external variables which are inputs to the problem. Similarly, the number of neurons in the output layer is the number of outputs of the problem. Thus, the architecture of a single-layer network is almost completely defined by problem specifications. In case of more than one layer, the external problem does not tell directly the number of neurons required in the hidden layers. A specific study has to be dedicated to this decision. As for the number of layers, most practical neural networks have just two or three layers. Regarding the bias, one can choose neurons with or without biases. The bias gives the network an extra variable, in fact, these networks are more powerful than those without bias [49].

F.5 Learning algorithms

Talking about machine learning, one refers to the ability of the machine to learn through experience from some input data. The core of the learning process in ANNs is the so called learning algorithm or law. There are several different classes of network learning laws. The one discussed here is the performance learning. In such a class the performance learning the network parameters are adjusted to optimize the performance of the network [49]. Hence, in order to optimize the performance of a model, at least one quantitative measure must be designed in order to evaluate the performance of the model. This parameter usually is computed using a testing set, different from the training set used to make the model learning. An example of performance parameter for a trained model can be the Mean Square Error (MSE) which is given by:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(\hat{\mathbf{y}} - \mathbf{y})^2 \tag{F.4}$$

Where $\hat{y}$ is the vector of the outputs of the model, y is the vector of the expected output values and m is the number of elements in the testing set. It's possible to state that a good learning algorithm must be able to improve the weight W in a way that reduces MSE_{test} when the algorithm is allowed to gain experience by observing a training set $(X(train); y(train))$. One intuitive way of doing this is just by minimizing the mean square error on the training set, MSE_{train}.

F.5.1 Optimization Algorithms

Any learning algorithm is based on an optimization method. Optimizing a problem $F(\mathbf{x})$ means finding the vector $\mathbf{x}$ which minimizes the given problem described by $F(\mathbf{x})$. Here

two main optimization algorithms are discussed: the steepest descend and the Newton's method. Both the two algorithms are iterative. This means that the algorithms begin from an initial guess x_0 and then update the guess in stages according to an equation of the form:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \tag{F.5}$$

or

$$\Delta\mathbf{x}_k = (\mathbf{x}_{k+1} - \mathbf{x}_k) = \alpha_k \mathbf{p}_k \tag{F.6}$$

Where the vector $\mathbf{p}_k$ represents a search direction, and the positive scalar α_k is the learning rate, which determines the length of the step. The optimization algorithms here described are different by the choice of the search direction and learning rate.

Steepest Descend

Steepest descent has the advantage that it is very simple, requiring calculation only of the gradient. It is also guaranteed to converge to a stationary point if the learning rate is small enough. The disadvantage of steepest descent is that training times are generally longer than for other algorithms.

For the steepest descend optimization algorithm is to choose the direction pk in order to have, for a small learning rate α_k, a decrease in the performance index $F(\mathbf{x})$. Let's consider the first-order Taylor's series expansion of the performance index about the old guess x_k:

$$F(\mathbf{x}_{k+1}) = F(\mathbf{x}_k + \Delta\mathbf{x}_k) \approx F(\mathbf{x}_k) + \mathbf{g}_k^T \Delta\mathbf{x}_k \tag{F.7}$$

where $\mathbf{g}_k$ is the gradient evaluated at the old guess $\mathbf{x}_k$:

$$\mathbf{g}_k = \nabla F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k} \tag{F.8}$$

As visible in Eq. F.7, in order to have a decrease in the performance index, the second term on the right-hand side of the equation must be negative:

$$\mathbf{g}_k^T \Delta\mathbf{x}_k = \alpha_k \mathbf{g}_k^T \mathbf{p}_k < 0 \tag{F.9}$$

Since the learning rate is always greater than zero this implies:

$$\mathbf{g}_k^T \mathbf{p}_k < 0 \tag{F.10}$$

Any vector $\mathbf{p}_k$ that satisfies the last equation is called a descent direction. In order to have the steepest descend, it will be needed that $\mathbf{g}_k^T \mathbf{p}_k$ is the most negative. Since this is an inner product between the gradient and the direction vector, the maximum of this product is met when the two are parallel (from linear algebra theory). In order to get the

most negative, the direction that points in the steepest direction is the negative of the gradient:

$$\mathbf{p}_k = -\mathbf{g}_k \tag{F.11}$$

Recalling equation F.5, the steepest method is given:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k \tag{F.12}$$

Newton's method

Newton's method is generally much faster than steepest descent. For quadratic functions it will locate a stationary point in one iteration. One disadvantage is that it requires calculation and storage of the Hessian matrix, as well as its inverse. In addition, the convergence properties of Newton's method are quite complex. The aim of optimizing the function $F(x)$ through Newton's method is accomplished by the principle of locating the stationary point of the quadratic approximation of the function $F(x)$. Considering the second-order Taylor series:

$$F\left(\mathbf{x}_{k+1}\right) = F\left(\mathbf{x}_k + \Delta\mathbf{x}_k\right) \approx F\left(\mathbf{x}_k\right) + \mathbf{g}_k^T \Delta\mathbf{x}_k + \frac{1}{2}\Delta\mathbf{x}_k^T \mathbf{A}_k \Delta\mathbf{x}_k \tag{F.13}$$

where $\mathbf{A}_k$ is the Hessian matrix evaluated at the old guess $\mathbf{x}_k$:

$$\mathbf{A}_k \equiv \nabla^2 F(\mathbf{x})\big|_{\mathbf{x}=\mathbf{x}_k} \tag{F.14}$$

The gradient of the function with respect to $\Delta\mathbf{x}_k$ must be set equal to zero:

$$\mathrm{g}_k + \ \mathrm{A}_k \Delta \mathrm{x}_k = 0 \tag{F.15}$$

and solving it for $\Delta\mathbf{x}_k$ it gives:

$$\Delta\mathbf{x}_k = -\mathbf{A}_k^{-1}\mathbf{g}_k \tag{F.16}$$

Finally, the Newton's method is expressed by:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{A}_k^{-1}\mathbf{g}_k \tag{F.17}$$

F.5.2 First application in artificial neural networks

One of the first artificial neurons was introduced by McCulloh and Pitts in 1943. The main feature of their model is that a weighed sum of input signals is compared to the threshold to determine the neuron output. These neurons could, in principle, be able to compute any logical or arithmetic function. On the other hand, these neurons were not trained with any training method. A new class of neural networks, called perceptrons, was proposed by Fank Rosenblatt [50] and other researchers in the late 50s. The main

innovation here was the use of a learning rule for the training. Rosenblatt's perceptron model limitation proposed as the transfer function the hard-limiting for the perceptron:

$$a = \text{ hardlim } (n) = \begin{cases} 1 \text{ if } n \geq 0 \\ 0 \text{ otherwise} \end{cases} \tag{F.18}$$

In 1960, two other researchers, Bernard Widrow and Marcian Hoff, introduced the ADALINE (ADAptive LInear NEuron) network, and a learning rule which they called the LMS (Least Mean Square) algorithm (the first implementation of descend gradient). Their ADALINE network is very similar to the perceptron of Rosenblatt with the difference that ADALINE's transfer function is linear and not hard-limiting. This makes the model able both of linear classification and linear function approximation still this is a single-layer network with R inputs and one output neuron. Least Mean Square (LMS) adjusts the weights and biases in the direction in which the mean square error goes down most steeply. The error used to compute the mean square error is the difference between the target output and the network output.

The LMS is more powerful than the perceptron learning rule and it found many practical uses, but both suffered the same limitation: they can only solve linearly separable problems. [43]

F.5.3 Backpropagation

Backpropagation is an approximate steepest descent algorithm that minimizes mean square error which is the performance index and can be used for training multi-layer networks overcoming the limitation of usage only in linearly separable problems. The difference between MSE and backpropagation is in the way the derivatives are computed. If for a single-layer linear network the error is an explicit linear function of the network weights and the derivative with respect to weights can be easily computed, in multi-layer networks with non-linear transfer functions, the relationship between the weights and the error is much more complex. In fact, in order to compute the derivatives, it's needed to use the chain rule of calculus. The first description of an algorithm to train multi-layer networks was given in 1974 but only in the mid of eighties the backpropagation algorithm was widely publicized. Today The multi-layer perceptron, trained by the backpropagation algorithm, is currently the most widely used neural network.The mathematical description of this algorithm is given.

The function to be minimized as objective is the mean square error. In LMS this function could be written as:

$$F(\mathbf{x}) = E\left[e^2\right] = E\left[(t-a)^2\right] \tag{F.19}$$

where $E[]$ is the expected value, t is the output value, a is the predicted value and $\mathbf{x}$ is defined as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} \tag{F.20}$$

given $\mathbf{w}$ the weight vector and b the bias. In backpropagation algorithm, since it's dealing with multi-layer networks, considering $\mathbf{t}$ as the output vector, $\mathbf{a}$ as the predicted output vector, the performance index can be seen as:

$$F(\mathbf{x}) = E\left[\mathbf{e}^T\mathbf{e}\right] = E\left[(\mathbf{t}-\mathbf{a})^T(\mathbf{t}-\mathbf{a})\right] \tag{F.21}$$

At this point one can replace the expected square error by the square error at iteration k. In this way one finds out:

$$\hat{F}(x) = (t(k)-a(k))^T(t(k)-a(k)) = e(k)^T e(k) \tag{F.22}$$

The steepest descend algorithm for the approximate mean square error is:

$$\omega_{i,j}^m(k+1) = \omega_{i,j}^m(k) - \alpha\frac{\partial\hat{F}}{\partial\omega_{i,j}^m} \tag{F.23}$$

$$b_i^m(k+1) = b_i^m(k) - \alpha\frac{\partial\hat{F}}{\partial b_i^m} \tag{F.24}$$

where α is the learning rate, m is the layer considered. For F.23 i and j for indicate respectively from which neuron the signal comes from and the neuron considered while for F.24 i indicates the considered neuron.

