Adaptive swarm optimisation assisted surrogate model for pipeline leak detection and characterisation.

ADEGBOYE, M.A.

2023

Adaptive Swarm Optimisation Assisted Surrogate Model for Pipeline Leak Detection and Characterisation

Adegboye, M. A.

2023

PhD 2023

Adaptive Swarm Optimisation Assisted Surrogate Model for Pipeline Leak Detection and Characterisation

Mutiu Adesina Adegboye

A Thesis submitted in partial fulfilment of the

requirements of the

Robert Gordon University

for the degree of Doctor of Philosophy

February 2023

MUTIU ADESINA ADEGBOYE

Adaptive Swarm Optimisation Assisted Surrogate Model for Pipeline Leak Detection and Characterisation

Supervisor Team: Dr Aditya Karnik
Dr Wai-keung Fung
Professor Radhakrishna Prabhu

School of Engineering, Robert Gordon University,
The Sir Ian Wood Building, Riverside East,
Garthdee Road, AB10 7GJ,
Aberdeen,
United Kingdom.

Declaration

I declare that this thesis, except where otherwise stated, is based on my own work. To the best of my knowledge and belief, this report contains no material previously published or written by another person except where due reference has been made. This thesis has not been submitted before for any degree or examination at any other University.

Dedication

This dissertation is dedicated to Almighty Allah, The All-Knowing, The All-Wise. A special feeling of gratitude to my parents, my ever-supportive wife, my adorable daughter, and my teachers.

Acknowledgement

I would like to offer my profound gratitude to my wonderful and supportive supervisory team: Dr Aditya Karnik for his guidance, encouragement and helpful suggestion throughout my PhD Journey; Dr Wai-keung Fung for starting me on the right footing, giving me inspiring suggestions and zest for perfection that brought the best out of me; Professor Radhakrishna Prabhu for his contributions in making this research a success.

I deeply appreciate the Federal Republic of Nigeria for allowing me to study for my Doctorate through the Petroleum Technology Development Fund (PTDF). I will not forget to thank Dr Lazaro Molina-Espinosa of Instituto Mexicano del Petróleo, Mexico, for making their datasets available for me despite the challenge he faced during the COVID-19 lockdown.
I will not forget to say a big thanks to my Parents, Brothers and Sisters for all your prayers and encouragement all throughout my study. I appreciate my loving wife Karamot for her patience and support and for holding our girl without complaining, even when I had to stay overnight at the University. To my little girl, Hikmah, I appreciate your patience and understanding.

I must say a big thank you to the support staff in the School of Engineering, particularly Dr Ros Shanks and Kirsty Stevenson. I appreciate the lecturers, friends, colleagues, technicians in the School of Engineering, and the library staff. Gratitude also goes to my friends Dr Auwalu Mohammed, Dr Maryam Heidarian, Dr Olawale Sanusi, Dr Aliyu Ibrahim, Dr Mansur Hamma-Adama, Dr Adamu Ali-Gombe, Badmus Taofeeq, Idris Zakariyya, Sani Lawal and Owolabi Abideen. A final thank you to all those who made this research possible in one way or another.

Abstract

Pipelines are often subject to leakage due to ageing, corrosion, and weld defects. It is difficult to avoid pipeline leakage as the sources of leaks are diverse. Various pipeline leakage detection methods, including fibre optic, pressure point analysis and numerical modelling, have been proposed during the last decades. One major issue of these methods is distinguishing the leak signal without giving false alarms. Considering that the data obtained by these traditional methods are digital in nature, the machine learning model has been adopted to improve the accuracy of pipeline leakage detection. However, most of these methods rely on a large training dataset for accurate training models. It is difficult to obtain experimental data for accurate model training. Some of the reasons include the huge cost of an experimental setup for data collection to cover all possible scenarios, poor accessibility to the remote pipeline, and labour-intensive experiments. Moreover, datasets constructed from data acquired in laboratory or field tests are usually imbalanced as leakage data samples are generated from artificial leaks. Computational fluid dynamics (CFD) offers the benefits of providing detailed and accurate pipeline leakage modelling, which may be difficult to obtain experimentally or with the aid of analytical approach. However, CFD simulation is typically time-consuming and computationally expensive, limiting its pertinence in real-time applications. In order to alleviate the high computational cost of CFD modelling, this study proposed a novel data sampling optimisation algorithm called Adaptive Particle Swarm Optimisation Assisted Surrogate Model (PSOASM) to systematically select simulation scenarios for simulation in an adaptive and optimised manner. The algorithm was designed to place a new sample in a poorly sampled region or regions in parameter space of parametrised leakage scenarios, which the uniform sampling methods may easily miss. This was achieved using two criteria: population density of the training dataset and model prediction fitness value. The model prediction fitness value was used to enhance the global exploration capability of the surrogate model, while the population density of training data samples is beneficial to the local accuracy of the surrogate model.

The proposed PSOASM was compared with four conventional sequential sampling approaches and tested on six commonly used benchmark functions in the literature. Different machine learning algorithms are explored with the developed model. The effect of the initial sample size on surrogate model performance was evaluated. Next, pipeline leakage detection analysis with much emphasis on a multiphase flow system was investigated in order to find the flow field parameters that provide pertinent indicators in pipeline leakage detection and characterisation. Plausible leak scenarios which may occur in the field were performed for the gas-liquid pipeline using a 3-Dimensional RANS CFD model. The perturbation of the pertinent flow field indicators for different leak scenarios is reported, which is expected to help in improving the understanding of multiphase flow behaviour induced by leaks. The results of the simulations were validated against the latest experimental and numerical data reported in the literature. The proposed surrogate model was later applied to pipeline leak detection and characterisation. The CFD modelling results showed that fluid flow parameters are pertinent indicators in pipeline leak detection. It was observed that upstream pipeline pressure could serve as a critical indicator for detecting leakage, even if the leak size is small. In contrast, the downstream flow rate is a dominant leakage indicator if the flow rate monitoring is chosen for leak detection. The results also reveal that when two leaks of different sizes co-occur in a single pipe, detecting the small leak becomes difficult if its size is below 25% of the large leak size. However, in the event of a double leak with equal dimensions, the leak closer to the pipe upstream is easier to detect. The results from all the analyses demonstrate the PSOASM algorithm's superiority over the well-known sequential sampling schemes employed for evaluation. The test results show that the PSOASM algorithm can be applied for pipeline leak detection with limited training datasets and provides a general framework for improving computational efficiency using adaptive surrogate modelling in various real-life applications.

Keywords: Adaptive surrogate model, CFD modelling, Data optimisation, Leak detection, Machine learning, Multiphase flow, Particle swarm optimisation, Pipeline leakage, Multiphase flow

List of Publications arising from this study

Journal articles:

1. **Adegboye, M. A.,** Fung, W. K., & Karnik, A. (2019). Recent advances in pipeline monitoring and oil leakage detection technologies: Principles and approaches. Sensors, 19(11), 2548.
2. **Adegboye, M. A.,** Karnik, A., & Fung, W. K. (2021). Numerical study of pipeline leak detection for gas-liquid stratified flow. Journal of natural gas science and engineering, 94, 104054.
3. **Adegboye, M. A.,** Karnik, A., Fung, W. K., and Prabhu, R. An adaptive surrogate modelling using swarm optimisation and its application in pipeline leakage detection and characterisation. Manuscript under preparation.
4. **Adegboye, M. A.,** Karnik, A., Fung, W. K., and Prabhu, R. An adaptive PSO-assisted surrogate modelling for computationally expensive problems. Manuscript under preparation.

Conference Proceedings/Seminar Presentations:

1. **Adegboye, M. A.,** Karnik, A., Fung, W. K. (2020). Increasing leak detection standards and innovations with pipeline leak detection methods to improve reliability. Practical applications of the latest technologies, techniques and innovations for oil and gas pipeline leak detection, January 29-30, Houston, Texas, USA.
2. **Adegboye, M. A.,** Karnik, A., Fung, W. K. (2021). Adaptive surrogate modelling for pipeline leak detection and localisation. An abstract submitted to Subsea Expo Conference, February 23-25, Aberdeen.

3. **Adegboye, M. A.,** Karnik, A., Fung, W. K., and Prabhu (2022). Pipeline leakage detection and characterisation with adaptive surrogate modelling using particle swarm optimisation. 9th International Conference on Soft Computing & Machine Intelligence (ISCMI), November 26-27, 2022, Toronto, Canada.
4. **Adegboye, M. A.,** Karnik, A., Fung, W. K., and Prabhu (2022). Numerical modelling of pipeline leak detection and characterisation using PSO-assisted surrogate model. 35th Scottish Fluid Mechanics Meeting, May 26th, SAMS, Obah.

Posters /Technical Presentation Session:

1. **Adegboye Mutiu Adesina** (2019) Oil Leak Prediction Model for Horizontal Pipeline in a Subsea Environment. Presented at Graduate School Seminar, Robert Gordon University on August 23, 2019.
2. **Adegboye Mutiu Adesina** (2020) Adaptive surrogate model for subsea pipeline leak detection. Presented at the School of Engineering Research Group Monthly Seminar, Robert Gordon University on November 25, 2020.
3. **Adegboye Mutiu Adesina** (2020) Numerical modelling of two-phase gas-liquid pipeline leakage detection. Presented at the Petroleum Engineering Research Group (PERG) Monthly Seminar on 30th July 2020.

Funding organisation

This research was funded by Petroleum Technology Development Fund (PTDF), Abuja, Nigeria. Grant number PTDF/ED/PHD/AMM/1385/18.

Contents

1.1 List of Figures

1.2 List of Tables

1.3 List of Appendices

1.4 List of Abbreviations

CFD	Computational fluid dynamics
PSOASM	Particle Swarm Optimisation Assisted Surrogate Model
ANN	Artificial Neural Network
DoE	Design of Experiment
MC	Monte-Carlo
SMCS	Stratified Monte Carlo sampling
LHS	Latin Hypercube Sampling
OLHS	Optimal LHS
HSS	Hammersley sequence sampling
QRLD	Quasi Random Low Discrepancy
GSA	Global Sensitivity Analysis
QBC	Query by Committee
LOOCV	Leave-one-out cross-validation
SA	Simulated Annealing
GA	Genetic Algorithm
PSO	Particle Swarm Optimisation
CPU	Central Processing Unit
GPU	Graphic Processing Unit
TLBO	Teaching Learning Based Optimisation
GSA	fuzzy surrogate-assisted
LSA	local surrogate-assisted
GSA	global surrogate-assisted
CD	Crowding Distance CD
SVMs	Support Vector Machines
MLP	Multilayer Perceptron
DT	Decision Trees
RF	Random Forest
OOB	out-of-bag
PR	Polynomial regression
MSE	Mean Square Error
RMSE	Root Mean Square Error
MAE	Mean Absolute Error

VOF	Volume of Fluid
SST	Shear Stress Transport
CSF	Continuum Surface Force
HRIC	High-Resolution Interface Capturing
IOGP	International Association of Oil and Gas Producers

Chapter 1 Introduction

1.1 Research Context and Motivation

The demand for energy is increasing worldwide. Completely substituting the hydrocarbon power source with renewable technology is not yet feasible. The 2020 annual energy report of the Energy Information Administration revealed that the natural gas supply as of 2020 is about 30% of the world's energy, and consumption of natural gas, petroleum, and other liquids will continue to increase until 2050 (Kim *et al*., 2021). The use of pipelines has extended over time because it provides an effective system to increase energy supply and has been considered the safest and the most economical and efficient means of petroleum transportation *(*Muggleton *et al*., 2020). Despite pipelines being considered the cheapest and safer than other modes of transportation, they are still subject to leakage due to ageing, corrosion, and weld defects. Therefore, a leak in the pipeline remains a major concern for both safety and contamination in daily operations (Li *et al.*, 2019). According to Li *et al.*(2018), the likelihood of pipeline developing leaks increases with age and service time and the consequence may lead to financial losses, human casualties and extreme environmental contamination, particularly when the leakage is not detected in a timely way (Bolotina *et al*., 2018). The cause of the pipeline damage varies. Figure 1.1 shows a pie chart that illustrates statistics of the major causes of pipeline leakage, including pipeline corrosion, human negligence, defects befalls during installation and erection work, flaws during manufacturing, and external factors (Bolotina *et al*., 2018).

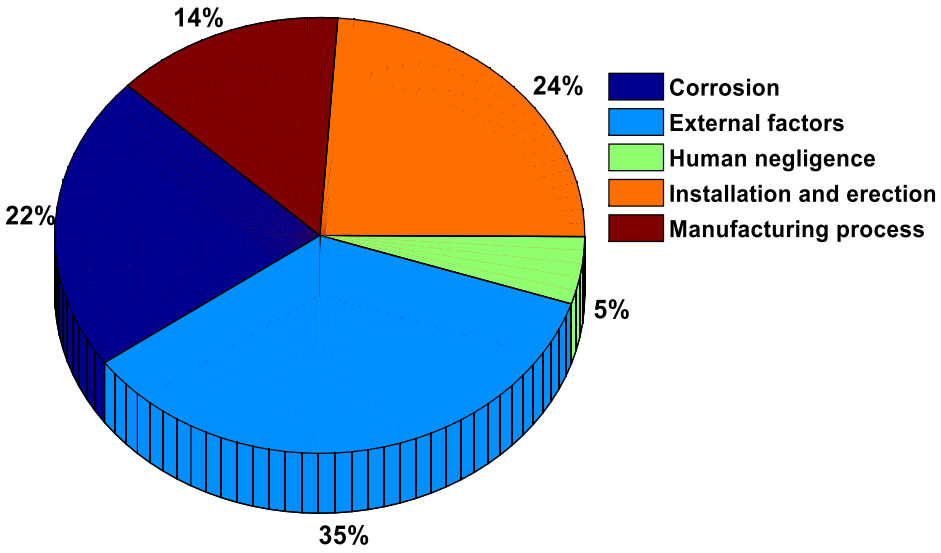


Figure 1-1: A pie chart for the statistics of the sources of pipeline failure. Data is obtained in (Bolotina et al., 2018).

Based on these statistics, incidents of pipeline leakage are hard to avoid as the sources of failures are entirely diverse. However, in order to reduce the impacts of oil spillage in society, it is essential to monitor pipelines for timely detection of leakage, as early detection of leaks will aid in quick response to stop oil discharge and proper pipeline maintenance. Many methods have been reported for pipeline leak detection and characterisations. The existing methods are typically classified into external, visual or biological and internal methods (Kim et al., 2021). The external methods include acoustic sensing (Mahmutoglu and Turk, 2018), ground penetration radar (Hoarau et al., 2017), fibre optic sensors (Png et al., 2018), etc., while the visual or biological methods include trained personnel, drones and trained Dogs (Q. Li et al., 2016). The internal methods such as mass-volume balance (Martins & Seleghim, 2010), pressure point analysis (He et al., 2017) and dynamics modelling (Kim et al., 2021) employed in pipe flow parameters such as pressure and flow rate to recognise anomalies that indicate a fluid loss. The internal methods have been regarded as promising approaches for detecting pipeline leakage (Pérez-Pérez et al., 2021). The major issue of traditional pipeline leakage detection methods is distinguishing the leak signal without giving false alarms (Feng & Zhang, 2006). Considering that the data obtained by these conventional methods are digital in context, the

machine learning model has been adopted to improve the accuracy of pipeline leakage detection. Machine learning algorithms are gaining widespread use in pipeline leak detection and characterisation as they can bring many benefits, such as low cost, accuracy, and timely pipeline leak prediction. However, existing studies typically depend on a large training dataset, which may not be possible to acquire in a physical pipeline due to the damaging impact and costs associated with experimentations. Moreover, datasets acquired in the laboratory or field tests are usually imbalanced as the data labelled leak is scarce in the field, and the cost of introducing and cleaning artificial leaks in controlled laboratory settings is usually high. Therefore, thorough measures are required to avoid pollution (Kim *et al*., 2021). Fortunately, dynamics modelling, also known as numerical modelling, can circumvent these challenges and has been widely used in the industry and by the research community (Ebrahimi-Moghadam *et al*., 2018; Fu *et al*., 2020; Yang *et al.*, 2017).

Numerical modelling provides an easy approach to creating and analysing models that mimic the actual pipelines in the field. The method can fit in various elements such as pipeline material, length, diameter, fluid type and inspect the relationship connecting the flow parameters such as pressure, flow rate, and temperature in the presence and absence of leakage through computationally intensive simulation. The drawback of numerical simulation, however, is the computational cost. A realistic fluid dynamics simulation can take days or even weeks to complete despite advancements in high-performance computing (Huang *et al*., 2021; C. Xu *et al*., 2015). The motivation for this study stems from the use of computational fluid dynamics modelling and machine learning algorithms to investigate fluid flow behaviours induced by leaks in a pipeline. Subsequently, develop a surrogate model to provide a fast-to-run approximation model for pipeline leakage prediction. A surrogate modelling approach that uses an effective sampling strategy and interpolation schemes will reduce the number of simulation trials required to construct a prediction model for computationally expensive problems like CFD simulation of pipelines without sacrificing model accuracy.

1.2 Gap in Knowledge

Large datasets are usually required for building machine learning models for engineering applications such as pipeline leakage detection and characterisations. However, these datasets are not always easy to acquire due to the cost, time, poor accessibility to the physical pipeline and so on (Kim *et al.*, 2021). Numerical modelling, also suggested as a good alternative for data generation, is computationally costly. Two possible solutions to this challenge are either to use coarse mesh to speed up the simulation, but it gives inaccurate results, which is usually unacceptable *(Sun et al.*, 2011) or to find a suitable way to minimise the simulation trials to build the machine learning model without sacrificing model accuracy. Therefore, this study proposed a surrogate model to systematically select simulation trials for CFD modelling. The surrogate model is an approximation function that mimics the behaviours of the original function but can be evaluated faster (Golzari *et al*., 2015). This approximation function is built by performing simulations at key points in the parameter space, analysing the outcomes and building a model that approximates the samples and the overall system behaviours well. A general description of the developing surrogate model is illustrated in Figure 1.2. The process consists of two steps: the first step is parameter space sampling, wherein a set of points is generated over the design space. The second stage is surrogate model fitting, in which the relevant black-box function is evaluated at each sample point to fit the surrogate model over the whole parameter space. It is important to highlight that the surface on the right side of Figure 1.2 is not searching for the optimal value.

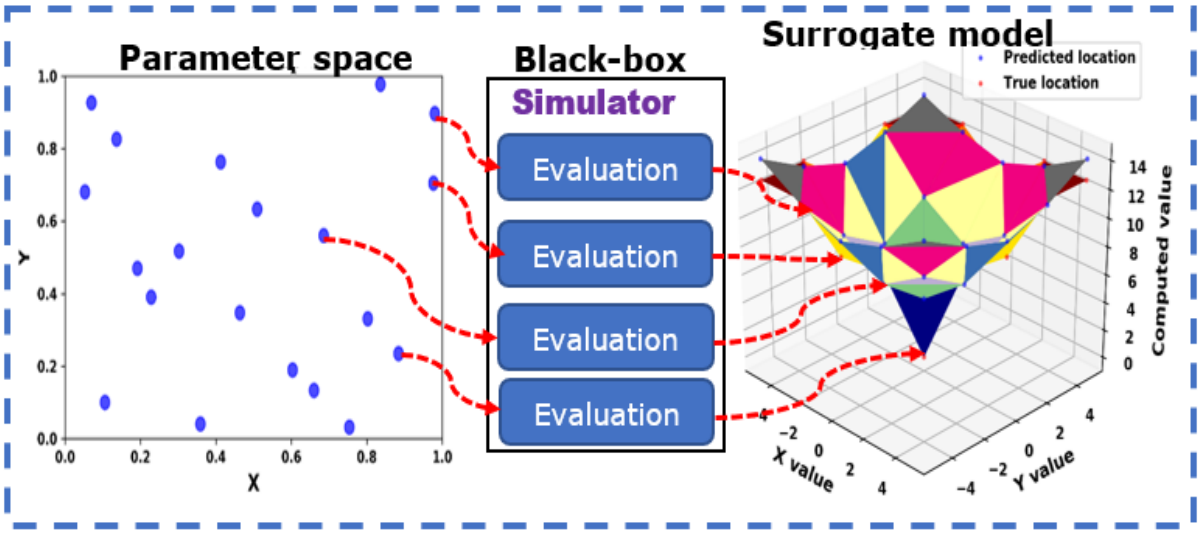
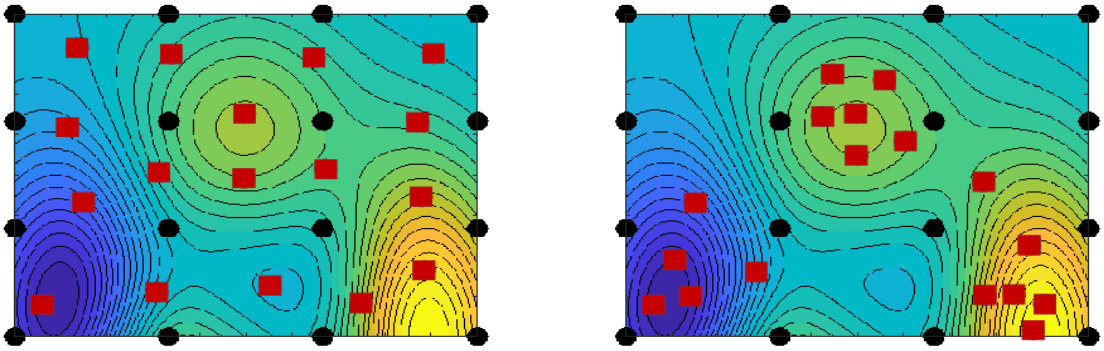


Figure 1-2: Surrogate modelling process

The accuracy of the surrogate model is highly dependent on two factors. The first factor is the number of sample points. If the density of samples in parameter space are not large enough, the error estimation results may be high. On the contrary, many samples mean additional computationally demanding simulations, which may be difficult to obtain. The second factor is the thoroughness of coverage in parameter space. It was reported by Fuhg et al.(2020) that proper sample point selection would reduce the number of training datasets as much as possible without affecting the accuracy of the surrogate model. In this regard, sequential sampling has become a keen research in surrogate modelling. This sampling technique is classified into space-filling and adaptive design, as illustrated in Figure 1.3.



(a) Space-filling design (b) Adaptive design

Figure 1-3: Illustration of adaptive sampling strategies adapted from Fuhg *et al.*(2020). Initial sample points in black dots and sequentially added samples in red squares.

The space-filling method is designed to spread the samples evenly in an iterative manner in parameter space, as shown in Figure 1.3(a). On the other hand, the adaptive design depicted in Figure 1.3(b) uses information extracted from the model prediction fitness to place the new sample in regions of high saliency and relevance to application of interest. As reported in the literature, the adaptive design provides a better approach to accelerating simulation-based optimisation tasks than the space-filling design (Fuhg et al., 2020; Shen et al., 2020; Tian et al., 2020). The key issue is the lack of a method to intelligently select a minimal number of parameterised simulation scenarios for data generation in surrogate modelling without sacrificing model accuracy. To tackle this problem, a novel data sampling optimisation algorithm, named adaptive particle swarm optimisation assisted surrogate model (PSOASM) is proposed in this study. The proposed PSOASM model incorporates the information (prediction fitness) extracted during the previous iteration and population density (also known as population distance) of the samples to place a new sample in a sparsely sampled region or regions in the parameter space of parametrised leakage scenarios.

The transportation of two-phase gas-liquid is mostly done through the pipeline that connects production facilities, refineries and in some cases, nuclear and chemical industries (de Vasconcellos Araújo et al., 2013; Knotek et al., 2021). Timely detection of pipeline leakage is important for preventing disasters in the nature and decreasing losses for industries. Various studies have shown that the flow field parameters provide pertinent indicators in pipeline leakage detection (Ebrahimi-Moghadam et al., 2018; Kim et al., 2021; Martins & Seleghim, 2010; Pérez-Pérez et al., 2021). However, not much is known regarding the pipeline conveying more than one phase at a time. A recent study by Behari et al.(2020) noted that the available leak detection techniques in the open literature fail to satisfactorily address multiphase pipeline leakage phenomena. There is no guarantee that the information available for single pipeline leak cases can be extended to the multiphase pipeline system. This is evident that more insight into pipelines transporting more than one phase is needed to understand

multiphase pipeline leakage thoroughly. In this study, plausible leak scenarios are investigated to improve the understanding of the multiphase pipeline leak prediction system. A comprehensive assessment of different leak conditions was performed for a gas-liquid pipeline using 3-D numerical modelling. Simultaneous flow of gas and liquid in a horizontal or slightly inclined pipeline often results in different flow patterns, such as annular, slug and stratified flows (Garbai & Sánta, 2012). This study considers stratified flow for investigation because it was reported as a basic flow pattern for gas-liquid and liquid-liquid two-phase flow pattern that is frequently encountered in various important industrial processes (Barmak et al., 2016) and long-distance horizontal flow lines such as steam, natural gas and oil flow, in petrochemicals, power generation and process plants (Ali, 2017; Cheng et al., 2020; Vlachos et al., 1999).

1.3 Research Aims and Objectives

The aim of this thesis is to develop an adaptive swarm optimisation-assisted surrogate model for pipeline leak detection using the minimum number of simulation trials to provide substantial computational speedup for rapid model construction without the sacrifice of model accuracy. To achieve this aim, the thesis objectives are:

1. To develop a novel parameter space sampling optimisation algorithm using particle swarm optimisation theory to systematically select simulation scenarios for numerical simulation in an adaptive and optimised manner.
2. To develop a surrogate model that forms the approximation function using an adaptive sampling model developed in objective 1 to optimise the surrogate training dataset.
3. To numerically study the effect of two-phase gas-liquid stratified flow behaviour induced by leaks in a horizontal pipeline.
4. To apply the surrogate model developed in objective 2 to the pipeline leakage detection and characterisation using flow field parameters obtained in objective 3.

5. To evaluate and compare the performance of the proposed model in terms of computational efficiency and accuracy with conventional sequential designs by considering several benchmark problems and 3-D pipeline leakage models.

1.4 Contributions of the thesis

This research work has been carried out to contribute to the body of knowledge in the following key areas:

- A novel computational framework for surrogate model development with intelligent sampling in parameter space using PSO for complex engineering applications like pipeline leakage detection and characterisation.

- A technique that incorporates two criteria (surrogate fitness value and population density of sample points) is proposed in data sampling optimisation.

- A systematic analysis of two-phase gas-liquid pipeline leakage behaviours using 3D CFD simulation. The perturbation of the pertinent flow field indicators provides a better understanding of multiphase flow behaviour induced by leaks and guidelines, which can be helpful for risk assessment and improving the emergency management level.

- Demonstration of pipeline leakage detection and characterisation using the proposed surrogate model. The practical application of the proposed surrogate model allows for pipeline leak prediction with a limited training dataset.

1.5 Research Methodology

The study started with a general review of the surrogate modelling. These initial steps offered an understanding of the technological trend behind surrogate modelling. A literature review on pipeline leakage detection and characterisation with specific emphases on multiphase gas-liquid two-phase flow and in pipes parameters associated with it was conducted. The review enables the identification of gaps in knowledge, the basis for appropriate method selection, strengths and weaknesses of the existing pipeline leakage detection methods, research gaps and open issues for the development of reliable pipeline leak detection technologies. Based on the limitation of the surrogate model identified in the literature, an adaptive surrogate modelling method that provides a framework for reducing computationally expensive problems was proposed. A numerical analysis of gas-liquid stratified pipeline leakage was conducted, covering different leakage scenarios that may occur on the 3D pipeline was performed. This analysis was conducted using ANSYS Fluent 18.1. The developed surrogate model was applied to 3D pipeline leakage detection and characterisation. In addition, verification and validation of the surrogate model were simulation data of the pipeline leakage obtained from the CFD model and experiment data from the literature. A schematic overview of the research methodology is presented in Figure 1.4.

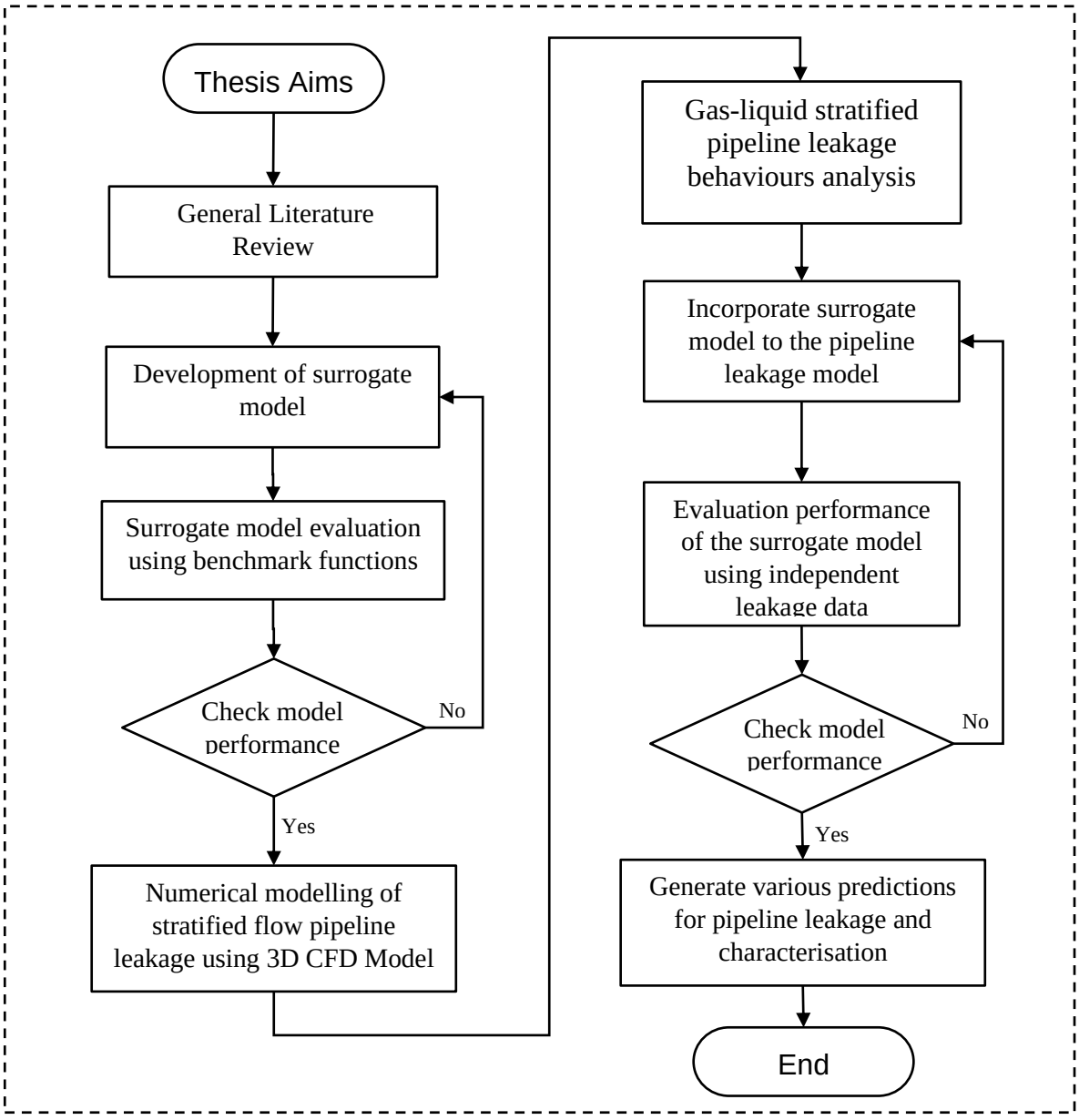


Figure 1-4: Schematic overview of research methodology

1.6 Thesis Structure

The remainder of the thesis is structured in six chapters given as follows:

Chapter 2: This chapter presents a comprehensive review of the background literature on the surrogate model to establish the knowledge gap in the field. This chapter also presents a review of related works on pipeline leakage detection and characterisation, emphasising the numerical modelling approach—the chapter ends with a review of modelling approaches for the multiphase system.

Chapter 3: This chapter proposes an adaptive surrogate model to optimise the machine learning training dataset involving computationally expensive simulations. The proposed model introduced two criteria, namely surrogate fitness value and population of the sample points to select new points for evaluation. The proposed surrogate model performance was evaluated on the various benchmark machine learning algorithms. This chapter further reports the result of the proposed surrogate model for different initial sample sizes.

Chapter 4: This chapter presents the computational fluid dynamics method with the governing equations and simulation approaches used for pipeline leakage detection and characterisation analysed in this study. The validation results of the simulations against the latest experimental and numerical data reported in the literature are also presented in this chapter.

Chapter 5: This chapter covers the simulation results for leak size effect and characterisation on two-phase gas-liquid stratified flow. The analysis is developed to account for the leak magnitude, longitudinal and axial leak positions, and multiple leakages.

Chapter 6: This chapter presents the practical application of the proposed surrogate model on single-phase and multiphase pipeline leakage detection and characterisation. The performance of the developed model with the several conventional sequential designs and experimental data obtained from the literature are also presented in this chapter.

Chapter 7: This chapter gives a summary of the key findings emerging from the preceding chapters and recommendations for future studies that could complement and extend the finding of this work.

Chapter 2 Literature Review

This chapter reviews elements of the literature relevant to the study. A general review of surrogate modelling, including the various surrogate modelling techniques are presented. The application of the design of the experiment for the surrogate model and comprehensive analysis of the popular approaches and their application to computationally expensive problems are also presented in this chapter. A Review of related works on pipeline leakage detection and characterisation, including dynamics modelling of single-phase and multiphase pipeline leakage detection and gaps in the existing studies, are presented. The major fluid flow characteristics in the pipeline are also discussed in this chapter. The chapter concludes by reviewing different flow patterns commonly encountered in the horizontal or slightly inclined multiphase pipeline.

2.1 Introduction

The term surrogate model can be regarded as an approximation of the original specialised models (Han and Zhang, 2012). The surrogate model is built to simplify the expensive model that cannot be easily simulated or experimented. The goal of developing the surrogate model is to provide a simpler and computationally efficient model that approximates the specified output of a complex model and its input parameters (Han and Zhang, 2012). The use of the surrogate model to replace complex phenomena has been studied by various researchers in different science and engineering disciplines, and it has generated satisfactory prediction accuracy (Bhosekar and Ierapetritou, 2018). Therefore, the execution of surrogates shall result in momentous saving of computational time, energy and resources. A description of the surrogate model can be presented most simply as follows: Assuming the engineering analysis that consists of complex computer code supply $\mathbf{x}$ as vectors of the design variables (inputs) and computing $\mathbf{y}$ as the vectors of the response variables (outputs). If the true function of the original computer code is given as:

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

then, a surrogate model, which is also known as metamodel or approximation model, can be derived as:

$$\hat{\boldsymbol{y}} = g(\mathrm{x}) \tag{2.2}$$

where g is the approximation function of $f(\mathrm{x})$, therefore, the original outputs $\mathbf{y}$ become:

$$\mathbf{y} = \hat{\boldsymbol{y}} + \varepsilon \tag{2.3}$$

where ε represents the approximation error.

The essence of developing the surrogate model is to determine a function g of a set of $\mathbf{x}$ input variables from the limited number of sample data obtained (by running numerical simulations or physical experiments) from the original model $\mathbf{y} = f(\mathbf{x})$, and then use any of the surrogate model techniques such as polynomial regression, radial basis function, artificial neural network, etc. to create an approximation of the expensive computer problem. This approximation function can replace the complete computer simulation while offering (Simpson *et al.*, 2001):

i. A better understanding of the relationship between input variables $(\mathbf{x})$ and the response variables $(\mathbf{y})$.
ii. Easier integration of the domain-dependent computer code.
iii. Fast analysis tools for optimisation and exploration of the design space.

2.2 Surrogate Model Techniques

Different methods have been used to develop a surrogate model, particularly when dealing with problems where obtaining samples is computationally expensive. These methods include Polynomial Regression (Han & Zhang, 2012), Kriging method (Bartz-Beielstein et al., 2016) and Artificial Neural Network (Ding et al., 2015). Other statistical methods, such as Multivariate Adaptive Regression Splines (Chua et al., 2021; W. Zhang et al., 2016) and inductive learning (S. S. Jin, 2021) have also gained insight into a number of studies. The illustration of these techniques is described as follows:

2.2.1 Polynomial Regression

Polynomial regression is one of the most commonly used methods for designing a response surface model. It has been widely employed because of its convergence speed and smoothing capability while maintaining the global trend of the variation, making it very robust and, therefore, well-appropriate for optimisation problems in engineering design (Han and Zhang, 2012). The first-order polynomial can be employed for the low curvature and is given as;

$$\hat{\boldsymbol{y}} = \beta_o + \sum_{i=1}^{k} \beta_i x_i \tag{2.4}$$

where β_i is the gradient in the direction x_i and β_o is the value of the model at the original space of basic random variables (Gaspar et al., 2014).

The second-order polynomial model, which includes all two-factor interactions, is expressed as;

$$\hat{\boldsymbol{y}} = \beta_o + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{i=1}^{k} \sum_{j=1, i<j}^{k} \beta_{ijxj} \tag{2.5}$$

where x is input variables, $i = 1, \ldots\ldots\ldots, k$, β_o, β_i, β_{ii} and β_{ij} are unknown coefficients which are normally determined using the least square method (Gaspar et al., 2014; Vakili and Gadala, 2013).

2.2.2 Kriging method

Kriging is an interpolating technique that features the data observed at all sample points. It offers a statistical prediction of the unknown functions by minimising its mean squared error. Kriging, also known as Gaussian process regression in the field of machine learning, is used to model values by the Gaussian process (Bhosekar and Ierapetritou, 2018). The idea of Kriging was first proposed in the field of geostatistics and was employed to model an error term instead of a linear coefficient. The simplest form of the Kriging is given as (Bartz-Beielstein *et al*., 2016):

$$\mathrm{y} = \beta_0 + \varepsilon \tag{2.6}$$

where β_0 is considered as a mean of the observed values, ε is a random error, which is expressed by the Gaussian stochastic process. In most general form, a Kriging surrogate can be formulated as;

$$\hat{f}(x) = \sum_{i=1}^{m} \beta_j f_j(x) + \varepsilon(x) \quad (2.7)$$

where β_j are the vector of regression coefficients, $f_j(x)$ are the m known independent basis functions that defined the trend of mean prediction at x location, $\varepsilon(x)$ is a random error at location x, which is usually distributed with zero mean (Haeri and Fadaee, 2016).