The error is an indirect function of the weights in the hidden layers. In order to compute the derivatives the *chain rule* is used. With the chain rule it's possible to write:

$$\frac{\partial\hat{F}}{\partial\omega_{i,j}^m} = \frac{\partial\hat{F}}{\partial n_i^m}\frac{\partial n_i^m}{\partial\omega_{i,j}^m} \tag{F.25}$$

$$\frac{\partial\hat{F}}{\partial b_i^m} = \frac{\partial\hat{F}}{\partial n_i^m}\frac{\partial n_i^m}{\partial b_i^m} \tag{F.26}$$

The latter term in each of these equations can be computed because the net input layer m is an explicit function of the weights and bias in that layer:

$$n_i^m = \sum_{j=1}^{S^{m-1}} \omega_{i,j}^m a_j^{m-1} + b_i^m \tag{F.27}$$

Therefore:

$$\frac{\partial n_i^m}{\partial\omega_{i,j}^m} = a_j^{m-1} \; and \; \frac{\partial n_i^m}{\partial b_i^m} = 1 \tag{F.28}$$

It's possible to state that the *sensitivity* of F to changes in the i-th element of the net input at a layer m is defined as:

$$s_i^m = \frac{\partial \hat{F}}{\partial n_i^m} \tag{F.29}$$

and equation F.25 and F.26 can be simplified to:

$$\frac{\partial \hat{F}}{\partial \omega_{i,j}^m} = s_i^m \; a_j^{m-1} \tag{F.30}$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = s_i^m \tag{F.31}$$

Now the steepest descent algorithm can be expressed as:

$$\omega_{i,j}^m (k+1) = \omega_{i,j}^m (k) - \alpha s_i^m \; a_j^{m-1} \tag{F.32}$$

$$b_i^m (k+1) = b_i^m (k) - \alpha s_i^m \tag{F.33}$$

Or in matrix form:

$$\mathbf{W}^m (k+1) = \mathbf{W}^m (k) - \alpha \mathbf{s}^m \; \left(\mathbf{a}^{m-1}\right)^T \tag{F.34}$$

$$\mathbf{b}^m (k+1) = \mathbf{b}^m (k) - \alpha \mathbf{s}^m \tag{F.35}$$

where

$$\mathbf{s}^m = \frac{\partial \hat{F}}{\partial \mathbf{n}^m} = \begin{bmatrix} \frac{\partial \hat{F}}{\partial n_1^m} \\ \frac{\partial \hat{F}}{\partial n_2^m} \\ \vdots \\ \frac{\partial \hat{F}}{\partial n_{S^m}^m} \end{bmatrix} \tag{F.36}$$

Now the point is to compute the sensitivity. Through the chain rule and through algebraic computations it's possible to demonstrate that:

$$\mathbf{s}^m = \frac{\partial \hat{F}}{\partial \mathbf{n}_i^m} = \left(\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m}\right)^T \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}} = \dot{\mathbf{F}}^m (\mathbf{n}^m) \left(\mathbf{W}^{m+1}\right)^T \frac{\partial \hat{\mathbf{F}}}{\partial \mathbf{n}^{m+1}} = \\ = \dot{\mathbf{F}}^m (\mathbf{n}^m) \left(\mathbf{W}^{m+1}\right)^T \mathbf{s}^{m+1} \tag{F.37}$$

where

$$\dot{\mathbf{F}}^m (\mathbf{n}^m) = \begin{bmatrix} \dot{f}^m (n_1^m) & 0 & \cdots & 0 \\ 0 & \dot{f}^m (n_2^m) & \cdots & 0 \\ \vdots & \vdots & & 0 \\ 0 & 0 & \cdots & \dot{f}^m (n_{S^m}^m) \end{bmatrix} \tag{F.38}$$

and

$$\dot{f}^m \left(n_j^m\right) = \frac{\partial f^m \left(n_j^m\right)}{\partial n_j^m} \tag{F.39}$$

The process of calculate the sensitivity is given the name propagation because of its recurrence relationship where in order to compute the sensitivity of a layer m, it's needed the sensitivity of the forward layer m+1. The sensitivity is propagated backward through the network from the last layer to the first layer.

Let's summarize the backpropagation algorithm:

1. Propagation of the input forward the through the network

$$\mathbf{a}^0 = \mathbf{p} \tag{F.40}$$

$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}\left(\mathbf{W}^{m+1}\mathbf{a}^m + \mathbf{b}^{m+1}\right) \; for \; m = 0, 1, \; ..., \; M-1 \tag{F.41}$$

$$\mathbf{a} = \mathbf{a}^M \tag{F.42}$$

2. Propagate the sensitivities backward through the network:

$$\mathbf{s}^M = -2\dot{\mathbf{F}}^M\left(\mathbf{n}^M\right)(\mathbf{t} - \mathbf{a}) \tag{F.43}$$

$$\mathbf{s}^M = \dot{\mathbf{F}}^m\left(n^m\right)\left(\mathbf{W}^{m+1}\right)^T \mathbf{s}^{m+1} \; for \; m = M-1, ...2, 1 \tag{F.44}$$

3. Update the biases and the weights using steepest descend rule:

$$\mathbf{W}^m(k+1) = \; \mathbf{W}^m(k) \; - \; \alpha\mathbf{s}^m\left(\mathbf{a}^{m-1}\right)^T \tag{F.45}$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha\mathbf{s}^m \tag{F.46}$$

This algorithm involves on-line or incremental training, in which the weights and biases are updated after each input is presented. Another approach often used is called batch training. In this case the gradient is computed after that all the inputs are presented to the network and before updating the weights and the biases [51].

F.5.4 Variation on backpropagation

Since the backpropagation algorithm is very slow for most practical applications, many variance and improvement have been done on the algorithm in order to make him faster and more practical. Two categories can be detected:

- The heuristic techniques: the study of backpropagation performance is accomplished and ideas like varying the learning rate, using momentum and rescaling variables.
- Standard numerical optimization techniques: since backpropagation network is only an optimization technique that uses steepest descend technique, the idea is to use other optimization algorithms like Newton's method and conjugate gradient, a compromise between Newton's method and steepest descend method [49]) maintain the backpropagation network architecture.

In this work, when ANNs are built, the training function used is the Levenberg-Marquardt which is a numerical optimized technique which implement Newton's method.

The Levenberg-Marquardt algorithm is a variation of Newton's method, designed for minimizing functions that are sums of squares of other nonlinear functions. This method suits perfectly in order to train neural network where the performance index is the mean square error. The main drawback of this method is the need to store a Hessian matrix $n \cdot n$ with n the number of parameters (weight and biases) of the network which leads to impossible usage of the algorithm for very large networks [49].

F.6 Generalization of an artificial neural network

One of the key issues in designing a multilayer network is determining the number of neurons to use. This directly will affect the performance and the complexity of the neural network. The number of neurons determines the number of parameters (weights and biases) that are adjusted by training algorithm to reach the optimal performance. There is not a specific rule for choosing it, but a wrong choice can lead to underfit or overfit of the data. The goal is the development of an ANN able to well perform in both new and already seen situations. This characteristics of an ANN is called generalization.

A quantitative measure of generalization is needed. Usually a neural network is trained on a dataset called training set and through the Levenberg-Marquardt learning algorithm, the means square error is decreased each iteration adjusting the weight and biases. Moreover another dataset is used as testing set. After the training of the network, the error done on this testing set can be considered as a kind of measure of generalization capability of the network. The testing set must contain points which are not present in the training set and that represent all the situations the ANN was trained for. There are methods to improve the generalization of a given model (with a number of neurons set). The main interesting for this thesis work is the early stopping. [49]

F.6.1 Early Stopping

The concept behind the early stopping method is that, in order to improve the generalization capability of a model, the training is stopped before the mean square error computed by the training set reaches the minimum. Early sopping works by selecting a random subset from the training set and computing periodically the so called validation error (mean square error computed on the validation set). While the mean square error from training continues to decrease, the error from the validation set at a certain point starts to increase. A stopping criterium for the training session which helps the problem to not overfit is selecting a threshold for consecutive validation errors which doesn't decrease [52].

Appendix G

Gaussian Process Regression

Gaussian process regression (GPR) model is a non-parametric kernel-based probabilistic model. [53] This kind of model has been originated in a Bayesian inference framework. This means that the model works by defining a prior over functions and then based on Bayesian inference convert it into a posterior once some data are observed [54]. This means that the algorithm doesn't try to fit the best according to given data (as in the case of ANN) but, given the combination of the prior and initial points, it fits the posterior distribution over functions. Then it's possible to compute posterior predictive distributions for new test inputs. The main advantage of these kinds of models is the possibility to quantify the uncertainty in the model estimates which gives the possibility to make more robust predictions [55]. Since the gaussian process regression is a kernel-based fully Bayesian regression algorithm, a review of Bayesian methods in the context of probabilistic linear regression is given. Afterwards the focus is on Gaussian processes and Gaussian process regression. Firstly however a briefly review about multivariate Gaussian distribution is given.

G.1 Multivariate Gaussians

Let's give the definition of multivariate gaussian distribution and properties which hold. Given a vector-valued random variable $x \in \mathbb{R}^n$ is said to have a multivariate normal distribution with a mean $\mu \in \mathbb{R}^n$ and covariance matrix Σ if

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{G.1}$$

In this case it's possible to write $x \sim \mathcal{N}(\mu, \Sigma)$. The covariance matrix is a square matrix that gives the covariance between each pair of elements. In particular covariance in the case of two random set of variables x and y can be expressed as:

$$\sigma(x, y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \tag{G.2}$$

This means that the variance of x can be seen as the covariance of x with itself by $\sigma(x, x)$. The elements of the covariance matrix can be expressed as $\Sigma_{i,j} = \sigma(x_i, x_j)$ where $C \in \mathbb{R}^{d\times d}$ and d is the dimension of random variables sets. In the case of two dimensions, the matrix is given by:

$$\Sigma = \begin{bmatrix} \sigma(x,x) & \sigma(x,y) \\ \sigma(y,x) & \sigma(y,y) \end{bmatrix} \tag{G.3}$$

In the case x and y independent the matrix takes the following shape:

$$C = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \tag{G.4}$$

Let's now consider a random vector $x \in \mathbf{R}^n$ with $x \sim \mathcal{N}(\mu, \Sigma)$. Let's suppose that the variable in x have been partitioned into two sets $x_A = [x_1 \cdots x_r]^T$ and $x_B = [x_{r+1} \cdots x_n]^T \in \mathbb{R}^{n-r}$ and similarly for μ and Σ such that

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x_A} \\ \boldsymbol{x}_B \end{bmatrix} \quad \boldsymbol{\mu} = \begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{bmatrix} \tag{G.5}$$

With $\Sigma_{AB} = \Sigma_{BA}^T$ since $\Sigma = E\left[(x-\mu)(x-\mu)^T\right] = \Sigma^T$. The following properties hold:

1. Normalization. The density function normalizes, i.e.

$$\int_x p(x; \mu, \Sigma) dx = 1$$

2. Marginalization. The marginal densities,

$$p(x_A) = \int_{x_B} p(x_A, x_B; \mu, \Sigma)\, dx_B$$

and

$$p(x_B) = \int_{x_A} p(x_A, x_B; \mu, \Sigma)\, dx_A$$

are Gaussian:

$$x_A \sim \mathcal{N}(\mu_A, \Sigma_{AA})$$

and

$$x_B \sim \mathcal{N}(\mu_B, \Sigma_{BB})$$

3. Conditioning. The conditional densities

$$p(x_A \mid x_B) = \frac{p(x_A, x_B; \mu, \Sigma)}{\int_{x_A} p(x_A, x_B; \mu, \Sigma)\, dx_A}$$

and

$$p(x_B \mid x_A) = \frac{p(x_A, x_B; \mu, \Sigma)}{\int_{x_B} p(x_A, x_B; \mu, \Sigma)\, dx_B}$$

are also Gaussian

$$x_A \mid x_B \sim \mathcal{N}\left(\mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}\left(x_B - \mu_B\right), \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}\right)$$

and

$$x_B \mid x_A \sim \mathcal{N}\left(\mu_B + \Sigma_{BA}\Sigma_{AA}^{-1}\left(x_A - \mu_A\right), \Sigma_{BB} - \Sigma_{BA}\Sigma_{AA}^{-1}\Sigma_{AB}\right)$$

4. Summation. The sum of independent Gaussian random variables (with the same dimensionality) $y \sim \mathcal{N}(\mu, \Sigma)$ and $z \sim \mathcal{N}\left(\mu', \Sigma'\right)$ is also Gaussian

$$y + z \sim \mathcal{N}\left(\mu + \mu', \Sigma + \Sigma'\right)$$

[56]

After this short review, the focus is moved to Bayesian linear regression.

G.2 Bayesian linear regression

Let's focus on a training set S given by $S = \left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1}^{m}$ with x which denotes an input vector and y denotes a scalar output. Following the standard probabilistic interpretation of linear regression, one can infer that:

$$y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}, \quad i = 1, \ldots, m \tag{G.6}$$

Where θ are linear parameters while $\epsilon^{(i)}$ are "noise" variables with independent $\mathcal{N}\left(0, \sigma^2\right)$ distributions. It follows that

$$y^{(i)} - \theta^T x^{(i)} \sim \mathcal{N}\left(0, \sigma^2\right) \tag{G.7}$$

Or

$$P\left(y^{(i)} \mid x^{(i)}, \theta\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right) \tag{G.8}$$

A prior distribution over parameters θ is also given. Usually a choice is $\theta \sim \mathcal{N}\left(0, \tau^2 I\right)$. A parameter posterior is then given using Bayes' rule which states that:

$$\text{posterior } = \frac{\text{likelihood } \times \text{ prior}}{\text{marginal likelihood}}$$

Hence:

$$p(\theta \mid S) = \frac{p(\theta)p(S \mid \theta)}{\int_{\theta'} p\left(\theta'\right) p\left(S \mid \theta'\right) d\theta'} = \frac{p(\theta)\prod_{i=1}^{m} p\left(y^{(i)} \mid x^{(i)}, \theta\right)}{\int_{\theta'} p\left(\theta'\right)\prod_{i=1}^{m} p\left(y^{(i)} \mid x^{(i)}, \theta'\right) d\theta'} \tag{G.9}$$

Given a new test point x_* , one can now predict, using the same noise model, a posterior predictive distribution over possible outputs y_* :

$$p\left(y_* \mid x_*, S\right) = \int_{\theta} p\left(y_* \mid x_*, \theta\right) p(\theta \mid S) d\theta \tag{G.10}$$

The integrals in the case of Bayesian linear regression can be algebraically treated and one can obtain:

$$\theta \mid S \sim \mathcal{N}\left(\frac{1}{\sigma^2} A^{-1} X^T \vec{y}, A^{-1}\right) \tag{G.11}$$

$$y_* \mid x_*, S \sim \mathcal{N}\left(\frac{1}{\sigma^2} x_*^T A^{-1} X^T \vec{y}, x_*^T A^{-1} x_* + \sigma^2\right) \tag{G.12}$$

With $A = \frac{1}{\sigma^2} X^T X + \frac{1}{\tau^2} I$ and $X = \begin{bmatrix} -\left(x^{(1)}\right)^T - \\ -\left(x^{(2)}\right)^T \\ \vdots \\ -\left(x^{(m)}\right)^T - \end{bmatrix} \in \mathbf{R}^{m \times n}$. As already said, this is the posterior distribution for the predictions $y^{(i)} - \theta^T x^{(i)} \sim \mathcal{N}(0, \sigma^2)$ [56].

Reviewed these basis concepts, let's focus on Gaussian processes.

G.3 Gaussian process

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. [55]. Gaussian processes can be seen as the extension of multivariate Gaussian to infinite-seized collections of variables. Moreover this extension may lead to think of Gaussian processes as distributions not just over random vectors but over random functions [56]. Hence if $\left\{f(x), x \in \mathbb{R}^d\right\}$ is a GP, than given n observations $x_1, x_2, \ldots, x_n$, the joint distribution of the random variables $f(x_1), f(x_2), \ldots, f(x_n)$ is Gaussian. A GP is defined by its mean function $m(x)$ and covariance function $k(x, x')$. That is, if $\left\{f(x), x \in \mathbb{R}^d\right\}$ is a GP, then $E(f(x)) = m(x)$ and $\mathrm{Cov}\left[f(x), f(x')\right] = E\left[\{f(x) - m(x)\}\{f(x') - m(x')\}\right] = k(x, x')$ [53].

A gaussian process can be written as

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_m) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_m) \\ \vdots & \ddots & \vdots \\ k(x_m, x_1) & \cdots & k(x_m, x_m) \end{bmatrix}\right) \tag{G.13}$$

Or more simply:

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right) \tag{G.14}$$

Recalling marginalization property for multivariate Gaussian, it's possible to obtain the marginal multivariate Gaussian density corresponding to any finite sub-collection of variables. While any real-valued function $m(.)$ is acceptable for Gaussian processing, for the covariance function (or kernel) $k(.,.)$, not every function can be used but there's the need that the function is positive semi-definite. In this way One can infer that the Gaussian processes are kernel-based probability distributions since any valid kernel can be used as a covariance function [56].

Usually covariance function is parameterized by a set of kernel hyper-parameters Θ and the kernel function is indicated as $k(x, x'|\Theta$ which indicates the dependence on Θ. One of the most used kernel function is the squared exponential kernel function which is defined as:

$$k\left(x_i, x_j \mid \theta\right) = \sigma_f^2 \exp\left[-\frac{1}{2}\frac{\left(x_i - x_j\right)^T \left(x_i - x_j\right)}{\sigma_l^2}\right] \tag{G.15}$$

For some $\tau > 0$. As an example, let's take a zero-mean Gaussian process

$$f(\cdot) \sim \mathcal{GP}(0, k(\cdot, \cdot)) \tag{G.16}$$

Defined for functions $h : \mathcal{X} \to \mathbf{R}$ where $\mathcal{X}$ some set of elements. The kernel function is set to be the squared exponential. In this way the expectation is that for the function values from GP will tend to be distributed around zero. Moreover for two points $x, x' \in \mathcal{X}$, $f(x)$ and $f(x')$ will tend to have high covariance if x and x' are lose in the input domain (from the kernel function which gives values close to one for close points). On the other hand if points are far apart in the domain, $f(x)$ and $f(x')$ will have low covariance [56].

Now that Gaussian process has been briefly exposed, it's possible to give a description about Gaussian process regression.

G.4 Gaussian process regression

Gaussian process can be used in the framework of Bayesian regression discussed previously.

Let's consider a training set $S = \left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1}^m$ of independent variables with some unknown distribution. Differently from linear regression, Gaussian process regression have

$$y^{(i)} = f\left(x^{(i)}\right) + \varepsilon^{(i)}, \quad i = 1, \ldots, m \tag{G.17}$$

Also in this case $\epsilon^{(i)}$ are "noise" variables with independent $\mathcal{N}\left(0, \sigma^2\right)$ distributions. Like done before for Bayesian linear regression, let's assume a prior distribution over functions $f(\cdot)$ in particular let's assume a zero-mean gaussian process prior for some valid covariance function $k(\cdot, \cdot)$:

$$f(\cdot) \sim \mathcal{GP}(0, k(\cdot, \cdot)) \tag{G.18}$$

Given the training data S and the testing inputs matrix $X_* \in \mathbf{R}^{m_* \times n}$, in order to compute the posterior predictive distribution over the testing outputs vector $\vec{y}_* \in \mathbf{R}^{m_*}$, one can follow the framework for Bayesian linear regression. Hence the steps to follow should be the computation of parameter posterior which allows the computation of the posterior predictive distribution $p\left(y_* \mid x_*, S\right)$ for a new test point x_*.

Alternatively for the definition of gaussian process, any set of input points should have a joint multivariate Gaussian distribution. In this way it's possible to infer:

$$\begin{bmatrix} \vec{f} \\ \vec{f}_* \end{bmatrix} \mid X, X_* \sim \mathcal{N}\left(\overrightarrow{0}, \begin{bmatrix} K(X,X) & K\left(X,X_*\right) \\ K\left(X_*,X\right) & K\left(X_*,X_*\right) \end{bmatrix}\right) \tag{G.19}$$

Where

$$\vec{f} \in \mathbf{R}^m \text{ such that } \vec{f} = \begin{bmatrix} f\left(x^{(1)}\right) & \cdots & f\left(x^{(m)}\right) \end{bmatrix}^T$$

$$\vec{f}_* \in \mathbf{R}^{m_*} \text{ such that } \vec{f}_* = \begin{bmatrix} f\left(x_*^{(1)}\right) & \cdots & f\left(x_*^{(m)}\right) \end{bmatrix}^T$$

$$K(X,X) \in \mathbf{R}^{m \times m} \text{ such that } \left(K(X,X)\right)_{ij} = k\left(x^{(i)}, x^{(j)}\right)$$

$$K\left(X,X_*\right) \in \mathbf{R}^{m \times m_*} \text{ such that } \left(K\left(X,X_*\right)\right)_{ij} = k\left(x^{(i)}, x_*^{(j)}\right)$$

$$K\left(X_*,X\right) \in \mathbf{R}^{m_* \times m} \text{ such that } \left(K\left(X_*,X\right)\right)_{ij} = k\left(x_*^{(i)}, x^{(j)}\right)$$

$$K\left(X_*,X_*\right) \in \mathbf{R}^{m_* \times m_*} \text{ such that } \left(K\left(X_*,X_*\right)\right)_{ij} = k\left(x_*^{(i)}, x_*^{(j)}\right)$$

From the previous assumption:

$$\begin{bmatrix} \vec{\varepsilon} \\ \vec{\varepsilon}_* \end{bmatrix} \sim \mathcal{N}\left(\overrightarrow{0}, \begin{bmatrix} \sigma^2 I & \overrightarrow{0} \\ \overrightarrow{0}^T & \sigma^2 I \end{bmatrix}\right)$$

The sums of independent Gaussian variables is also Gaussian, so

$$\begin{bmatrix} \vec{y} \\ \vec{y}_* \end{bmatrix} \mid X, X_* = \begin{bmatrix} \vec{f} \\ \vec{f}_* \end{bmatrix} + \begin{bmatrix} \vec{\varepsilon} \\ \vec{\varepsilon}_* \end{bmatrix} \sim \mathcal{N}\left(\overrightarrow{0}, \begin{bmatrix} K(X,X) + \sigma^2 I & K\left(X,X_*\right) \\ K\left(X_*,X\right) & K\left(X_*,X_*\right) + \sigma^2 I \end{bmatrix}\right) \tag{G.20}$$

Now, through the rules for conditioning Gaussians, it follows that

$$\overrightarrow{y_*} \mid \vec{y}, X, X_* \sim \mathcal{N}\left(\mu^*, \Sigma^*\right) \tag{G.21}$$

where

$$\mu^* = K\left(X_*,X\right)\left(K(X,X) + \sigma^2 I\right)^{-1} \vec{y}$$

$$\Sigma^* = K\left(X_*,X_*\right) I - K\left(X_*,X\right)\left(K(X,X) + \sigma^2 I\right)^{-1} K\left(X,X_*\right)$$

Hence Gaussian process regression models lead to simple and straightforward linear algebra implementation.Moreover as Bayesian linear regression discussed above, these models' output is a probability distribution which hence give information about an average prediction and an estimated error. Gaussian process regression models can also take advantage of data structure by a careful choice of kernel function [56].

Some reasons can lead to the necessity to explicitly model a mean function which is a way to not consider GPs model with zero mean functions. A first and trivial way to specify a non-zero mean over functions is to use a fixed mean function $m(x)$. In this way the

Gaussian process becomes $f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k\left(\mathbf{x}, \mathbf{x}'\right)\right)$ while the predictive mean becomes $\overline{\mathbf{f}}_* = \mathbf{m}\left(X_*\right) + K\left(X_*, X\right) K_{\boldsymbol{y}}^{-1}(\mathbf{y} - \mathbf{m}(X))$ with $K_y = K + \sigma_n^2 I$ while the predictive variance remains unchanged. Another way to explicitly model the mean function is through the definitions of few fixed basis functions whose coefficients are estimated from the data. The model becomes

$$g(\mathbf{x}) = f(\mathbf{x}) + \mathbf{h}(\mathbf{x})^\top \boldsymbol{\beta}, \text{ where } f(\mathbf{x}) \sim \mathcal{GP}\left(0, k\left(\mathbf{x}, \mathbf{x}'\right)\right) \tag{G.22}$$

with $\mathbf{h}(\mathbf{x})$ are a set of fixed basis functions and β are additional parameters. During the fitting of the model beta could be optimized together with the hyper-parameters of the covariance function [55].Matlab implementation allows to train Gaussian process regression models by estimating the basis function coefficients β, the noise variance, σ^2, and the hyper-parameters, θ, of the kernel function from training set.

In the follow a brief description about parameter estimation is given.

G.5 Parameters' estimation

In order to recap, an instance of response y modelled through GPR with few fixed basis functions is given by

$$P\left(y_i \mid f\left(x_i\right), x_i\right) \sim N\left(y_i \mid h\left(x_i\right)^T \beta + f\left(x_i\right), \sigma^2\right) \tag{G.23}$$

So the following parameters are needed for fitting the model:

- Coefficient vector β of fixed basis functions
- Hyper-parameters θ for kernel function
- Noise variance σ^2

The strategy implemented by Matlab to estimate the parameters is given by maximizing the likelihood $P(y \mid X)$ as a function of β, θ and σ^2. Hence:

$$\widehat{\beta}, \hat{\theta}, \hat{\sigma}^2 = \underset{\beta, \theta, \sigma^2}{\arg\max} \log P\left(y \mid X, \beta, \theta, \sigma^2\right) \tag{G.24}$$

Since:

$$P(y \mid X) = P\left(y \mid X, \beta, \theta, \sigma^2\right) = \mathcal{N}\left(y \mid H\beta, K(X, X \mid \theta) + \sigma^2 I_n\right) \tag{G.25}$$

Then the marginal log likelihood function is [55]

$$\begin{aligned} \log P\left(y \mid X, \beta, \theta, \sigma^2\right) = & -\frac{1}{2}(y - H\beta)^T \left[K(X, X \mid \theta) + \sigma^2 I_n\right]^{-1} (y - H\beta) \\ & - \frac{n}{2} \log 2\pi - \frac{1}{2} \log \left|K(X, X \mid \theta) + \sigma^2 I_n\right| \end{aligned} \tag{G.26}$$

First of all $\hat{\beta}(\theta, \sigma^2)$ which maximizes the log-likelihood function is computed. Then the estimate is used to calculate the beta-profiled likelihood. At this point, after algebraical computations, *beta*-profiled log-likelihood over θ and σ^2 is optimized and the estimates are found.

From a computational point of view, the training of a GPR model in an exact mode, following the procedure previously exposed can be difficult to handle by the machine. Among all, the inversion of an nxn kernel matrix and the evaluation of likelihood are very huge to handle if n becomes large. Moreover also the predictions are heavy from a computational point of view. In order to avoid this problem, approximation methods can be used in Matlab.

Appendix H

Support Vector Machines

H.1 Classification algorithms

Machine learning (ML) algorithms can be recognized because they lead to a self-learning procedure. These algorithms can learn from data without being explicitly programmed. The two main branches of ML are the supervised training and the un-supervised training. For the former branch three categories of algorithms (ANN, GPR and SVM) have been exploited in this work while for the latter branch k-means clustering has been applied for the novel adaptive algorithms proposed. Focusing on the first category, supervised learning tasks can be grouped into regression problems where outputs are continuous and in classification problems where output are categorical.

Given a dataset whose members are classified according to some given label or category, a classifier algorithm learns from the training set and then assigns new data point to a particular class. Binary classification or multi-label classification are common examples of tasks that a classifier handles.

Five main categories can be identified for classification algorithms:

- Logical-based learning algorithm whose main representative is Decision Tree algorithm. By generating a set of decision sequences, it leads to predicting the label of an unlabelled data.
- Support Vector Machine.
- Statistics based algorithms which generalize problems with the help of distributive statistics and look into the distribution structure to continue the predicting task. Naïve Bayes is the most popular example of statistics-based algorithm.
- Lazy learning algorithms whose main representative is K-nearest neighbour algorithm. They fall under the title of statistical methods but they are called 'lazy' due to their ability to well predict labels with low computational cost.

- Artificial Neural Networks (ANN) [18]

In this work the focus is set on binary classification while the algorithm chosen for handling the classifier problem has been the SVM.

H.2 Support Vector Machines

Support Vector Machines (SVMs) is a popular machine learning method for classification and regression.

SVMs are used in many applications: from stock marketing to applications in bioinformatics, face detection, classification of images, handwriting recognition, classification of images. [18] Moreover SVM is used for binary classification in tasks like detecting anomalies and faults in many engineering problems like nuclear industry, and automotive industry.

In this work a SVM classifier (SVM-C) has been used for the assembling of a composite model with the aim of increasing the performance for low CDT events. The composite model assembled is made by a classifier which labels the points in two categories (the unsafe region, labelled with 1, and the safe region, labelled with 0) and by two sub-models. Afterwards two novel adaptive methods have been applied in order to assess the possibility to enhance the performance of composite predictions in unsafe region.

H.2.1 Theory

SVMs are a class of powerful, highly flexible modelling techniques. The theory behind SVMs was originally developed in the context of classification models but later has been broadened also for regression tasks. Since in this work SVM has been mainly used for classification purpose, let's focus on the theory behind SVM-C. Initially developed in the mid-sixties by Valdimir Vapnik, the model was continuously developed and it became one of the most flexible and effective machine learning tools available [57].

Let's consider the case of binary classification with a training set given by

$$\{(\mathbf{y}_1, x_1)\,,\ \ldots,\ (\mathbf{y}_{\mathbb{J}}, x_J)\}$$

with $\mathbf{y}_j \in \mathbb{R}^M$ and $x_j \in \pm 1$. SVM represents the dataset as points in a n-dimensional space segregated into classes by a clear margin widest possible known as the distance of the closest positive and negative training points. In order to set the margin, hyperplanes are chosen with some constraints: they must pass at least through one of the training samples of each class while no samples can be found within the margin. The points through which the support hyperplanes pass are called Support Vectors while the hyperplanes are

called Support Hyperplanes. The algorithm creates maps to predict if a point falls in an hyperplane or in another.

Let's start considering a linear SVM case with a linear boundary that can be written as:

$$\langle \boldsymbol{w}, \boldsymbol{y}_j \rangle + b = 0 \tag{H.1}$$

Where $\langle \cdot . \cdot \rangle$ is the dot product in $\mathbb{R}^M$. For points in the training set with $x_j = +1$ the linear expression gives a positive value, on the contrary points with $x_j = -1$ show negative values.