2.2.3 Artificial Neural Network

Artificial Neural Network (ANN) is a mathematical modelling structure inspired by biological neural networks (Gershenson, 2003). It is a universal approximation for modelling nonlinear relationships between input and output data and learning the dataset's underlying patterns (Haykin, 2009). ANN is made up of a group of interconnected neurons organised in the form of layers: input layer, hidden layers, and output layer, where each layer comprises a group of neurons. A typical multilayer feedforward neural network is illustrated in Figure 2.1.

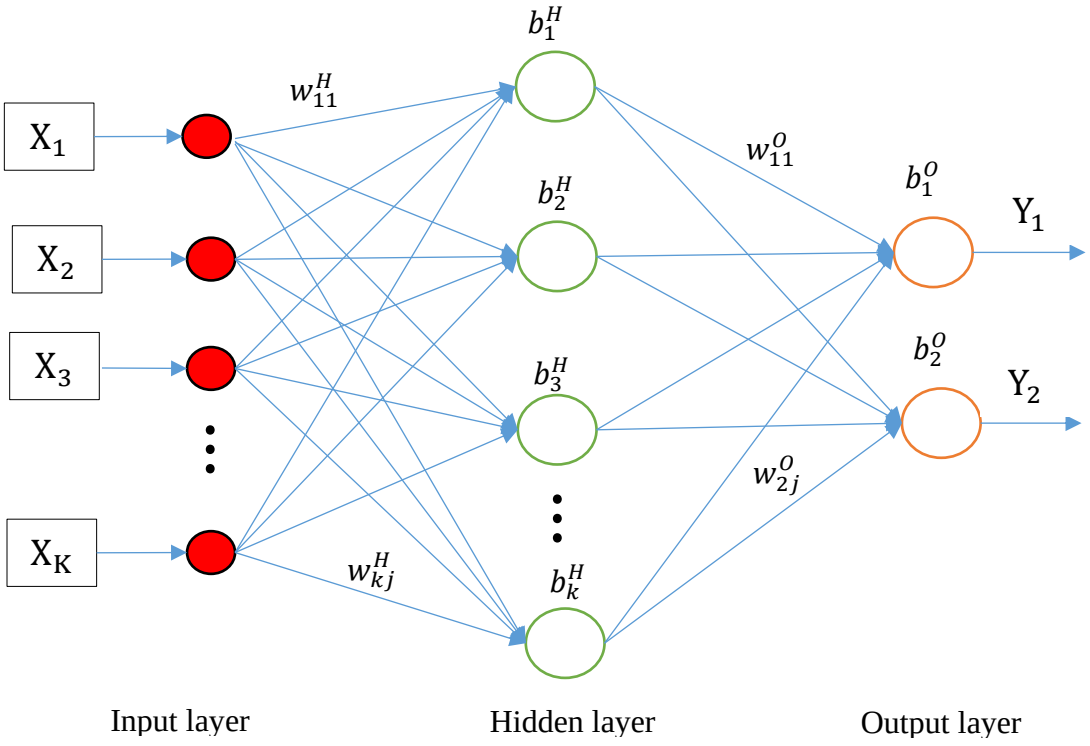


Figure 2-1: Architecture of multilayer feed forward neural network.

The symbol w_{kj}^{H} denotes the synaptic weight between the output of the jth neuron in the hidden layer and the input of the kth neuron in the output layer. The symbol b_k^H denotes the bias of the kth neuron in the hidden layer. The superscript O stands for output layer. As shown in Figure 2.1, each input signal x_k is primarily multiplied by the corresponding weight value w_{Kj} and the resultant products are summed up to generate a total weight in the form of $w_{j1}\,x_1 + w_{j2}\,x_2 + \ldots\ldots + w_{jm}\,x_m$. The sum of the weighted inputs and the bias ($S_j = \sum_{k=1}^{m} w_{kj}\,x_k + b_j$) forms the input to the activation function, φ. An activation function processes this sum and gives out the output, O_j. The resulting sum is processed by a neuron activation function to obtain the ultimate output of the neuron as follows (Ahmadi, 2015):

$$O_j = \varphi(S_j) = \varphi\left(\sum_{k=1}^{m} w_{kj}\,x_k + b_j\right) \tag{2.8}$$

where x_k is the input to the neuron, k is the number of inputs to the neuron, w_{kj} is the input connection matrix, b_j is the bias terms and φ is the activation function. Commonly used activations are softplus, ReLU, linear, softmax, sigmoid and Tanh (Yan et al., 2020). In a mathematical form, the output value of the network can be computed as:

$$y = f(S_j) = \begin{cases} 1 & if\ w^T x \geq \delta \\ 0 & otherwie \end{cases} \tag{2.9}$$

where δ is known as a threshold level; in this case, this type of node is referred to as a linear threshold unit. The weight factors are generally considered as the adaptive parameters in the network to obtain the strength of the input signals, while the bias is characterised by weight except that it has a constant input of 1. Overfitting and underfitting are one of the concerning issues in training ANN. Overfitting occurs when the model (ANN architecture) is not designed to catch the underlying relationship of the function. In other words, overfitting implies that the model is well on the training data but has poor performance on new data. On the contrary, under-fitting refers to a model that is not good on the training data and cannot be generalised to predict new data. A few strategies to avoid the problem of overfitting and underfitting include the penalty method and early

stopping, which are widely used for training neural networks (Swathi, 2018).

The aforementioned surrogate model techniques have been widely used by the research community, and they have been proved to be alternative approaches to overcoming a computationally expensive model (Eason and Cremaschi, 2014; Manoochehri and Kolahan, 2014; Mengistu and Ghaly, 2008). However, most of the studies selected ANN for the surrogate model construction because of its applicability to higher dimensional problems. It is also available in several software packages such as Neural Designer, Neuroph, Darknet, Keras, Deepy, etc. Since the construction of ANN or any surrogate model needs input-output data from the underlying model, such as CFD, Eason and Cremaschi (2014) reported that the number and location of the data points are important to controlling the computational expense of the surrogate model and its overall accuracy. Similarly, it was reported that the performance of the surrogate model is influenced by the design of the experiment (Garud *et al*., 2017; Straus and Skogestad, 2019), while the efficiency of sample size is important to maximise the surrogate model performance as well as minimise the computational cost of expensive numerical problems (Davis *et al*., 2018). One of the objectives set in this thesis is to develop a novel adaptive sampling method to select samples for surrogate model training. The detail of the proposed design space sampling model is presented later in Chapter 3. Several design of experiment methods have been proposed in the literature (Straus and Skogestad, 2019). A general review of design of experiment methods including their merits and shortcoming are presented in Section 2.3.

2.3 Design of Experiment

The goal of Design of Experiment (DoE), also called sampling, is to maximise the amount of information achieved from the limited sample points. DoE is generally categorised into two groups: predefined (one-shot) and sequential sampling (Straus and Skogestad, 2019). Figure 2.2 illustrates different typical DoE approaches. The sample points and locations are determined in a single stage in the predefined sampling schemes. Although predefined sampling is the easiest method but does not guarantee

accuracy, as it is challenging to have prior knowledge of the sample sizes and locations required to design an efficient surrogate (Straus and Skogestad, 2019). The use of predefined sampling can lead to under-sampling or oversampling. The problem of oversampling is the increase in computational burden due to the sampling of the points that do not improve the accuracy of the model. To overcome the drawbacks of the one-shot sampling, flexible sequential sampling, such as space-filling and adaptive sequential samplings were proposed (Garud *et al*., 2017). In the space-filling method, as the name implies, sample points generated are over the entire space. However, space-filling sampling techniques are typically developed based on the predefined approach. Contrary to the space-filling sequential sampling methods, an adaptive sequential sampling techniques, also known as active learning, is developed to utilise the information provided by the surrogate responses and consequently, exhibit better performance with fewer data points.

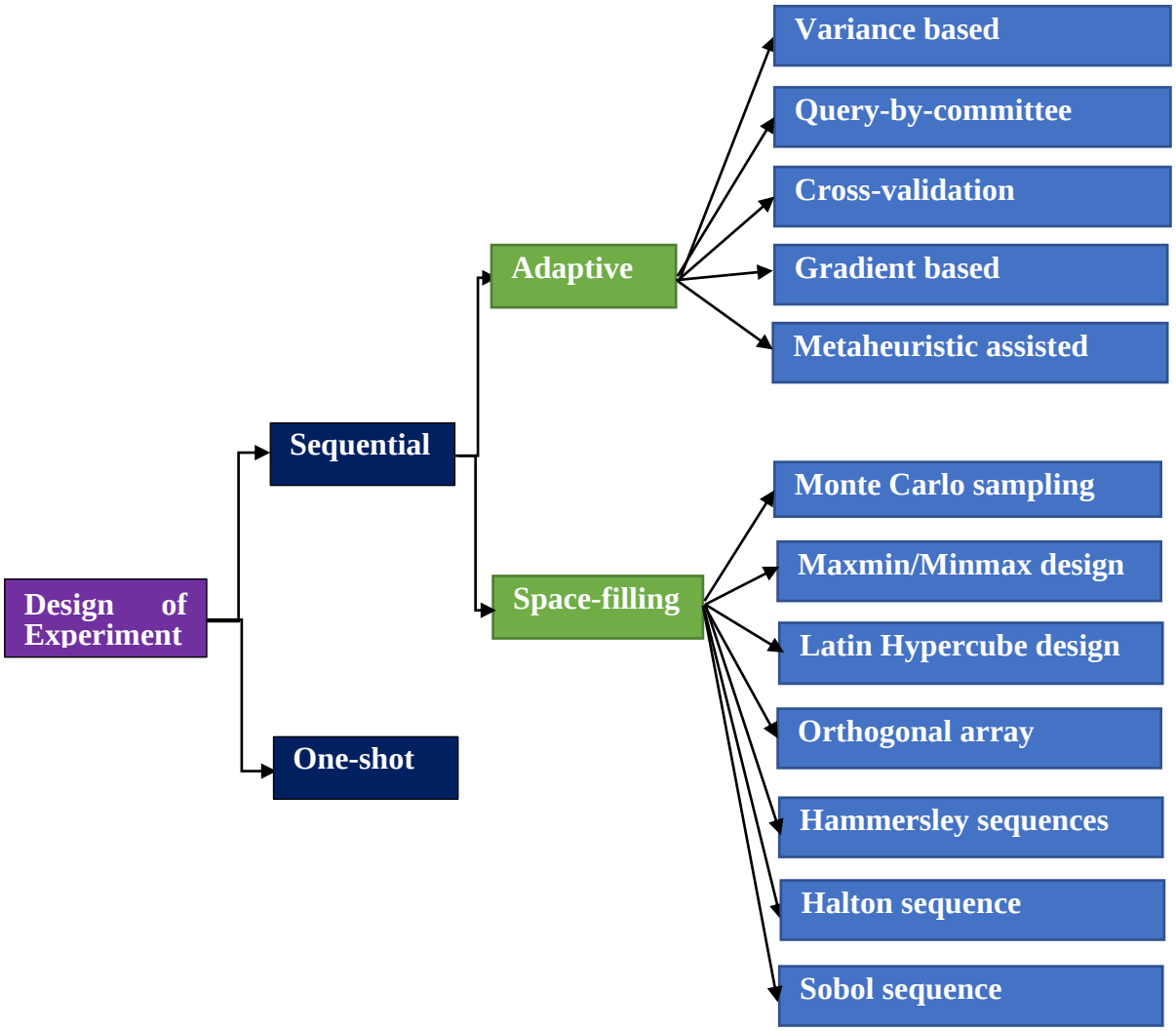


Figure 2-2: Classification diagram of different designs of experiments

2.3.1 Review of Space-filling methods

Space-filling sampling methods have a long history in the DoE field. Generally, they are designed to spread the sample over the entire domain rather than concentrating on a specific area. The widely used space-filling sampling approaches include Monte Carlo sampling (Ghojogh et al., 2020), maxmin/minmax design (Garud et al., 2017), Latin hypercube design (Doyle and Defoe, 2020), Orthogonal array (Giunta et al., 2003), Sabol's sequence (X Sun et al., 2017), Hammersley sequence (Steponavičė et al., 2016) and Halton sequence (Hess et al., 2006). Some of these algorithms have been widely used by the research community. The deficiency of space-filling, however, is usually run in a one-stage fashion. The summary of

space-filling techniques and their application to the surrogate model is provided in sections (2.3.1.1) to (2.3.1.7).

2.3.1.1 Monte-Carlo sampling

Monte-Carlo (MC) was the first modern DoE proposed by Metropolis and Ulam in (1949) to generate samples from a distribution. The method employed pseudo-random numbers to generate K samples with the intent that the generated points would lead to space-filling. The algorithm generates sample points blindly such that every iteration or step does not take the previous iterations into consideration (Ghojogh *et al*., 2020). For example, given an interval $[x_L, x_U]$, where x_L represent lower bound of the design space and x_U is the upper bound, MC sampling select a random number that lies in the interval.

MC sampling is simple to implement. However, a set of MC samples often leave the large region of the design space un-explored resulting from the random and independent nature of the simple sites produced by a random number generator. To overcome the deficiency of MC sampling, the idea of Stratified Monte Carlo sampling (SMCS) was proposed, where space-filling is accomplished by dividing the bounded domain into non-random divisions. Each of the n intervals of $[x_L, x_U]^n$ is divided into bins of equal distribution in SMCS. The deficiency of SMCS, however, is that samples number scales at best as n^2 (Sushant S Garud *et al*., 2017; Giunta *et al*., 2003).

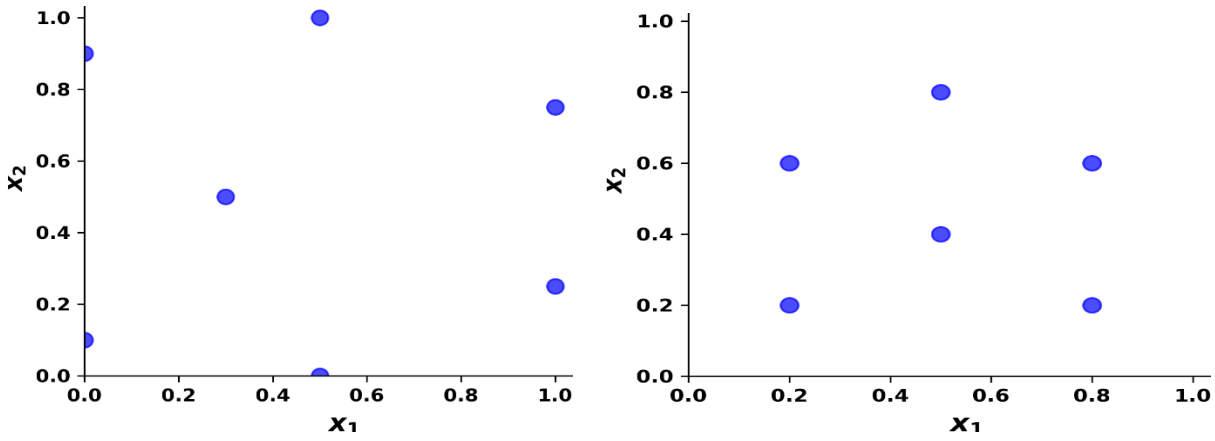
2.3.1.2 Maximin/Minimax distance method

Maximin and minimax sampling use Euclidean distance technique to maximise or minimise the distance of the sample in the design space. This space-filling method was originally proposed by Johnson *et al.*(1990) using two distance systems known as maximin (Mm) and minimax (mM) techniques. The maximin was employed to maximise the minimum distance between the two points. This is given in equation (2.10). On the contrary, the minimax criterion is proposed to minimise the maximin distance between the two points as denoted in equation (2.11).

$$\mathrm{Mm}\left(x_N^{(K)}\right) = \underset{x \in D}{max}\left[\underset{j \neq k}{min}[d\left(x^{(j)}, x^{(k)}\right)]\right] \tag{2.10}$$

$$\mathrm{mM}\left(x_N^{(K)}\right) = min\left[\begin{matrix} max \\ x \in D \end{matrix}\left[\begin{matrix} min \\ j \neq k \end{matrix}[d(x^{(j)}, x^{(k)})]\right]\right] \tag{2.11}$$

where the input variables defined as $x = \{x_j | \, j = 1, 2, \ldots\ldots\ldots, N\} \in \mathbb{R}^N$, the collection of sample points denoted as $x_N^{(K)} = \{x^{(k)} | \, k = 1, 2, \ldots\ldots\ldots, K\}$ is the set of sample points of size K and $d(\,,\,)$ denotes the distance function. The parameter space defined by the bound: $D = \; x^L \leq x \leq x^U$, while the sample point is a precise instant of $x \; in \; D$. The results of the maximin and minimax designed for the $\mathbb{R}^N$ using equation. (2.10) and equation (2.11) is illustrated in Figure 2.3 (a) and (b), respectfully.



(a) (b)

Figure 2-3: Illustration of maxmin and minmax for D^2 (a) Maxmin design (b) Minmax design (Garud *et al*., 2017).

A similar study by Quirante and Caballero (2016), utilised the maxmin technique for the placement of points such that the maximum distance between the sampling point is minimised. A recent study conducted by Jiang *et al.*(2018) reported that a better uniformity sample points projection can be obtained using this sampling scheme. This method has been considered as one of the sampling strategies to obtain an even coverage of the design space. Its drawback, however, is that the maxmin optimisation tends to favour decision vectors that are located near the boundary of the decision space. In contrast, minimax favour decision vectors that are

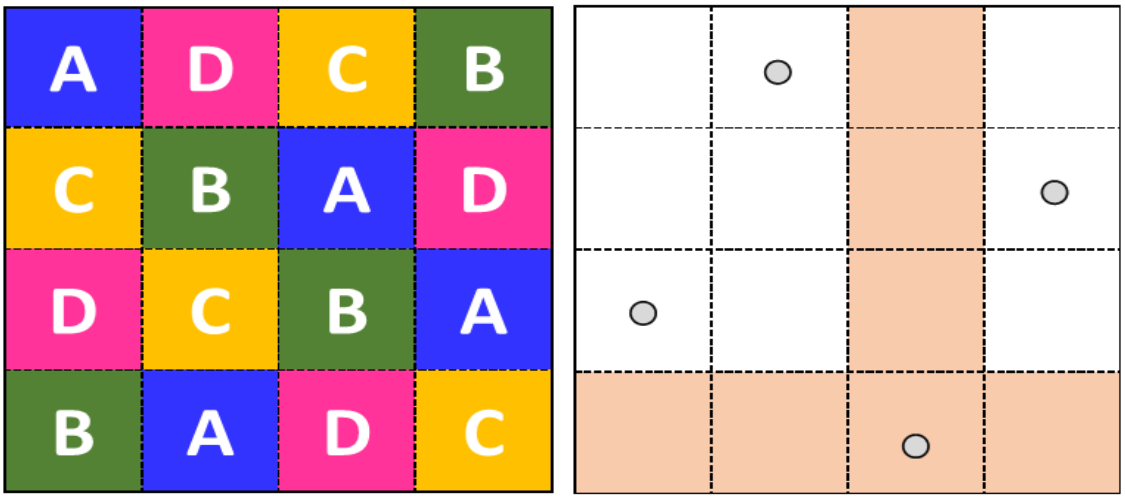
located near the centre of the design space (Steponavičė et al., 2016). This can be observed in Figure 2.3.

2.3.1.3 Latin Hypercube Design

Latin Hypercube Sampling (LHS) was developed to improve Monte-Carlo sampling and its variations (Garud *et al*., 2017). LHS is one of the most popular DoE that has been found useful in many computational applications. It was originally developed by Mckay *et al.*in 1979 (McKay *et al*., 2000). Even though the study by McKay *et al.* is considered as the inventor of LHS, the existence of LHS can be traced back to the quota sampling proposed in the work of Steinberg (1963), which was inspired by a work of Latin square in 1968 (Garud *et al*., 2017). In practice, LHS operates by dividing an empirical distribution function of a variable X into n equiprobable, non-overlapping strata and then drawing one random value from each stratum (Clifford et al., 2014). suppose J variables are $X_1, X_2, \ldots\ldots\ldots, X_j$, the n random values drawn for variable X_1 are combined randomly with the n random values drawn for variable X_2, and so on until n J-tuples are generated.

The basic concept of LHS for a two-dimensional space is illustrated in Figure 2.4. In Figure 2.4(a), an n-by-n matrix is filled with n different characters such that each character appeared exactly once in each row and exactly in each column. A two-dimensional example of LHS with four sample points is shown in Figure 2.4(b). Each row and each column has one point (the darkened displayed row and column taken by one of the sample points). LHS allows the user to select the number of samples to run based on the available computational budget. But the limitation of LHS is that it is not reproducible because sample points are generated based on random combinations (Doyle and Defoe, 2020). Besides, the lack of reproducibility of LHS can lead to almost nearly co-linear sampling, which can result in poor performance when used in conjunction with other systems. The drawback of LHD necessitates an enhancement which led to the advent of Optimal LHS (OLHS) for providing uniform distribution of sample points for the entire regions of the design space (Doyle and Defoe, 2020). OLHS employed satisfactory criteria to ensure that points are uniformly

distributed evenly within the design space while following the basic LHS procedure.



(a) (b)

Figure 2-4: Illustration of basic LHS concept (Sheikholeslami and Razavi, 2017).

2.3.1.4 Orthogonal array

Orthogonal array is typically viewed as a generalisation of LHS. It shares many similarities with LHS, which are also based on the basic idea of random sample placement within the bins (Lin *et al*., 2001). Orthogonal array can provide a set of samples that produce uniform sampling in any N-dimensional design space. The algorithm for generating Orthogonal array sample can be expressed (Giunta *et al*., 2003):

$$x_j^{(i)} = \frac{\pi_j^{(i)} + U_j^{(i)}}{P} \tag{2.12}$$

for $1 \leq i \leq k$ and $1 \leq j \leq n$ where k is the number of samples, n is the number of design variables. U is a random value on array [0, 1], π is an independent random permutation of the sequence of integers $0, 1, \ldots\ldots, K-1$, the superscript (i) represents the sample number while subscript j represents the dimension index. The construction of an Orthogonal array depends on four selected parametric integers: the number of sample points K, the

dimensions of the domain n, the strength of the array and the number of bins per domain (Garud et al., 2017). The total dependent on the parametric integers makes Orthogonal array less flexible than other space-filling sampling methods (Giunta *et al*., 2003; Viana, 2016).

2.3.1.5 Hammersley sequence sampling

Hammersley sequence sampling (HSS) is one of the first high dimensional quasi-random low discrepancy sequences for sample point generation. It was initially proposed by Kalagnanam and Diwekar (1997) to address the issues associated with the MCS and SMCS. The study indicated that HSS provides better uniformity over LHS, and the probability of samples with clustered decision vectors is minimal (Steponavičė *et al*., 2016). Steponavičė *et al.*reported that HSS requires significantly fewer sample points to approximate the mean and variance of distributions than other conventional techniques. It was reported that HSS is more efficient than many conventional space-filling sampling algorithms such as random sampling, Monte Carlo, etc. The drawback of HSS, however, includes flexibility in implementation *(Garud et al*., 2017). The largest base employed for the generation of sequence increase as the space dimensions increase.

2.3.1.6 Halton sequence

Halton sequence sampling was implemented based on the deterministic technique that uses different prime radices for different dimensions to generate a d-dimensional low discrepancy sequence (Halton, 1960). This sampling algorithm is inspired by the Hammersley sequence that uses a Quasi Random Low Discrepancy (QRLD) sequence to tackle the issues associated with the MCS and SMCS. The term quasi-random refers to the deterministic nature of the sequence, while the term low discrepancy means its nearness to a uniform distribution of the sample points in a domain (Garud *et al*., 2017). In order to construct high-speed space-filling, Halton proposed modification to the Hammersley sequence to overcome the shortcoming. Contrary to the Hamersley sequence, where the radices are strictly prime numbers, Halton indicates that the radices should be only

mutually prime or co-prime. Therefore, the large radices and slow space-filling allied to Hammersley sequence were avoided via the modification. The Halton sequence is easy to construct and implement. However, studies show that it faces serious drawbacks for N-dimensional design space greater than 14 (Hess *et al*., 2006; Ökten *et al*., 2012).

2.3.1.7 Sobol sequence sampling

Sobol sequence was proposed as an improved version of Hammersley and Halton sampling strategies. It was proposed to address these two sampling algorithms' common pitfalls (performance degrades in higher dimensions). Performance comparisons of the Sobol sequence with other space-filling have been carried out by the research community (Sun *et al*., 2017; Sun *et al*., 2021; Tarantola *et al*., 2012). Tarantola *et al.*compared Sobol quasi-random with the Latin hypercube at increasing sample size and dimension against analytical values. The authors reported that the Sobol sequence provides thorough coverage similar to Latin hypercube sampling in most cases investigated. The effectiveness of the random, Latin hypercube and Sobol sampling strategies was investigated by Sun *et al*. (2017). It was reported that the Sobol sequence provides the highest accuracy in most experiments. The common issue raised about Sobol sequence structure was that Sobol might result in occasional large errors in sensitivity indices of input factors. Some basic issues in the uses of the Sobol sequence and its current randomisation methods for sensitivity analysis were investigated by Sun *et al.*(2021) to provide some insights into its practical use. The outcome of their finding leads to an alternative method of randomising Sobol sequences called the Column shift method (Sun et al., 2021).

The performance comparison of the reviewed non-adaptive sequential sampling strategies is shown in Table 2.1. Complexity, low computational cost, high dimensional suitability, uniform coverage and stochastic are the criteria employed to evaluate the characteristic of the review sampling methods. As shown in Table 2.1, none of the techniques satisfies all the characteristics.

Table 2-1: Non-adaptive sequential sampling characteristic

Non-adaptive sequence sampling strategies	**Performance comparison metric**				
	Complexity	Low computational cost	High dimensional suitability	Uniform coverage	Stochastic
Monte Carlo sampling	No	Yes	No	No	Yes
Maximin/minimax design	Yes	No	No	Yes	Yes
Latin hypercube design	No	Yes	No	No	Yes
Orthogonal array	No	No	No	Yes	No
Hamersley sequence	Yes	Yes	No	Yes	No
Halton sequence	Yes	Yes	No	Yes	No
Sabol sequence	No	Yes	Yes	Yes	No

2.3.2 Review of Adaptive Sampling Approaches

The sampling strategies described so far are one-stage or static, where sample points generation is done at once. Though these sampling methods are very popular and widely used by the research community, the problems associated with these techniques are under-sampling or oversampling. Therefore, it resulted in poor system approximation (Crombecq, 2011; Garud et al., 2017b). In order to address these issues, adaptive sequential sampling strategies have been proposed. Adaptive sampling can be regarded as an improvement to static or space-filling sampling approaches. In this case, the method incorporates system information to place the sampled points using exploration and exploitation concepts. Crombecq *et al.*(2011) reported the basis and importance of adaptive sampling over static methods. They indicated that adaptive sampling strategies produce improved surrogate approximations than the static or space-filling methods. The two major advantages of adaptive sampling over space-filling are

highlighted as low computational budget and better approximations. Different adaptive sampling strategies have been reported in the literature. These sampling have been classified into five categories according to the way actual predicting errors is represented (Liu et al., 2018). A review of adaptive sequential sampling techniques and their application to the surrogate model is presented in section 2.3.2.1 to 2.3.2.5.

2.3.2.1 Adaptive variance sampling

Adaptive variance is deeply combined with Gaussian process regression, which considers output response as the realisation of Gaussian response (Liu et al., 2018). This model is constructed by applying the Bayesian rules to provide a posterior Gaussian distribution $y(x) \backsim GP(\hat{f}(x), \hat{\sigma}^2(x))$ using prior system respond obtained from the observed data points. The GP symbol denotes gaussian process also known as Kriging model, $\hat{\sigma}^2(x)$ represent the variance while $\hat{f}(x)$ is the prediction response. Many studies have employed adaptive variance to perform the sampling process. Gratiet and Cannamela (2015) considered the variance provided by the Kriging and the observed variance of a leave-one-out cross-validation procedure to select a new data point using function $f(x) = (\sin(7x) + \cos(14x))x^2e^{-4x}, x \in [0,4]$ to demonstrate the efficiency of the proposed approach. Figure 2.5 illustrates a 1D example for selecting new sample points by minimising prediction variance while adjusting the prediction variance of the Kriging model. As shown in Figure 2.5(a), six sample points was initially used to build the Kriging model. The adjustment of prediction variance was performed by computing the variance between the true function and Kriging model shown as red circle in Figure 2.5(a) and (b), respectively. It was reported that the proposed approach allowed the new observations set at the locations where the model error is large.

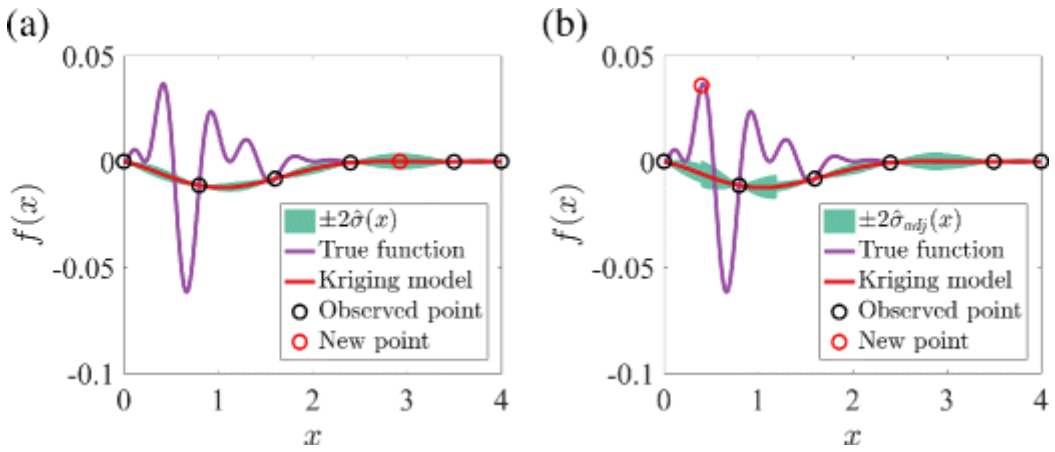


Figure 2-5: A 1D example of the selection of a new point by maximising (a) the prediction variance and (b) the adjusted prediction variance of Kriging, respectively (Liu et al., 2018).

Farhang-Mehr and Azarm (2005) proposed a modification of prediction variance via the localisation of local optima on the Kriging model to identify irregular regions. Liu *et al.*(2016) stated that the Farhang-Mehr and Azarm approach heavily depends on the Kriging model value. Therefore, a poor model may result in an erroneous identification of an interesting region. A similar study by Busby *et al.*(2007) employed an adaptive gridding algorithm to decompose the design domain into disjoint cells. The cell edges are of the order of correlation lengths of different variables. This is followed by maximum entropy and cross-validation criterion to select cells with large prediction variances for further sampling. More recently, Menz *et al.*(2021) investigated the effect of Monte Carlo sampling and the Gaussian process surrogate model on the probability of failure estimator using variance decomposition. Ameryan *et al.*(2022) combined the Kriging meta-model and the conventional sequential space-filling method. It was reported that the proposed hybridisation algorithm provides better performance with fewer function calls than the sequential space-filling method. A Global Sensitivity Analysis (GSA) method that is based on the surrogate metamodeling and theory of active subspaces proposed by Zhou *et al.*(2022). They show that three GSA measures, namely activity score, sobal total indices and derivative-based sensitivity measure can be obtained at different steps of the proposed method. The research community's studies show that Kriging model variance is a powerful tool to improve an

experimental design set. However, when the accuracy of the Kriging model is not homogenous over the input parameter space, the variance method can suffer from an important flaw. Moreover, the model solely determined by the distance between the prediction and the design points which might not reflect the true model errors.

2.3.2.2 Query-by-committee based adaptive sampling

The Query by Committee (QBC) scheme was first proposed for problem classification in the field of machine learning (Freund et al., 1993; Seung et al., 1992). When employing QBC on metamodels, new samples are selected based on a set of randomly proposed sample points, sorted using a metamodels committee (Fuhg et al., 2020). The QBC sampling procedure are as follows: (1) a committee C consists of n_C members ($C = \{\widehat{M}_i^C\} i = 1, \dots\dots\dots .. n_C$) represents different computing surrogate models, (2) each surrogate model is designed to predict response at a point x candidate, (3) the new candidate is selected at a point for which the committee members have maximum disagreement. The level of agreement is evaluated based on the prediction fluctuation F_{QBC} provided by C committee members of different metamodel as (Fuhg et al., 2020):

$$F_{QBC}(x) = \frac{1}{n_C}\sum_{i=1}^{n_C}(\widehat{M_i^C}(x) - \overline{\widehat{M^C}}(x))^2 \tag{2.13}$$

where $\overline{\widehat{M^C}}(x) = \frac{1}{n_C}\sum_{i=1}^{n_C}\widehat{M_i^C}(x)$ is the average estimation output considering the different committee members. An illustration of the QBC scheme using four Kriging models, exponential basis function, namely Gaussian basis, spline basis function, and cubic spline basis function for 1D is presented in Figure 2.6. The shadow represents the prediction response plus/minus twice the QBC variance. It was observed that the new point falls into the region with a large prediction error via maximisation of QBC variance. QBC is more generic than variance adaptive sampling, which solely depends on the specific type of surrogate model. However, Fuhg *et al.*(2020) highlighted that committee members should exhibit differences in order to reduce the efficiencies of surrogate model error. They discourage a QBC approach that is based on only one type of surrogate model, as this might be problematic.

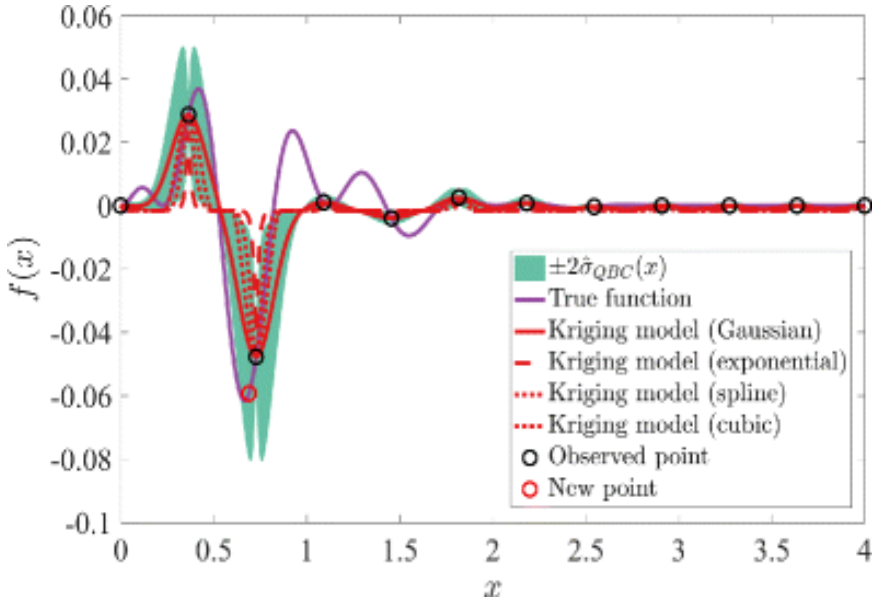


Figure 2-6: Numerical illustration of 1D QBC variance (Liu et al., 2018)

2.3.2.3 Cross-validation base adaptive sampling

Cross-validation based sampling is a scheme for analysing the surrogate model response with respect to unknown sample points. Different variations of cross-validation, including k-fold cross-validation and leave-one-out cross-validation, are available in the literature (Fuhg et al., 2020; Meckesheimer et al., 2001). The k-fold cross-validation is usually used to estimate the prediction error. It is generally applied in the surrogate model by chosen $K-1$ subsets to establish the training subset, while the remaining subset is used to validate and estimate the performance score. The process is then repeated in K times such that all the subsets are successively used for validation. The cross-validation is then evaluated as the mean of the K results. However, it is generally discouraged for adaptive metamodels as the k-fold cross-validation bias may become a problem when K is small (Fushiki, 2011). Hence, the well-known leave-one-out cross-validation (LOOCV) becomes a special case for the general k-fold cross-validation with $K=m$, where m is the number of training data size. For every $i \in [1,m]$, an auxiliary metamodel $\widehat{M}_{-i}$ is trained on $m-1$ observation. The accuracy of the surrogate model of interest $\widehat{M}$ is then evaluated via the cross-validation error e_{LOOCV} at point x^i, as given in equation (2.14):

$$e_{LOOCV}(x^i) = \left|\widehat{M}(x^i) - \widehat{M}_{-i}(x^i)\right|, \forall i \in [1, m] \quad (2.14)$$

Some studies have classified LOOCV into continuous CV and discontinuous adaptive methods, also called discrete adaptive sampling (Fuhg et al., 2020). The errors obtained using continuous CV based adaptive sampling can adaptively guide local exploitation. However, It was reported that the new sample points obtained with the continuous LOOCV usually cluster around some observed points if one directly maximises the CV based criteria, as its focus is purely local exploitation (Aute et al., 2013; G. Li et al., 2010). In order to tackle this issue, distance-based space-filling was introduced to prevent the new sample from being too close to each other. However, Xu *et al.*(2014) and Jiang *et al.*(2015) indicated that it is hard to determine an appropriate threshold d value because large d value force the new sample points to spread evenly over the entire design space, while small d value does not help to avoid the clustering phenomenon. Note that the auxiliary metamodel $\{\widehat{M}_{-i}\} i\ in\ [1, m]$ in Equation (2.14) is built to evaluate local error for every available observation, which may be computationally expensive. Hence, Jin *et al.*(2002) proposed equation (2.15) to obtained an error at any point x as the superposition of the relative errors between the current surrogate model and the leave-one-out surrogate models.

$$e_{LOOCV}^{count}(x) = \sqrt{\frac{1}{m}\sum_{i-1}^{m}\left(\widehat{M}(x) - \widehat{M}_{-i}(x)\right)^2} \quad (2.15)$$

In the discontinuous CV-based adaptive sampling technique, the general approach is to partition the design space into discrete cells. Each cell is assigned a variant of $e_{LOOCV}(x^i)$ according to some closeness metrics. The cell associated with the highest error is then recorded as a priority cell for further sampling. This strategy is the most widely used for CV based adaptive sampling. For example, Devabhaktuni and Zhang (2000) employed it to determine a region with the largest prediction error and divided it into 2^n regions. A similar study by Braconnier *et al.*(2011) divided the design space into a set of hypercubes and used a quad-tree algorithm to partition the region with the largest CV error into 2^n equal hypercubes. Although this method seems simple and effective, its limitation is that it

easily leads to the over-sampling problem (Fuhg et al., 2020). To circumvent the drawback of 2^n partitioning, Xu *et al.*(2014), Jiang *et al.*(2018), and other related studies employed the Voronoi diagram algorithm to partition the design space into the Voronoi cells. Subsequently, the cell with the largest CV error was selected as the sensitive region. Considering that this approach first required building a true model with lower error precision to guide the surrogate model (metamodel) with the lower accuracy. Therefore, this approach can be considered impractical as the test points are usually unavailable in practice, particularly in the highly expensive simulation model.

2.3.2.4 Gradient-based adaptive sampling

The general assumption about the gradient-based method is that system response may produce high prediction errors in regions with large gradients than in corresponding subdomains with low gradients (Bouhlel & Martins, 2019; Van Beers & Kleijnen, 2003). Therefore, many researchers proposed an adaptive surrogate model around the premises of gradient information to discover the important regions. An adaptive sampling strategy for enhancing the Kriging model was proposed by Rumpfkeil *et al.*(2011). In this study, Dutch interpolation and discrepancy measurements between the local and global Kriging models were employed to build a local surrogate model. The region with the largest discrepancy was selected as a new point for further scrutiny. Liu *et al.*(2018) observed that this approach has a considerable restriction due to its derivatives requirement.

In a similar study, the regions of interest were exploited by Yao *et al.*(2009) using first-order gradient information $\partial\hat{f}/\partial x$ of the radial basis function neural network. They observed that gradient information is more suitable for geomery exploitation. Therefore, employed the optimal LHS strategy to provide additinal points whenever the performance of the response surface is no longer improve through local exploitation. Mackman and Allen (2010); Mackman *et al.*(2013), and Wei *et al.*2012) claimed that curve curvature usually provides more information than the first-order gradients, thus employing the Hessian matrix $\partial^2\hat{f}/\partial x^2$ as an alternative to first order gradient to guide the local exploitation. While, the global exploration was

achieved through a distance term. Liu *et al.*(2018) reported that the gradient sampling strategies are highly depends on the quality of the surrogate model. They noted that the choice of model type, model parameters may influence the estimation of the gradient information and subsequently affect the selection of new sample points.

2.3.2.5 Optimisation assisted approach

The emergence of the adaptive sampling technique has attracted the attention of researchers in the last decade (Crombecq et al., 2011; Y. Jin, 2011; Junior et al., 2022; Regis, 2014a; Vu et al., 2017). The finding revealed that adaptive sampling yields better approximation results than the predefined or space-filling methods and holds much promise. However, it was reported that the focus should shift to how to sample the minimum number of points intelligently so that the surrogate model would reflect the actual black-box function in the fields of interest (Dong et al., 2018). Moreover, an intelligent sample scheme would further advance the metamodel efficiencies. To this end, different optimisation algorithms have been introduced along with space-filling sapling techniques. The commonly employed optimisation algorithms are Simulated Annealing (SA), Genetic Algorithm (GA) and Particle Swarm Optimisation (PSO) based methods.

Simulated annealing as an optimisation model formulates a sequence of steps, (1) defining a minimisation process by random changes of the design variables, (2) computation of a probabilistic acceptance criterion, (3) evaluation of design vector is accepted or rejected following the criterion (Tzannetakis & Van de Peer, 2002); accept a design if:

$$f(X_{i+1}) - f(X_i) \leq 0 \; and \; set \; X_{i+1} = X_i, \tag{2.16}$$

otherwise, accept the design with the probability:

$$P[f(X_{i+1}) - f(X_i)] = e^{-[f(X_{i+1}) - f(X_i)]/k_b T} \tag{2.17}$$

where the factor k_b is the Boltzmann constant, T is the temperature. Manoochehri and Kolahan (2014) integrated an artificial neural network with SA to optimise the deep drawing process. Four parameters, namely punch radius, die radius, blank holder force, and frictional conditions, are considered input parameters while thinning, which is one of the major failure modes in deep-drawn parts considered as an output parameter. The

authors stated that good accuracy was obtained after 75 iterations. The authors further compared the developed optimisation model results with the LHS-based surrogate model. In this case, the optimal design of experiments was implemented, where only 81 simulations were run and used to develop the ANN model. It was reported that computational results of the SA-based optimised model demonstrated quite the capability of obtaining high-quality solutions (optimal or near-optimal) within reasonable computational times. A similar study by Maakala *et al.*(2018) proposed a framework to optimise the geometry of the superheater region. The framework was implemented as a surrogate-based optimisation method that combined simulated annealing and polynomial regression surrogate models. The developed model was used to quantify the connection between geometry and heat transfer. It was reported that the uniformity of the flow field improved while the heat transfer rate also increased. According to Luo *et al.*(2021), SA is employed because it can find global optimisation when given sufficient time. However, SA was observed to be slower than many optimisation algorithms as it requires a longer time to find a near-optimal solution. Moreover, it's a local search method, and neighbouring point selection is based on random.

More recently, Luo *et al.*(2021) combined gradient-based optimisation with simulated annealing to address the convergence rate issues associated with the SA. The proposed framework was applied to the line design of an underwater vehicle, where energy consumption and resistance reduction is the optimisation goal. The CFD-based simulation was conducted to optimise the parameters of the underwater vehicle. A comparison was made between the proposed optimised and static surrogate models (Latin hypercube-based DOE combined with RBF model). It was reported that both resistance and energy consumption descend on the basis of optimal lines in the static and optimised surrogate model. But the optimised surrogate model outperforms the static surrogate model in terms of minimisations of iterations.

Chu *et al.*(2015) proposed a reliability-based optimisation model for structural design under uncertainty. This study used a surrogate model to

search for the best compromise between the cost and safety while considering system uncertainties by incorporating reliability measures within the optimisation. First, the LHD was employed in the structural finite element model to acquire an effective database for building the surrogate model, followed by incorporations of SA and GA to prevent the surrogate model from converging prematurely. The SA was implemented using the procedure described in equations (2.16) and (2.17). At the same time, GA optimisation begins from a random initial population. Each individual, also known as the chromosome, is encoded into a structure denoting its properties and progresses through successive generations. Lastly, the chromosomes are rated with respect to their fitness in each generation. The performance comparisons of SA against GA were carried out. The study revealed that SA converged to a better design than GA, which led to more material savings and higher structural reliability but required a much higher computational cost.

Mengistu and Ghaly (2008) coupled GA with the ANN-surrogate model to optimise a single-point aerodynamic transonic turbine stator and multi-point NACA65 subsonic compressor rotor. The optimisation objective is a weighted sum of the performance objectives and penalised with the constraints in order to achieve better aerodynamic performance at the design point or over the full operating range by reshaping the blade profile. However, the common drawback of GA highlighted in the literature is numerous objective function evaluations for convergence (Bates et al., 2004; Fonseca et al., 2009; Mahulja et al., 2018; Wang & Sobey, 2020; J. Zhou et al., 2006). GA often require many iterations, typically taking many generations, sometimes thousands or hundreds of evaluations to converge (Wortmann et al., 2015). Some studies have proposed a modification to the generic GA to tackle its shortcoming. Deb and Myburgh (2016) developed a customised GA to find near-optimal solutions. Some of the modifications to generic GA in this study include more than one solution employed in each iteration to create a new population collectively. The initialisation and population update methods are customised to exploit the linearity aspect of the problem. At the same time, a small population of solutions was used in

each iteration. Parallel implementations of GA on the Central Processing Unit (CPU) and Graphic Processing Unit (GPU) have also been employed in some studies to improve generic algorithm performance (Iturriaga and Nesmachnow, 2012; Wang & Sobey, 2020). Although the above-reviewed methods have achieved various levels of success, however, computationally expensive links to SA and GA models, especially when the problem requires computationally intensive simulations motivate the search for alternative optimisation techniques such as PSO (Felkner et al., 2013; Wortmann et al., 2015).

2.3.3 Particle Swarm Optimisation

Particle swarm optimisation (PSO) is a popular search optimisation algorithm commonly used to explore the search space of a given problem to find the optimal parameters or settings required to maximise or minimise a specific objective. PSO is a population-based stochastic optimisation algorithm inspired by the behaviours of birds or fish schooling within a flock in evolutionary computation. The collection of particles in a search space where each particle represents a potential solution of the optimisation task is referred to as a swarm. The PSO algorithm was initially proposed by Eberhart and Kennedy (1995) in 1995 for solving an unconstrained optimisation problem. Since then, it has been successfully applied to a series of optimisation problems such as nonlinear, non-differentiable, and multiple optimisation tasks in engineering design optimisation fields (Cheng et al., 2020b). In PSO, each particle communicates and exchanges information about the search space and moves in the multi-dimensional solution space according to some velocity. The movement of PSO is influenced by two factors: the local best solution of each particle and the global best solution of all particles involved in the solution space. The local best position of each particle and swarm global best positions are updated during the iteration of the algorithm if a better solution is found. The process is repeated till the desired results are attained or it reaches the specified number of iterations. The illustration of the PSO process is presented in Algorithm 2.1.

Algorithm 2.1: PSO algorithm

Input:	Initial number of iterations (t); the maximum number of iterations (t_{max})
Step 1:	
Step 2:	Initialise a swarm ($\boldsymbol{S}$) with $\boldsymbol{n}$ particles, and give each particle in $\boldsymbol{S}$ a random velocity.
Step 3:	Evaluate all particles in $\boldsymbol{S}$, and take the populations of all particles in $\boldsymbol{S}$ as the optimal solutions they find, which is typified as $\boldsymbol{P_i}(\boldsymbol{i} = \boldsymbol{1}, \boldsymbol{2}, \ldots\ldots, \boldsymbol{n})$; the solution with the best fitness is typified as $\boldsymbol{P_g}$.
Step 4:	**while** $\boldsymbol{t} < t_{max}$ **do**
Step 5:	Update the velocity and positions of all particles in $\boldsymbol{S}$
Step 6:	Evaluate all particles in $\boldsymbol{S}$
Step 7:	Update $\boldsymbol{P_i}$ and $\boldsymbol{P_g}$
Step 8:	$t = t + 1$
Step 9:	end **while**
Step 10:	Output $\boldsymbol{P_g}$, the solution with the best fitness

PSO algorithm is widely used for optimisation because of its excellent global search capacity and simplicity (Li et al., 2019). A framework for expensive optimisation that coupled PSO with radial basis function surrogates was proposed by Regis (2014b). In this study, the proposed framework called Optimisation by particle swarm using surrogates used multiple trial positions and velocities for each particle in the swarm in every iteration. PSO generally requires many fitness evaluations to find a good solution. Secondly, some of the fitness values are not important in the iterative process of PSO (Li et al., 2020). Thus, various improvements were made to the PSO algorithm to be suitable for the surrogate model. Sun *et al.*(2015) implemented a two-layer surrogate-assisted PSO algorithm to address the issue of a large number of fitness evaluations in PSO. They proposed a pre-screening criterion in which a global and a number of local surrogate models are employed for fitness approximation. A local surrogate model was constructed using the data samples near each particle where improved

fitness estimation is achieved. A global surrogate model was employed to guide the swarm to fly quickly to an optimal or global minimum.

It was reported by Li *et al.*(2020) that many fitness evaluations are consumed in the work of Sun *et al.*(2015). Li *et al.*(2020) applied the radial basis function surrogate model to update the global best solution of the population. They used a selected subset of samples to build a local surrogate, which aims to model the promising sub-regions of the design space. They reported that the proposed algorithm might effectively solve medium-scaled computationally expensive problems as a small number of candidate solutions are required to be evaluated at each iteration.

More recently, some studies proposed the hybridisation of PSO with the other evolutionary algorithm. For example, Li *et al.*(2020) employed the learner phase of teaching-learning-based optimisation (TLBO) to explore design space and used PSO to speed up the convergence. A fuzzy hierarchical surrogate-assisted probabilistic PSO proposed by Chu *et al.*(2021) applied fuzzy surrogate-assisted (FSA), local surrogate-assisted (LSA) and global surrogate-assisted (GSA) models to fit the fitness evaluation function individually. Furthermore, a probabilistic PSO was implemented to predict the trained model and update the samples.

PSO is adopted in this study to find the optimal data point for surrogate model accuracy maximisation. PSO is a population-based optimisation technique initialised with a population of random solutions, and the search for the optimal solution is performed by updating generations. Unlike GA, PSO has no evolution operators, such as crossover and mutation. Additionally, PSO is computationally more efficient in terms of both speed and memory requirements (Gad, 2022). AbWahab *et al.*(2015) demonstrated that PSO can have better results faster and cheaper than other methods such as GA and SA. Moreover, it does not use the gradient of the problem being optimised. In other words, unlike the gradient descent optimisation method, PSO does not require the problem to be differentiable. The advantage of PSO was summarised in (Freitas et al., 2020) as follows: (i) it does not make assumptions about the continuity and differentiability

of the objective function to be optimised; (ii) it does not need to compute the gradient of the error function; and (iii) it does not need good initial starting points or deep a priori knowledge about the most promising areas of the search space.

2.4 Review of Pipeline Leakage Detection

Over the last decades, several studies have been proposed on pipeline leakage detection using different approaches. Existing leak detection and characterisations are classified into software and hardware methods (Kim et al., 2021). In an effort to classify these technologies based on their technical nature, further research efforts were made and led to the classification into three groups, namely external, visual or biological and internal methods (Cramer et al., 2015). The external technologies include acoustic sensing, accelerometer, fibre optics, vapour sampling, infrared thermography, ground penetration radar and electromechanical impedance that utilise man-made sensing devices to achieve leak detection tasks at the exterior part of the pipeline (Hoarau et al., 2017; Png et al., 2018). The visual-based methods employ experienced personnel, trained dogs, pigs and drones to inspect and detect pipeline leakage. The visual-based methods appear to be the most suitable for leak detection and characterisation. However, the operational time of these techniques is based on the frequency of inspection. For the internal-based leak detection methods, many researchers have reported a collection of techniques to detect and characterise pipeline leakage (Pérez-Pérez et al., 2021). Generally, these methods employ computational algorithms in conjunction with various sensors for monitoring parameters that characterise the fluid flow within pipelines. Some commonly used techniques include mass-volume balance, negative pressure wave, pressure point analysis, state estimator and numerical modelling. The scope of the method reviewed in this study is confined to numerical modelliof pipeline leakage. More details on the review of other leak detection methods can be found in Adegboye *et al.*(2019).

2.4.1 Numerical Modelling of Pipeline leakage Detection

The numerical modelling method, also known as transient modelling, is reported as the most sensitive leak detection method (Moore, 1999; Liu et al., 2019). This method employs conservation equations for the fluid mass, momentum and energy to model the flow within a pipeline and compares the predicted values with the measured data to determine and characterise leakages. The flow parameters monitored in this method are flow rate, pressure, and other fluid flow parameters. Pipeline leak detection using the transient-based leak detection approach has been extensively adopted in the research community. It has been shown to detect and locate pipeline leak positions successfully.

Technological advancements have benefited the petroleum industry to a large extent that it is possible to transport un-separated gas-liquid mixture over a long distance. The economic impact of this technology is enormous to the degree that some offshore developments and multiphase flow lines have, in some cases, replaced their topside installations (Bratland, 2010). In fact, in some areas, such as offshore oil and gas section of the petroleum industry, multiphase flow line provides the most economical option to transport oil and gas from underwater wells to onshore separation facilities (Xuejie Li et al., 2022). Considering the harsh external environment and complex internal flow of multiphase pipelines, it is important to monitor multiphase pipelines for timely and accurate detection of leakage to reduce economic loss and environmental damage. However, most of the works reported in the literature are limited to the single-phase pipeline leakage. For example, de Sousa and Romero (2017) investigated the effect of the leaks on a monophase pipeline with particular attention to the pressure and flow rate characteristics using ANSYS Fluent. Three different leak sizes were studied using a 1 m pipeline length with a diameter of 0.15 m in an onshore environment. The obtained results revealed that the occurrence of the pipe leakage impacted both pressure and flow rate within the vicinity of the leak regions (Figure 2.7). The fluid discharged from the pipeline perforated, increasing the pipe friction and resulting in more pressure drop. The pipeline leakage conditions considered in this study are limited to the only leak sizes

effect. It is also difficult to admit that the observed results can be applicable to multiphase flow systems.

Own to the fact that multiphase pipelines involve simultaneous transportation of two or more fluids; it is difficult to conclude that only pressure and flow rate sufficient to accurately detect multiphase pipeline leakage. Therefore, determining the leak effect on the multiphase pipeline and the interaction between the in-pipe flow parameters is of great importance.

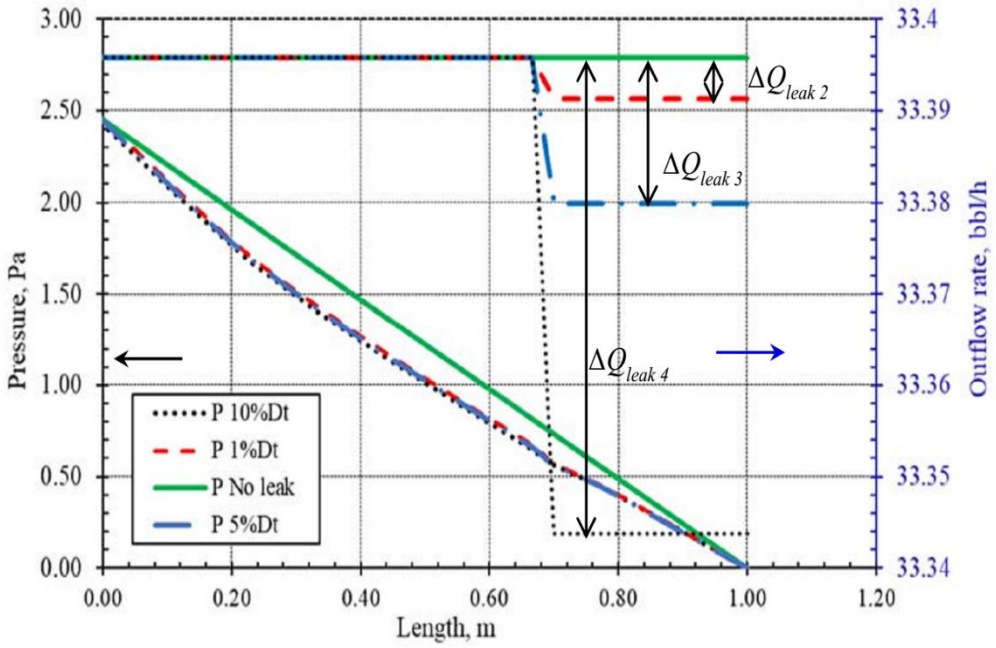


Figure 2-7: Change in pressure and flow rate along the pipeline due to the occurrence of leakages (de Sousa and Romero 2017).

Molina-Espinosa *et al.*(2013) carried out numerical modelling backed up by physical experiments for pipe leakage. This study investigated the transient modelling of incompressible flow in pipes with leaks. The experiments were performed using a 2.33 m long horizontal pipe of 0.0127 m internal diameter. The fluid considered for the experiment is water, while the static pressure tapping points ($P_1 - P_6$) distributed along the main measurement section. A discharged valve was used to create leakages. A series of experiments were carried out and compared with the numerical solution. The obtained results revealed good correlations between the simulation and

experimental data in terms of pressure drop within the vicinity of the leakages. The researchers reported that the pressure decreases due to a leak and its influence on the overall pressure distribution is a function of the leak size and the pipe inlet flow rate. However, this study only considered pipeline leakage as a function of leak sizes. Exploring other leak scenarios would be useful for pipeline engineer's emergency planning. A relevant study proposed by Zhu *et al.*(2014) modelled oil released from submarine pipelines subjected to different leak sizes. This study investigated the effects of oil leak rate, leak sizes, oil density, and water velocity on the oil spill behaviour using the Volume of Fluid (VOF) method (Chinello et al., 2019). This study revealed that small leak size, slow leaking and high fluid density led to a long period for oil to reach the maximum horizontal migrate distance. In a similar study by Li *et al.*(2018), a numerical investigation of submarine pipeline spillage was carried out using ANSYS Fluent to forecast the trajectory movement of oil spills.

The quantity and trajectory of spilt oil under various operating pressure, current sea velocities and wavelengths were analysed and compared. Li *et al.*(2017) employed CFD models to describe the underwater oil release rate and its trajectory movement from the damaged subsea pipeline to the free surface of the water. The simulated results revealed that the developed model could provide a detailed understanding of pipeline leakages, such as gas release rate, horizontal dispersion distance and gas rising time in a subsea environment, and hence reduced the cost and number of physical experiments. However, gas movement trajectory behaviour can only be predicted in a shallow ocean as the sea wave can easily alter the dispersion movement of the leaking fluid. The numerical modelling of pipeline leakages reported in the literature (Wei and Masuri, 2019; Li et al., 2018; Zhu et al., 2014) shows that dynamics modelling can provide an easy method for creating and analysing models that mimic the actual pipeline in the fields.

The extensive review reveals that, at present, the literature on a multiphase pipeline system is rather limited. Research on multiphase pipeline leakage lacks attention. Most of the available literature focuses on single-phase flow

systems, and few studies on the system that simultaneously conveys more than one hydrocarbon fluid have been reported. Multiphase flow systems are inherently nonlinear due to the continuous interaction of the participating phases; hence, it is challenging to accurately capture their dynamics. Multiphase flow systems are commonly found in nuclear reactors, chemical processes and the oil and gas industry. As such, the development of an accurate leak prediction model is timely and essential as this will aid in advancing rapid pipeline leak detection technologies for these critical applications. In the context of multiphase pipeline leak detection, the computational study by Kam (2010) investigated the influence of leak sizes and the longitudinal locations of the leak on fluid flow parameters. However, this study was only limited to a 1-D pipeline, assuming that the pipeline was made up of a series of small segments in which each node along the pipe modelled the local flow characteristics.

Figueiredo *et al.*(2017) investigated the effect of leakage on two-phase flow behaviour in nearly horizontal pipeline of 45 km long with a diameter of 0.450 m. Their study examined the effect of longitudinal leak location on stratified flows. A mineral oil with a $719.7\ kg/m^2$ density and vapour were the fluids used. The study assumed that the leaks occurred through a circular hole with a diameter of 0.0138 m and positions at 12.5 km, 22.5 km and 32.5 km. The obtained results revealed that the leak localisation strategy based on the upstream and downstream pressure profiles commonly employed in mono-phase flow leakage could be extended to the two-phase stratified flow pattern, typically observed in many production gas pipelines. The limitation of this work, however, is that the study is limited to a 1-D pipeline. The empirical models do not adequately capture all the dynamics of the multiphase flow behaviour. The assumptions of these analytical solutions limit their capability to consider different scenarios in which leaks may occur in 3-D pipelines.

The 3-D CFD modelling approach promises to be an effective tool to investigate complex multiphase flow problems (Alghurabi et al., 2021; Demirel et al., 2017; Saeedipour et al., 2019). It avoids unrealistic assumptions usually adopted in the empirical models for multiphase pipeline

leakage. In addition, CFD models provide an opportunity to incorporate intricate pipeline configurations and offer detailed information on multiphase flow systems, which may be challenging to obtain using analytical models or physical experiments. In particular, 3-D CFD models can readily investigate the influence of the radial position of the leak along the circumference of the pipeline relative to the gas-liquid interface. Araújo *et al.*(2014) investigated the influence of leaks in the hydrodynamics of oil-water two-phase flow in a horizontal pipeline. The simulation was performed in ANSYS CFX using the Eulerian-Eulerian model by considering the oil as a continuous phase and water as a dispersed phase. The authors varied the volume fraction of oil at the inlet of the pipeline and observed that the amount of oil discharged from the leak region reached a stable value after around 0.4 s for all the simulations reported in their study. However, their study is limited to the leak effect prior to the flow stability time. Also, the applicability of their study may be limited since they did not report a particular flow pattern. Besides, the effects of radial and longitudinal leak locations, leak opening sizes and multiple leakages remain to be investigated.

Therefore, to better understand the fluid flow behaviour induced by leaks for the aforementioned effects, one of the objectives set in this thesis is to extend the multiphase pipeline leakage to both before and after the flow stability state. A comprehensive assessment of different leak sizes, longitudinal leak locations, radial positions and simultaneous occurrence of leakages is performed for a gas-liquid pipeline system. The perturbation of the pertinent flow field indicators for different leak scenarios is essential to improving the understanding of multiphase flow behaviour induced by leaks.

2.5 Fluid Flow Characteristics

Classification of fluid flow is usually based on common characteristics. Some of the major categories are steady versus transient flow, laminar versus turbulent flow and compressible versus incompressible flow.

2.5.1 Laminar and Turbulent Flow

Fluid flowing in a pipeline could either flow smoothly or chaotic manner. There are three flow regimes, namely laminar, turbulent and transitional flows (Suman, 2014). It is important to identify the flow regime present in the pipeline to accurately model pipeline leakage. A parabolic velocity profile characterises laminar flow in a pipe (Vítkovský et al., 2003). The turbulent flow, however, is chaotic, with perpendicular movement, swirls and random fluctuations. Between laminar and turbulent flow, with laminar flow near the edge of the pipe and turbulent in the pipe, the centre is known as transitional flow. In general, these flow regimes (laminar, turbulent and transitional) describe fluid flow in a duct. The curve in Figure 2.8 shows the time dependence between the fluid velocity at point A in the flow regime. The velocity component that exists in laminar flow is in only one dimension $\vec{V} = u\hat{\imath}$, and random 3D velocity components $\vec{V} = u\hat{\imath} + v\hat{\jmath} + w\hat{k}$ are predominant for turbulent flow. When flow is laminar in state, there are disturbances that occasionally dampen (Suman, 2014); the Reynolds number, Re, plays a key role as a parameter for deciding the characteristic of the flow regime in a pipeline. With the moderate Reynolds number (10^2< Re <10^3) the flow may be laminar, but as the Reynolds number increases, the flow becomes lost in the orderly low pattern and velocity fluctuation, and subsequently, the flow becomes turbulent. Reynolds number is the key parameter in analysing different flow types when there is a substantial velocity gradient (shear) and is expressed as (Birkeland, 2014):

$$Re = \frac{\rho UD}{\mu} = \frac{UD}{v} \tag{2.18}$$

where ρ is the density of the fluid (kg/m^3), U is the flow velocity of the fluid (m/s), D is a linear dimension characteristic (m), μ is the fluid dynamic viscosity ($Pa.s\ or\ N.s/m^2$), and v is the kinematic viscosity of the fluid (m^2/s) obtained as μ/ρ. The characteristic of the flow regime in a duct can be approximated into different ranges, as shown in Table 2.2.

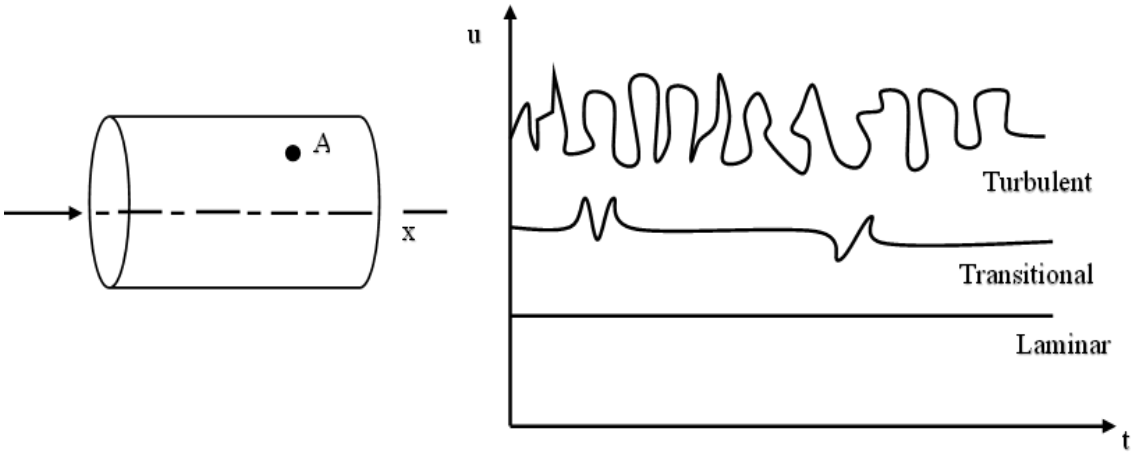


Figure 2-8: Time dependent fluid velocity at a point (SUMAN, 2014)

Table 2-2: Reynolds number ranges for different flow characteristics in a duct (Suman 2014)

Characteristics	Reynolds number
Highly viscous laminar motion	0 < Re < 1
Laminar and Reynolds number dependence	0 < Re < 100
Laminar boundary layer	10^2< Re <10^3
Transition to turbulence	10^3< Re <10^4
Turbulent boundary layer	10^4< Re <10^6

Pipeline leakage modelling can be challenging if the flow in the pipe is within the turbulent flow regime (Birkeland, 2014). Some researchers have investigated the unsteady behaviour of the pipeline leakage for various Reynolds numbers. Silva et al. (1996) developed a computational approach to analyse hydraulic transients caused by pipeline leakage. Different liquid flow rates and Reynolds numbers ranging from 2000 to 13000 were considered. Figure 2.9 illustrates obtained maximum pressure deviation as a function of leak magnitude for various Reynolds numbers. The experiments were carried out using 433 m and 1248 m long pipelines. The study indicates that turbulence modelling is more difficult than the laminar model. However, leak detection in turbulent flow is easier than in laminar

flow. Figure 2.9 shows leaks are detected when their magnitude is higher than 20% at the low Reynolds numbers (laminar flow), while 5% of leaks were easily detected in the turbulent regime.

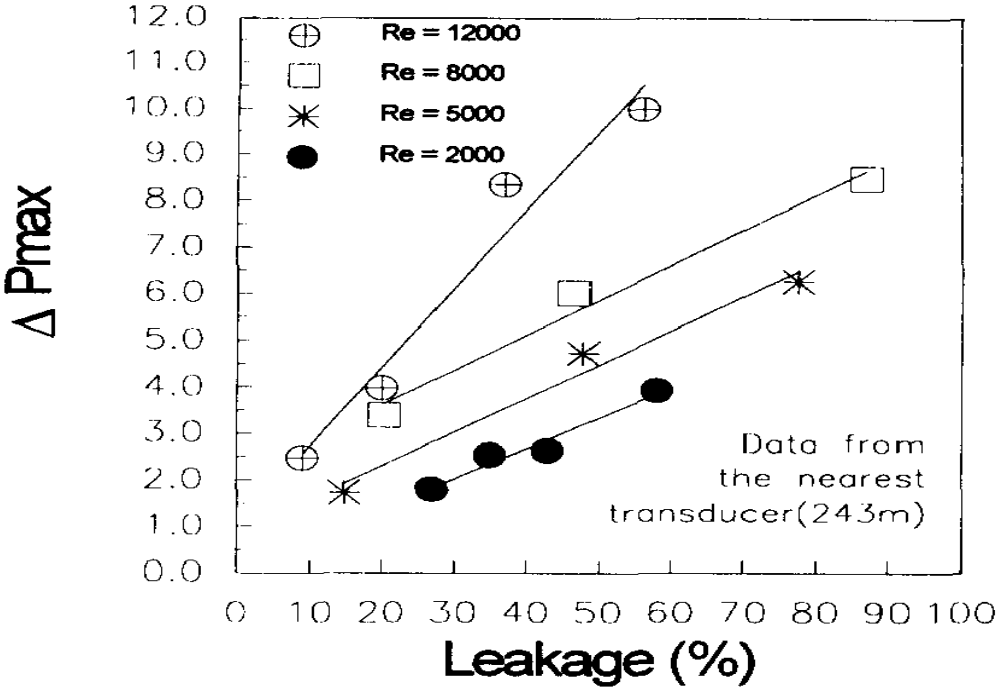


Figure 2-9: Pressure deviation profiles for various leak rate magnitudes (Silva et al., 1996)

Vardy and Brown (1996) derived a weighted function for smooth-pipe turbulent flow using a frozen viscosity model. The viscosity distribution was based on a core region with a constant turbulence viscosity and an outer shear layer with a linear change in viscosity from laminar (at the pipe wall) to turbulent viscosity (at the core or shear-layer interface). The weighted function is given as:

$$\mathrm{W}(\tau) = \frac{1}{\sqrt[2]{\pi\tau}} e^{\left(-\frac{1}{C^*}\tau\right)} \tag{2.19}$$

where τ is the dimensionless time defined as $\frac{4Vt}{D^2}$, C^* is the shear decay coefficient. The shear Deacy coefficient for laminar flows is defined as 0.99476. However, the turbulent flow is dependent on the Reynolds number of the mean flow given as:

$$C^* = \frac{7.41}{Re^k} \tag{2.20}$$

and

$$\mathrm{k} = log_{10}\left(\frac{14.3}{Re^{0.05}}\right) \tag{2.21}$$

where $\boldsymbol{Re}$ is the Reynolds number. Vítkovský et al. (2003) present a pipeline leakage detection model using Vardy and Brown (1996) weighting function model in the frequency domain. Vítkovský et al. (2003) employed frequency components of steady friction to investigate pipeline leakages using a pipeline length of 2000 m with a diameter of 0.3 m. An analytic solution for unsteady friction in both time and frequency domains was performed on a pipeline between the two tanks, as shown in Figure 2.10. A valve located at the downstream tank was used to introduce transients into the system through a small disruption in the valve position. The flow in the pipeline has a Reynolds number of 46,044 at steady state. A minimal non-linear behaviour was created by fluctuating the magnitude of the valve. The behaviour of the unsteady friction-weighted function for both laminar and turbulent flows created is shown in Figure 2.11. It was reported that weighted friction for turbulent flow is less dependent on historical acceleration than laminar flow due to a more uniform velocity distribution. The leakage analyses conducted demonstrated that the time and frequency domain could benefit from better treatment of unsteady friction. Similarly, the leakage detection method implemented could potentially be benefited from better treatment of unsteady friction.

These studies have demonstrated the effectiveness of detecting pipeline leakage in laminar and turbulent flows. However, more pilot studies using dynamics modelling is essential to evaluate the feasibility and improve upon the current studies. Pipeline leakage detection and, particularly the multiphase flow system warrant further study due to the limited studies currently available in the literature (Behari *et al.*, 2020). In this regard, this study is proposed to investigate the effect of leakage on multiphase pipelines considering different leakage conditions such as leak sizes, leak locations, axial leak positions and multiple leakages.

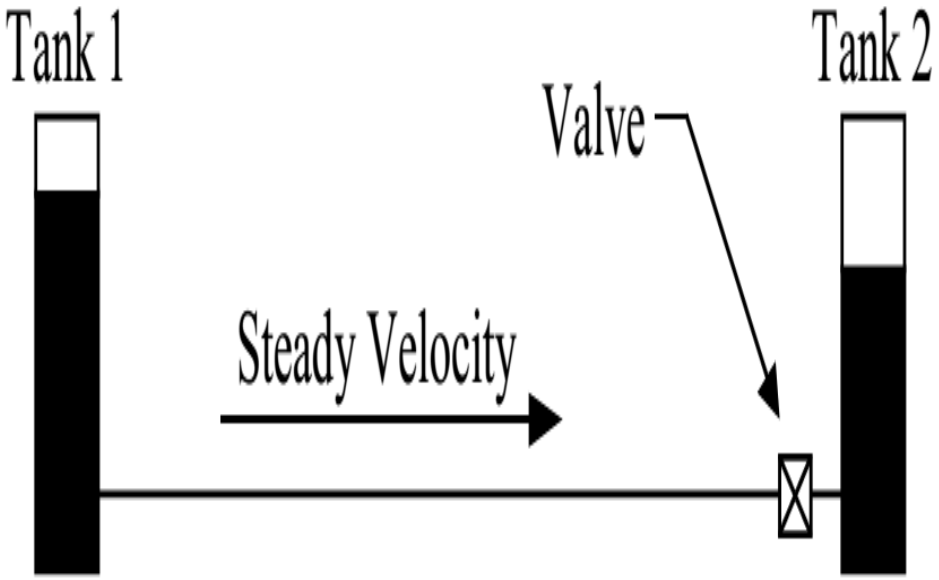


Figure 2-10: Illustration of pipeline considered for numerical analysis by Vítkovský et al. (2003)

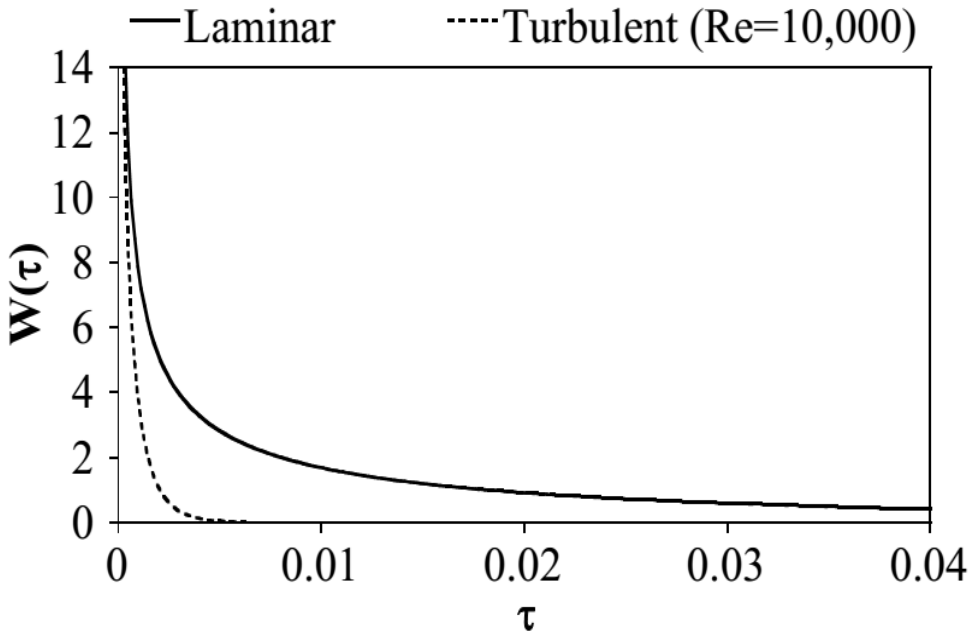


Figure 2-11: Unsteady friction-weighted behaviour of laminar and turbulent flows (Vítkovský et al., 2003) (W is is define as weighting function and τ is dimensionless time)

2.5.2 Steady and Transient Flow

A state where the various parameters associated with fluid transport at any point do not change with time is referred to as a steady state. In a steady operation, the mass, density, volume and total energy content of a steady flow section or device remain constant, and the local derivative of the velocity is zero at a steady state (Xu and Karney, 2017). The term transient flow is referred to as the intermediate state flow that describes the transition between two steady states. Any disturbance or variation between the two steady states, whether it is accidental or deliberate, can lead to transient conditions. Different scenarios initiate different transient conditions in a pipeline network, including operating conditions such as starting or stopping of pumps, sudden variations in valve or pump settings and sudden changes or anomalies incidents such as rupture along the pipeline networks. Characteristically, the term transient flow in a pipeline describes an unsteady flow phenomenon (Xu and Karney, 2017).