From the definition of the margin, algebraically it can be written:

$$\min_{j=1,\dots,J} \frac{|\langle \mathbf{w}, \mathbf{y_j} \rangle + \mathrm{b}|}{\|\mathbf{w}\|} \tag{H.2}$$

By scaling the problem $\min_{j=1,\dots,J} \left| \langle \boldsymbol{w}, \boldsymbol{y}_j \rangle + b \right| = 1$. Hence the margin can be reduced to the expression of $\frac{1}{||\mathbf{x}||}$. In the same way considering two linear hyperplanes, one can algebraically write:

$$\langle w, y_1 \rangle + b = +1 \tag{H.3}$$

$$\langle w, y_2 \rangle + b = +1 \tag{H.4}$$

And by making the difference and scaling:

$$\langle w, y_1 - y_2 \rangle + b = 2 \tag{H.5}$$

$$\left\langle \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}, \boldsymbol{y}_1 - \boldsymbol{y}_2 \right\rangle + b = \frac{2}{\|\boldsymbol{w}\|} \tag{H.6}$$

Since +1 and -1 are arbitrary values, one can state that the margin can be reduced to the expression of $\frac{1}{||\mathbf{x}||}$ which has to be maximized. Moreover it's possible to infer that $x_j \left(\langle \boldsymbol{w}, \boldsymbol{y}_j \rangle + b \right) \geq 1$ for all the j. Hence, the mathematical expression for the maximization problem for the margin can be written as:

$$\begin{gathered} \max_{\mathbf{w},\mathbf{b}} \frac{1}{\|\mathbf{w}\|} \\ \text{s.t. } \mathrm{x_j} \left(\langle \mathbf{w}, \mathbf{y_j} \rangle + \mathrm{b} \right) \geq 1 \text{ for all} \end{gathered} \tag{H.7}$$

or

$$\begin{gathered} \min_{\mathbf{w},\mathbf{b}} \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t. } \mathrm{x_j} \left(\langle \mathbf{w}, \mathbf{y_j} \rangle + \mathrm{b} \right) \geq 1 \text{ for all j} \end{gathered} \tag{H.8}$$

The problem can be considered as a constrained optimization problem with a quadratic objective function and linear constraints.

However not always is possible to linearly separate the points. An help is given by adjusting the constraints by a slack variable ξ_j that can be taken large enough for each

training point. Moreover there's the need for a penalization on ξ_j by a factor $C > 0$. In this way the modified optimization problem is obtained:

$$\begin{gathered} \min_{\mathbf{w},\mathbf{b},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{\mathrm{J}} \sum_{j=1}^{\mathrm{J}} \xi_j \\ \text{s.t. } \mathrm{x}_j \left(\langle \mathbf{w}, \mathbf{y}_\mathrm{j} \rangle + \mathrm{b}\right) \geq 1 - \xi_j \text{ for all j} \\ \xi_j \geq 0 \end{gathered} \tag{H.9}$$

From a mathematical point of view, such a problem can be solved by adopting the Lagrangian method of multipliers [58]. In terms of SVM optimization problem, the Lagrangian by introducing non-negative Lagrange multipliers α_j and β_j becomes:

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{J}\sum_{j=1}^{J} \xi_j + \sum_{j=1}^{J} \alpha_j \left(1 - \xi_j - x_j \left(\left\langle \boldsymbol{w}, \boldsymbol{y}_j \right\rangle + b\right)\right) - \sum_{j=1}^{J} \beta_j \xi_j \tag{H.10}$$

It's needed now minimize the maximum of $L\left(\mathbf{w}, b, \xi, \alpha, \beta\right)$.

As from theory, the gradients of the Lagrangian with respect to w, b and ξ_j are set to zero to find the optimal point:

$$\begin{gathered} \nabla_{\mathrm{w}} \mathrm{L} = \mathrm{w} - \sum_{\mathrm{j}=1}^{\mathrm{J}} \alpha_\mathrm{j} \mathrm{x}_\mathrm{j} \mathrm{y}_\mathrm{j} = 0 \\ \nabla_{\beta} \mathrm{L} = -\sum_{\mathrm{j}=1}^{\mathrm{J}} \alpha_\mathrm{j} \mathrm{x}_\mathrm{j} = 0 \\ \nabla_{\xi_\mathrm{j}} \mathrm{L} = \frac{\mathrm{C}}{\mathrm{J}} - \alpha_\mathrm{j} - \beta_\mathrm{j} = 0 \end{gathered}$$

And substituting these equations into the optimization problem for Lagrangian:

$$\begin{gathered} \max_{\alpha} -\frac{1}{2} \sum_{\mathrm{j,i}} \mathrm{x_i x_j} \alpha_\mathrm{i} \alpha_\mathrm{j} \left\langle \mathbf{y}_\mathrm{i}, \mathbf{y}_\mathrm{j} \right\rangle + \sum_{\mathrm{j}}^{\mathrm{J}} \alpha_\mathrm{j} \\ \text{s.t. } \sum_{\mathrm{j}=1}^{\mathrm{J}} \alpha_\mathrm{j} \mathrm{x}_\mathrm{j} = 0 \\ 0 \leq \alpha_\mathrm{j} \leq \frac{\mathrm{C}}{\mathrm{J}}, \mathrm{i} = 1, \ldots, \mathrm{J} \end{gathered} \tag{H.11}$$

Where $\langle \mathbf{y_i}, \mathbf{y_j} \rangle$ is called the Gram matrix. Considering Karush-Kuhn-Tucker (KKT) [59] complementarity conditions, required for nonlinear programming solutions to be optimal, algebraic computations lead to the derivation of the hyperplane:

$$\langle \mathrm{w}, \mathbf{y} \rangle + \mathrm{b} = \sum_{j=1}^{\mathrm{J}} \alpha_\mathrm{j} \mathrm{x}_\mathrm{j} \left\langle \mathbf{y}_\mathrm{j}, \mathbf{y} \right\rangle + \mathrm{b} \tag{H.12}$$

In the case non-linear classification, needed when the points could not be separated by a hyperplane, there's the need of mapping the input data into a so called Reproducing Kernel Hilbert Space (RKHS) using a map$\phi : \mathbb{R}^M \longrightarrow \mathbb{R}^N$ with $N \geq M$. This procedure is called Kernel trick and allows to evaluate the Gram matrix through a kernel associated with the RKHS. This is given by $K(\mathbf{y}_j,\ \mathbf{y})$ and eventually the decision hyperplane becomes:

$$\langle \mathrm{w}, \mathbf{y} \rangle + \mathrm{b} = \sum_{j=1}^{\mathrm{J}} \alpha_\mathrm{j} \mathrm{x}_\mathrm{j} \mathrm{K}\left(\mathbf{y}_\mathrm{j}, \mathbf{y}\right) + \mathrm{b} \tag{H.13}$$

As already stated, α_j are the Lagrange multipliers obtained from the quadratic programming optimization problem used to construct the SVM. These parameters are positive in the case they refer to support vectors, equal to zero on the contrary. That is, a SVM classifier can be trained only by the support vectors which usually are much less than the number of the training points. Using the equation, the label of a point can be given by the sign of S for a given point $\mathbf{x}$ [19].

For the linear case where the Gram matrix could be defined as $K(\mathbf{y}_i, \mathbf{y_j}) = \langle \mathbf{y_i}, \mathbf{y_j} \rangle$, while the usual non-linear transformation that can be applied are:

- Polynomial: $K(\mathbf{x}_i, \mathbf{x}) = (1 + \mathbf{x}_i \prime \mathbf{x})^p$, with p as integer.
- Radial basis function (or gaussian): $K(\boldsymbol{x}_i, \boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}\|^2}{2\sigma^2}\right)$ with $\sigma > 0$ the parameter controlling the kernel width.
- Hyperbolic tangent: $K(\mathbf{x}_i, \mathbf{x}) = tanh(1 + \mathbf{x}_i \prime \mathbf{x})$

H.2.2 Class imbalance

The main challenge of this work was the creation of a surrogate model with high performance above all for low values of cool down time predictions. Due to the physics of the problem, the initial dataset presented cool down time distribution highly imbalanced. This means that only few points give low cool down time. This leads to difficulties for the classifier to reliably assign the labels to new points which happen to belong to the critic region's class.

In literature there are many examples of situations with highly class-imbalanced dataset: faults detection problems of high reliable industries as nuclear systems or high speed trains are examples. Usually class-imbalance problems are often related to anomaly and outlier detection. In such cases standard classification techniques which assumes the training dataset well-balanced, can't achieve high prediction performance: the points belonging to minority class, usually the class which refers to failures, are misclassified. Hence new strategy should be set up to enhance prediction capabilities for the minority class. Two solutions applied in this work are the cost-sensitive option and two approaches of adaptive sampling techniques.

H.2.3 Cost-sensitive SVM

In order to improve the performance of the classifier in the presence of an imbalanced training set the cost-sensitive approach can be considered. The main idea of this option is to assign a different penalty to the classification errors on the minority class than the classification errors on the majority class. In particular for the majority class $c^- = \frac{1}{J^-}$ is assigned with J^- number of training points in the minority class while for the minority

class $c^+ = \frac{1}{J^+}$ is assigned and also in this case J^+ is the number of training points in the majority class. This means that if the points in the majority class are much more than the points in the minority class, it leads to $c^+ >> c^-$. These parameters are introduced in the expression to be minimized:

$$\begin{gathered} \min_{\boldsymbol{w},\boldsymbol{b},\xi} \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{J}\left(\sum_{j=1}^{J} \xi_j \left(c^+\delta_j^+ + c^-\delta_j^-\right)\right) \\ \text{s.t. } x_j^{CTI}\left(\langle \boldsymbol{w}, \boldsymbol{y}_j\rangle + b\right) \geq 1 - \xi_j \text{ for all } j \\ \xi_j \geq 0 \end{gathered} \tag{H.14}$$

With

$$\delta_j^{+(-)} = \begin{cases} 0, & \text{if pattern } j \text{ belongs to majority (minority) class} \\ 1, & \text{if pattern } j \text{ belongs to minority (majority) class} \end{cases}$$

H.2.4 Performance Parameters

Once a classifier has been trained, a testing set with new points can be used in order to assess how it well predicts new points. The performance are then evaluated commonly through the confusion matrix and by some performance parameters.

The simplest metric that can be applied is the overall accuracy rate so the ratio between well predicted labels and the total number of points in the testing set. The main drawback of this indicator is that it doesn't make any distinction about the type of errors being made. This is why other indicators are preferred

The confusion matrix for binary classification is a 2x2 matrix which indicates true and false positive responses (respectively TP and FP), true and false negative responses (respectively TN and FN) (fig []). The responses given by the model can be in fact true or false. From these values, the following performance parameters are computed.