2.5.3 Compressible and Incompressible Flow

In general, fluid density can vary due to temperature or pressure variations. When fluid density varies significantly within the flow channel, the flow can be regarded as a compressible flow. On the contrary, the flow is mainly treated as incompressible flow when the variation of density in the flow domain is negligible. This is true for liquids based on the fact that the density of the liquid decreases moderately with pressure and slightly with temperature across a spectrum of operating conditions (Wesseling, 2009). The flow velocity is usually stated in terms of the dimensionless Mach number (Ma) when analysing high speed gas flows, and is expressed as:

$$Ma = \frac{V}{c} = \frac{\text{Speed of flow}}{\text{Speed of sound}} \tag{2.22}$$

where c is the air sound speed at room temperature at sea level with the value of $346\,m/s$. Gas behaviour is usually incompressible at low speeds. However, as the Mach number increases above 0.3, the effect of compressibility becomes essential. The relationship between the Mach number and the equivalent flow regimes is shown in Figure 2.12.

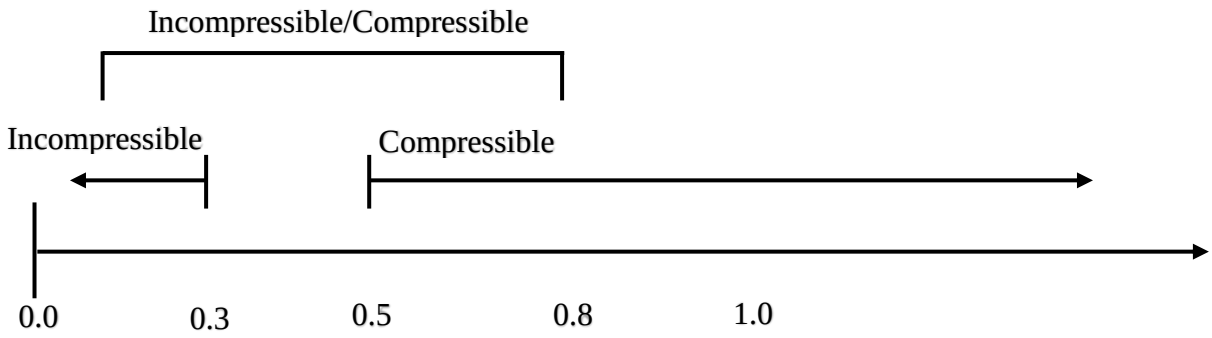


Figure 2-12: Flow regime Mach number values (Denk, 2007)

2.6 Gas-liquid Flow Regimes in Horizontal Pipes

Multiphase flow is often characterised by gas and liquid simultaneously occurring. They are sometimes included solid in the mixture. In the case of gas-liquid flow, different flow patterns are observed in a horizontal or slightly inclined pipeline. Different flow patterns or flow regimes are encountered when gas and liquid flow simultaneously flow inside a pipe. These regimes depend on different factors, including the pipeline diameter, fluid properties and flow rates of each of the phases. Two-phase flow patterns in horizontal pipelines are similar to the patterns in vertical pipes, except that the liquid's distribution is influenced by gravity that acts to stratify the gas to the top and liquid to the bottom of the pipe (Garbai & Sánta, 2012). The simultaneous flow of gas and liquid in a horizontal pipe often results in various flow regimes, as shown in Figure 2.13 and summarised following the description of Garbai and Sánta (2012).

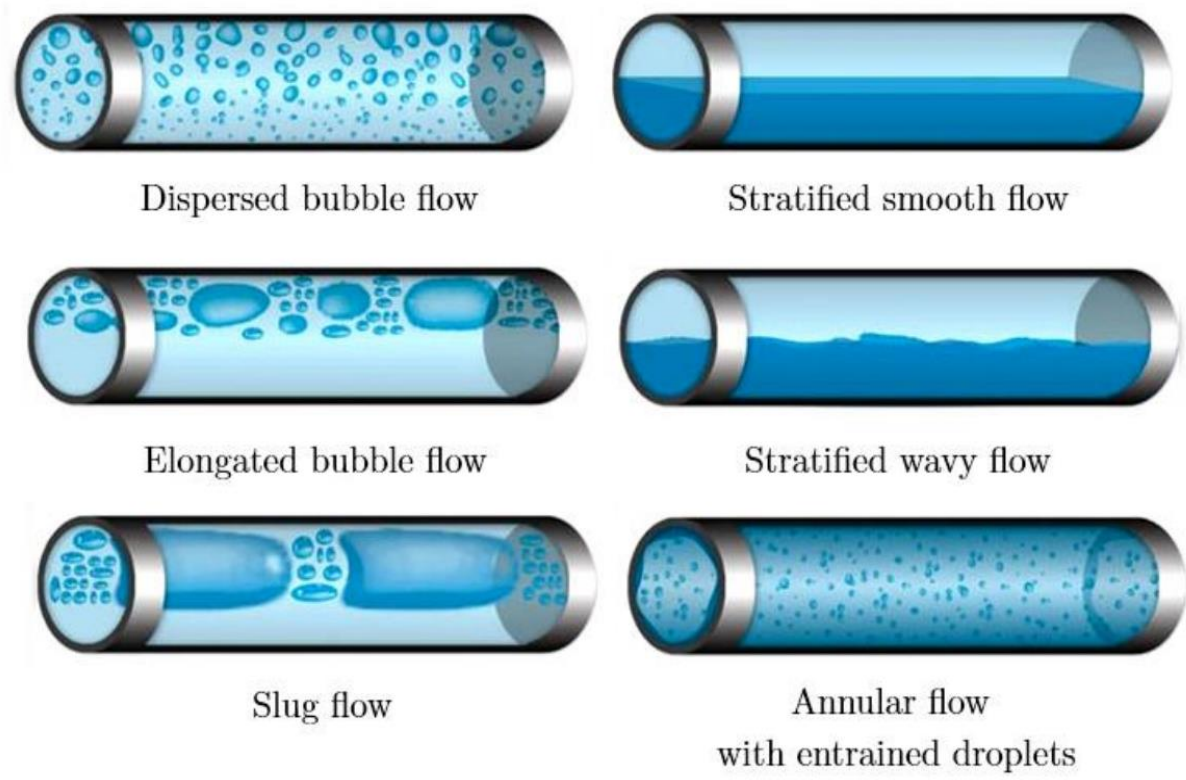


Figure 2-13: Gas-liquid flow regimes in the horizontal pipe (Garbai and Sánta, 2012)

Dispersed bubble flow is commonly occurring at a very high liquid flow rate. This flow regime is characterised by the flow where one phase is dispersed in the other continuous phase. The elongated bubble flow is regarded as a special case of plug flow. The bubbles are longer in this flow regime and may be differentiated by a tail and a nose. In the dispersed bubble flow, the phases appear fairly distributed uniformly within each other in axial and radial directions of the flow. The annular flow regime occurs at high gas flow rates. In this flow pattern, the liquid phase is forced to form a continuous annular layer around the pipe perimeter. While the interface between the vapour core and the liquid annulus is distraught by small-amplitude waves and results in droplets and scatter in the gas core. A stratified flow pattern occurs for relatively low gas and liquid flow rates. Complete separation of the gas and liquid is generally observed in which gas moves to the top and liquid to the bottom of the pipe and is separated by a horizontal interface. An increase in the stratified flow's gas velocity resulted in the waves forming at the gas and liquid interface and travelling in the direction of flow called stratified-wavy flow. Most gas-phase accumulates to form large bullet-

shaped bubbles in slug flow. A slug pattern exists at relatively high gas flows, and the slug of highly aerated liquid moves downstream pipe at the gas velocity on average. Considering the importance of multiphase flow in the oil and gas industry, it is essential, especially from the economic point of view, to investigate the effect of leakage on multiphase flow pipes. Therefore, this study considers stratified flow for investigation as it is one of the most frequently encountered in long-distance horizontal flow lines, such as steam, natural gas and oil flow, in petrochemicals, power generation and process plants (Ali, 2017; Cheng et al., 2020; Vlachos et al., 1999). The benefit of the study would aid not only the efficient operation of two-phase systems but also help in improving the understanding of multiphase flow behaviour induced by pipeline leakage, which can be useful for risk assessment and improve the emergency management level.

2.6.1 Review of two-phase flow pattern maps in horizontal pipe

Various experimental and theoretical analyses have been performed over the years to predict the two-phase flow patterns present in the flow regime map as accurately as possible (Baker, 1954; Scott, 1964; Beggs and Brill, 1973). Many of these studies presented the pattern maps in terms of dimensional parameters, such as physical properties of the working fluids, while some literature used non-dimensional properties. A two-phase flow pattern map proposed in (Baker, 1954), which is also regarded as one of the oldest pattern maps employed superficial mass velocities $(G_G^*/\lambda \; and \; G_L^*/\Psi)$ and fluids properties, as shown in Figure 2.14.

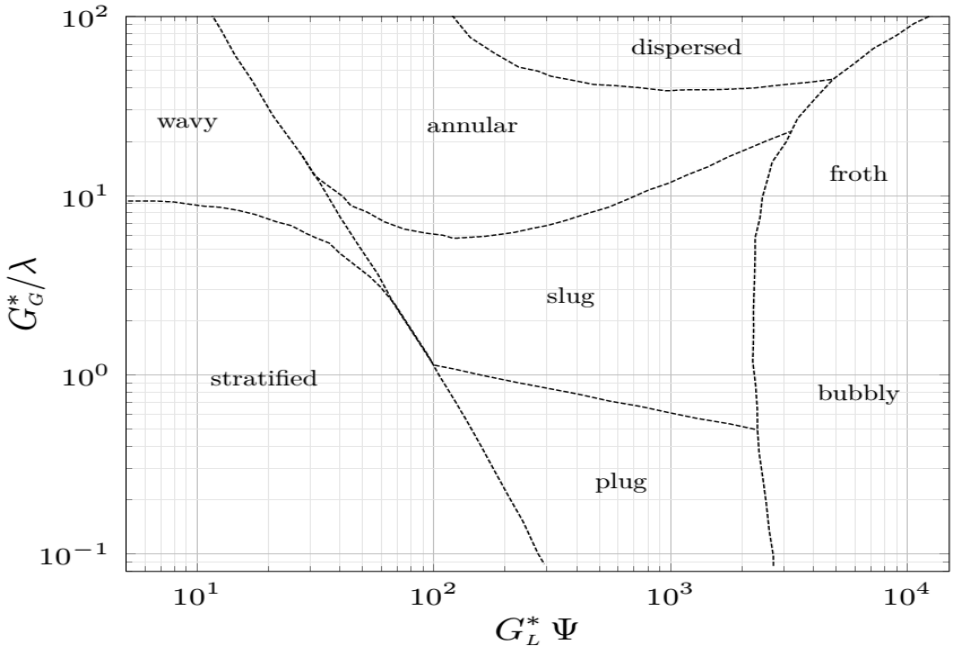


Figure 2-14: Baker prediction map for two-phase flow patterns along horizontal pipes (Baker, 1954).

A wide range of working fluids was covered in the flow patterns map of Baker (1954) using the fluids property correlation factors (λ and ψ) with gas-phase velocity on the y-axis and the liquid-phase velocity on the x-axis. The λ and ψ parameters were defined in terms of the physical properties of the working fluids and given as:

$$\lambda = \left[\left(\frac{\rho_G}{\rho_{GB}}\right)\left(\frac{\rho_L}{\rho_{LB}}\right)\right]^{1/2} \tag{2.23}$$

$$\Psi = \frac{\sigma_B}{\sigma}\left[\left(\frac{\mu_L}{\mu_{LB}}\right)\left(\frac{\rho_{LB}}{\rho_L}\right)^2\right]^{1/3} \tag{2.24}$$

Where μ_L is the dynamic viscosity of the liquid, σ is the surface tension, ρ_L and ρ_G are liquid and gas density, respectively. The other parameter values are $\mu_{LB} = 1.0x10\ Ns/m^2$, $\rho_{LB} = 997.9kg/m^2$, $\sigma_B = 0.073\ N/m$, $\rho_{GB} = 1.201$kg, while Ψ and λ are equal to 1 for air-water system at atmospheric pressure. In the study of Baker (1954), the pattern map was divided into the plug, slug, stratified, annular, wavy, dispersed and bubble flow regions. It was reported in a later study by Scott (1964) that the effects of pipe diameter were not considered in the Baker map and, therefore, may not represent the major

parameters affecting the transition from one phase to another. Scott (1964) also reported that Baker map did not account for a number of interacting forces in two-phase flow, such as surface tension and gravity. This led to an alternative transition map proposed in (Scott, 1964). Scott modified Barker's map to illustrate the transition regimes as regions instead of lines to exemplify the levels of uncertainty, as illustrated in Figure 2.15. Also excluded from Baker map was the transition line between the annular and dispersed flow.

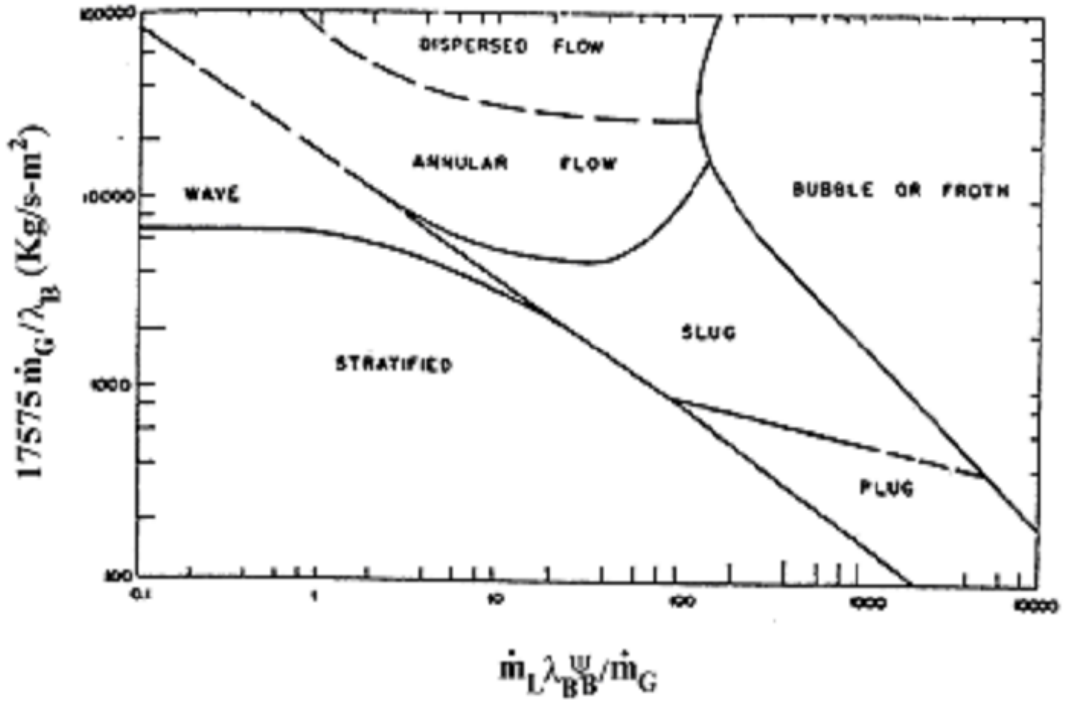


Figure 2-15: An extension of Baker (1954) flow pattern map, proposed by Scott (1964).

In an attempt to simplify the flow pattern maps, Beggs and Brill (1973) considered only three regions, namely separated flow (stratified, wavy and annular flow), intermittent flow (plug and slug flow) and distributed flow (bubble flow). The Beggs and Brill map was based on a system of fixed properties and reported failed to account for viscosity, density and interfacial tension variation. Similarly, their transition lines are based on best fits. Therefore only truly applicable to the systems similar to those in which they are reported (Emamzadeh, 2012). In a similar study carried out

by Mandhane et al. (1974) at the University of Calgary, a multiphase pipe flow data bank was employed to generate a basic flow pattern map of superficial gas velocity versus superficial liquid velocity. The illustration of Mandhane et al. (1974) flow pattern map is presented in Figure 2.16. The proposed map was divided into six regimes: slug, dispersed, stratified, annular, wave and elongated bubble flow. Extensive experiments were conducted using different pipe diameters, superficial gas velocities (v_{sq}), superficial liquid velocities (v_{sl}), surface tensions, and liquid and gas phase densities. It was reported that despite Mandhane et al. (1974) being limited totally to the correlational approach, it provided a better prediction than many conditions in (Beggs and Brill, 1973) map.

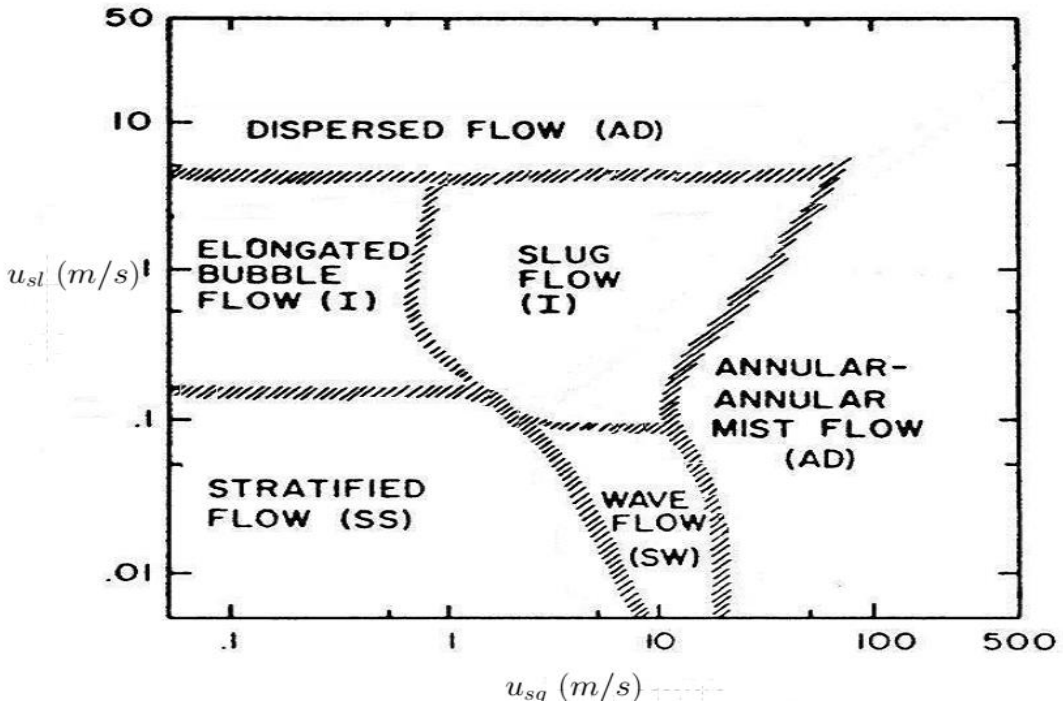


Figure 2-16: The Mandhane *et al.* flow pattern map (Mandhane et al., 1974)

The flow regime transitions reported by Taitel and Dukler (1976) used a semi-theoretical approach. They classified their map into stratified wavy, annular, stratified smooth and dispersed bubble flow. With a one-dimensional steady-state flow model adopted, they derived separate momentum balances for the gas and liquid phases and then combined them by eliminating the pressure gradient terms. By assuming that the liquid

layer is of constant height, with a smooth gas-liquid interface, and that the interfacial shear term is equal to the gas-wall shear term, they then derived a non-dimensional form of this combined momentum (Emamzadeh, 2012).

To improve the transition prediction between the intermittent and annular flow reported by Taitel and Dukler (1976), Hale (2001) suggested that transition should occur where $\widetilde{h_l} < 0.5h_l$. Using the slug body holdups raging from $0.7\ to\ 1.0$ suggested in (Dukler and Hubbard, 1975) led to liquid height (h_l) over pipe diameter $\left({}^{h_l}/_{D}\right)$ values to the range of 0.35 and 0.5 at the transition, which reported agrees better with the experimental observations of Mandhane et al. (1974) and Barnea (1987). This was demonstrated in (Taitel and Dukler, 1976) flow map shown in Figure 2.17 as the transition between intermittent and annular is indicated as a region between the two curves (B).

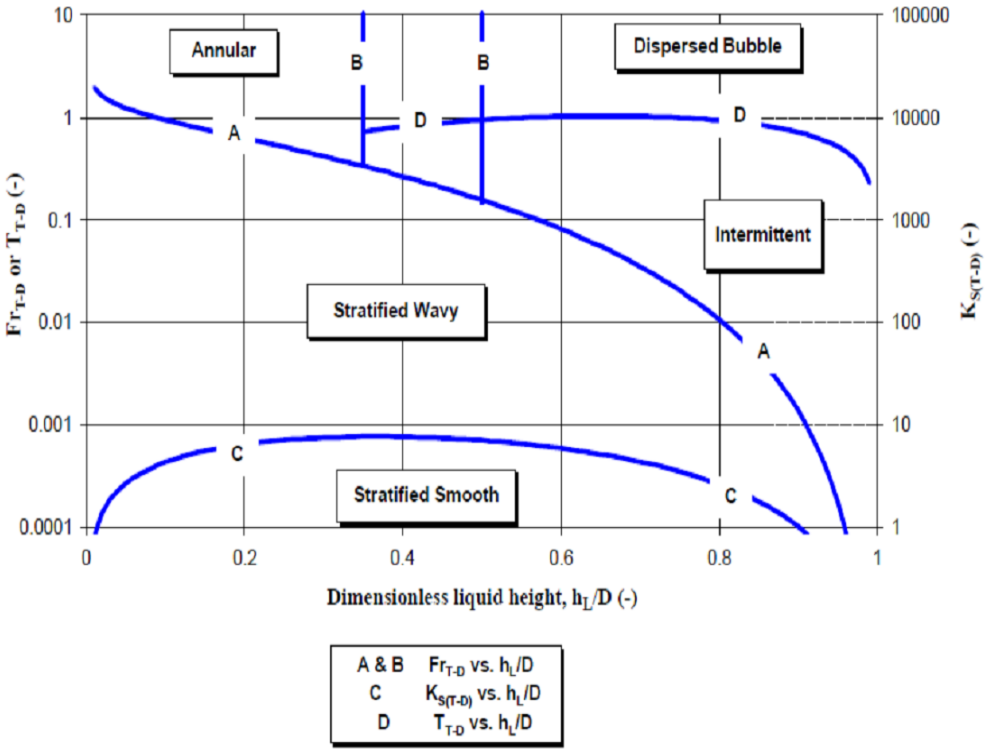


Figure 2-17: The Taitel and Dukler (1976) flow pattern map.

A method to account for variations in fluid properties and different pipe diameters was proposed by Weisman et al. (1979). The study aims to provide a method where changing one fluid property would not significantly affect the other fluid properties. A wide range of fluid velocities with different pipe diameters, including 12 mm, 25 mm and 51 mm, were employed for the experimentations. The flow patterns included in their map are dispersed, stratified smooth, stratified wavy, intermittent and annular regimes. The proposed transitions are for stratified smooth to stratified wavy flow, the separated to intermittent flow, the transition to disperse flow and the onset of annular flow. The proposed transition was compared with the study of Mandhane et al. (1974) and Taitel and Dukler (1976). It was reported that the annular-intermittent boundary provided the most notable feature as it exhibits an opposite trend to their predictions. The illustration of the Weisman et al. (1979) flow pattern map is presented in Figure 2.18. A number of dimensionless correlations for the flow regime transition boundaries were proposed in (Weisman et al., 1979), as given in Table 2.3.

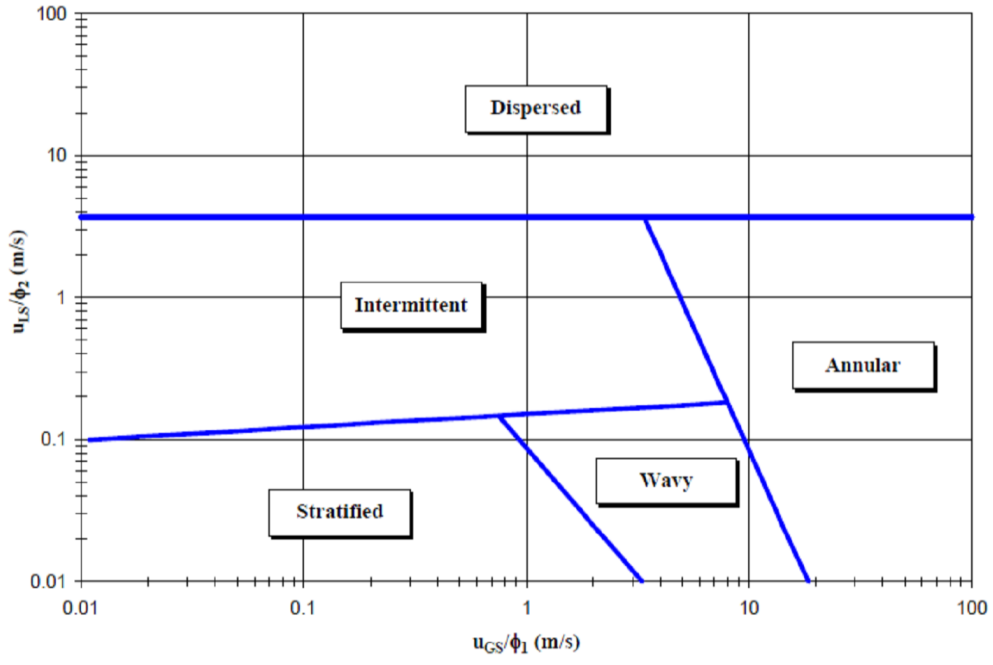


Figure 2-18: The Weisman et al. (1979) flow pattern map

Table 2-3 Weisman et al. (1979) transition boundaries. Note that these parameters' definitions and values are the same as Baker (1954) parameters presented on the previous pages.

Stratified smooth to stratified wavy flow: $$\left[\frac{\sigma}{(\rho_L-\rho_G)\mathrm{g}D^2}\right]^{0.20}\left[\frac{\rho_G U_{SG} D}{(\mu_G)}\right]^{0.45} = 8\left(\frac{U_{SG}}{U_{SL}}\right)^{0.16}$$
Separated-intermittent transition: $$(Fr_G)^{1/2} = 0.25\left(\frac{U_{SG}}{U_{SL}}\right)^{1.1}$$
Transition to dispersed flow: $$\left[\frac{\left(dp/dx\right)_{SL}}{(\rho_L-\rho_G)\mathrm{g}}\right]^{0.50}\left[\frac{\sigma}{\left((\rho_L-\rho_G)\mathrm{g}D^2\right)}\right]^{-0.25} = 9.7$$
Transition to annular flow: $$1.9\left(\frac{U_{SG}}{U_{SL}}\right)^{1/8} = Ku_G 0.2_{Fr_G{}^{0.18}}$$

A similar transition criterion proposed by Taitel and Dukler (1976) was employed by Barnea (1987) to predict the transition from annular and dispersed bubbles to intermittent flow. An extensive model for predicting steady-state transition boundaries for the whole range of inclinations pipe is presented. It was reported that the transition from annular flow occurs when the gas core is blocked at any location by liquid. Similarly, the interfacial shear stress has its minimum value when the transition to unstable annular flow occurs. Figure 2.19 illustrates the Barnea (1987) flow pattern map computed for air-water flow at 1 bar pressure. Barnea (1987) suggested that the transition from dispersed bubble to intermittent flow occurs when one of the following conditions is satisfied: (a) agglomeration of large distorted bubbles and (b) migration bubbles due to buoyancy effects to the upper part of the pipe. It was observed that Barnea (1987) map is slightly different from that of Taitel and Dukler (1976) map.

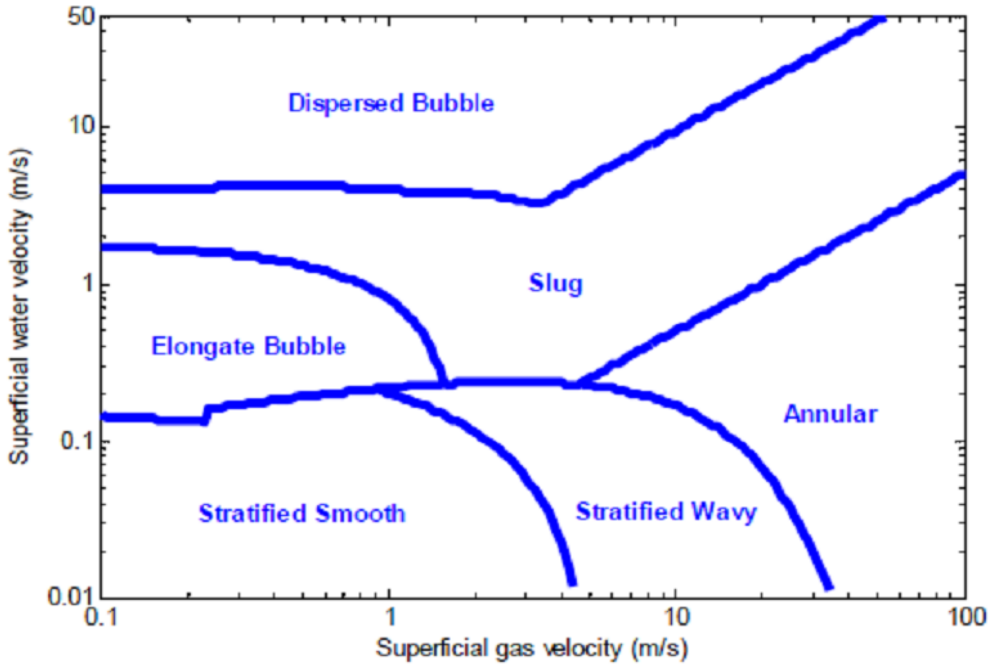


Figure 2-19: The Barnea (1987) flow pattern map

A unified model for gas-liquid flow was proposed in (Zhang et al., 2003) to predict flow pattern transition at the pipe inclination from -90^0 to 90^0. The flow patterns considered were classified into the following groups: elongated bubble, bubble and dispersed flow, stratified and annular, and slug and churn flow. The proposed model was compared with the series of experimental measurements, and it was reported that the proposed model provided a good prediction for horizontal and upward-inclined flows. A comprehensive study of the flow regime transitions for air and non-Newtonian fluid systems were reported by Xu et al. (2007). The proposed study was carried out using 10 m long and 20 to 60 mm diameter tubes inclined at different angles for shallow inclinations. Their measurements were reported in good agreement with the (Barnea, 1987).

More recently, an experimental study on adiabatic and condensation two-phase flow patterns map and their transitions in a horizontal tube with an inner diameter of 4 mm was presented in (Song et al., 2018). The experiments were conducted using mass fluxes ranging from 200 to 650 $\mathrm{kg}/(m^2\mathrm{s})$. The effects of mass flux, saturation pressure and heat flux on flow pattern transitions were studied and analysed. The observed results were compared with six well-known flow pattern maps. It was reported that only

some of the obtained results could predict all the transition lines accurately. The study also proposed a new dimensionless number S_1, which takes into account the inertia force, gravity force, shear force and surface tension to develop the new adiabatic flow pattern transition criteria. The new adiabatic flow pattern map was compared with eight data groups. It was reported that most of the flow pattern data were predicted accurately.

2.6.2 Two-phase pressure drop prediction models

Three methods are widely used for predicting two-phase pressure drop. These methods are empirical, analytical and phenomenological (Al-Tameemi, 2018). The empirical models are simple to implement and usually provide good accuracy by using a range of available databases to develop correlations. The drawback of the empirical technique includes its total dependency on experimental data. It is also mostly limited to special experimental conditions. An analytical approach usually involves complex calculations and requires more time to obtain one solution. Most analytical approach models depend on mathematical assumptions and are limited to specific flow conditions (Al-Tameemi, 2018). The phenomenological method used theoretical analysis and experimental data to incorporate the flow pattern effect into the two-phase pressure drop.

Many scholars have proposed different empirical models to predict two-phase pressure drop along the straight pipe. These models are groups based on their assumption to express the two-phase flow. The commonly quoted empirical methods are reviewed as follows:

2.6.2.1 Homogeneous models

In the homogeneous models, two-phase is assumed to flow at the same velocity. Therefore, treated as an equivalent mono-phase flow with the specific volume of the mixture (gas-liquid flow) defined as (Filip et al., 2014):

$$v_H = xv_G + (1-x)_{v_L} \tag{2.25}$$

The density of the mixture is given as:

$$\rho_H = \frac{1}{v_H} = \left(\frac{x}{\rho_G} + \frac{1-x}{\rho_L}\right)^{-1} \tag{2.26}$$

where x is the gas mass quality, v_G is the volume of gas, v_L is the volume of liquid, ρ_G is the density of gas and ρ_L is the density of liquid.

By using homogeneous mixture dynamic viscosity, the Reynolds number is determined as:

$$Re = \frac{G * D}{\mu_H} \tag{2.27}$$

where G is the total mass flow rate per unit area and D is the internal pipe diameter.

The dynamic viscosity of the mixture is defined as:

$$\mu_H = \left(\frac{x}{\mu_G} + \frac{1-x}{\mu_L}\right)^{-1} \tag{2.28}$$

The friction factor is considered as:

$$f = \begin{cases} 16Re^{-1}, & \text{for } Re < 2000 \\ 0.079Re^{-0.25}, & \text{for } 2000 \leq Re < 20000 \\ 0.046Re^{-0.2}, & \text{for } \geq 20000 \end{cases} \tag{2.29}$$

2.6.2.2 Separated models

In the separated flow models, each phase is assumed to flow separately at different velocities in the pipe. Lockhart and Martinelli (1949) developed a two-phase flow pressure (p) drop model. This study is regarded as one of the earliest models developed for separate flow. The model suggested four different flow regimes: (1) flow is turbulent for both gas and liquid phases (tt), (2) flow is laminar for both gas and liquid phases (vv), (3) the liquid phase is laminar and the gas-phase is turbulent (vt), (4) the liquid phase is turbulent and the gas phase is laminar (tv). The pressure drop model is proposed based on the concept of different two-phase friction multipliers for the gas $(\emptyset_G^2)$ and liquid $(\emptyset_L^2)$ as follows (Filip et al., 2014):

$$\left(\frac{dp}{dz}\right) = \emptyset_L^2 \left(\frac{dp}{dz}\right)_L = \emptyset_G^2 \left(\frac{dp}{dz}\right)_G \tag{2.30}$$

$$\left(\frac{dp}{dz}\right)_L = \frac{2f_L G^2 (1-x)^2}{\rho_L D} \tag{2.31}$$

$$\left(\frac{dp}{dz}\right)_G = \frac{2f_L G^2 x^2}{\rho_G D} \tag{2.32}$$

where z is the tube length and f_L is the Froude number

The two-phase friction multipliers ($\emptyset^2$) were first presented in a graphical form and later expressed by Chisholm (1967) as the following dependence:

$$\emptyset_L^2 = 1 + \frac{C}{x} + \frac{1}{x^2} \tag{2.33}$$

$$\emptyset_G^2 = 1 + Cx + x^2 \tag{2.34}$$

where the Lockhart-Martinelli parameter x is defined as:

$$x^2 = \frac{\left({}^{dp}/_{dz}\right)_L}{\left({}^{dp}/_{dz}\right)_G} \tag{2.35}$$

The empirical parameter C values are defined by Chisholm (1967) for different flow conditions, as presented in Table 2.4. Illustrated in Figure 2.20 is the two-phase flow multiplier of Lockhart and Martinelli (1949) model for a wide range of flow conditions.

Table 2-4 Experimental values of the parameter C (Chisholm, 1967)

Gas-phase	Liquid-phase	Symbol	C
turbulent	Turbulent	(tt)	20
turbulent	laminar	(vv)	12
laminar	Turbulent	(vt)	10
laminar	laminar	(tv)	5

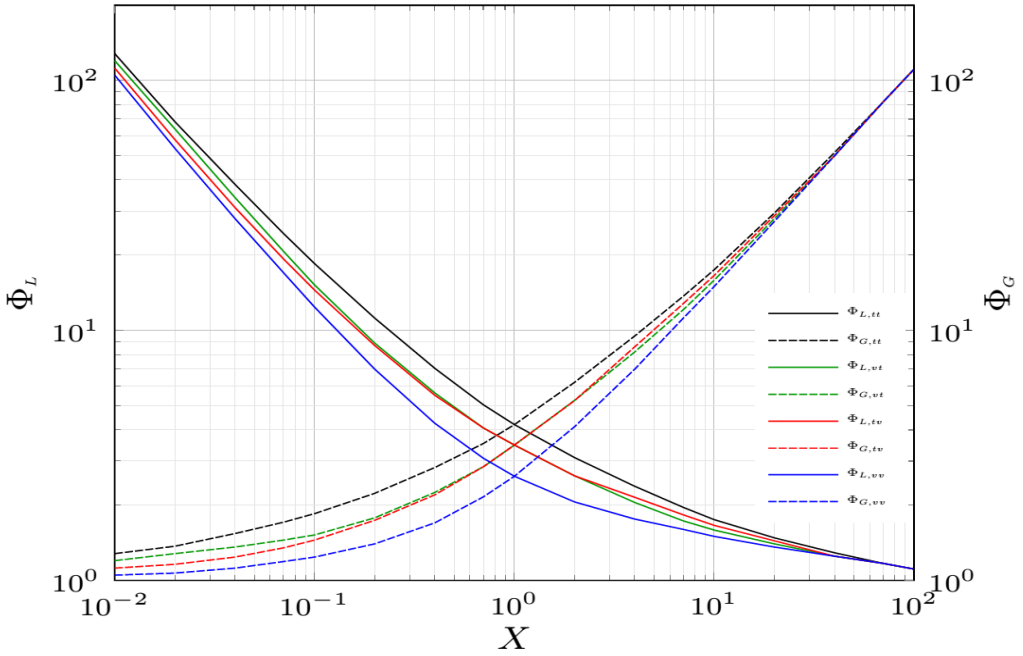


Figure 2-20: Two-phase flow multiplier Φ_k values in terms of Martinelli parameter for wide range of flow conditions (Al-Tameemi, 2018)

2.6.2.3 Friedel (1979) model

Friedel developed an improved friction pressure drop correlation for horizontal and vertical two-phase pipe flow in 1979 (Friedel, 1979). In the Friedel model, a liquid-only multiplier was proposed for the friction and defined as follows:

$$\left(\frac{dp}{dz}\right) = \left(\frac{dp}{dz}\right)_L * \emptyset_L^2 \tag{2.36}$$

The multiplier is defined as a function of gas-liquid properties of a mass vapour quality, gravity and surface tension effect using Froude and Weber:

$$\emptyset_L^2 = (1-x)^2 + \left(\frac{\rho_L}{\rho_G}\right)\left(\frac{f_G}{f_L}\right) + 3..24x^{0.78}(1-x)^{0.224}\left(\frac{\rho_L}{\rho_G}\right)^{0.91}\left(\frac{\mu_G}{\mu_L}\right)^{0.19}\left(1-\frac{\mu_G}{\mu_L}\right)^{0.7} Fr_{tp}^{-0.045} We_{tp}^{-0.035} \tag{2.37}$$

The Weber (We) and the Froude (Fr) were determined using the homogenous mixture density and expressed as follows:

$$We_{tp} = \frac{G^2 D}{\sigma \rho_H} \tag{2.38}$$

$$Fr_{tp} = \frac{G^2}{\mathrm{g}D\rho^2{}_H} \tag{2.39}$$

2.6.2.4 Chen et al. (2001) model

Chen et al. (2001) introduced dependence on Weber and Bond numbers as a correction of the homogeneous model. The pressure drop is expressed as:

$$\left(\frac{dp}{dZ}\right) = \left(\frac{dp}{dZ}\right)_{hom} \Omega_{hom} \tag{2.40}$$

where the correction factor is defined as:

$$\Omega_{hom} = \begin{cases} 1 + [0.2 - 0.9\exp{(Bo)}], & \text{for } Bo \geq 2.5 \\ \dfrac{We^{0.2}}{[\exp(Bo)]^{0.3}} - 0.9\exp(-Bo), & \text{for } Bo \geq 2.5 \end{cases} \tag{2.41}$$

The Bond number is expressed as:

$$Bo = \mathrm{g}(\rho_L - \rho_G)\frac{(D/2)^2}{\sigma} \tag{2.42}$$

The Weber number is considered with the homogeneous mixture density and given as:

$$We = \frac{G^2 D}{\sigma \rho_H} \tag{2.43}$$

Summary

This chapter presents a review of related studies on the surrogate model, including metaheuristic-assisted methods and pipeline leakage detection technologies.

The literature review has revealed that numerous studies have been carried out to investigate pipeline leakage detection and characterisation via parameters that characterise the fluid flow using different approaches, with dynamic modelling reported as the most sensitive method (Liu et al., 2019; Moore, 1999). The gap in knowledge is; however, most of the works reported focused on single-phase pipeline systems. Studies on multiphase pipeline systems are rather limited to a 1-D pipeline (Figueiredo et al., 2017; Kam, 2010) or do not consider a particular flow pattern (Araújo et al., 2014). Therefore, one of the objectives of this study is to investigate the accidental leakage of pipelines as a multiphase flow system. Studies

have shown that CFD is an effective but computationally intensive tool for modelling complex multiphase flow problems and provides wealthy data about the flow characteristic that can be translated into knowledge using algorithms such as machine learning. However, CFD model is highly expensive to modelled. It can take days or weeks to run a single simulation despite advancement in computing capacity. The surrogate model has been proposed to address the challenges of computationally expensive problems of CFD and has been proven faster in many orders of magnitude with satisfactory prediction accuracy. Nevertheless, the technical challenge for developing an efficient surrogate model is the appropriate selection of data points for fitness evaluation. This study proposed a new data sampling optimisation methodology to optimally select training datasets in machine learning applications involving computationally expensive problems like CFD simulation in the pipeline. The approach taken in this study incorporates the system fitness value information and population density of sample points to select candidate solutions for fitness evaluation. These two criteria would control the search direction of the algorithm to select data points in the region of high interest for fitness evaluation.

Chapter 3 Adaptive Surrogate Model Design and Optimisation

3.1 Introduction

The background and literature review related to this chapter are presented in Chapter 2. As cited, the PSO algorithm is typically used for optimisation because of its excellent global search capacity and simplicity. PSO has been introduced along with the surrogate model algorithm. Like many evolutionary algorithms, the PSO suffers from slow convergence and can easily fall into local optima, particularly in a non-linear region. This chapter proposed a novel PSO-assisted surrogate model called the adaptive PSO-assisted surrogate model (PSOASM) to optimise computationally expensive problems that are difficult to simulate or experiment. The proposed approach introduced two criteria: the fitness value information and population density of sample points to select the candidate solution for evaluations. The introduced technique is implemented to perform two functions – Guiding the exploitation move and the exploratory move. The candidate solution in a region with less population density is selected for function evaluation during the exploratory movement. If the fitness value yields improved results, the candidate solution would replace the global best position and archive for further exploitation. This process is repeated until any additional sample in that region does not improve the algorithm fitness value. The introduced candidate solution with the best fitness value can aid the proposed model in exploiting the promising region, while the distance measure can reduce the chance of the proposed model being stuck at the local optimum. Consequently, attain a better solution with the limited number of function evaluations. In addition, different machine learning algorithms are explored to evaluate the performance of the proposed surrogate model. Specifically, five different machine learning algorithms and four sequential sampling schemes are compared using different benchmark functions.

3.2 Proposed adaptive PSO-assisted surrogate model

The proposed PSOASM framework is described in this section. Figure 3.1 depicts the overall framework of the proposed surrogate model. The framework mainly revolves around three major concepts: design space sampling, sample points placement optimisation and construction or updating of the surrogate model. As illustrated in Figure 3.1, the red and black arrow lines represent the data flow and algorithm flow, respectively. The dashed arrow indicates the calling and retrieving of data from the expensive computational model like CFD simulation. The sample locations and their corresponding CFD simulation results are stored in the database. The execution process of the algorithm framework is described as follows: At the beginning, a set of algorithm parameters, including total initial sample size, iterations, PSO parameters and algorithm termination criteria, are defined, followed by initialisation of the initial training sample points using the Latin hypercube sampling (LHS) method. It is important to highlight that LHS was adopted over the other sequential samplings to generate initial training data in this study because it provides better thoroughness in coverage (Neto et al., 2014). Additionally, its uniform distribution in the design domain can aid in better estimation of the global accuracy of the surrogate model (Kang et al., 2016; S. Li et al., 2022; Zhang et al., 2022). The CFD simulation is performed based on the scenario represented by the generated points in the parameterised design parameter space. The initial sample points and obtained simulation results are stored in the database. A course surrogate model is then built with the stored samples and their corresponding simulation data. Note the surrogate model is constructed using machine learning techniques described in section (3.6), and the performance is evaluated using error threshold mechanics (mean squared error). The error threshold is used to determine if additional sample point improve the surrogate accuracy and, at the same time, output the surrogate model if the error reach or is less than the threshold. The samples in the archive are used to initialise the PSO-assisted surrogate population. The population then evolves N iteration by the optimisers based on two criteria: population density of sample points and surrogate fitness value.

These two criteria are employed alternatively when no better candidate solution can be found. The PSO-assisted method using fitness value information can search around the best individual candidate solution, while the samples distance criteria guide the algorithm toward better global searching ability. The fitness value of the surrogate is mainly used to exploit the current global best region, and global exploration is performed to find unexplored regions.

By integrating these two criteria, the proposed algorithm can combine the advantages of the two schemes to balance exploitation and exploration capabilities (Viana, 2016). At the beginning of the iteration, a particle with the utmost distance to the other particles is picked to form the global best of the swarm. Then, the velocity and position of the particles are updated. The new global best position is evaluated. If added sample point improves the surrogate model accuracy, then the algorithm enters the local surrogate stage. At this stage, the algorithm performs velocity and position updating according to equations (3.10) and (3.12), respectively and selects the new global position for fitness evaluation. If there is no improvement and the algorithm has not satisfied the termination conditions (i. minimum specified error value, ii. maximum iterations N_{max} or iii. when the algorithm does not improve after five successful iterations), update the PSO population with the entire sample points in the database to find the region that are not well explored in parameter space (using the population density of swarm). Then, select the sample with the utmost distance to the other samples to replace the global best position and performs the fitness function. Check whether the termination condition is satisfied. If it is satisfied, output the surrogate model; if not satisfied, then increment the iteration and execute the next step. Based on this explanation, the detailed description of the proposed surrogate model is presented in detail in sections 3.3 to 3.7. These include problem formulation, initial training data sampling generation, sample points placement optimisation, model development and model performance evaluation.

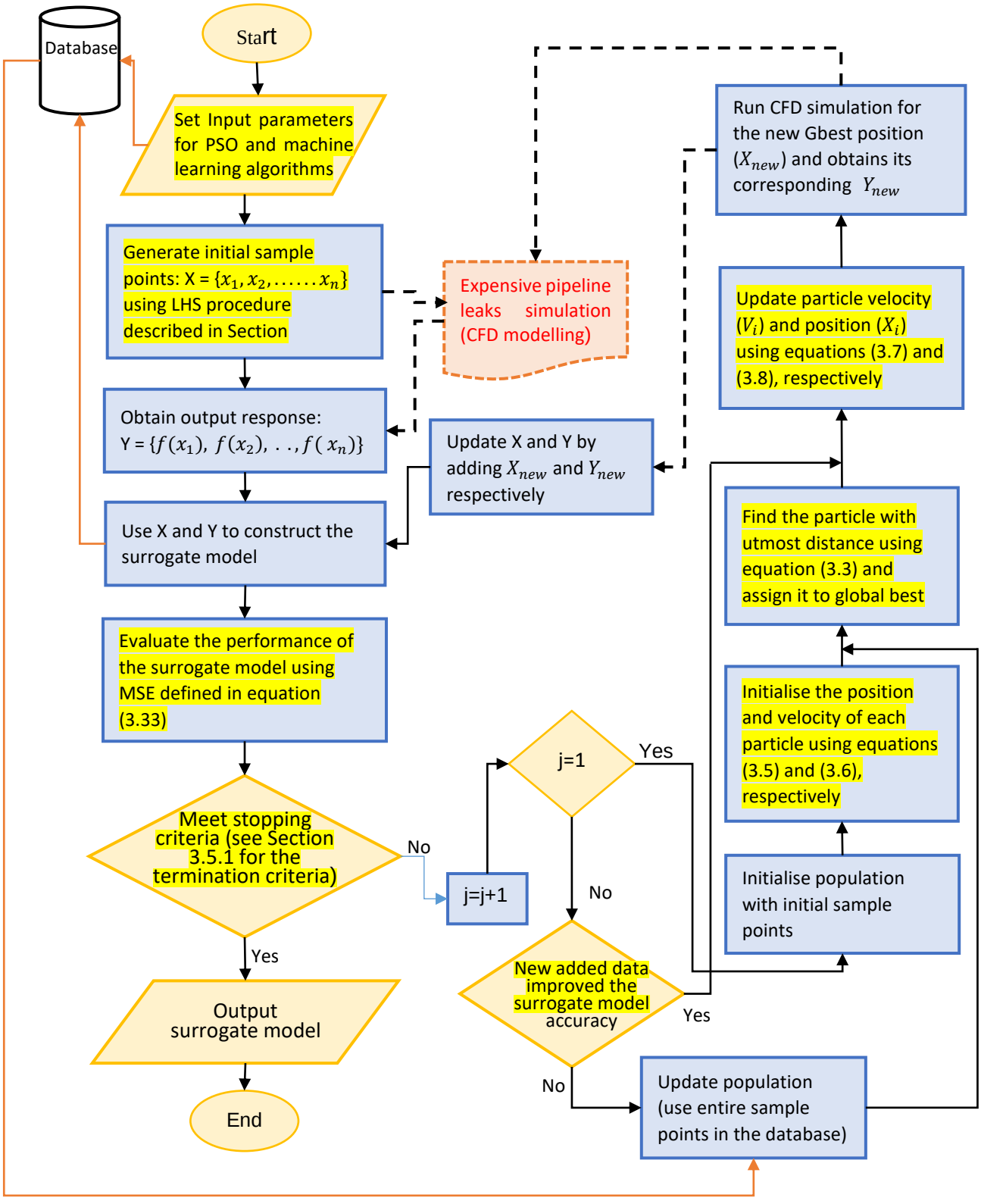


Figure 3-1: The adaptive POS-Assisted surrogate model (PSOASM) framework. The symbol X denotes the training input data, Y denotes the training output data, X_{new} and Y_{new} are the adaptive added input and output data, respectively and j denotes counter or iteration.

One of the crucial features of the surrogate model developed for optimisation of the computationally expensive problem is to ensure it does not overfit during the training process and robustness in the prediction of unseen data. To test this capacity, three phases of tests were performed, as shown in Figure 3.2. The stage 1 test involved PSOASM hyperparameters tuning before arriving at the model parameters presented in Table 3.3. The stage 2 test examines the performance of the developed model on six benchmark problems with different characteristics. Various machine-learning algorithms were also explored with the developed model and compared with conventional sequential sampling models in stage 2. The stage 3 test assesses the model's performance on the data independent of the original dataset used in the development of the surrogate model.

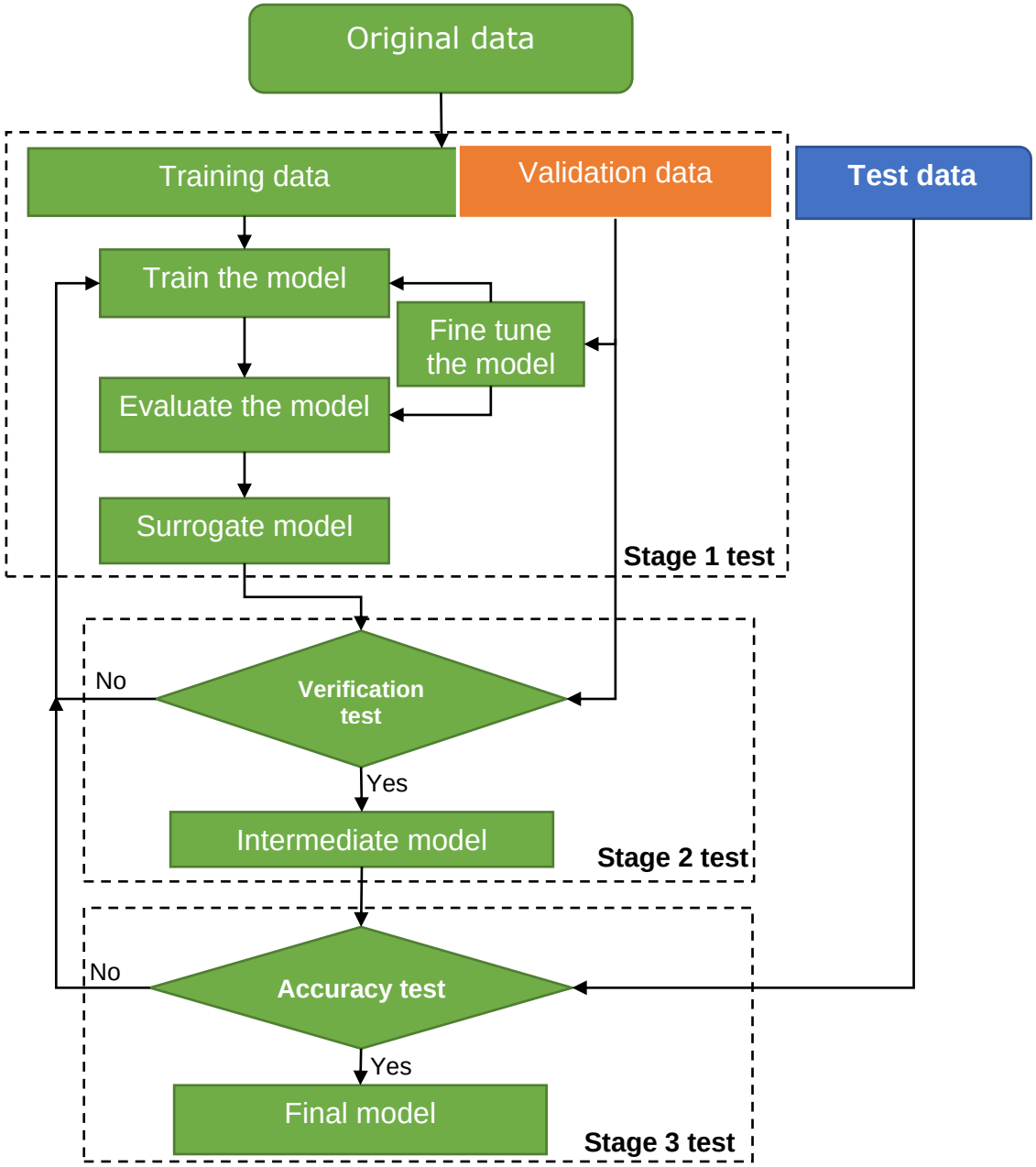


Figure 3-2: The flow chart of the PSOASM testing process

3.3 Problem Formulation

The rationale of this study is to develop a leak prediction model (surrogate model) with a limited number of simulation trials while at the same time attaining maximum accuracy. Given a parameter space $\mathbb{R}^N$ which can be considered as a domain of possible simulation conditions of an n-dimensional random input vector, where N denotes the dimensionality of the parameter space. For example, when $N = 2$, the parameters that characterise pipeline leakage scenarios are leak size and location. Let a vector $X = \{x_1, x_2, \ldots, x_m\}^T$

represents the leak scenarios in the parameter space $x_i \in \mathbb{R}^N, i = 1,2,\ldots.m$, where m is the total number of inputs. A vector $y = \{y_1, y_2, \ldots., y_m\}^T$of the numerical simulation of X that is computationally expensive is denoted as $y = f(x_i); f: \mathbb{R}^N \to \mathbb{R}$. To develop a pipeline leak detection model using a machine learning algorithm, a large data set with thorough data space coverage $y = f(X)$ is required, which is computationally costly. Therefore, an approximated surrogate model is needed to minimise the number of simulation trials to generate the training dataset. The surrogate model$y = g(X)$, which is the approximation of $y = f(X)$ is developed based on the selected input-output data samples. This problem can be formulated as follows:

$$\text{Given function: } f: \mathbb{R}^N \to \mathbb{R}^{\cdot} \tag{3.1}$$

$$\text{find } X \in \mathbb{R}^N s.t. g(X) \leq f(X), \forall X \in \mathbb{R}^N \tag{3.2}$$

For the target function f defined in $f: \mathbb{R}^N \to \mathbb{R}^{\cdot}$, the initial sampling process begins with a set of sample pairs $(X_i, f_i), i = 1,2,3,\ldots\ldots., I$with valid bounds $X^L \leq X \leq X^U$ where $X^L$and X^U represents the lower and upper bounds of parameter space, respectively. The maximum number of sampling points for which the desired accuracy of the developed surrogate model obtain denotes I_{max}; $g(X), i = 1,2,3,\ldots\ldots., I_{max}$ In this study, an approximation model $g(X)$ is implemented using five machine learning algorithms. The key concept of these machine learning models is presented in section 3.6.

3.4 Generation of initial training data samples

The sample size and location of the samples govern the trained surrogate model's computation cost and overall performance. Therefore, the selection of data samples for training is essential in enhancing the surrogate modelling process. A space-filling design is generally employed to design a coarse surrogate model and then refine it until a finer surrogate model is attained. Space-filling designs are designed to spread sample points as evenly as possible over the design parameter domain. In this study, the initial training data points are generated using the LHS method (Golzari et al., 2015). Many researchers have widely used LHS to generate the initial samples for the surrogate model. Its advantages include uniform sample

distribution in the design domain and aid in better estimation of the global accuracy of the surrogate model.

3.4.1 Parameters space sampling concept

The surrogate model act as a prediction model operating on the three basic sets of data: input data or parameters $x_i(t)$, the true response at the observed points $y_i(x)$ (usually the results of a numerical simulation or physical experiment) and the surrogate response $g(X)$. The general assumption about the surrogate model is that its performance depends on the distribution of the training sample set (Garud et al., 2016). Zhang *et al.*(2012) observed that a large crowding distance value of a sample point reflects low sample density (fewer points around that point). The surrogate accuracy is likely relatively lower around the region with fewer points. Therefore, more sample points will probably need in such regions. On the contrary, a small crowding distance value of sample points reflects high sample density (more points around that region), and the accuracy of the surrogate measure is expected to be relatively higher around that region. Therefore, balance is required between the less and highly populated sample regions. The proposed approach is to optimise the sample points by considering a crowding distance factor (also known as population distance of the sample points), which minimises the surrogate training sample size while at the same time achieving maximum accuracy. Based on this concept, the crowding distance of the sample sets and the developed surrogate model's performance (fitness value) are combined in placing the new sample points.

Consider I sample points $X_{C_i}, i = 1,2,\ldots\ldots,I$, the Crowding Distance (CD) of an arbitrary point X_C in the domain of $x_i(t)$ can be measured as:

$$CD\ (X_C) \quad = \sum_{i=1}^{I} \| X_C - X_{C_i} \|^2 \qquad (3.3)$$

3.4.2 Departure function

The departure function is implemented using the fitness value information of the surrogate. The function is employed for local exploitation, aiming to exploit further the region that improves surrogate accuracy. The function is defined as follows:

Let $S_L(X_I)$ represent the surrogate model constructed from I sample points $(i = 1,2,3\ldots\ldots, I)$. Generally, a simulation scenario may generate a group of data patterns for surrogate training. That is, an additional parameter sample point in parameter space will give multiple training patterns from one simulation scenario for surrogate model training. However, in the pipeline leakage considered in this study, only one training data pattern is generated for surrogate model training. Therefore, $S_L(X_{I+1})$ represent the surrogate being built after an additional sample (I +1). The departure function, which evaluates the changes or improvements in the successive iteration, is given as:

$$S_L\Delta_{I+1}^{I}(x) = S_L(X_{I+1}) - S_L(X_I) \tag{3.4}$$

The concept of crowding distance of the existing sample points discussed in section 3.4.1 (equation 3.3) and the fitness value of the surrogate model in this section (equation 3.4) is combined to place a new sample point in an adaptive manner using the PSO technique to optimise the process. In other words, the global exploration is achieved via the crowding distance criteria strategy, and the surrogate model's fitness value benefits the local searching. A good trade-off between exploration and exploitation can be achieved by combining global and local search strategies. The description of sample points placement using the two concepts with PSO to search for a potential optimal size is given in Section 3.5.

3.5 Sample Points Placement Optimisation

The main goal of developing the surrogate model in this study is to predict the location of the new or next sample points in a region of interest. The numerical simulation of the predicted point is then fitted with the machine learning and continues iteratively until when the optimal solution is attained. For the sample points placement optimisation proposed in this study using PSO. Particle swarms comprise of particles, and the position of each particle is described as follows:

$$X_i(t) = [X_{i,1}(t), X_{i,2}(t), \ldots\ldots\ldots, X_{i,j}(t)], X_{i,j}(t) \in \mathbb{R}^N, (i = 1,2,\ldots., Z), (j = 1,2,\ldots., N) \tag{3.5}$$

where X_i is a vector of a current position (particle position), N is the dimension of a particle and Z is the size of a swarm. Each particle has a velocity, which is denoted as:

$$V_i(t) = [V_{i,1}(t), V_{i,2}(t), \ldots\ldots, X_{i,j}(t)] \quad (3.6)$$

During the movement, the best previous position of the particle is recorded as:

$$Pbest_i(t) = [Pbest_{i,1}(t), Pbest_{i,2}(t), \ldots\ldots, Pbest_{i,j}(t)] \quad (3.7)$$

and the best historical position obtained by the swarm is represented as:

$$gbest(t) = [gbest_1(t), gbest_2(t), \ldots\ldots, gbest_{,j}(t)] \quad (3.8)$$

For particles in the search space to represent their feature, each particle in the swarm is iteratively revised based on three attributes: current velocity V_i, current position X_i and previous best position $Pbest_i$. The new velocity of each particle is updated based on the aforementioned attributes as:

$$V_i(t+1) = V_i(t) + \eta_1 r_1(Pbest_i(t) - X_i(t)) + \eta_2 r_2(gbest_i(t) - X_i(t)) \quad (3.9)$$

where t and $t+1$ indicate two successive iterations of the algorithm, $\eta_1$and $\eta_2$are the acceleration constants, r_1 and $r_2$are the random values uniformly distributed in the range [0, 1].

To avoid the swarm divergence due to the velocity exploration, Shi and Eberhart (Shi & Eberhart, 1998) introduce the time-varying inertia weight w_t to improve the convergence of the PSO algorithm. The idea was used to control the particle's momentum by moderating the contribution of the velocity at the previous iteration to the definition of the real particle velocity. Accordingly, the velocity updating equation described in equation (3.9) is revised and updated as follows:

$$V_i(t+1) = wV_i(t) + \eta_1 r_1(Pbest_i(t) - X_{i,d}(t)) + \eta_2 r_2(gbest_i(t) - X_i(t)) \quad (3.10)$$

Diversity plays an important role in improving the effectiveness of evolution in PSO. Nezami *et al.*(2013) observed that the lack of diversity in PSO might be the main reason for the premature convergence of PSO. Generally, in an adaptive sampling method, initial sampling begins with the generation of some initial sample points to fill the entire domain evenly. Consequently, the adaptive surrogate model can better represent the underlying function and guide the subsequent sampling process to find regions that require

more samples. Since the initial sample points, which are the initial positions of the particle, are randomly generated, their distribution may be uneven over the searching space. For instance, good local searching performance is guaranteed if the initial weight is fairly small. However, if the initial weight is large enough, the algorithm will perform a good global searching ability (Nezami et al., 2013). Therefore, to improve the algorithm diversity, the initial weight should be adjusted in accordance with the situation of the particles to balance the trade-off between its global and local searching abilities. Shi and Eberhart (1998) suggested the use of weight w_t between 0.8 and 1.4 ($0.8 < w_t < 1.4$), starting with a large value of weight (a more global search behaviour) that is dynamically reduced (a more local search behaviour) during the optimisation. Borrow idea of Shi and Eberhart, the scheme which dynamically adjusts the values of the weight is introduced as follows:

$$w_t = w_{max} - \left(\frac{\mathrm{W}_{max} - \mathrm{W}_{min}}{iter_{max}}\right) iter_{cur} \tag{3.11}$$

where $iter_{max}$ represent the maximum number of iterations, $iter_{cur}$ is the current iteration numbers; w_{max} is 0.9, and w_{min} is 0.4. During the updating period of the particles, the new position of the ith is determined as:

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{3.12}$$

The personal best of the individual particle is updated as:

$$Pbest_i(t+1) = \begin{cases} Pbest_i(t),. & if \quad f(X_i(t+1)) \geq f(Pbest_i(t)) \\ X_i(t+1),. & otherwise \end{cases} \tag{3.13}$$

where, $f(.)$is the fitness function reflects the quality of the solution. The global best position $gbest$ of the swarm is updated as:

$$gbest(t+1) = arg.\max_{Pbest_i} \quad f(gbest_i(t+1)),. \quad 1 \leq i \leq n \tag{3.14}$$

3.5.1 Algorithm termination criterions

An adaptive surrogate model involves the addition of sample points adaptively to enable a reasonable judgment of the added point. The performance of this kind of model is strongly affected by the number and location of sample points. If the number of sample points is not sufficiently large, the accuracy of the model may be very low. On the other hand, insufficient data points in a region with relatively high uncertainty in the

input domain may result in large prediction uncertainty. Therefore, three different termination criteria are proposed to determine the algorithm termination conditions. The **first** one is surrogate accuracy. The algorithm will be terminated if an error (MSE) is equal to or lower than the specified error value. The **second** one is to monitor the successive relative surrogate improvement. In other words, when the best solution is not improved for several consecutive iterations. In this case, the termination criterion is defined when the surrogate performance (error) is not reduced after N successful iterations (this is important to avoid convergence failure). The N was chosen as five in this study through trial and error. That is, N was chosen as five based on observed convergence performance after multiple analyses were carried out. The **third** termination criterion is when the maximum iteration of PSO N_{max} is attained. The motivation for this work comes from two aspects. The first aspect is to minimise the surrogate training dataset and, at the same time, maximise the model accuracy. The optimal simulation trial (data size) is determined in a manner in which simulation points are chosen systematically in an adaptive and optimised manner, assuring that the sample points are placed in relatively high uncertainty regions in the input domain to determine a promising area for model refinement. Based on the previous description, the pseudo-code of the proposed PSOASM model is summarised in Algorithm 3.1.

Algorithm 3.1. Adaptive PSOASM Algorithm

1: **Database Initialisation:** Generate an initial sample set $X_N = \{x_1, x_2, \ldots\ldots\ldots, x_N\}$using LHS to fill the entire domain evenly

2: Compute expensive CFD simulation modelling to obtain the response $y_.(x_i)$ for the generated sample points, and archive them for surrogate fitting.

3: Construct initial surrogate $S_L(x_i)$ with lower precision using initial simulation data

4: **Set termination criterion:** maximum number of iterations $N\,I$, specified accuracy to be attained, no positive gain in the successful relative improvement.

5: **While** stopping criterion is not reached, **do**

6: **Particles Initialization/updating:** Select initial simulation data to form the initial population $pop(t)$,

7: Set $t = 1$

8: Find the $Xbest$ best position in the swarm

9: **For** each particle i in the swarm, **do**

10: **Behavioural learning**: find the particle's utmost distance to the other particles using crowding distance (CD) defined in equation 3.3.

11 Compute true response for the new sample X_C and obtain the corresponding response Y_{new}

12: **Updating**: update the surrogate model $S_{L+1}(x_i)$ using the new set of data$X_{N+1} = X_C \cup X_N = \{x_1, x_2, \ldots\ldots\ldots, x_{N+1}\}$ and $Y_{N+1} = Y_{new} \cup Y_N$.

13: **Fitness estimation**: Evaluate the fitness of the added particle by computing departure function $S_L\Delta_{I+1}^I(x)$ defined in equation (3.4).

14: **If** $S_L(X_{I+1})$ greater than $S_L(X_I)$

15: **then** Update $X_C Pbest$

16: **end if**

17: **If**$f(X_C Pbest, S_{L+1}(x_i)$ better than $f(X_C Gbest, S_L(x_i)$

18: **then** Update $X_C Gbest$

19: **end if**

20: Apply local sampling search around $X_C Gbest$

21: **end for**

22: $t = t + 1$

23: **End while**

24: Output the final solution

3.6 Model development

This section introduces different machine learning architectures employed to construct the surrogate model. These architectures include support vector machine, multilayer perceptron, decision tree, random forest, and polynomial regression. The key concept of these models is presented in the subsequent section. Most of these architectures depend on

hyperparameters. The values of these parameters are set at the beginning of the training process through systematic testing.

3.6.1 Support vector machine

Support Vector Machines (SVMs) is one of the machine learning algorithms capable of both classification and regression. It was developed based on statistical learning theory by Vapnik V.N (Cheng et al., 2020) and has been widely used in different fields such as function approximation, signal processing and time series prediction. SVMs are attractive to the research community due to their good generalisation performance and ability to handle nonlinear problems using kernel methods. In SVMs, original data is projected into a high-dimensional feature space using the kernel function, subsequently searching for the best prediction function in the feature space. The error is fitted within a margin or threshold through the boundary line. The data points closest to this boundary is called support vectors. Salehi *et al.*(2019) reported that SVM is effective and robust when dealing with noise, insufficient information and uncertainty.

In this study, Support vector regression is employed for the surrogate construction. It has been applied to the construction of a surrogate model in many studies (Cheng et al., 2017, 2020; Jiang et al., 2018). Given a set of data $\{(X, Y), X \in \mathbb{R}^N, Y \in \mathbb{R}\}$, where X denotes the input vector, Y represents the target and $\mathbb{R}^N$ is the input design space. Considering a nonlinear mapping ϕ, X is mapped into a particular space in which a linear function is expressed as (Gholizadeh et al., 2020):

$$y = w^T \phi(X) + c \tag{3.15}$$

where $w$is the weight matrix and c denotes the bias value which is 1 in this study. Given a dataset $(X_1, Y_1), \ldots\ldots, (X_k, Y_k) \in \mathbb{R}^N \times \mathbb{R}$, the function in equation (3.16) can be estimated by exploring a decision function with a small risk (test error). Calculation of risk function is performed using the risk function given as:

$$F = \frac{1}{2}||w||^2 + \frac{D}{K}\sum_{k=1}^{K} |Y_k - g(X_k, w)|_\varepsilon \tag{3.16}$$

where Drepresents the regularisation parameter, ε is a positive number, g is a regularise risk function and K denotes the number of support vectors. The second term in equation (3.16) is expressed as:

$$|Y_k - g(X_k, w)|_\varepsilon = \begin{cases} 0. \quad . \quad . \quad if_. |Y_. - g(X_., w)| < \varepsilon \\ |Y_. - g(X_., w)| - \varepsilon. \quad otherwise \end{cases} \tag{3.17}$$

To transform the function in equation (3.17) to the dual space using the Lagrange multiplier technique, which is the most common technique in the literature (Gholizadeh et al., 2020) employed and expressed as follows:

Maximise

$$Z(\sigma^{(*)}) = -\varepsilon \sum_{k=1}^{K} (\sigma_k^* + \sigma_K) + \sum_{k=1}^{K} (\sigma_k^* - \sigma_K)\, Y_k - \frac{1}{2} \sum_{k=1}^{K} \sum_{i=1}^{K} (\sigma_k^* - \sigma_K) \times_. (\sigma_i^* - \sigma_i) K(X_{k'.} X_i) \tag{3.18}$$

Subject to:

$$\textstyle\sum_{k=1}^{K} (\sigma_k^* - \sigma_K) = \sigma_.^{(*)} \quad . \tag{3.19}$$

where $\pi(X_{k'.} X_i)$ is the kernel function, σ_k^* and σ_K are the coefficients selected at the training stage. Accordingly, the SVR formula is given as (Gholizadeh et al., 2020):

$$g(X) = \textstyle\sum_{k=1}^{K} (\sigma_k^* - \sigma_K)\, K(X_., X') + C \tag{3.20}$$

The kernel function is vital in designing the SVR as the choice of kernel function plays a significant role in SVR response. Different kernel types have been used in the literature. These include linear, Radial Basis Function (RBF), Gaussian RBF, polynomial, etc. Gaussian RBF kernel is employed in this study and defined as:

$$K(X_., X') = exp(-\gamma (X_., X')^2) \tag{3.21}$$

where $\gamma$is a constant number selected through the systematic test.

3.6.2 Multi-layer perceptron

Multilayer Perceptron (MLP) is the most popular and classical type of artificial neural network design to find a functional dependency between variables (Gholizadeh et al., 2020). It is capable of modelling complex functions, good at dealing with noise and can adapt its weight. The structure of the MLP is based on three layers: input layer (input variables), one or more hidden layers, and an output layer (target variable). Figure 3.3 depicts the schematic of the MLP model used in this study. The number of neurons

in the input and output layers corresponds to the dimensions of the input and output variables data, respectively. Each neurone in the hidden layer is connected to every neurone in the input and output layer via weights (w). The weights are regulated using the backpropagation technique during the training process. The sum of the weighted inputs in each neuron and bias (B) is then passed through an activation function that produces an output value. A single hidden layer with a range between 8 and 20 neurons is used in this study to construct the surrogate model.

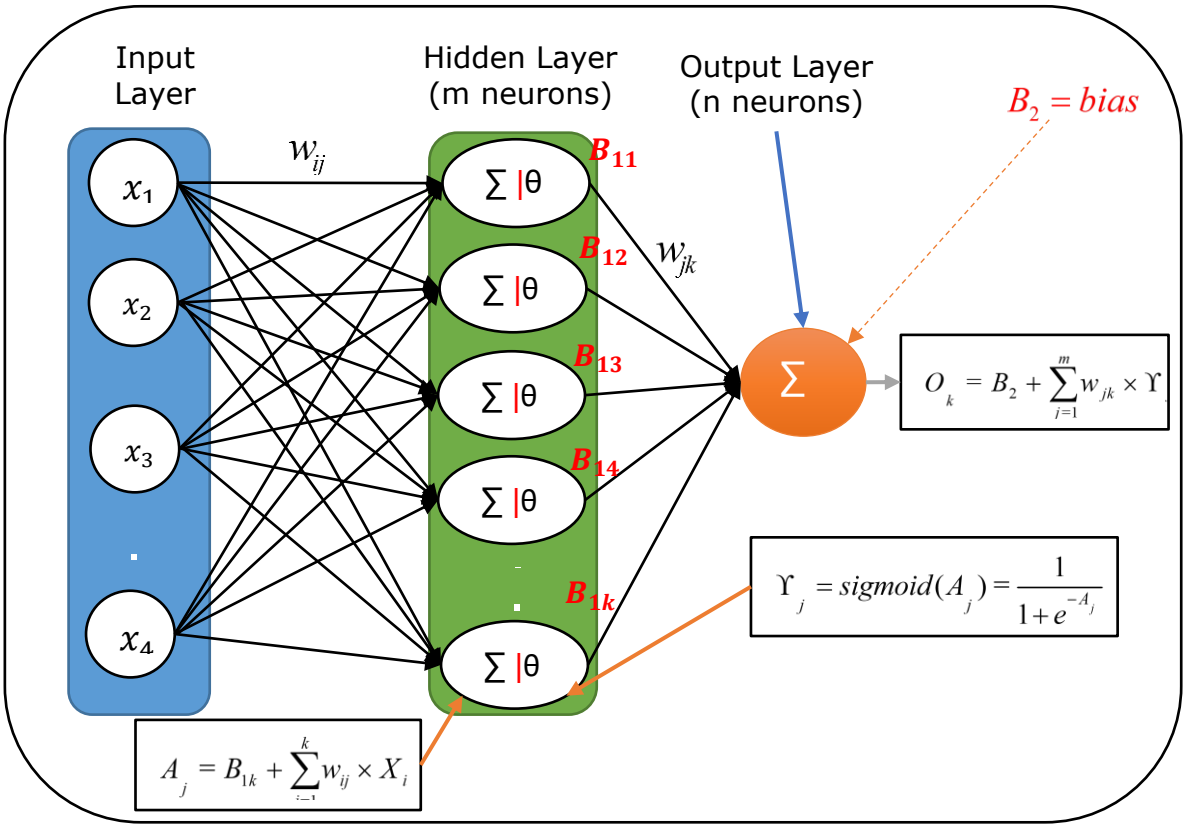


Figure 3-3: A systematic of the MLP model

Given the inputs (X_i), biases and weights, the MLPs are computed through the following steps:

The weighted sums of the inputs are computed by:

$$A_j = B_{1k} + \sum_{i=1}^{k}.w_{ij} \times X_i \qquad (3.22)$$

where X_i is the input to the input node(s), w_{ij}denotes connection weight from the i_{th}node in the input layer to the j_{th} node in the hidden layer, k is

the total number of the nodes in the input layer and B_{1k} is the bias term of each hidden neuron. The output of each hidden node is computed as:

$$\Upsilon_j = sigmoid(A_j) = \frac{1}{1+e^{-A_j}} \quad (3.23)$$

Finally, the output of the MLPs is calculated from the hidden nodes as follows:

$$O_k = B_2 + \sum_{j=1}^{m}.w_{jk} \times \Upsilon_j \quad (3.24)$$

$$f(O_k) = sigmoid(O_k) = \frac{1}{1+e^{-O_k.}} \quad (3.25)$$

where $B_2$is the bias term associated with the output neuron, w_{jk} denotes the connection weight from thej_{th} hidden node to the k_{th}output node, and m is the number of hidden nodes. The activation function adopted for this study is logarithmic sigmoid (logsig). It is selected among the various activation functions available for the ANN because the model output values are in the ranges of [0,1].