- True positive rate (TPR) (or sensitivity): $TPR = \frac{TP}{TP+TN}$
- True negative rate (TNR) (or specificity):$TNR = \frac{TN}{TN+FP}$
- Positive predictive value (PPV) (or precision): $PPV = \frac{TP}{TP+FP}$
- Negative predictive value (NPV) : $NPV = \frac{TN}{TN+FN}$
- F or F1 score, that is the harmonic mean of precision and sensitivity:

$$F = 2\frac{PPV * TPR}{PPV + TPR} = 2\frac{TP}{2TP + FP + FN}$$

- Gmean or Fowlkes–Mallows index:$G_{mean} = \sqrt{PPV * TPR}$
- z, that is the mean of F and Gmean: $z = \frac{F+G_{mean}}{2}$

Moreover for better assessing the quality of classifier prediction the Receiver Operating Characteristic (ROC) curve can be built. This graph is a technique for visualizing, organizing and selecting classifiers based on their performance. The ROC Curve relies on the fact that for a point the classifier predicts the label given a probability value. For default, the probability threshold for the assignment of a label is 50%. ROC curve aims at providing a more comprehensive view of the model performances for different classification thresholds. ROC graphs are two dimensional graphs in which TP rate is plotted on the Y axis and FP rate is plotted on the X axis. Each point in the space may represent a point prediction. The diagonal line represents the strategy of randomly guessing a class. In order to be better than a randomly guessing one, a classifier should be on the left side of the diagonal. Changing the probability threshold of the classifier, many points are predicted and, if printed on the plot, they give the ROC plot. Higher the area between the curve and the diagonal line, better is the quality of the classifier. [60]

Appendix I

K-means Clustering

Before discussing and giving a briefly review about the technique of k-means clustering, a review about cluster analysis is outlined.

I.1 Cluster Analysis

Cluster analysis is a set of techniques that aims at dividing data into groups, called clusters, that are meaningful and/or useful. The fields of applications of cluster analysis is wide. Talking about clustering for understanding, one can state that clusters are conceptually meaningful groups of objects that share a common characteristics and help in analysing and describing the world. Examples are found in Biology where taxonomy for classifying living things in clusters of different hierarchical grade has always been of first importance. Moreover nowadays clustering is applied in the analysis of genes with the novel availability of genetic information. Clustering analysis is also applied in web applications, in climate science, in psychology and medicine and of course in business. Finding patterns in atmospheric pressures, in diseases or in costumers is of huge advantage and cluster analysis plays a huge role. On the other hand, clustering for utility is meant to cluster data for a number of additional data analysis or data processing techniques. In this context cluster analysis can be used for example with the aim of summarization (so for reducing the amount of data to be analysed) or for compression (applied for example in image, sound or video data).

The basis of cluster analysis is the grouping of objects based on information found only in the data that describe the objects and their relationships. The similar and related objects will be accorporated within a group, on the contrary different or unrelated objects will not be accorporated. Of course there's the need to understand how to divide and to define clusters. In fact the same dataset can be divided into clusters in very different ways. Cluster analysis divides in two main categories (as for the entire machine learning world): supervised classification and unsupervised classification, based on the fact of dataset being

labelled or not. Supervised category has been already discussed in appendix H while in the follow, the focus is on unsupervised clustering.

There are different kinds of clustering.

- Partitional versus hierarchical: the first is a simply division of data while the second has subclusters, with a set of nested clusters that are organized as a tree.
- Exclusive versus Overlapping versus Fuzzy: the first kind of clustering happens when each object is assigned to a single cluster, for the second clustering, an object is simultaneously labelled into more than one cluster while for the third clustering, each object belongs to every cluster with a membership weight.
- Complete vs Partial: the first assign every object to a cluster while the second does not.

Additionally, cluster can be distinguished in different types.

- Well separated: the distance between any two points in different groups is larger than the distance between any two points within a group. They can have any shape.
- Prototype-based: the objects of a cluster are closer to the prototype (usually the centroid) that defines the cluster than to the prototype of any other cluster.
- Graph-based: the components of a cluster are objects connected together but not connected to the components of other clusters.
- Density-based: a cluster is defined as a dense region of objects that is surrounded by a region of low density (DBSCAN algorithm is an example of this) [61].

Now that an overview has been outlined, the algorithm used in this work for clustering, k-means clustering, is briefly reviewed.

I.2 K-means Clustering

K-means algorithm is a prototype-based and partitional clustering technique. The algorithm tries to find k clusters where k is an user-specified number of clusters represented by the centroids. Even if the history of k-means algorithm is very rich, the most common k-means method is the one proposed by Lloyd in [62]. The simplicity and the convergence time of this method are the main advantages of the method even if at cost of accuracy. Nevertheless k-means clustering remains one of the most used and famous tool for clustering. The Lloyd's algorithm and the k-means++ algorithm proposed in [63] are presented along with the strength and weaknesses points related to these.

I.2.1 K-means Clustering Algorithm

Kk-means clustering, or Lloyd's algorithm, is an iterative data-partitioning algorithm that assigns n observations to exactly k clusters defined by centroids with k chosen by the user before the algorithm starts. Hence, the algorithm goes in this way: it selects K points as initial centroids, it assigns each point to the closest centroid of each cluster forming k clusters; it then updates the centroids of each cluster based on the points of each cluster. The latter step is repeated until no point changes clusters or until the centroids remain the same. Hence the alg0rithm can be summarized in the following algorithm.

1. Select K points as initial centroids
2. repeat
   (a) Form K clusters by asignig each point to its closest centroid
   (b) Recompute the centroid of each cluster
3. until centroids do not change

The focus should be addressed to two main topics of the algorithm:

- The objective functions for the updating of the centroids
- The choice of initial centroids

I.2.2 Proximity measure and objective functions

In order to upgrade the centroids at each iteration, an objective function which can depend on the proximities of the points to one another or to the cluster centroid have to be specified. Moreover this objective function is based on the proximity measure. For example an usual objective function, which measures the quality of a clustering and which has to be minimized for each cluster, is given by the sum of the squared error (SSE):

$$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{\mathrm{x} \in C_i} dist\left(\mathrm{c}_i, \mathrm{x}\right)^2 \tag{I.1}$$

where $dist$ is the standard Euclidean distance between the point and the centroid of the cluster in the Euclidean space. In this case the proximity function can be considered to be the squared Euclidean distance. Given this, the updated centroid is taken as the mean of the i^{th} point:

$$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{x \in C_i} \left(c_i - x\right)^2 \tag{I.2}$$

with C_i the i^{th} and c_i the centroid of the i^{th} cluster. Now let's solve for the k^{th} centroid c_k which minimizes equation I.2 by differentiation the SSE, setting it equal to 0 and solving [61].

$$\begin{aligned}\frac{\partial}{\partial c_k}\text{SSE} &= \frac{\partial}{\partial c_k}\sum_{i=1}^{K}\sum_{x\in C_i}(c_i - x)^2 \\ &= \sum_{i=1}^{K}\sum_{x\in C_i}\frac{\partial}{\partial c_k}(c_i - x)^2 \\ &= \sum_{x\in C_k} 2\times(c_k - x_k) = 0\end{aligned}$$

$$\sum_{x\in C_k} 2\times(c_k - x_k) = 0 \Rightarrow m_k c_k = \sum_{x\in C_k} x_k \Rightarrow c_k = \frac{1}{m_k}\sum_{x\in C_k} x_k \tag{I.3}$$

It has been shown that the best centroid for minimizing the SSE of a cluster is the mean of the points in the cluster.

General speaking, different proximity function and objective function can be defined and used in the basic k-means algorithm [61] and [64]. For this project, Matlab's default options for k-means clustering have been chosen. Hence the SSE and the Squared Euclidean distance have been used in the algorithm.

I.2.3 Choosing initial Centroids

The main concept to be underlined is that choosing the proper initial centroids is the key step for k-means clustering. The simplest option of taking the initial centroids is by random choosing which is not of course not the best strategy. One can improve the results by re-running the code more than once. However this will come back with some drawbacks. Another idea is to perform a preliminary clustering phase on a random subset. The centroids of this sub-clusters are taken as initial centroids. Another effective approach for selecting initial centroids consists in the selection of a random point which is the first centroid, then for the following selections the centroids are taken as the farthest point from all the other centroids. Unfortunately outliers can be selected in this approach. [61]. An evolution of this strategy is the k-means++ algorithm proposed in [63] which is also the default option selected by Matlab because of the demonstrated improvement of such a method with respect to the Lloyd's one. This algorithm improves the performance of the clustering process in terms of SSE. The main idea of K-means++ is picking centroids incrementally until k centroids have been picked. At each step, each point has a certain probability to be chosen as new centroid. This probability is proportional to the square of its distance to its closest centroid. This avoids the problem of selecting outliers as centroids (since they are rare) solving the problem of the previously described methodology. The following algorithm shows the main steps for K-means++ initialization algorithm.

1. For the first centroid, pick one of the points at random
2. for i=1 to number of trials, do:
   (a) Compute the distance, $d(x)$, of each point to its closest centroid.
   (b) Assign each point a probability proportional to each point's $d(x)^2$.
   (c) Pick new centroid from the remaining points using the weighted probabilities
3. end for

I.2.4 Strengths and weaknesses

The simplicity of K-means method can be considered the most important point in favour for the algorithm. Moreover the efficiency of the method is not bad and there's the possibility to process a wide variety of data types (even if only the types of data for which there is the notion of centroid). On the other hand the algorithm has troubles clustering data that contains outliers and it cannot handle non-globular clusters or clusters of different sizes and densities. Some related techniques and some variants exist however even if usually they are more computational expensive. For this work k-means algorithm is used in the context of adaptive sampling when there is the necessity of selecting points among the candidate ones to be added to the training folds in the next iteration. Future work can implement different clustering algorithms in order to try to enhance the performance of the adaptive sampling methodologies.

Appendix J

Design of Experiment & Cross-Validation

J.1 Design of Experiment

Design of Experiment (DoE) is a procedure to plan and define conditions for performing controlled experimental trials. While the classical branch of DoE refers to physical experiments and it has very ancient roots, the modern branch of DoE is linked with the advent of computers. In fact researchers are increasingly replacing the time-consuming and monetary expensive physical experiments by faster and cheaper computer simulations which also allow experimentation not feasible in the practice. The primary aim of DoE is to decide which points should be simulated/analysed by the system. While the ancient branch of DoE was dealing with physical experiments whose main property was given by the stochastic nature related to a variety of unknowns and uncontrolled factors resulting in random errors, the modern DoE deals with deterministic situation. From this, since the application of the same techniques is not possible, new techniques had to be developed to cope with the new trend of computer simulations. It can be stated that the key aim of DoE is to generate sample points to fill the domain.

One can classify all the techniques into two main categories: static and adaptive. The first category ignores the system under study and focuses on the special distribution of sample points in the domain. The latter category is used by researcher taking into account the system under study which is integrated in the DoE technique. This approach has the aim to get the best design with the smallest sample set.

Among the static DoE techniques one can cite Monte Carlo sampling (MCS), stratified Monte Carlo sampling, Quasi-Monte Carlo sampling with its integrations given by Hammersley sequence, Halton sequence and Sobol sequence, Latin Hypercube design or sampling (LHS) and orthogonal array sampling.

In the first part of the work LHS has been chosen for initial design of the points to

be simulated by the physics – based model (OLGA). LHS is one of the most popular DoE techniques to overcome the issues associated with the MCS and its variations. LHS performs uniformly well over a range of dimensions and it can be used without too much drawbacks. LHS works in the following way. Let's consider an $\mathcal{N}$-dimensional design space $\mathcal{D}^{\mathcal{N}}$. Each dimension is dived into $\mathcal{K}$ equal bins of edge length $\frac{1}{\mathcal{K}}$. This results in $\mathcal{K}^{\mathcal{N}}$ hypercubes. Let's arrange $\mathcal{K}$ samples points as a $\mathcal{K} \times \mathcal{N}$ matrix $\mathbf{L} = \left[x^{(1)}, x^{(2)}, ...x^{(k)}\right]$ where each column represents a variable and each row represents a sample point. Then, $\mathbf{L}$ is an $\mathcal{N}$-dimensional LHS of size $\mathbf{K}$, if for each column of $\mathbf{L}$, no two elements in that column fall in the same bin [65].

J.2 Cross Validation

Cross validation (CV) is one of the resampling methods which are essential in the modern statistics even if they can be computationally expensive. These methods involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain additional information about the fitted model. Probably cross validation is the simplest and most widely used method for estimating prediction error of a learning model [66]. As reported before, an example of prediction error that can be used as performance parameter is the error done on the testing set. Given a learning method, in order to evaluate its performance, or to select an appropriate level of flexibility, an idea can be holding out a subset of the training observations from the fitting process and then applying the learning method to those computing the error every time.

A first kind of cross validation approach is called "the validation set approach" and it consists in randomly dividing the dataset into two parts: a training set for fitting the model and a validation set (or hold-out set) in order to compute the performance parameter through the trained model. The validation set approach is conceptually simple and is easy to implement. However the validation estimate of the performance parameter can be highly variable depending on which observations are included in the two subsets. Moreover in the validation approach only a subset at the time is used to fit the model. Since the learning methods tend to perform worse when trained with fewer observations, this suggests that the validation set error may overestimate the test error of the model fit with the entire data set.

In order to address this drawbacks, Leave-one-out cross-validation (LOOCV) has been proposed. Also this method involves splitting the set of observations into two parts. In this case the validation set is composed only by a single observation while the remaining observations make up the training set. Iteratively n (where n is the number of observations) models are trained and n performance parameter are collected. If the performance parameter is the mean square error (MSE), at the end n $MSEs$ are collected.

$$MSE_i = (\hat{y}_i - y_i)^2 \tag{J.1}$$

The LOOCV's estimate for the test MSE is the average of the estimated n test errors:

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{MSE}_i \tag{J.2}$$

LOOCV approach tends to not overestimate the test error rate as much as the validation set approach does. Moreover performing LOOCV multiple times always yield the same result: there is no randomness in the training/validation set splits. It's possible to state that LOOCV is a very general method and can be used with any kind of predictive modelling but the huge drawback is that training the model n times is not computational feasible if n is high and the learning model not very fast.

Another method, the reference on for this work, is the k-fold cross validation. The method is based on a random division of the set of observations into k groups of folds of approximately equal size. The first fold is treated as a validation set on which the performance index is computed, and the method is fit on the remaining $k-1$ folds. The process is repeated k times with k different validation sets. At the end k error's estimators are found. If the error's estimate is given by MSE, the k-fold CV estimate is computed by:

$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\mathrm{MSE}_i \tag{J.3}$$

Usually the number of folds is five or ten which is considered a good compromise [66] (in this project $k = 10$ has been chosen). This gives the K-fold cross validation the main advantage to be much more computational feasible than LOOCV because the learning procedure is repeated only 5 or 10 times and not the number of observations. When performing k-fold cross validation, the MSE for each iteration is different due to the different random split of the observations (of course the variation is less than than validation set approach). In this project 10-folds cross-validation has been used in order to get a robust estimation of the performance index for different kind of learning techniques.

Acronyms

AI	Artificial Intelligence
ML	Machine Learning
FA	Flow Assurance
WC	Water Cut
IR	Imbalance Ratio
EI	Expected Improvement
PI	Probability of Improvement
PR	Peng Robinson
GP	Gaussian Process
FP	False Positive
TP	True Positive
TN	True Negative
FN	False Negative
CV	Cross Validation
CDT	Cool-Down Time
SVM	support Vector Machines
GPR	Gaussian Process Regression
ANN	Artificial Neural Network
LHS	Latin Hypercube Sampling
IEA	International Energy Agency
NTT	No Touch Time
LTT	Light Touch Time
CIR	Circulation
MCS	Monte Carlo Sampling
MSE	Mean Square Error
KMC	K-Means Clustering
CEI	Constrained Expected Improvement
EFF	Expected Feasibility Function

GOR	Gas Oil Ratio
PVT	Pressure Volume Temperature
IPR	Inflow Performance Relationship
VFP	vertical Flow Performance
TPL	Tensioned Leg Platform
SKR	Soave Redlich Kwong
MEG	Mono Ethylene glycol
DEG	Diethylene glycol
TEG	Triethylene glycol
IFE	Institute for Energy Research
LMS	Least Mean Square
TPR	True Positive Ratio
TNR	True Negative Ratio
PPV	Positive Predictive Value
NPV	Negative Predictive Value
ROC	Receiver Operating Characteristic
SSE	Sum of Squared Errors
THI	Thermodynamic Inhibitors
OPEX	Operative Expenditure
RMSE	Root Mean Square Error
NRMSE	Normalized Root Mean Square Error
FPSO	Floating Production Storage and Offloading unit
RKHS	Reproducing Kernel Hilbert Space
DoE	Design of Experiment
DOCE	Design of Computer Experiment

Bibliography

[1] *World Energy Outlook 2018*. Tech. rep. International Energy Agency, Nov. 2018. URL: http://www.iea.org.

[2] Wenyuan Liu et al. "Assessment of hydrate blockage risk in long-distance natural gas transmission pipelines". In: *Journal of Natural Gas Science and Engineering* 60 (2018), pp. 256–270. ISSN: 1875-5100. DOI: https://doi.org/10.1016/j.jngse.2018.10.022. URL: http://www.sciencedirect.com/science/article/pii/S1875510018304931.

[3] Sampath K. Bukkaraju et al. *OPEX Savings in Pipeline Blockage Remediation - Lessons Learnt from West Africa Experience*. OTC. Houston, Texas, USA, Apr. 2018. DOI: 10.4043/28919-MS. URL: https://doi.org/10.4043/28919-MS.

[4] Yong Bai and Qiang Bai. "Chapter 12 - Subsea System Engineering". In: *Subsea Engineering Handbook*. Ed. by Yong Bai and Qiang Bai. Boston: Gulf Professional Publishing, 2010, pp. 331–347. ISBN: 978-1-85617-689-7. DOI: https://doi.org/10.1016/B978-1-85617-689-7.10012-3. URL: http://www.sciencedirect.com/science/article/pii/B9781856176897100123.

[5] Eni S.p.A. *Operating philosophy manual*. Manual for the development of an Offshore African field.

[6] *OLGA 2016 Version 2016.2 User manual*. Schlumberger. 2016.

[7] Tarek Ganat, Meftah Hrairi, and MNA Hawlader. "Validation of ESP Oil Wells Measured Parameters Using Simulation Olga Software". In: *IOP Conference Series: Materials Science and Engineering* 184 (Mar. 2017), p. 012057. DOI: 10.1088/1757-899X/184/1/012057.

[8] Rui Teixeira et al. "Reliability assessment with density scanned adaptive Kriging". In: *Reliability Engineering & System Safety* 199 (2020), p. 106908. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2020.106908. URL: http://www.sciencedirect.com/science/article/pii/S0951832019307434.

[9] Ove Bratland. *Pipe Flow 2: Multi-phase Flow Assurance*. Jan. 2010.

[10] Dwight Janoff, Nigel McKie, and Janardhan Davalath. *Prediction of Cool Down Times and Designing of Insulation for Subsea Production Equipment*. OTC. Houston, Texas, Jan. 2004. DOI: 10.4043/16507-MS. URL: https://doi.org/10.4043/16507-MS.

[11] J. Davalath and K. Stevens. "Cool-Down Thermal Performance of Subsea Systems Based on Gulf of Mexico Field Experience". In: (Jan. 2006). DOI: 10.4043/17972-MS.

[12] Xiaying Du et al. *Research on Prevention and Elimination of Hydrate After Subsea Wet-Gas Pipeline Shut-Down*. ISOPE. Sapporo, Japan, June 2018.

[13] Bjarne Grimstad. "Daily Production Optimization for Subsea Production Systems - Methods based on mathematical programming and surrogate modelling". PhD thesis. Oct. 2015. DOI: 10.13140/RG.2.1.4511.1447.

[14] Jeremie Bruyelle and Dominique Guerillot. "Proxy Model Based on Artificial Intelligence Technique for History Matching - Application to Brugge Field". In: Jan. 2019. DOI: 10.2118/198635-MS.

[15] Guilherme A. Polizel, Guilherme D. Avansi, and Denis J. Schiozer. *Use of Proxy Models in Risk Analysis of Petroleum Fields*. SPE. Paris, France, June 2017. DOI: 10.2118/185835-MS. URL: https://doi.org/10.2118/185835-MS.

[16] Denis Igorevich Zubarev. *Pros and Cons of Applying Proxy-models as a Substitute for Full Reservoir Simulations*. SPE. New Orleans, Louisiana, Jan. 2009. DOI: 10.2118/124815-MS. URL: https://doi.org/10.2118/124815-MS.

[17] Z. Di et al. "An Adaptive Pre-clustering Support Vector Machine for Binary Imbalanced Classification". In: *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2018, pp. 681–686. DOI: 10.1109/SMC.2018.00124.

[18] Hong Yan Hao et al. "Reliability Analysis Method Based on Support Vector Machines Classification and Adaptive Sampling Strategy". In: *Advances in Product Development and Reliability III*. Vol. 544. Advanced Materials Research. Trans Tech Publications Ltd, Sept. 2012, pp. 212–217. DOI: 10.4028/www.scientific.net/AMR.544.212.

[19] Anirban Basudhar and Samy Missoum. "An improved adaptive sampling scheme for the construction of explicit boundaries". In: *Structural and Multidisciplinary Optimization* 42.4 (Oct. 2010), pp. 517–529. ISSN: 1615-1488. DOI: 10.1007/s00158-010-0511-0. URL: https://doi.org/10.1007/s00158-010-0511-0.