3.6.3 Decision tree

Decision Trees (DT) is a decision support tools that use a tree-like model to make predictions. DT comprises arbitrary numbers of leaf nodes and branch nodes. The leaf node represents a decision, while the branch node represents a choice between several alternatives. These nodes are determined based on the given instances. DT initiates an algorithm that partitions the given dataset into different splitting points for each variable and calculates the error between the true and predicted values at each split point. Different types of decision learning algorithms have been developed to build the decision tree, which includes Iterative Dichotomiser 3 (ID3), CHi-squared Automatic Interaction Detector (CHAID), and Classification and Regression Tree (CART). In this study, CART is employed to build a decision tree. CART operates on the principle of recursive partitioning, aiming to increase the similarity in successive daughter nodes—hence CART is more efficient and reliable for building decision tree regression (Pathak et al., 2018; Pekel, 2020). As with other repressors, the decision tree takes $X = \{x_{1,}x_{2,}\ldots\ldots,x_k\}^T$as input variables and $Y = y_{1,}y_{2,}\ldots\ldots,y_j$ as a target variable, where k and j are the total number of predictor variables and observation,

respectively. Mathematically, the structure of decision tree regression can be expressed as follows:

Let$\gamma = (vf, th_t)$ denotes candidate split where $vf$is a feature variable, t is a node, and th represents the threshold value (MSE). The decision tree on the left side (Q_L) and decision tree on the right side found by splitting the data (predictor variables) into candidate split (γ) can be expressed as (Pekel, 2020):

$$Q_L(\gamma) = (x, y)|x_{vf} \leq th_t \tag{3.26}$$

$$Q_r(\gamma) = (x, y)|x_{vf} > th_t \tag{3.27}$$

The sum of squared error for the tree (T) is calculated as:

$$S = \sum_{c \hat{I}(T)} \sum_{j \hat{I} c} (y_j - m_c)^2 \tag{3.28}$$

where $c$is the number of a sample at the current node, $m_c$is the calculated mean predicted value at terminal nodes and expressed as:

$$m_c = \frac{1}{c} \sum_{j \in (c)} y_j \tag{3.29}$$

The procedures for the constructing decision tree regression is described in (Pathak et al., 2018; Pekel, 2020)

3.6.4 Random forest

Random Forest (RF) is an effective learning model employing an ensemble of decision trees for classification or regression tasks. It was developed by Breiman (2001) and is widely used for data prediction and interpretation purposes. RF model is developed using a combination of decision trees, where each tree is generated using the random bootstrap samples of the input datasets. Unlike a decision tree where the individual node is split using the best split among all the samples, in the RF model, each node is split using the best among a subset of predictors randomly selected. This strategy allows RF to perform very well when dealing with high-dimensional data, complex interaction and correlation, and robust against overfitting (Pourghasemi & Kerle, 2016). As the tree grows on a bootstrap sample, the error rate for observations left out of the bootstrap sample is monitored using an out-of-bag (OOB) error rate. The OOB basically indicates the accuracy of the RF predictor (generalisation error).

As reported in (Rahmati et al., 2016), the OOB is an unbiased estimate of the generalisation error in RF. Some of the advantages of OOB include low bias and low variance due to averaging over a large number of trees, higher prediction performance and no overfitting (Rahmati et al., 2016). To run the RF model, two sets of parameters are necessary to be defined: the number of trees to be built in the forest to run (n_{tree}) and the number of variables or samples to be in each tree-building process (m_{try}). Likewise, these parameters should be optimised in order to minimise the generalisation error. However, it was reported by Gholizadeh *et al.*(2020) that there is no specific rule to determine these parameters. Although Breiman (2001) and Liaw and Wiener (2002) stated that a variable ($m_{try} = 1$) could generate good accuracy, Grömping (2009) reported that there is a need to use at least two variables samples (i.e. $m_{try} = 2,3,4,\ldots\ldots,m$) to avoid a weaker prediction model. In this study, these parameters were determined using the internal optimiser RF function (TuneRF) that recognises the optimal number of samples in each tree-building process (Gholizadeh et al., 2020; Rahmati et al., 2016; Taalab et al., 2018). In order to check the performance of the surrogate model as the number of simulation trials increases, the MSE of the OOB is computed for the additional data point and calculated as follows:

$$MSE_{OOB} = \frac{1}{n}\sum_{i=1}^{n}(\mu_i - \hat{\mu}_i)^2 \quad (3.30)$$

where n is the number of data points, and $\hat{\mu}$ donates the mean predicted data points and μ_i is the observed data points.

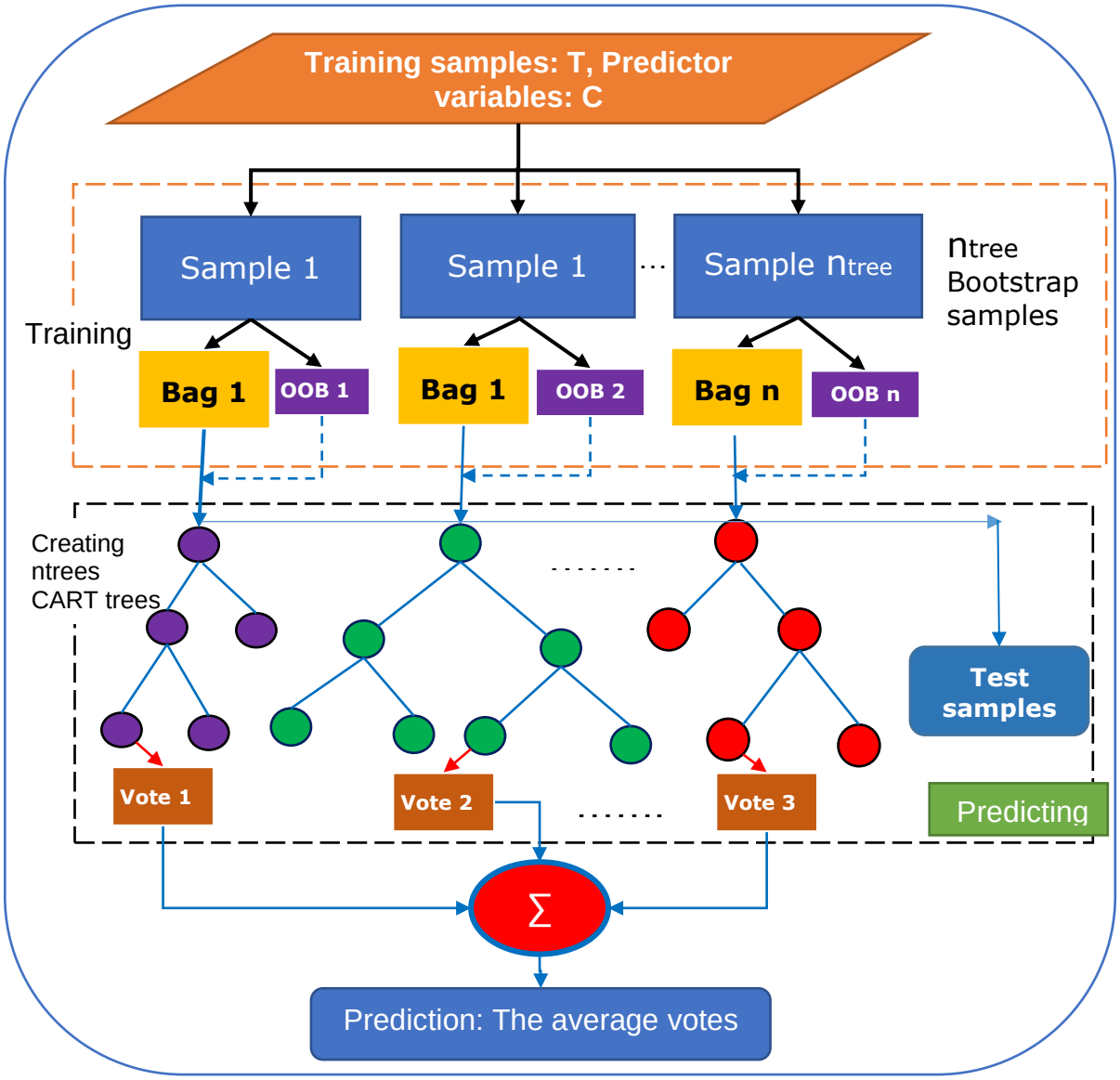


Figure 3-4: A systematic of the Random forest model

3.6.5 Polynomial regression

Polynomial regression (PR) is a special case of multiple regression where the relationship between the independent variable x and the dependent variable Y is modelled as an nth order of the polynomial. Polynomial regression is generally used when linear regression fails to describe the best results. This algorithm provides the best approximation of the relationship between the independent and dependent variables, and a broad range of functions can be fitted. The equation described polynomial regression model with nthorder can be expressed as:

$$Y_X = \beta_0 + \beta_1 X + \beta_2 X^2 + \ \ldots\ldots\ldots + \beta_n X^n \ + \ e_X \qquad (3.31)$$

Where βs βs are unknown parameters determined by the least squares method, Y_X is the observed variable at a point X. The e_X is the error term, which also follows the probability distribution of Y_X and n is the order of the polynomial.

3.7 Model performance evaluation

Considerable prediction uncertainty may likely occur in a surrogate model built with a limited number of sample points if the sample locations are not appropriately selected (Jiang et al., 2020). Applying such an imprecise surrogate model in design and optimisation may lead to misleading predictions. Therefore, it is essential to verify the overall assessment of the constructed surrogate model before using it for prediction or uncertainty qualification. The overall assessment is defined here in terms of the generalisation ability of the surrogate model. That is its ability to predict well over unknown data points (external data). The performance of the surrogate model is evaluated by applying the following standard performance metrics to test data points and overall performance:

3.7.1 Mean Square Error

Mean Square Error (MSE) is the average squared difference between estimated and actual values. MSE is the mean of the overall squared prediction errors and can be expressed as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \widehat{Y}_i)^2 \qquad (3.32)$$

where N is the number of data points, Y_i is the observed values, and $\hat{Y}$is the predicted data values. The more MSE closer to the zero means, the better the estimator can perfectly predict the response of a parameter. That is, the smaller the MSE is, the better the quality of the surrogate model.

3.7.2 Root mean square error

Root Mean Square Error (RMSE) is the standard deviation of the residuals. It can be regarded as the average vertical distance of the actual observation from the fit line (Jiang et al., 2020). The formula for computing the RMSE is given as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \widehat{Y}_i)^2} \quad (3.33)$$

Mathematically, the RMSE is the square root of the MSE. Therefore, it is also a global error metric. Its value ranges from 0 to 1, meaning the smaller RMSE value represents a higher surrogate accuracy.

3.7.3 R-squared

R-squared R^2 is the proportion of the variation in the independent variable that is predictable from the independent variable. In other word is a systemic error measure that reveals how good the best fit line is from the baseline model. R^2 is a percentage number whose value range from 0 to 1, with 0 signifying that surrogate model prediction does not improve over the mean model, and 1 signifying perfect prediction value. The formula for computing R^2 is defined as (Jia et al., 2020):

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(Y_i - \widehat{Y}_i)^2}{\sum_{i=1}^{N}(Y_i - \bar{Y}_i)^2} \quad (3.34)$$

where Y_i is the observed data values, $\widehat{Y}_i$represents the predicted data value, $\bar{Y}_i$ denotes the mean of the observed data values and Nstands for the number of testing data.

3.7.4 Mean absolute error

Mean Absolute Error (MAE) is the mean value of the error at all verified data points with the same weight assigned to all the errors. It measures the average magnitude of the errors in a set of predictions without considering their direction. MAE is usually regarded as the true error of the surrogate model when enough verification points are considered (Jiang et al., 2020). The MAE can be expressed as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|Y_i - \widehat{Y}_i\right| \qquad (i = 1, 2, ., ., ., N) \quad (3.35)$$

3.8 Numerical Test

In order to evaluate the effectiveness of the developed PSOASM for optimising the sample size for an expensive simulation or experiment problems, various numerical studies were conducted using six benchmark functions with different characteristics. The benchmark functions considered include Rosenbrock, Ackley, Rastrigin, Beale, Goldstein, and

Schaffer functions. These functions are widely used in the literature (Kaveh & Dadras, 2017; J. Zhang et al., 2012) to assess new algorithms developed for optimisation problems. The description of these benchmark problems is presented in equations (3.36) to (3.41); their three-dimensional diagrams are also exemplified in Appendix 1. In this study, several factors that may influence the evaluation results are considered, such as the initial sample size, total number of samples and generalisation of the developed algorithm in different machine learning algorithms. The model was tested on external data to verify the possibility of overfitting during the training process. Finally, the performance of the developed model (PSOASM) is compared with the several well-known state-of-the-art sampling methods, which include the Halton sequence, Hammersley sequence, Sobol, and Latin hypercube sampling (LHS). The PSOASM has shown better performance than all these sampling algorithms in all the analyses carried out.

All the algorithms constructed in this study and the analysis performed are programmed using Python (Python 3.4.10). Python is an open-source programming language developed by Guido Van Rossum (Severance, 2015). It is commonly used in scientific computing and chosen in this study because it allows the user to benefit from modern API releases in areas like machine learning, quantum computing and optimisation with up-to-date community support for libraries and the only open sources language available script ANSYS space claim. All the computation run on an Intel(R) Xeon(R) Gold 6230 CPU @ 2.10 GHz processor, 16 Cores, 64.0 GB RAM in operating Windows 10.

Function 1 (F1): Rosenbrock function

$$f(x,y) = (1-x)^2 + 10(y-x^2)^2 \qquad (3.36)$$

$x \in [-1.5,2.0], y \in [-1.5,3.0]$

Function 2 (F2): Ackley function

$$f(x,y) = -20\,exp\left[-0.2\sqrt{0.5(x^2+y^2)}\right] - exp[0.5(cos\,2\pi x + cos\,2\pi y)] + exp(1) + 20 \qquad (3.37)$$

$x \in [-5.0,5.0], y \in [-5.0,5.0]$

Function 3 (F3): Rastrigin function

$$f(x,y) = 20 + x^2 - 10\cos(2\pi x) + y^2 - 10\cos(2\pi y)$$
$$x \in [-5.12, 5.12], y \in [-5.12, 5.12] \quad (3.38)$$

Function 4 (F4): Beale function

$$f(x,y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$$
$$x \in [-4.5, 4.5], y \in [-4.5, 4.5] \quad (3.39)$$

Function 5 (F5): Goldstein-Price function

$$f(x,y) = [1 + (x + y + 1)^2(19 - 14x + 3x^2 - 14y + 6xy + 3y^2)]$$
$$[30 + (2x - 3y)^2(18 - 32x + 12x^2 + 48y - 36xy + 17y^2)] \quad (3.40)$$
$$x \in [-2.0, 2.0], y \in [-2.0, 2.0]$$

Function 6 (F6): Schaffer function N.2

$$f(x,y) = 0.5 + \frac{sin^2(x^2 - y^2) - 0.5}{[1 + 0.001(x^2 + y^2)]^2} \quad (3.41)$$
$$x \in [-100, 100], y \in [-100, 100]$$

3.8.1 Parameter setting

The following parameter values were generally found to produce accurate function estimation through numerical experiments. The PSO acceleration constants η_1 and η_2 are selected as 1.2 and 1.5, respectively, which also fall into the range of values commonly used in the literature (F. Li, Shen, et al., 2020; Nickabadi et al., 2011). The maximum inertia weight (w_{max}) is set to 0.9, and the minimum inertia weight (w_{min}) is set to 0.4. The initial sample size is 10, which is also used as swarm size and its effect on the developed surrogate model is analysed in section (3.8.2). The additional samples (N) generated via the surrogate model is allowed to be determined by the algorithm through various analysis. The summary of the adjustable parameters used for the machine learning algorithms is presented in Table 3.1. The other parameter settings of the machine-learning model are used as the default values of the python program.

Table 3-1: Adjusted parameters used for the machine-learning algorithms

Method	Adjusted parameters and values
SVM	Kernel function: Gaussian RBF
MLP	Input layer: number of input variables, one hidden layer: 8 to 20 neurons, momentum=0.094, learning rate =0.017.
Decision tree	Minimum sample leaf: 1, minimum sample split: 2
PR	Number of degrees: 3 to 5
Random forest	Minimum sample leaf: 1, minimum sample split: 2

3.8.2 Influence of initial sample size

An adaptive surrogate model generally begins with the initial samples that are recommended to fill the design space evenly. The influence of this sample size on the performance of the developed surrogate model was investigated. The decision about the initial sample size is undefined in the literature. Some studies reported that small initial samples could lead to surrogate model focus at inappropriate locations, which could mislead the first steps of the adaptive procedure (Fuhg et al., 2020; H. Liu et al., 2018). On the other hand, a large initial sample size can cause high computational costs. Most of the computational budget may be spent on space-filling samples, which could be effectively used up on the adaptively added samples. In this regard, some surrogate modelling literature suggested few empirical formulas or guidelines for determining the initial sample points. Table 3.2 illustrates different empirical formulas suggested to determine the proper initial sample sizes through an extensive literature survey presented in (Liu *et al.*2018), where n is the dimension of the parameter space, which is two in this study.

Table 3-2: Empirical formulas to determine the initial sample size (H. Liu et al., 2018)

Authors	Initial sample size
Regis and Shoemaker (2007)	N = 2(n+1)
Busby *et al.*(2007)	N = (n + 2)(n + 1)/2 +10
Jones *et al.*(1998); Loeppky *et al.*(2009)	N = 10n
Xu *et al.*(2014)	N = 5n
(Gutmann, 2001)	N = n^2

The illustration of different initial sample sizes using empirical formulas in Table 3.2 and samples adaptively added for the Ackley test function is shown in Figure 3.5. A similar figure for the Rosenbrock function is shown in Figure 3.6. The initial samples are indicated in red dots, while adaptive added samples are designated square blue. It is important to highlight that sampling locations for each benchmark function may be different despite using the same sampling strategy. The reason is that sampling locations depend on the surrogate model's prediction values. Figure 3.7 illustrates the convergence profiles for different initial sample sizes for the Ackley and Rosenbrock functions. The figures show the prediction performance of the PSOASM on different initial sample sizes as the number of added sample points increases. The lower the fitness value (MSE), the better the surrogate model performance. The surrogate performance shows that the higher the initial sample size, the higher the accuracy of the surrogate model at the beginning. However, the difference between the accuracy of the 10, 16, and 20 initial sample sizes is not significant after additional 20 sample points. Therefore, ten initial sample sizes is chosen for the surrogate model developed in this study as it saves considerable computational costs and performs similarly to both 16 and 20 initial sample sizes.

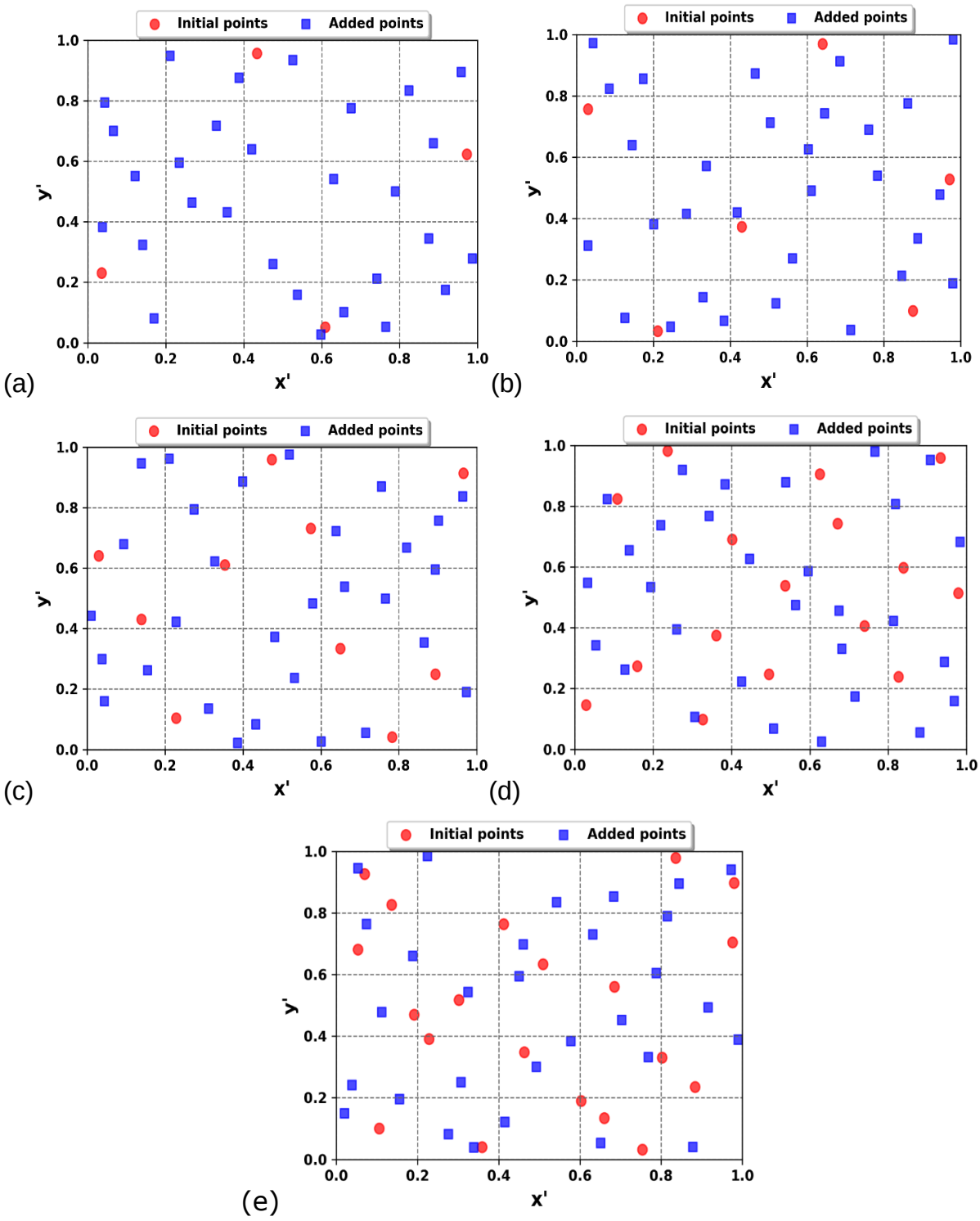


Figure 3-5: Illustration of different initial samples sizes and adaptively added samples for Ackley function: (a) 4 initial samples, 30 adaptive added samples, (b) 6 initial samples, 30 adaptive added samples, (c) 10 initial samples, 30 adaptive added samples, (d) 16 initial samples, 30 adaptive added samples, (e) 20 initial samples, 30 adaptive added samples (initial samples in red dots, adaptive added samples in square blue).

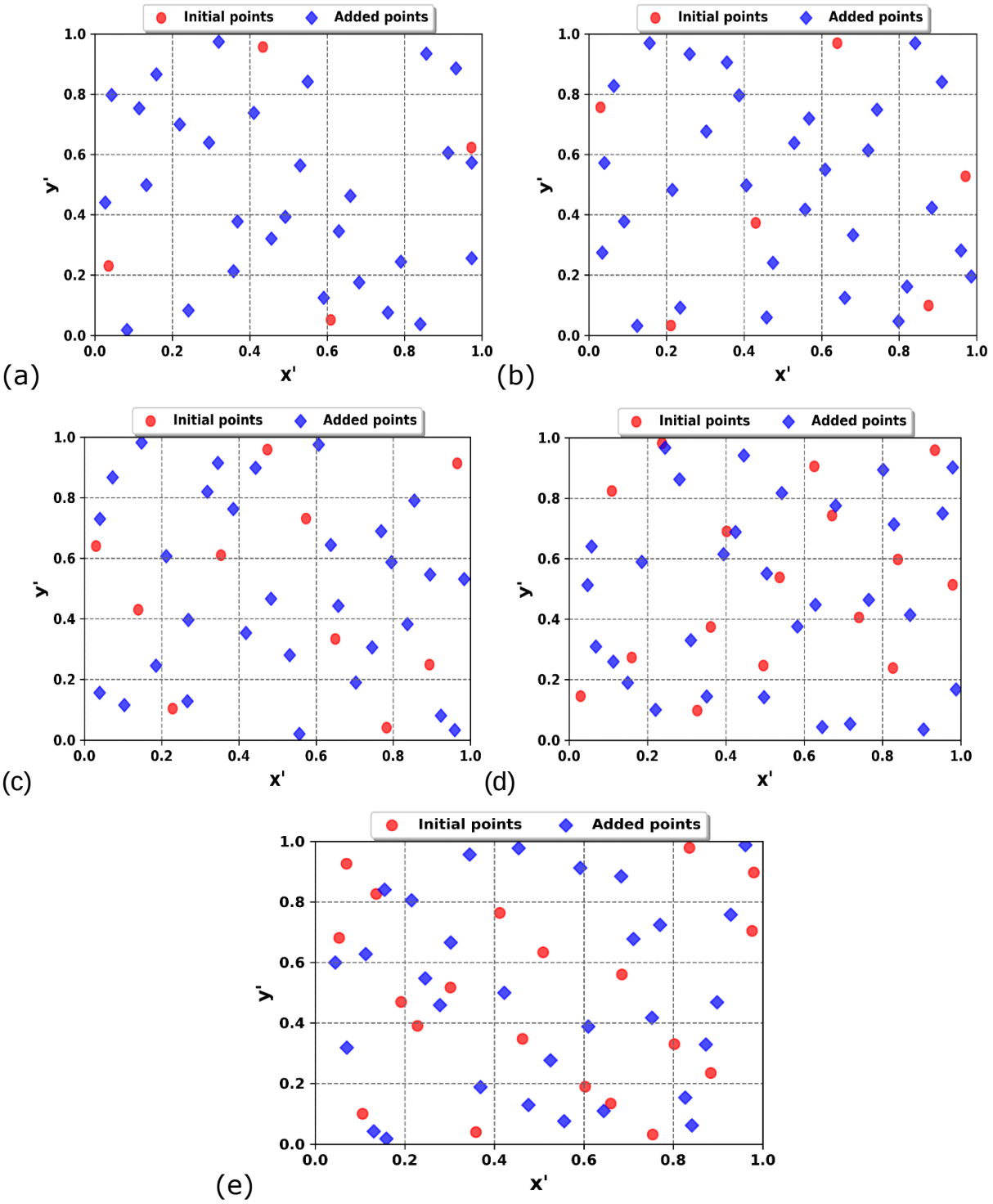
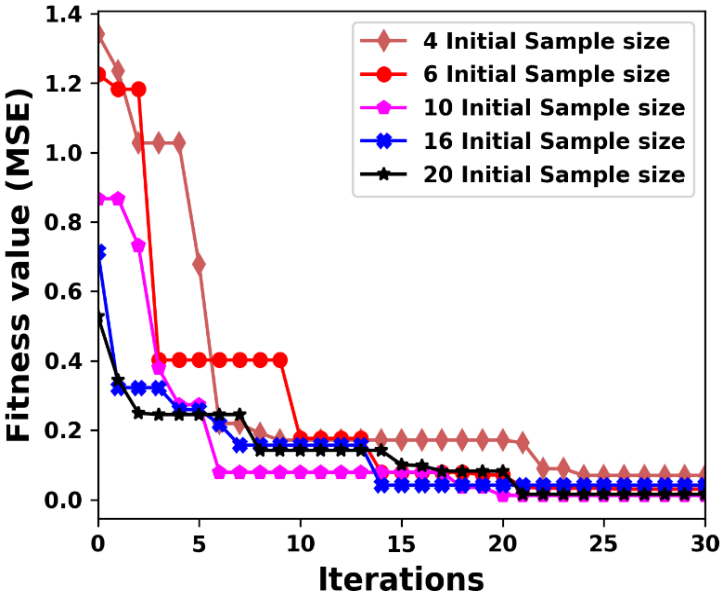
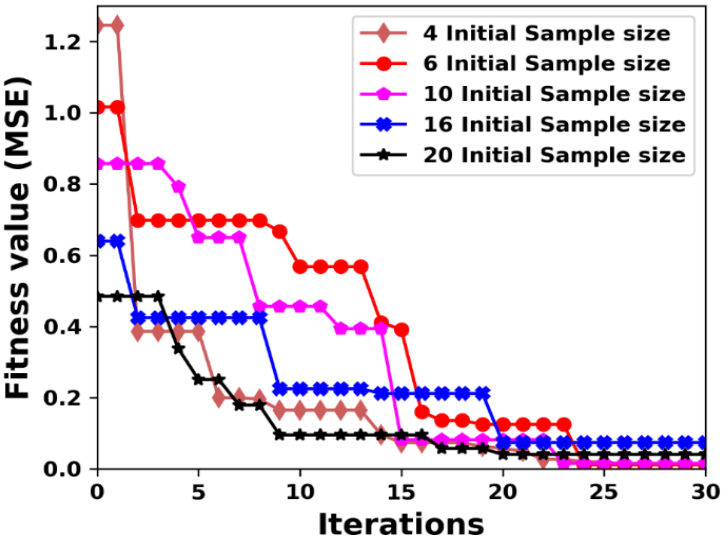


Figure 3-6: Illustration of different initial samples sizes and adaptively added samples for Rosenbrock function: (a) 4 initial samples, 30 adaptive added samples, (b) 6 initial samples, 30 adaptive added samples, (c) 10 initial samples, 30 adaptive added samples, (d) 16 initial samples, 30 adaptive added samples, (e) 20 initial samples, 30 adaptive added samples (initial samples in red dots, adaptive added samples in rhombus blue).



(a)



(b)

Figure 3-7: Convergence profiles of the different initial sample sizes: (a) Ackley test function, (b) Rosenbrock test function.

The impact of chosen initial sample size (10 samples) as the number of additional samples increases is shown in Figure 3.8. The number of added samples varies from 0 to 200. It was observed that as the added sample point increases, the prediction error of the surrogate model decreases sequentially up to 30 iterations. After the 30 iterations, adding more sample points contributes little to the surrogate model's accuracy. The prediction quality of the PSOASM in terms of optimal convergence at 30 iterations was further investigated using five benchmark problems. Figure 3.9 shows the

progression of estimated error for different test benchmark functions. From the convergence profiles presented in Figure 3.9, one can see that PSOASM shows no significant improvement after 25 iterations for all the cases tested.

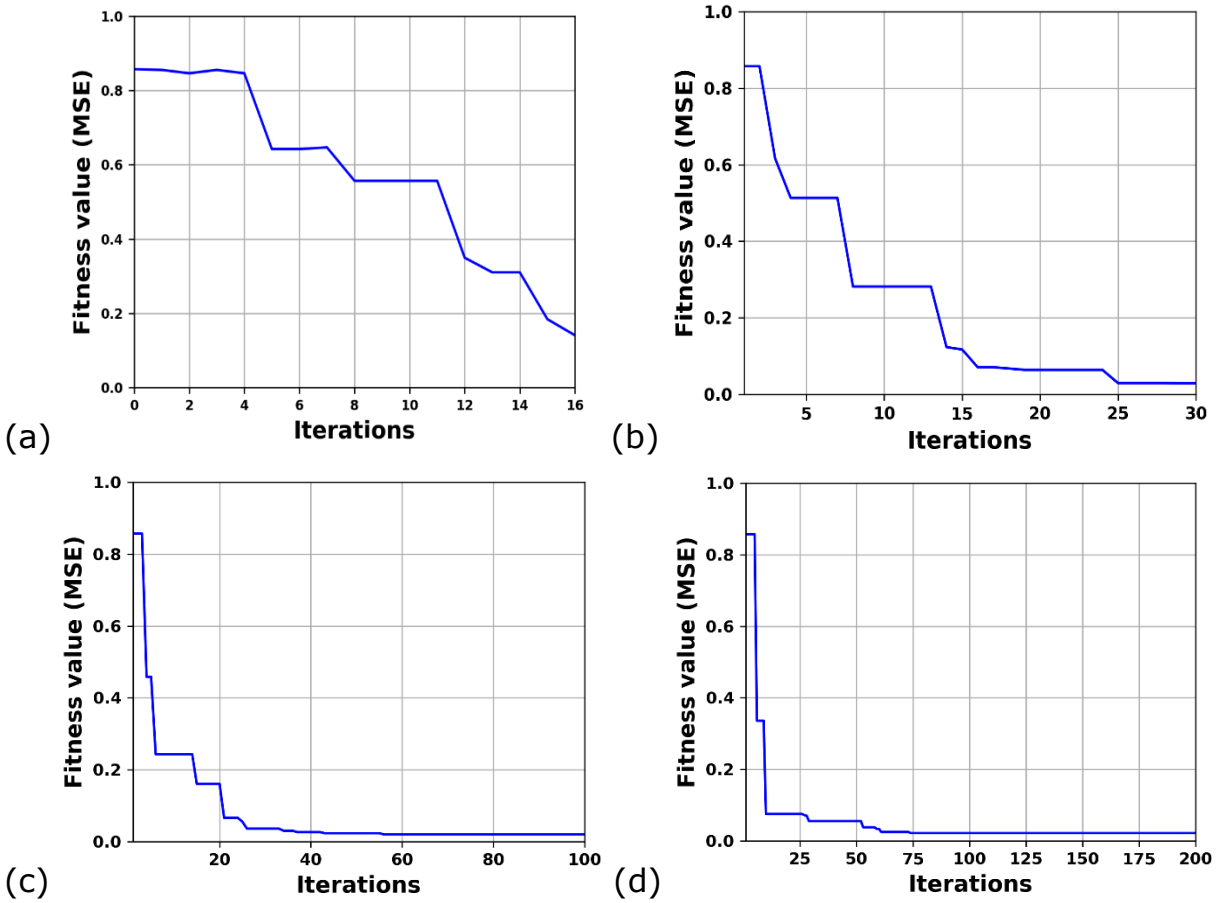


Figure 3-8: Convergence profiles of 10 initial sample sizes under different sample sizes (iterations indicate sample sizes).

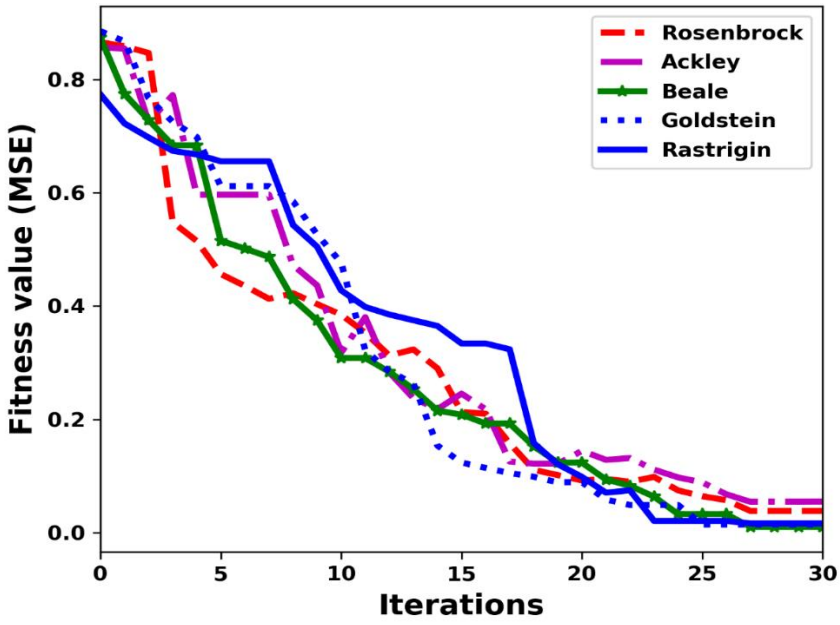
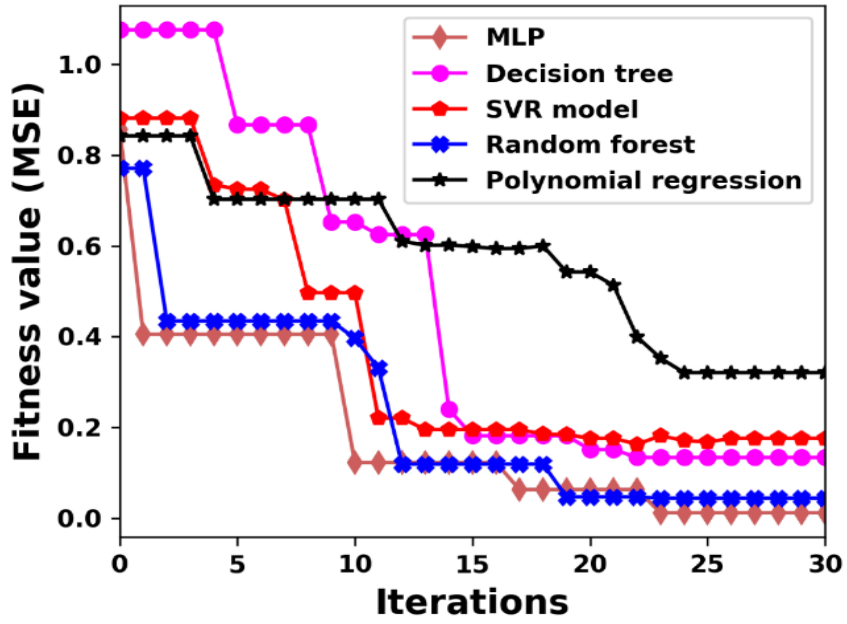


Figure 3-9: Convergence profiles of the PSOASM as a function of 30 samples for different benchmark problems.

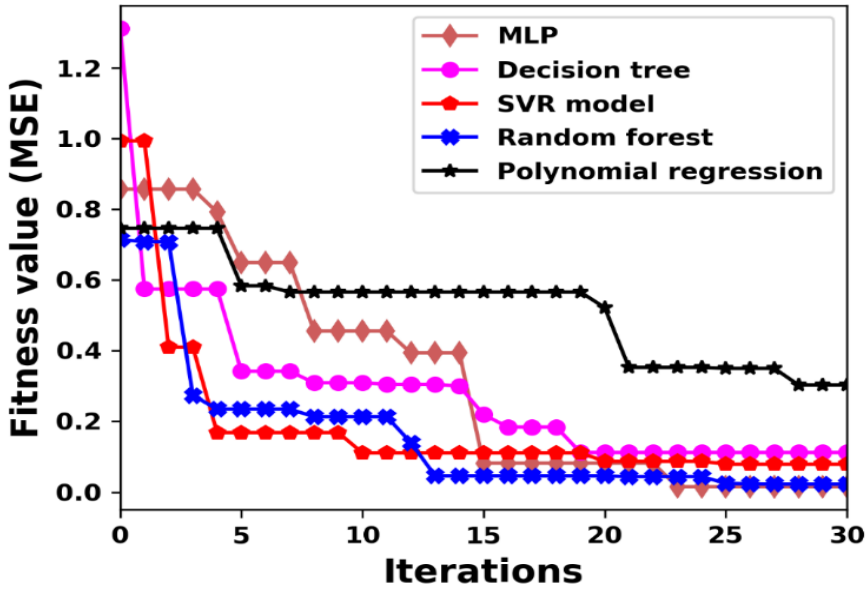
3.8.3 Performance comparison of different machine learning methods on the developed PSOASM

To further examine the performance of the proposed surrogate model on different machine learning algorithms, five machine-learning algorithms presented in Section 3.6 were used for the evaluations. After initial training with the 10 data points, the developed surrogate model is employed to adaptively select sample points in which computation is performed for a new sample added to the training dataset. The process is repeated until maximum prediction accuracy is achieved. Figure 3.10 shows the obtained prediction profiles for the machine learning method using Rosenbrock and Ackley test functions. The PSOASM performs well in all machine learning algorithms except the polynomial regression. MLP and random forest results for the two cases tested (Rosenbrock and Ackley functions) demonstrate good performance, while decision tree and SVR have mixed results for the two cases. The error of polynomial regression is far higher than the other machine learning models as samples are added. This might be attributed to underfitting due to one or two outliers in the training dataset, which can seriously affect the performance of polynomial regression. That means

there is a room for significant optimisation with the choice of polynomial degree for good bias or variance tradeoff. MLP outperform all the machine learning methods compared in all the cases tested. Therefore, MLP is used for the rest of this study.



(a)



(b)

Figure 3-10: Performance evaluation of the PSOASM on different machine learning models: (a) Rosenbrock test function, (b) Ackley test function.

3.8.4 Comparison of the PSOASM with conventional sequential sampling algorithms

The performance of the PSOASM is compared with the other conventional sequential sampling schemes, namely Halton sequence, Latin hypercube sampling (LHS), Hammersley sequence, and Sobol sequence. The same number of training points used for the developed surrogate model was employed for the conventional sampling approaches and tested on six different benchmark functions. Figure 3.11 illustrates the Performance comparison of PSOASM against other sequence sampling algorithms as their sample size grows. As can be seen in Figure 3.11, most of the tested sampling schemes had performance comparable to or slightly different from PSOASM for the initial training data points. However, as the iteration increases, the proposed method's accuracy rapidly increases more than the conventional sequential sampling methods. Overall, PSOASM outperforms the conventional sampling methods by providing lower prediction error than conventional sampling methods employed for comparisons. The error obtained with the LHS and Halton were lower than both Harmersley and Sobol sequences in all the problems tested except the Rastrigin function.

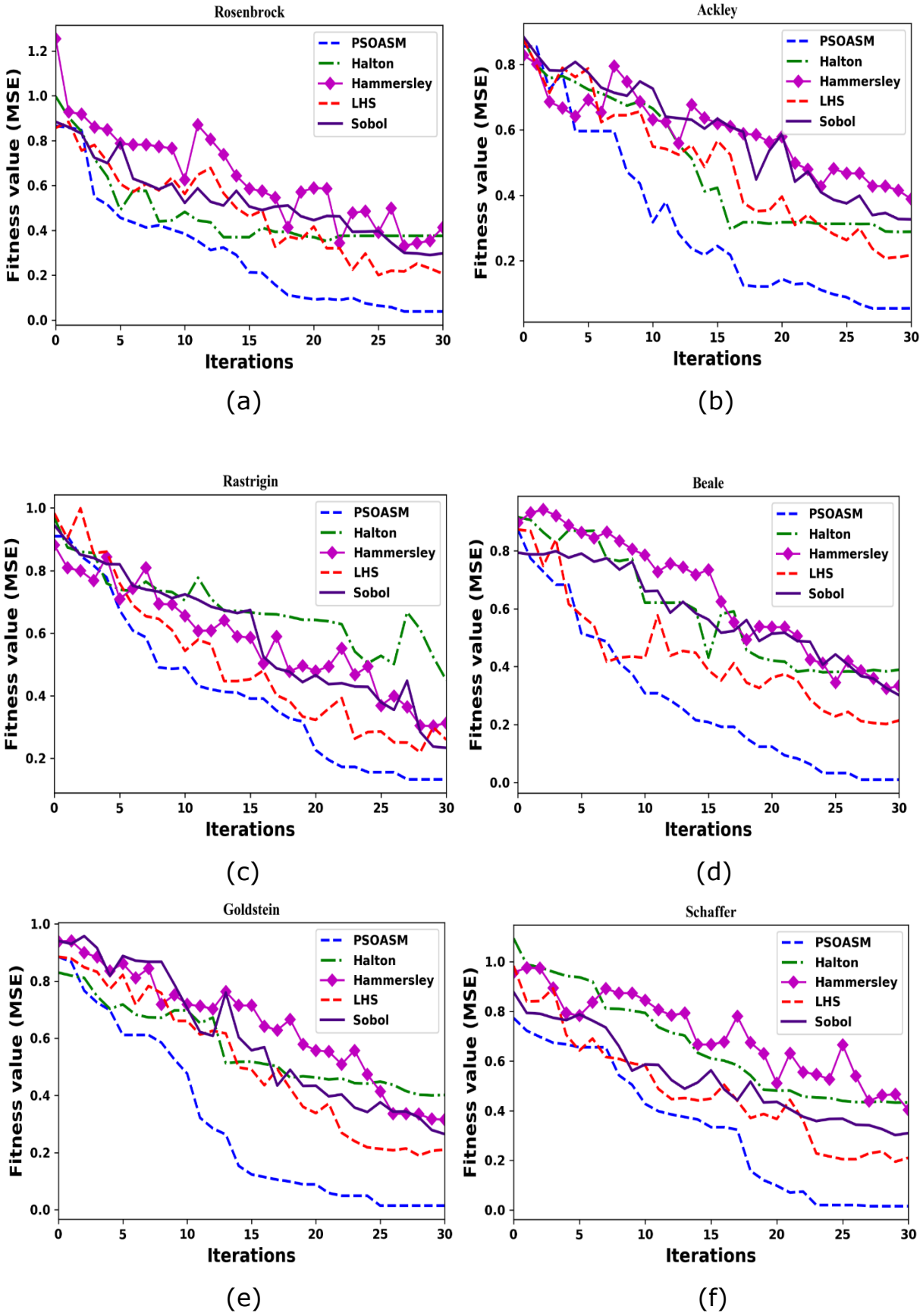


Figure 3-11: Performance comparison of PSOASM against conventional sequential sampling algorithms: (a) Rosenbrock, (b) Ackley, (c) Rastrigin, (d) Beale, (e) Goldstein, and (f) Schaffer functions

3.8.5 Prediction quality assessment of PSOASM on the new dataset

The effectiveness of the PSOASM model on different machine algorithms has been observed consistently adequate and proved to outperform several sequential sampling algorithms for different benchmark problems. To further scrutinise the robustness of the PSOASM model, more assessment is performed to test if its predictive accuracy is robust beyond the data it was developed in. New sets of datasets, including 16, 25, 36, 64, 100, and 144 samples, were generated in one stage in a space-filling manner. Appendix B presents the sample sets of new data employed for the testing. Figure 3.12 and Figure 3.13 exemplify the comparison of ground truth and predicted values for Rosenbrock and Ackley functions, respectively. It can be observed that the figures demonstrate good matches between the ground truth and the predicted values. It is important to highlight that computational time is not considered in this thesis because PSOASM iteration takes lesser computational time (within 1 to 10 seconds) compared to hours or weeks for a CFD simulation trial.

Four standard metrics of model prediction error, namely MSE, RMSE, R^2 and MAE discussed in Section 3.7 are computed to examine how the PSOASM prediction accuracy differs as the size of new test data increases. The summary of the estimated errors as the number of training data points grown for different new data employed for testing is presented in Table 3.3. The least error for each trained sample is highlighted in bold. The errors are presented in the form of mean values of ten repetitions tests performed for each case. For all the test cases performed, the worst results are obtained when the training sample size is ten but improve as the training samples increasing. PSOASM outperforms the conventional sampling methods by providing lower MSE, RMSE, MAE and higher R^2errors in all the cases tested.

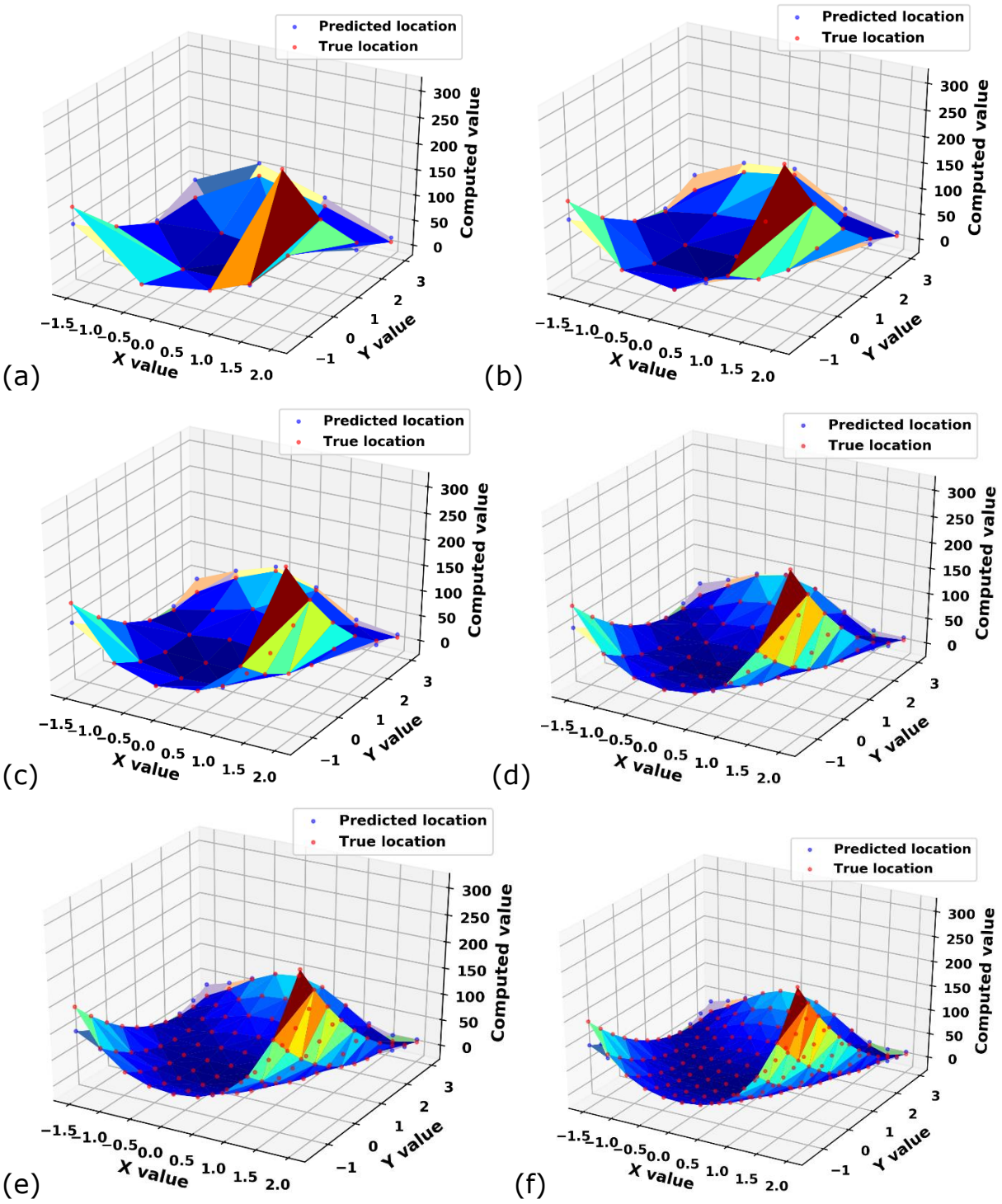


Figure 3-12: Overview of PSOASM model response surface versus ground truth for Rosenbrock function: (a) 16 samples, (b) 25 samples, (c) 36 samples, (d) 64 samples, (e) 100 samples, (f) 144 samples (true values in red-purple colour, predicted value in blue colour).

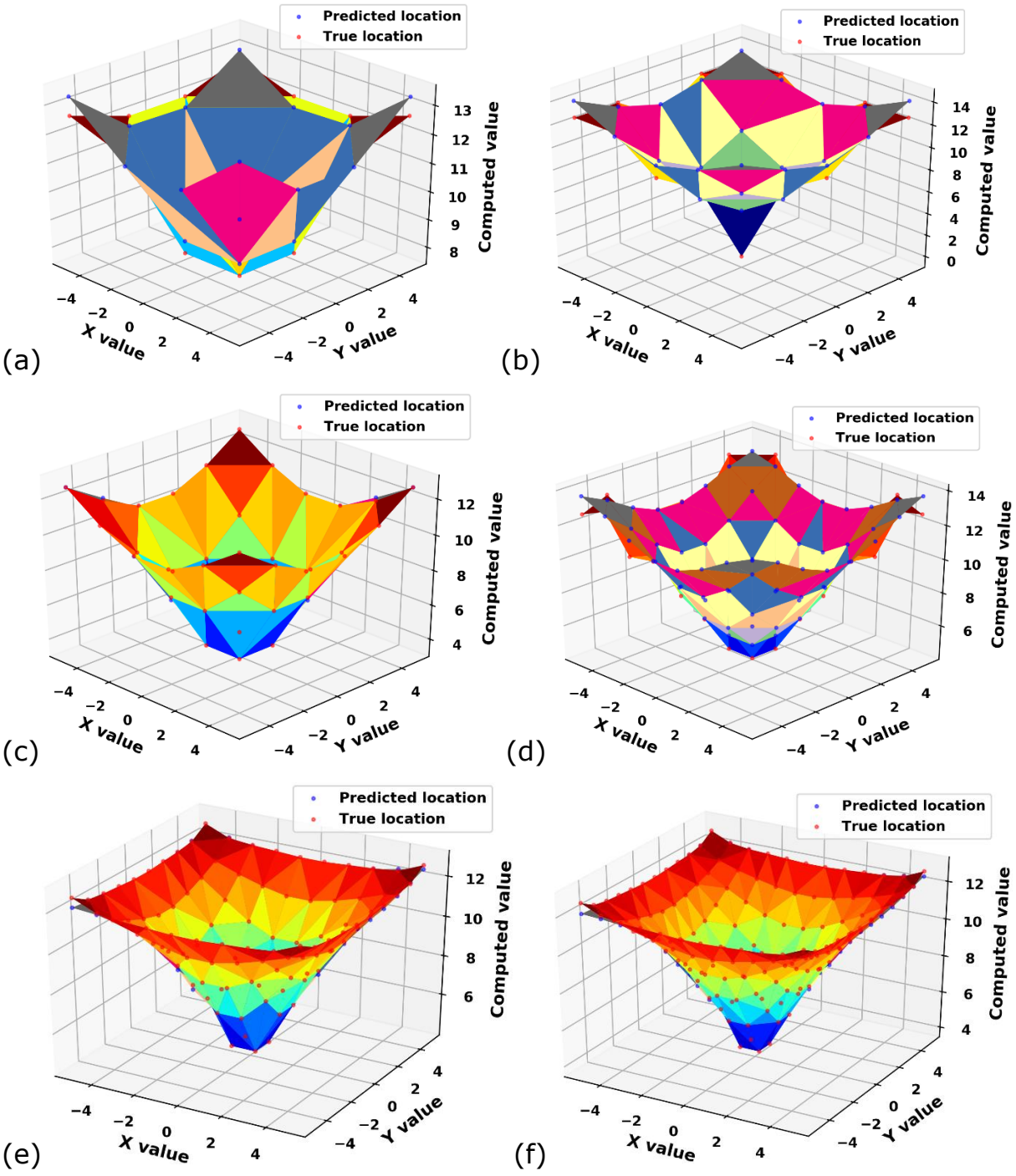


Figure 3-13: Overview of PSOASM model response surface versus ground truth for Ackley function (a) 16 samples, (b) 25 samples, (c) 36 samples, (d) 64 samples, (e) 100 samples, (f) 144 samples (true values in red-purple colour, predicted value I blue colour).

Table 3-3: Test results of PSOASM on different new data sizes (16, 25, 36, 64, 81, 144, and 2500) as a function of different training samples

Training sample size	Testing sample sizes	Rosenbrock function				Ackley function			
		MSE	RMSE	R^2	MAE	MSE	RMSE	R^2	MAE
20	16	**0.447**	**0.668**	**0.552**	**0.320**	**0.364**	**0.604**	**0.635**	**0.383**
	25	0.456	0.675	0.543	0.316	0.443	0.665	0.556	0.385
	36	0.463	0.681	0.536	0.306	0.348	0.589	0.651	0.427
	64	0.472	0.687	0.527	0.317	0.367	0.606	0.632	0.423
	81	0.475	0.689	0.524	0.320	0.385	0.620	0.614	0.409
	144	0.479	0.692	0.520	0.326	0.352	0.623	0.647	0.406
	2500	0.485	0.697	0.517	0.329	0.341	0.629	0.648	0.401
25	16	**0.233**	**0.483**	**0.766**	**0.284**	**0.199**	**0.446**	**0.800**	**0.388**
	25	0.244	0.494	0.755	0.264	0.290	0.538	0.709	0.333
	36	0.252	0.502	0.747	0.248	0.160	0.400	0.839	0.318
	64	0.261	0.511	0.738	0.241	0.193	0.439	0.806	0.370
	81	0.264	0.514	0.735	0.239	0.213	0.462	0.786	0.311
	144	0.268	0.518	0.731	0.239	0.195	0.442	0.804	0.304
	2500	0.269	0.522	0.728	0.242	0.197	0.445	0.813	0.300
30	16	**0.219**	**0.468**	**0.780**	**0.277**	**0.122**	**0.349**	**0.877**	**0.276**
	25	0.221	0.470	0.778	0.252	0.217	0.466	0.782	0.280
	36	0.224	0.474	0.775	0.235	0.084	0.291	0.915	0.209
	64	0.228	0.478	0.771	0.213	0.112	0.336	0.887	0.262
	81	0.229	0.479	0.770	0.210	0.131	0.363	0.868	0.184
	144	0.231	0.481	0.768	0.203	0.115	0.339	0.884	0.237
	2500	0.235	0.487	0.752	0.201	0.112	0.344	0.871	0.241
35	16	**0.119**	**0.345**	**0.880**	**0.228**	**0.062**	**0.250**	**0.937**	**0.226**
	25	**0.113**	**0.337**	**0.886**	**0.204**	0.195	0.442	0.804	0.325
	36	**0.113**	**0.337**	**0.886**	**0.191**	0.021	0.145	0.978	0.106
	64	0.115	0.339	0.884	0.168	0.061	0.247	0.938	0.209
	81	0.116	0.340	0.883	0.163	0.089	0.299	0.910	0.190
	144	0.117	0.342	0.882	0.156	0.057	0.238	0.942	0.170
	2500	0.119	0.344	0.881	0.157	0.052	0.242	0.944	0.173

40	16	**0.104**	**0.323**	**0.895**	**0.233**	**0.051**	**0.227**	**0.948**	**0.218**
	25	0.102	0.320	0.897	0.213	0.032	0.181	0.967	0.150
	36	0.104	0.322	0.895	0.201	0.033	0.183	0.966	0.147
	64	0.107	0.327	0.892	0.180	0.061	0.247	0.938	0.213
	81	0.108	0.328	0.891	0.176	0.077	0.278	0.922	0.150
	144	0.109	0.331	0.890	0.166	0.071	0.267	0.928	0.198
	2500	0.106	0.329	0.893	0.167	0.067	0.269	0.931	0.201

3.9 Summary

This chapter proposed a novel surrogate model named adaptive particle swarm optimisation assisted surrogate model (PSOASM), which trained the machine learning models with a limited dataset and achieved good accuracy. The method present in this study is unique, as it combined two criteria: fitness value of the surrogate model and the population density of the sample points to select the candidate solution for fitness evaluations. Various machine learning methods were explored with the proposed algorithm. Specifically, five different machine learning algorithms and four sampling schemes were compared, considering six benchmark problems of various characteristics chosen for evaluation. The result of the approach proposed in this chapter through extensive evaluations supports the use of a surrogate model to bypass the large number of datasets needed to train machine-learning algorithms. The developed surrogate model performed well in all the machine learning algorithms employed for evaluation and can predict up to 98% accuracy with an average of 40 data points in all the benchmark functions tested. The possibility of integrating the developed surrogate model into engineering problems are explored. The application of the surrogate model proposed in this chapter to a 3-D pipeline leakage detection and characterisation is presented in Chapter 6.

Chapter 4 Numerical Modelling of Pipeline Leakage

4.1 Introduction

Computational fluid dynamics (CFD) is a valuable tool that can improve understanding of fluid flow in a pipeline and the consequences of the leak in various ranges under different conditions. With CFD, one can solve a set of partial differential equations that represent fluid systems over the region of interest and determine how various fluid properties, such as density, pressure, velocity, etc., vary. These mathematical models include equations representing the principle of mass, momentum and energy conservations. Depending on the problems at hand, additional equations that describe other physical phenomena can also be included to model the transport of given properties, such as turbulent kinetic energy k and its dissipation rate ε using the standard $k-\varepsilon$ model (Chinello et al., 2019) when solving turbulence model problems.

CFD provides the opportunity for engineers and designers to evaluate the design's values easily. It is useful to lessen the need to build and evaluate prototypes at the early stage of the design process. Besides, the level of detail that can be obtained in the CFD model is tremendous. A lot of results can be generated in a simulation without additional simulation costs. These enable more complicated and complete studies requiring physical experiments to be performed using CFD modelling. CFD has been widely used to study pipeline leakage, and it has been shown to be successful in detecting and characterising pipeline leakage. However, the available literature is mostly focused on single-phase flow systems.

This chapter presents a computational model for pipeline leakage detection as a multiphase flow system. It also explains the approaches taken for the numerical simulation of pipeline leakage. Specifically, Reynolds-averaged Navier-Stokes equations were solved using the VOF approach (Chinello et al., 2019). The simulation results were validated against the numerical simulation by Chinello *et al.* (2019) and experimental data reported in (Espedal, 1998). In particular, the comparison was made between mono-

phase and stratified flow behaviours induced by leaks and validated with the physical experimental data reported by Molina-Espinosa *et al.*(2013).

4.2 Two-phase flow modelling

A two-phase flow involves gas and liquid phase flows simultaneously within the pipeline. As the mixture of these fluid phases flows along a pipeline, separation between the phases occurs and eventually forms a specific flow configuration known as flow regimes or patterns such as bubble, slug, annular or stratified (Raimondi, 2019). Among these flow regimes, stratified gas-liquid is reported as one of the common flow regimes typically encountered in the multiphase pipeline (Chinello et al., 2019). Various conditions govern the establishment of these regimes, but superficial flow velocity stands above them all. The increases or decreases in superficial gas-liquid velocity can bring about the transition from one regime to another.

Several studies have been performed on stratified two-phase flow modelling using Reynolds-averaged Navier-Stokes equations. Some of these studies are based on the VOF method, where a single set of momentum equations is solved and these equations are shared between the two phases by means of volume-averaged properties in each cell (Holmås et al., 2005). On the other hand, the second method is the two-fluids formulation model, where the mass and momentum equations are solved separately for each of the phases. Most of the reviewed literature have employed the VOF method as it provides an appropriate method for separated flows such as stratified and annular flows (Chinello et al., 2019; Lo & Tomasello, 2010; Terzuoli et al., 2008). The two-fluids approach, on the contrary, is more suitable for dispersed flow, such as gas bubbles carried by a liquid flow or liquid droplets carried by a gas flow. Holmås et al.(2005) performed stratified two-phase flow simulation using the Volume of Fluid (VOF) method. The $k-\varepsilon$ renormalization group (RNG) and $k-\omega$ Shear Stress Transport (SST) models were employed to solve behaviours of turbulences. Lo and Tomasello (2010) simulated stratified gas-liquid flow in a three-dimensional pipe using commercial Star-CD code. The VOF method was employed, and three different turbulence models, including the standard $k-\varepsilon$, $k-\omega$, and

$k-\omega$ SST, were investigated. The obtained liquid holdup and pressure drop were compared with the experimental data reported in Espedal (1998). The results revealed that liquid holdup was underestimated. At the same time, the pressure drop was overestimated for the three aforementioned models when the turbulence viscosity was not artificially reduced near the interface. Recent work by Chinello *et al.*(2019) conducted extensive studies on stratified gas-liquid flow with ANSYS Fluent 17.1 using the VOF method. The authors solved Reynolds-averaged Navier-Stokes equations with the VOF approach and evaluated the performance of the $k-\omega$ SST turbulence model with and without damping of the turbulence at the gas-liquid interface. The simulated pressure drop and liquid holdup were compared against the experimental data reported in Espedal (1998) for stratified air-water flow in a pipe. The obtained results show that proper damping of the turbulence close to the interface is required to obtain good agreement with the experimental pressure drop and liquid holdup.

4.3 Computational Model

In order to describe multiphase flow modelling, it is required to solve the flow governing equations together with the turbulence model. In this context, the flow governing equations and turbulence model for gas-liquid simulation are presented in this section.

4.3.1 Governing Equation

The VOF method and $k-\omega$ SST turbulence models are applied for modelling stratified gas-liquid flow in a pipeline. The flow is assumed to be incompressible, isothermal and adiabatic. The VOF method, which is a one-fluid approach, comprises the continuity and momentum equations which are given in Equations (4.1) and (4.2), respectively (Chinello et al., 2019):

$$\frac{\partial \rho}{\partial t} + \nabla \bullet (\rho \vec{\boldsymbol{v}}) = 0 \tag{4.1}$$

$$\frac{\partial}{\partial t}(\rho \vec{\boldsymbol{v}}) + \nabla \bullet (\rho \vec{\boldsymbol{v}}\, \vec{\boldsymbol{v}}) = -\nabla p + \nabla \bullet (\overline{\tau} + \overline{\tau}_t) + \rho \vec{g} + \overline{F} \tag{4.2}$$

where ρ is the density of the mixing fluids, $k\,g/m^3$; t is time, s; $\vec{\boldsymbol{v}}$ is velocity vector after Reynolds averaging, m/s; p is static pressure, Pa; $\vec{\boldsymbol{g}}$ is gravity force, m/s^2; $\overline{F}$ is a source term accounting for the effect of surface tension.

The molecular stress tensor $\overline{\tau}$ is given as (Chinello et al., 2019; X. Li, Chen, Zhang, et al., 2019):

$$\overline{\tau} = \mu\left[(\nabla\vec{\boldsymbol{v}} + \nabla\vec{\boldsymbol{v}}^{T}) - \frac{2}{3}\nabla \bullet \vec{\boldsymbol{v}}I\right] \tag{4.3}$$

where $\vec{\boldsymbol{v}}^{\boldsymbol{T}}$ is the transpose of the velocity vector, m/s .The turbulent stress tensor for the Reynolds stress $\overline{\tau}_t$ defined with the Boussinesq eddy viscosity approximation is defined as (Chinello et al., 2019):

$$\overline{\tau}_t = \mu_t\left[(\nabla\vec{\boldsymbol{v}} + \nabla\vec{\boldsymbol{v}}^{T}) - \frac{2}{3}(\nabla \bullet \vec{\boldsymbol{v}} + \rho k)I\right] \tag{4.4}$$

where I is unit tensor, $\vec{\boldsymbol{v}}^{T}$ is the transpose of the velocity vector, m/s. The surface tension force, $\overline{\boldsymbol{F}}$ is modelled using the Continuum Surface Force (CSF) method (Chinello et al., 2019). The VOF model concept is applied to treat the two-phase gas-liquid as one single mixture in accordance with the previous studies by Salem and Tomaso (2018) and Chinello *et al.*(2019). The density (ρ) and viscosity (μ) are volume fraction weighted mixture quantities:

$$\rho = \alpha_1\rho_1 + \alpha_2\rho_2 \tag{4.5}$$

$$\mu = \alpha_1\mu_1 + \alpha_2\mu_2 \tag{4.6}$$

where α_1 and α_2 are the volume fractions of the primary and secondary phases, respectively.

$$\alpha_1 + \alpha_2 = 1 \tag{4.7}$$

The volumetric transport equation for the secondary phase is determined using the following equation:

$$\frac{\partial\alpha_2}{\partial t} + \vec{\boldsymbol{v}} \bullet \nabla\alpha_2 = 0 \tag{4.8}$$

4.3.2 Turbulence modelling

Selection of an appropriate turbulence model is highly crucial in two-phase gas-liquid modelling. Chinello *et al.*(2019) compared numerical simulations with the physical experiment data conducted by Espedal (1998), which revealed that the $k-\omega$ SST model yields better results than both $k-\omega$ and $k-\varepsilon$ models for the air-water flow simulation if turbulence is properly damped at the gas-liquid interface. Therefore, the $k-\omega$ SST model is employed in this study, and its constitutive equations are defined as follows: The turbulence viscosity is given as (Chinello et al., 2019):

$$\mu_t = \frac{\rho k}{\omega} \frac{1}{max\left[\frac{1}{\alpha^*}, \frac{SF_1}{a_1 \omega}\right]} \tag{4.9}$$

where k is turbulent kinetic energy, $\mathrm{J/kg}$; ω is specific dissipation rate, S is the strain rate magnitude and is defined as:

$$S = \sqrt{2S_{ij}S_{ij}} \tag{4.10}$$

$$S_{ij} = \left(\frac{1}{2}\right)\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}\right) \tag{4.11}$$

where S_{ij} is the average strain rate, V_i and V_j are the velocity components in x_i and x_j axis, respectively. The transport equation for the turbulent kinetic energy; k and the specific dissipation rate ω is defined as:

$$\frac{D\rho k}{Dt} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + min\left(\mu_t S^2, 10\rho\beta^* k\omega\right) - \rho\beta^* k\omega \tag{4.12}$$

$$\begin{aligned}\frac{D\rho\omega}{Dt} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\omega}\right)\frac{\partial \omega}{\partial x_j}\right] + \frac{\alpha}{V_t} min\left(\mu_t S^2, 10\rho\beta^* k\omega\right) - \rho\beta\omega^2 \\ + 2(1-F_2)\rho\frac{1}{\sigma_\omega, 2\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} + S_\omega\end{aligned} \tag{4.13}$$

where S_ω is the additional source term, β is turbulence model constant. The blending functions F_1 and F_2 are defined as follows:

$$F_1 = tanh\left[max\left(\frac{2\sqrt{k}}{0.09\omega y}, \frac{500\mu}{\rho y^2 \omega}\right)\right]^2 \tag{4.14}$$

$$F_2 = tanh\left\{min\left[max\left(\frac{\sqrt{k}}{0.09\omega y}, \frac{500\mu}{\rho y^2 \omega}\right), \frac{4\rho k}{\sigma_{\omega,2} D_\omega^+ y^2}\right]\right\}^4 \quad (4.15)$$

where y is the distance to the closest wall surface, D_ω^+ is the dimensionless specific dissipation rate. The model constants are selected according to the $k-\omega$ SST model of Chinello *et al.*(2019).

4.4 Solution Procedure and Validations

The procedure for the numerical simulation of the CFD model used in this study is presented in Figure 4.1. These steps include the creation of geometry, mesh generation, boundary condition definitions, numerical method and code validation. For the results presented in this study, the pipeline inlet is treated as the reference location, and all distances are measured relative to the pipeline inlet.

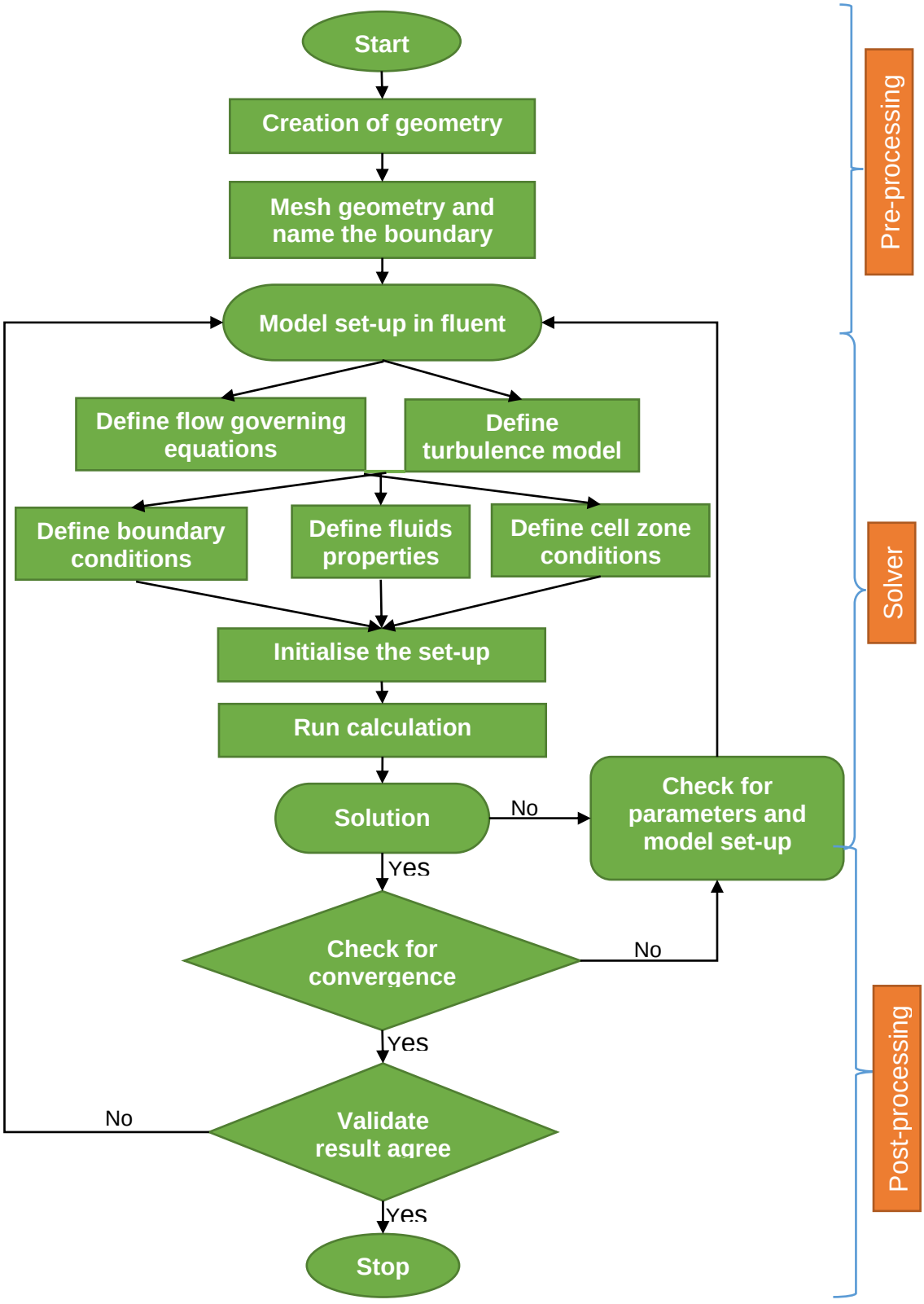


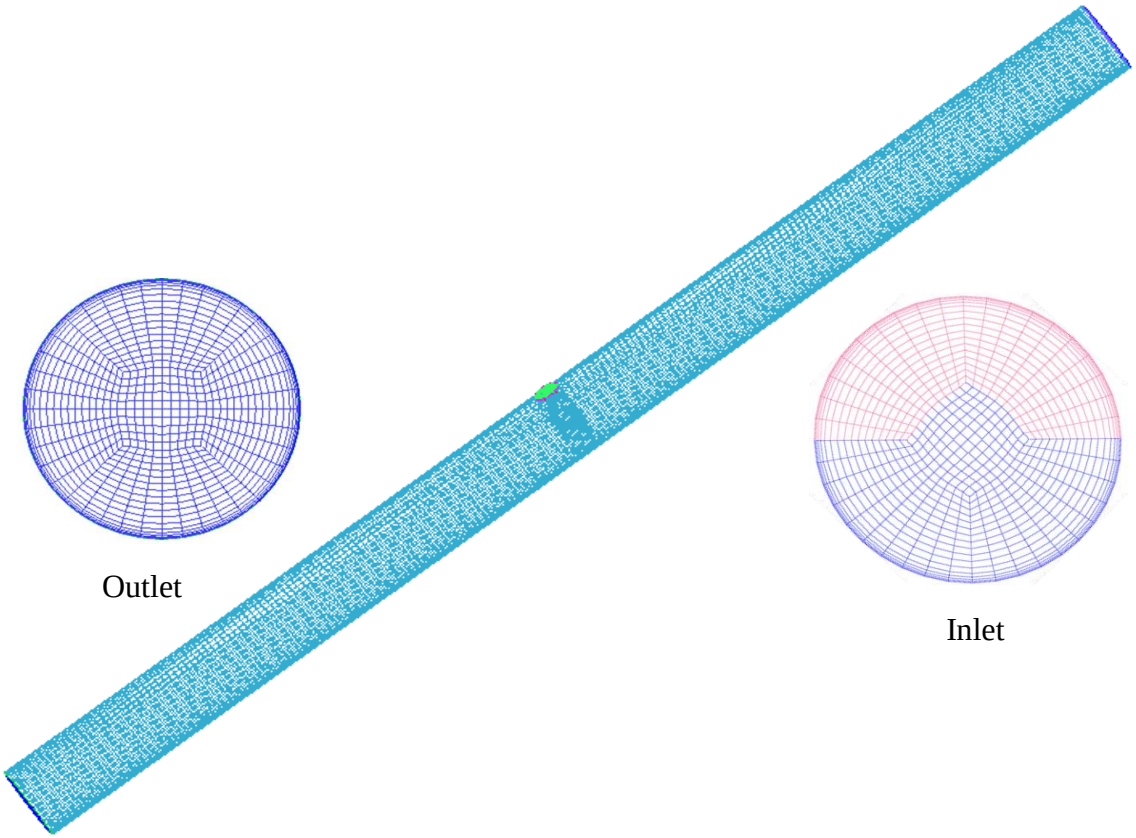
Figure 4-1: Flow diagram of the CFD modelling Procedure and Validations

4.4.1 Geometry Design

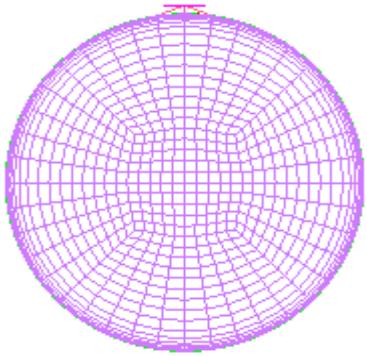
The geometry used in this study was created using ANSYS workbench 18.1. A pipe 3-D horizontal pipe was created on the x-z axis with the face along the x-y plane, as shown in Figure 4.2(a). The pipe leakage was generated using a circular opening size whose hole diameters (d_{hole}) are determined in accordance with the international oil and gas production recommended hole size distributions for subsea pipelines (Li et al., 2018). The generated geometry was then exported for the mesh generation.