[20] Jan N. Fuhg, Amélie Fau, and Udo Nackenhorst. "State-of-the-Art and Comparative Review of Adaptive Sampling Methods for Kriging". In: *Archives of Computational Methods in Engineering* (Aug. 2020). DOI: 10.1007/s11831-020-09474-6. URL: https://doi.org/10.1007/s11831-020-09474-6.

[21] P.O. Hristov et al. “Adaptive Gaussian process emulators for efficient reliability analysis”. In: *Applied Mathematical Modelling* 71 (2019), pp. 138–151. ISSN: 0307-904X. DOI: https://doi.org/10.1016/j.apm.2019.02.014. URL: http://www.sciencedirect.com/science/article/pii/S0307904X19300915.

[22] Siu-Kui Au and James L. Beck. “Estimation of small failure probabilities in high dimensions by subset simulation”. In: *Probabilistic Engineering Mechanics* 16.4 (2001), pp. 263–277. ISSN: 0266-8920. DOI: https://doi.org/10.1016/S0266-8920(01)00019-4. URL: http://www.sciencedirect.com/science/article/pii/S0266892001000194.

[23] Hong-Shuang Li and Zi-Jun Cao. “Matlab codes of Subset Simulation for reliability analysis and structural optimization”. In: *Structural and Multidisciplinary Optimization* 54.2 (Aug. 2016), pp. 391–410. ISSN: 1615-1488. DOI: 10.1007/s00158-016-1414-5. URL: https://doi.org/10.1007/s00158-016-1414-5.

[24] Ahmed Shokry and Antonio Espuna. “Applying Metamodels and Sequential Sampling for Constrained Optimization of Process Operations”. In: *Artificial Intelligence and Soft Computing*. Ed. by Leszek Rutkowski et al. Cham: Springer International Publishing, 2014, pp. 396–407.

[25] B. Echard, N. Gayton, and M. Lemaire. “AK-MCS: An active learning reliability method combining Kriging and Monte Carlo Simulation”. In: *Structural Safety* 33.2 (2011), pp. 145–154. ISSN: 0167-4730. DOI: https://doi.org/10.1016/j.strusafe.2011.01.002. URL: http://www.sciencedirect.com/science/article/pii/S0167473011000038.

[26] *The Schlumberger Oilfield Glossary*. Online resource. URL: https://www.glossary.oilfield.slb.com/.

[27] M. Blunt. *Imperial College Lectures In Petroleum Engineering, The - Volume 2: Reservoir Engineering*. The Imperial College Lectures in Petroleum Engineering. World Scientific Publishing Company, 2017. ISBN: 9781786342119. URL: https://books.google.it/books?id=24pEDwAAQBAJ.

[28] T. Ahmed. *Reservoir Engineering Handbook*. Elsevier Science, 2010.

[29] Shell. URL: https://www.shell.com.

[30] Boyun Guo, Shanhong Song, and Ali Ghalambor. “Offshore Pipelines (Second Edition)”. In: ed. by Boyun Guo et al. Second Edition. Boston: Gulf Professional Publishing, 2014, pp. 1–10. ISBN: 978-0-12-397949-0. DOI: https://doi.org/10.1016/B978-0-12-397949-0.00001-7. URL: http://www.sciencedirect.com/science/article/pii/B9780123979490000017.

[31] Yong Bai and Qiang Bai. "Chapter 15 - Hydrates". In: *Subsea Engineering Handbook*. Ed. by Yong Bai and Qiang Bai. Boston: Gulf Professional Publishing, 2010, pp. 451–481. ISBN: 978-1-85617-689-7. DOI: https://doi.org/10.1016/B978-1-85617-689-7.10015-9. URL: http://www.sciencedirect.com/science/article/pii/B9781856176897100159.

[32] Zheng Rong Chong et al. "Review of natural gas hydrates as an energy resource: Prospects and Challenges". In: *Applied Energy* 162 (Jan. 2016), pp. 1633–1652. DOI: 10.1016/j.apenergy.2014.12.061.

[33] Yannick Beaudoin et al. "Frozen Heat: A UNEP Global Outlook on Methane Gas Hydrates. Volume 1." In: (Mar. 2015).

[34] K Hartono et al. "Hydrate mitigation for subsea production multiphase pipeline by flow assurance approach". In: *IOP Conference Series: Materials Science and Engineering* 434 (Dec. 2018), p. 012188. DOI: 10.1088/1757-899x/434/1/012188.

[35] Dinesh Herath. "A probabilistic approach to assess hydrate formation and design preventive measures". In: 2016.

[36] E. G. Hammerschmidt. "Formation of Gas Hydrates in Natural Gas Transmission Lines". In: *Industrial & Engineering Chemistry* 26.8 (Aug. 1934), pp. 851–855. ISSN: 0019-7866. DOI: 10.1021/ie50296a010. URL: https://doi.org/10.1021/ie50296a010.

[37] W. Ertel and N.T. Black. *Introduction to Artificial Intelligence*. Undergraduate Topics in Computer Science. Springer London, 2011. ISBN: 9780857292995. URL: https://books.google.it/books?id=viqIIjGWqt0C.

[38] Jiankun Yang, Marcelo Igor Lourenco, and Segen F. Estefen. "Thermal insulation of subsea pipelines for different materials". In: *International Journal of Pressure Vessels and Piping* 168 (2018), pp. 100–109. ISSN: 0308-0161. DOI: https://doi.org/10.1016/j.ijpvp.2018.09.009. URL: http://www.sciencedirect.com/science/article/pii/S030801611830098X.

[39] Nadege Bouchonneau et al. "Thermal Insulation Material for Subsea Pipelines: Benefits of Instrumented Full-Scale Testing To Predict the Long-Term Thermomechanical Behaviour". In: (Jan. 2007). DOI: 10.4043/18679-MS.

[40] Andrea Molaro. "Analysis of an OffShore Pipeline: Fluid Properties and Flow Modelling". MA thesis. Politecnico di Milano, 2017.

[41] *National Oceanic and Atmospheric Administration Ocean Service*. Online resource. URL: https://oceanservice.noaa.gov/.

[42] Treccani. URL: http://www.treccani.it/enciclopedia.

[43] Marco Giuliani. "Computational intelligence: a hybrid technique for process modeling and production optimization of an oil field". MA thesis. Politecnico di Milano, 2017.

[44] URL: https://www.britannica.com.

[45] S. Skansi. *Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence*. Undergraduate Topics in Computer Science. Springer International Publishing, 2018. ISBN: 9783319730042. URL: https://books.google.it/books?id=5cNKDwAAQBAJ.

[46] Nils J. Nilsson. *Introduction to Machine Learning: An Early Draft of a Proposed Textbook. Pages 175-188. http://robotics.stanford.edu/people/nilsson/mlbook.ht ml.* 1996.

[47] T.M. Mitchell. *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, 1997. ISBN: 9780071154673. URL: https://books.google.it/books?id=EoYBngEACAAJ.

[48] C.C. Aggarwal. *Neural Networks and Deep Learning: A Textbook*. Springer International Publishing, 2018. ISBN: 9783319944630. URL: https://books.google.it/books?id=achqDwAAQBAJ.

[49] M.T. Hagan et al. *Neural Network Design*. Martin Hagan, 2014. ISBN: 9780971732117. URL: https://books.google.it/books?id=4EW9oQEACAAJ.

[50] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological Review* 65.6 (1958), pp. 386–408. DOI: 10.1037/h0042519. URL: https://doi.org/10.1037/h0042519.

[51] Martin T. Hagan et al. *Neural Network Design 2nd Edtion*. Martin Hagan, 2014. ISBN: 0-9717321-1-6.

[52] G. James et al. *An Introduction to Statistical Learning: with Applications in R*. Springer Texts in Statistics. Springer New York, 2013. ISBN: 9781461471387. URL: https://books.google.it/books?id=qcI%5C_AAAAQBAJ.

[53] Matlab. URL: https://www.mathworks.com/help/stats/gaussian-process-regression-models.html.

[54] Mohammad Shekaramiz, Todd K. Moon, and Jacob H. Gunther. *A note on kriging and gaussian processes*.

[55] C.E. Rasmussen et al. *Gaussian Processes for Machine Learning*. Adaptive computation and machine learning. MIT Press, 2006. ISBN: 9780262182539. URL: https://books.google.it/books?id=Tr34DwAAQBAJ.

[56] Standford.edu. URL: http://cs229.stanford.edu/section/cs229-gaussian_processes.pdf.

[57] M. Kuhn and K. Johnson. *Applied Predictive Modeling*. SpringerLink : Bücher. Springer New York, 2013. ISBN: 9781461468493. URL: https://books.google.it/books?id=xYRDAAAAQBAJ.

[58] M. Bramanti, C.D. Pagani, and S. Salsa. *Analisi matematica 2*. Zanichelli, 2009. ISBN: 9788808122810. URL: https://books.google.it/books?id=GBIEngEACAAJ.

[59] Christian Bauckhage and Daniel Speicher. *Lecture Notes on Machine Learning: The Karush-Kuhn-Tucker Conditions (Part 1)*. Aug. 2019.

[60] Tom Fawcett. "ROC Graphs: Notes and Practical Considerations for Researchers". In: *Machine Learning* 31 (Jan. 2004), pp. 1–38.

[61] Pang-Ning Tan et al. *Introduction to Data Mining (2nd Edition)*. 2nd. Pearson, 2018. ISBN: 0133128903.

[62] S. Lloyd. "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137. DOI: 10.1109/TIT.1982.1056489.

[63] David Arthur and Sergei Vassilvitskii. "K-Means++: The Advantages of Careful Seeding". In: vol. 8. Jan. 2007, pp. 1027–1035. DOI: 10.1145/1283383.1283494.

[64] Matlab. URL: https://www.mathworks.com/help/stats/gaussian-process-regression-models.html.

[65] Sushant S. Garud, Iftekhar A. Karimi, and Markus Kraft. "Design of computer experiments: A review". In: *Computers & Chemical Engineering* 106 (2017). ESCAPE-26, pp. 71–95. ISSN: 0098-1354. DOI: https://doi.org/10.1016/j.compchemeng.2017.05.010. URL: http://www.sciencedirect.com/science/article/pii/S0098135417302090.

[66] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York, 2013. ISBN: 9780387216065. URL: https://books.google.it/books?id=yPfZBwAAQBAJ.