4.4.2 Mesh generation

The numerical simulations are conducted on the created pipe with and without a leak. The flow domain is divided into small discrete cells and meshed using structured mesh. This grid type allows the mesh refinement to be closer to the pipe wall and provides an opportunity to prevent singularities in the middle of the flow domain (Akhlaghi et al., 2019). The mesh is generated such that the coarse mesh is in the centre while the fine mesh is at the region near the pipe wall, as recommended by Akhlaghi *et al.*(2019). The mesh was developed using advanced functions, which resulted in its high quality with an average orthogonal quality of 0.99 (closer to 1.0) and skewness of 0.06. A grid dependence test was performed using various grid sizes to identify the most efficient grids for this study. In the grid independence study, superficial gas and liquid velocities were chosen as 3.0 m/s and 0.32 m/s, respectively, which are also the same with the values used for numerical simulation in (Chinello et al., 2019) and the physical experiment on stratified flow conducted by Espedal (1998) employed for comparisons. These values also represent the highest liquid hold-up and Reynolds number used in these studies.



(a) Mesh generation for modelling pipeline leakage



(b) Cross-section view of the leakage **(c)** Top view of the leakage

Figure 4-2. Depiction of the mesh duct and detail of (a) Mesh generation for modelling pipeline leakage, (b) Cross-section view of the leakage.

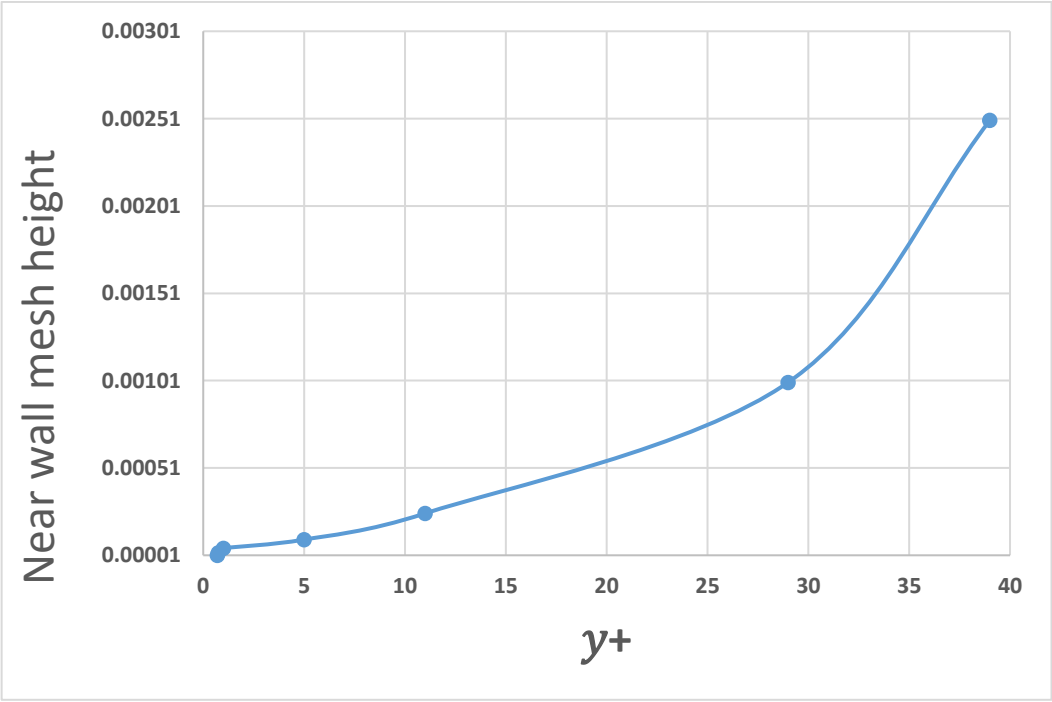
The mesh independence analysis was performed by running simulations on grids with the smaller cells number. The grids size was further reduced, which subsequently led to the increases in grids number. Note that a mesh independent solution exists once changes in mesh size does not affect the final simulation. The grids sensitivity was conducted by increasing the mesh sizes at the cross-section of the pipe and along the pipe axis. The details

specifications of the grids employed for analyses, including their mesh sizes, y^+ and obtained pressure drop and liquid hold-up are presented in Table 4.1. Seven different mesh sizes are applied in the grid test, as shown in Table 4.1, to demonstrate the influence of y^+ as a means of identifying the appropriate near-wall treatment. This was achieved by refining the mesh, with particular attention to the near-wall region of the geometry. The simulation results show that increases in grid numbers from mesh 4 to mesh 7 have little changes in the pressure drop, whereas the difference between mesh 1 and the other mesh 4 is massive. The pressure drop was computed from the obtained numerical solution by recording the time-averaged static pressure at different successive points on the pipe wall (planes cut at different pipe locations) after the flow was ensured to reach a steady state. This approach followed the common practice reported in the literature, in particular for stratified flows (Ali et al., 2022; Newton & Behnia, 1996).

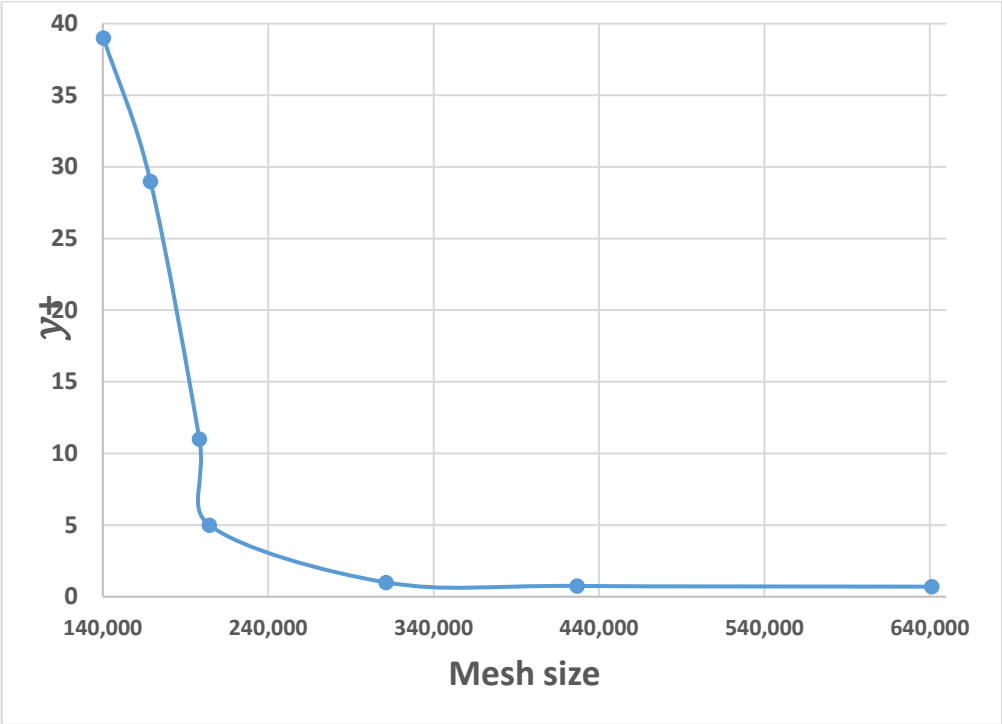
The near-wall mesh size against y^+ for different grids test conducted is shown in Figure 4.3(a), while the y^+ versus total mesh size illustrated in Figure 4.3(b). Figure 4.3(c) displays the y^+ effect on liquid hold-up obtained at the centre of the pipe length ($X = 1.3\,m$). The figure indicates variation in liquid level from meshes 5 to 7 is negligible. Therefore, mesh 5 was chosen for the numerical simulation as it demonstrates the optimal cells number for this study. Besides the simulation results' accuracy, simulation cost is essential to consider before one chosen mesh sizes for the simulation study. Therefore, mesh 5 demonstrates the optimum mesh size for the present study as it satisfies both computational cost and accuracy. The cross-sectional slices of the four structured grids that provided closer results are presented in Figure 4.4.

Table 4-1: Grids specification for mesh sensitivity analysis

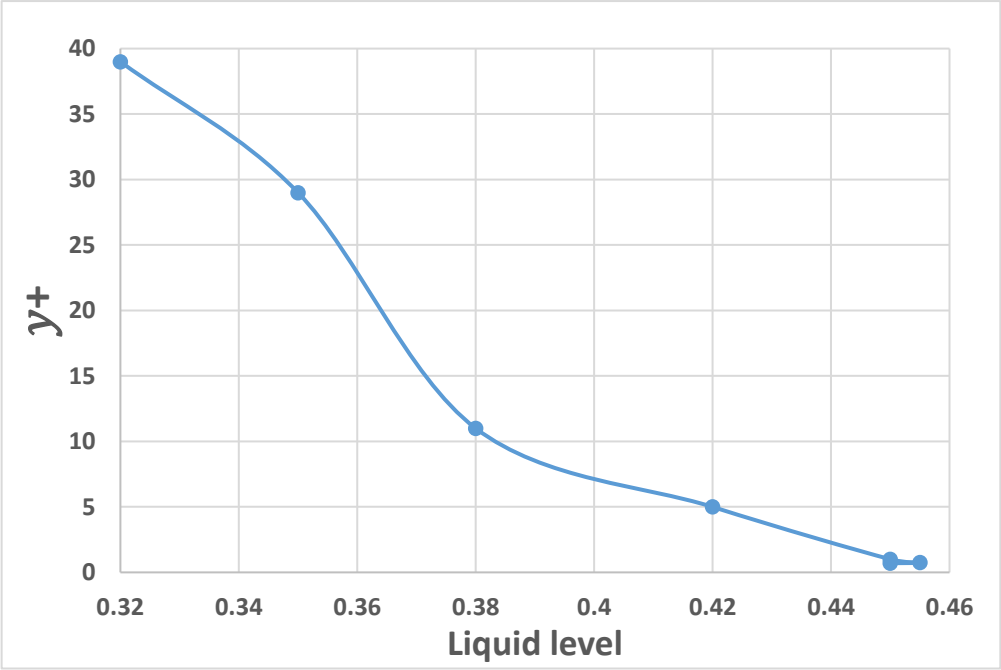
Mesh name	Near wall cell height	Mesh size	y^+	Computed Pressure drop (Pa/m)	Computed liquid hold-up
Mesh 1	0.0025	140,400	39.00	7.91	0.32
Mesh 2	0.001	168,800	29.00	8.72	0.35
Mesh 3	0.00025	198,400	11.00	9.18	0.38
Mesh 4	0.0001	204,400	5.00	10.22	0.42
Mesh 5	0.00005	311, 200	1.00	10.95	0.45
Mesh 6	0.000025	426,800	0.75	10.97	0.45
Mesh 7	0.00001	641,200	0.70	10.98	0.45



(a) Near wall mesh size against y^+



(b) y^+ against total mesh



(c) Effect of y^+ on liquid hold-up.

Figure 4-3: Different mesh configurations and corresponding y^+ values for; (a) near wall mesh size versus y^+, (b) y^+ versus total mesh and (c) effect of y^+ on liquid. The liquid holdup obtained at the centre of the pipe length ($X = 1.5\,m$).

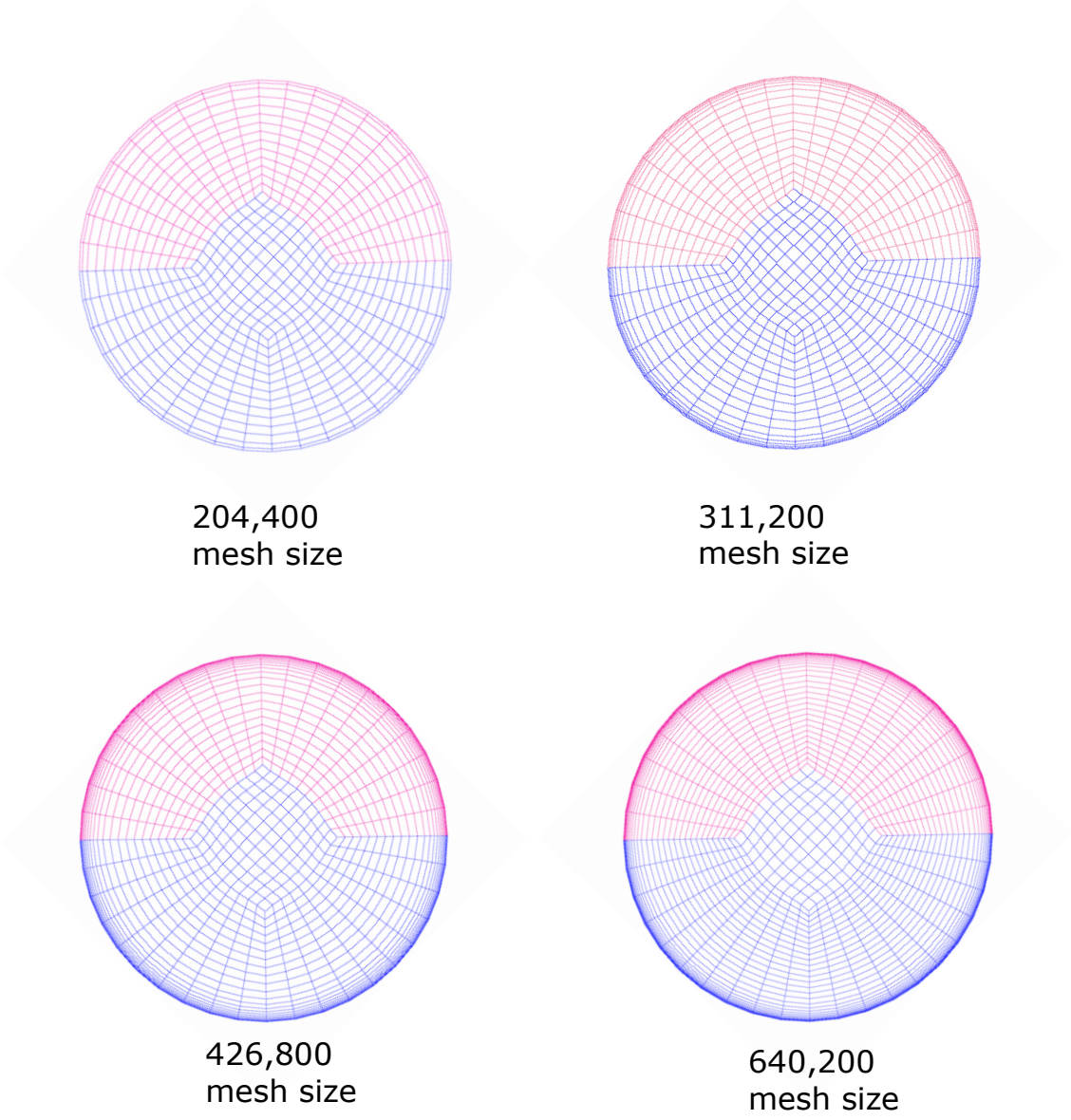


Figure 4-4: Cross-sectional slices of the grids tested at the pipe upstream.

4.4.3 Boundary conditions

The pipeline inlet is set as a velocity inlet boundary defined by gas and liquid superficial velocities. Injection of the two-phase into the computational domain can be done in two ways. One method is to inject the liquid into the pipe peripherally using power law velocity and gas with a uniform velocity profile in the centre region of the pipe (Akhlaghi et al., 2019). After some distance, the separation between the mixed phases initiates along the length of the pipe and distributes fluids into a specific pattern. In the second approach, which is the method used in this study, the two phases are separately injected at the pipe inlet. One significant advantage of this

method is that flow can reach the fully developed condition sooner. The gas is injected from the upper half cross-section of the pipe, while the liquid is injected from the bottom half cross-section of the pipe. This resembles a separate flow structure, where each phase is separated into different layers, with the lighter fluid flowing on top of the denser fluid. The gas and liquid velocities at the inlet are specified to attain the targeted superficial velocities of the phases based on experimental data.

The leak boundary is set as pressure outlet. The no-slip condition is applied to the pipe wall. Since the flow is assumed to be fully developed at the pipeline outlet, the pressure outlet is imposed. The pipe is assumed to be in underwater condition, and the leak orifice and pipeline outlet pressures are defined constant, similar to that reported in (Kam, 2010) for pressure at 100 m below the sea surface (Wei and Masuri, 2019). In this case, the pipeline outlet and leak surrounding pressure is considered as 400 Pa. The physical properties of the fluid phases are presented in Table 4.2. The temperature of the property of the selected fluid is assumed to be 25^0C.

Table 4-2: Fluid phases of physical properties

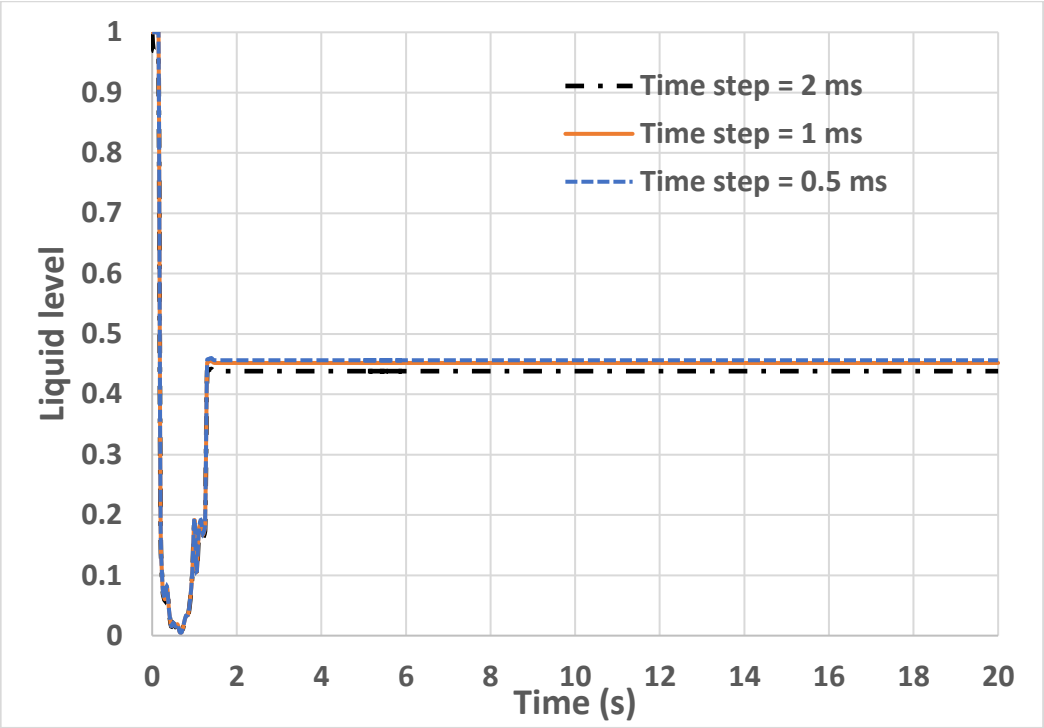
Property	Gas-phase	Liquid-phase
Density (ρ), kg/m^3	1.225	998.2
Dynamic viscosity (μ), Pa.s	0.00001823	0.00091
Interfacial tension, N/m	0.0715	

4.4.4 Numerical method

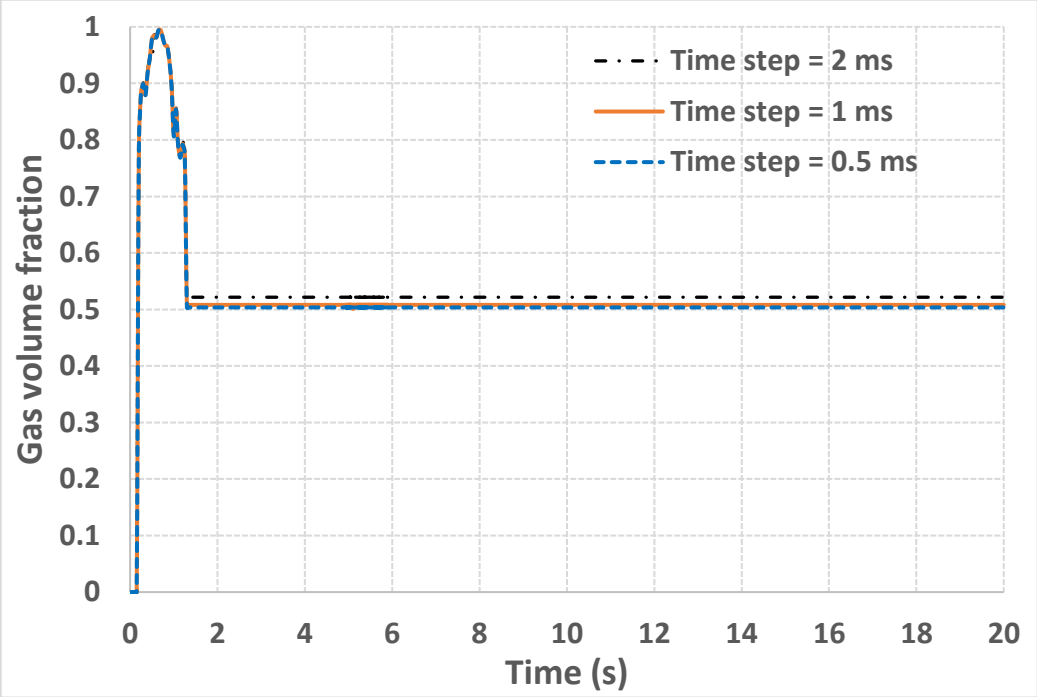
The VOF modelling method is employed to simulate stratified gas-liquid flows. The computation is performed using a pressure-based solver, while the pressure fields are coupled with the velocity fields using SIMPLE pressure-velocity coupling scheme. The turbulence is modelled using the $k-\omega$ SST model. The momentum, turbulent kinetic energy and specific dissipation rate equations are discretised in space for the advection terms using a second-order upwind scheme in accordance with the study of Chinello *et al.*(2019). The discretisation of the volume fraction is performed using high-resolution interface capturing (HRIC) scheme. A first-order

implicit temporal discretisation scheme is used to solve the governing equations. This method has been demonstrated to be reliable for evaluating pressure gradients and flow rates, which are of interest in this work. The implicit algorithm is applied because it allows the numerical calculation to stabilise unconditionally with respect to the time-step size (Ali, 2017).

The time step used in the simulations is 1 ms and simulated for 20 seconds, which is 20,000 iterations. Note that different time steps are tested before arriving at the 1 ms. The analysis of time step size on the liquid holdup profile over time is reported in Figure 4.5. The figure illustrates the liquid level fields obtained with three different values of time steps. When a time-step of 1 ms is simulated, the liquid holdup does not differ from the one gotten from 0.5 ms. A slightly different liquid level is obtained when the time-step is reduced to 2 ms. With a time-step of 1 ms, the steady state condition is observed at 5000 iterations (5 seconds). Therefore, 20,000 iterations, which is equivalent to 20 seconds is utilised to run the simulations. The predicted liquid height can be visualised more clearly using contour plots of the liquid volume fractions. Figure 4.6 displays the contour plots of the liquid heights for the three-time steps tested. The cross-section plane was obtained at the middle of the pipe length ($\mathrm{X} = 1.5\ \mathrm{m}$). One explanation that can the noted in this figure is that the height of the liquid appeared to be almost the same. Therefore, 1 ms time step is used for the present simulation in this study.



(a) Time step impact on liquid hold up



(b) Time step impact on the gas volume fraction

Figure 4-5: Results of the time-step convergence analysis; (a) Time step impact on liquid hold up, (b) Time step impact on gas volume fraction. Calculated liquid hold-up and gas volume fraction from cross-section area-weighted average at 1.5 m away from the pipe upstream after the stratified flow is fully developed.

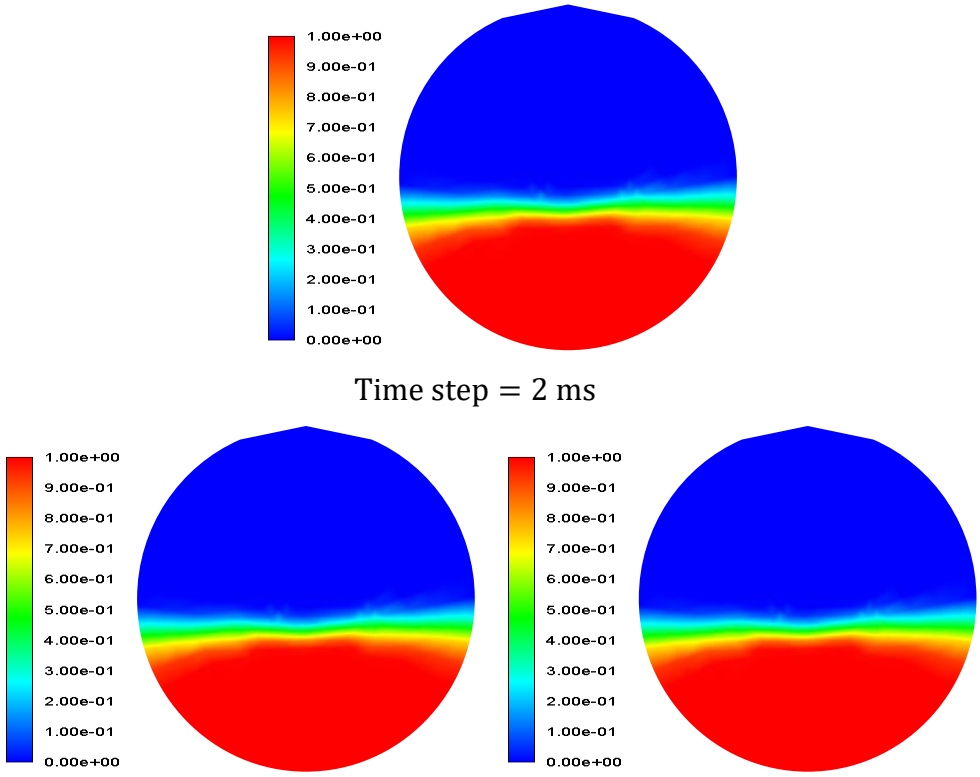


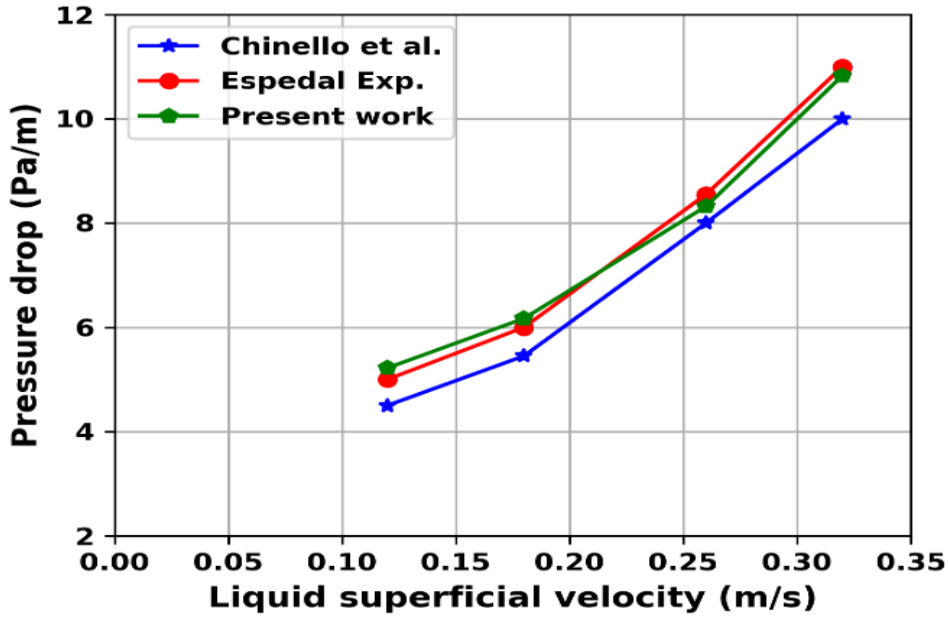
Figure 4-6: Contours of the gas-liquid volume fraction field in cross-section plane at the centre of the pipe length ($\mathrm{X} = 1.5\ \mathrm{m}$) for the different time steps. The blue colour represents the gas phase and the red colour represents the liquid phase.

4.4.5 Code validation

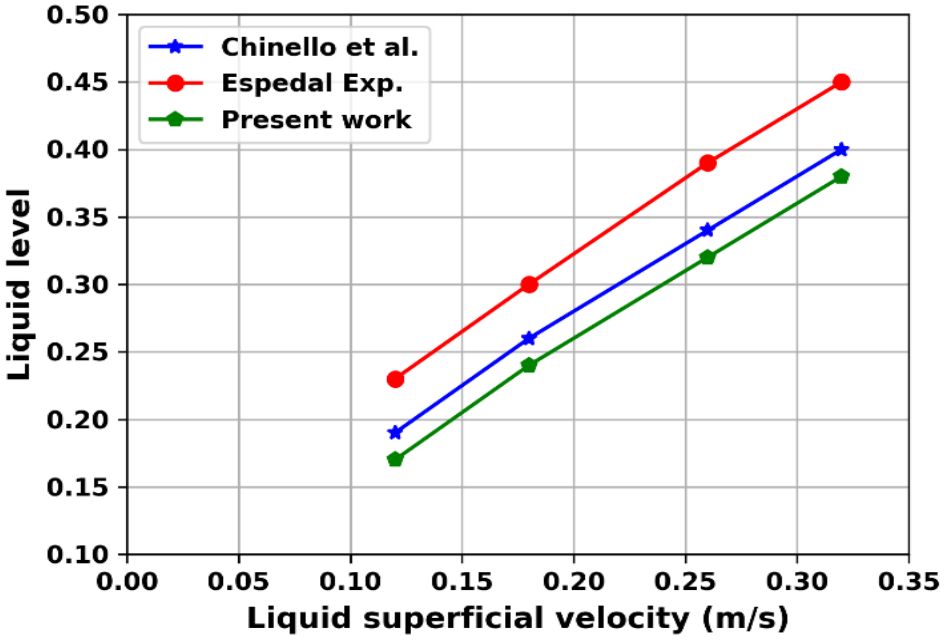
The CFD code used in this study has been validated against the published experimental data in (Espedal, 1998) and numerical simulations reported in (Chinello et al., 2019), which also employed the VOF model in ANSYS. Simulations are conducted using the VOF model for stratified air-water flow in a 3D pipe with the same experimental conditions as in these studies. The pipe used for the simulations is 18 m in length with a diameter of 0.06 m. It is important to highlight that the mesh employed for the 18 m pipe is the same as the one used for grid convergence with additional mesh on the axial cells. The values of the model parameters for the density, interfacial tension and dynamic viscosity are given in Table 2. The $k-\omega$ SST

turbulence model is applied to simulate the air-water field. Four sets of numerical simulations were performed using the superficial gas velocity of 3 m/s, while the superficial liquid velocities were chosen as 0.12 m/s, 0.18 m/s, 0.26 m/s and 0.32 m/s. The pressure gradients are computed and compared against the experimental data. Please note that the pressure gradient was computed from the obtained numerical solution by recording the time-averaged static pressure at different successive points on the pipe wall after the flow is ensured to reach a steady state. This approach followed the common practice reported in the literature, in particular for stratified flows (Ali et al., 2022; Newton & Behnia, 1996). Figure 4.7(a) shows the comparison of the present simulation results against the numerical simulations reported in (Chinello et al., 2019), and experimental data reported in (Chinello et al., 2019).

The obtained results demonstrate good agreement with the published CFD simulation results and experimental data. As shown in Figure 4.7(a), the pressure gradient in the present simulation is more consistent with the experimental data than the simulation results reported by Chinello *et al.,* with little underestimation of liquid levels. The reason for the underestimation of liquid levels in Figure 4.7(b) could be inherent from the liquid injection surface area of the pipe (see Figure 4.2 for the inlet cross-section plane in boundary condition). Therefore, it should be admitted that there is a discrepancy in liquid levels obtained in both simulation and experiments due to the possible difference in the surface area of injection of the liquid phase. This validation has been undertaken to demonstrate the adequacy of the mesh and numerical schemes employed. The liquid-phase levels are obtained as an area-weighted average of the liquid volume fraction across a section at 9 m away from the pipe upstream (middle of the pipe length). Figure 4.8 displays the cross-section view of the gas-liquid volume fraction field at the centre of the pipe length ($X = 9\,m$). It can be seen that different superficial liquid velocities yield different liquid levels. A lower liquid level was obtained for a lower superficial liquid velocity, while a higher superficial liquid velocity yielded a higher liquid level.



(a)



(b)

Figure 4-7: Validation of numerical simulation model against experimental data reported in (Espedal, 1998) and numerical simulation results in (Chinello et al., 2019); (a) pressure drop (Pa/m), (b) Liquid level.

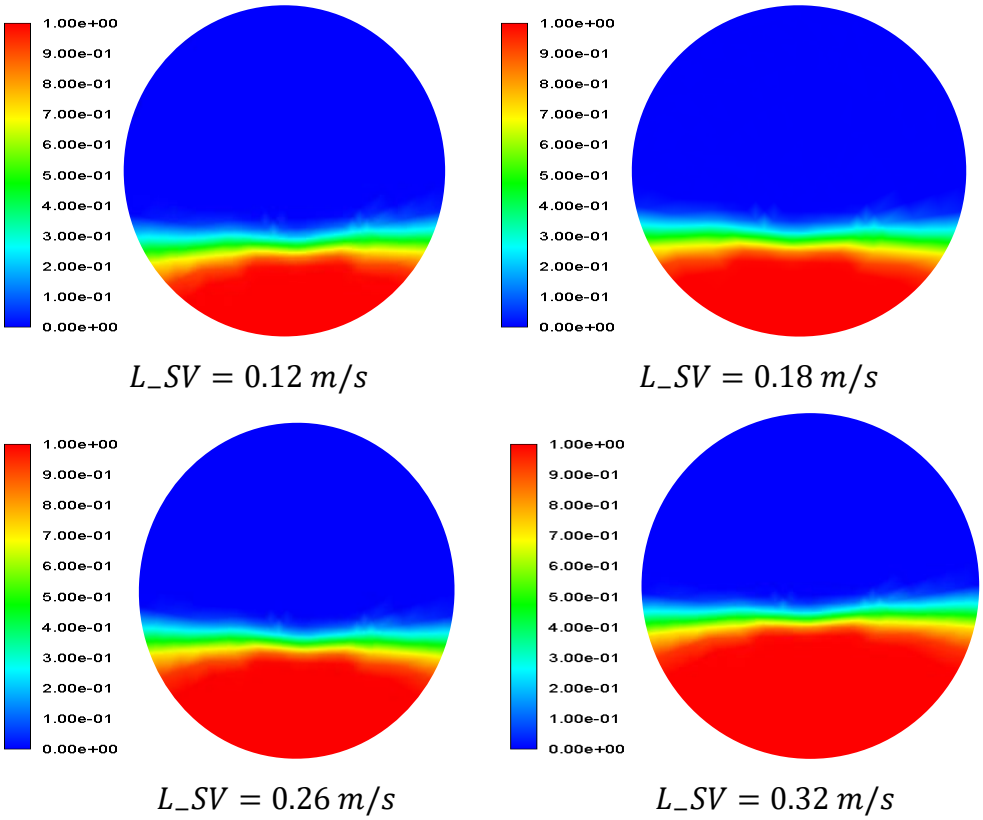


Figure 4-8: Contours of the gas-liquid volume fraction field in the cross-section plane at the centre of the pipe length ($X = 9\ m$) for the different superficial liquid velocities. The L_SV represent superficial liquid velocity, the blue colour represents the gas phase and the red colour represents the liquid phase.

The present numerical model is also validated against experimental data reported by Strand (1993) to further ascertain our model's validity. Strand (1993) performed pressure drop measurements in a two-phase gas-liquid stratified flow using a 50 mm internal diameter. Air and water were the fluids used. The pressure drop with a constant superficial gas velocity and different superficial liquid velocities were measured by sample five probes simultaneously in a pipe during experiments. The superficial gas velocity is 8.6 m/s, while the superficial liquid velocities were chosen as 0.243 m/s, 0.259 m/s, 0.278 m/s, 0.327 m/s, 0.354 m/s, 0.396 m/s, 0.484 m/s and 0.678 m/s. The pressure gradient was computed by probing the planes at five different locations with equal intervals on the pipe. The comparisons of the pressure gradient between the current simulation and the corresponding

experimental data of Strand (1993) is shown in Figure 4.9. As shown in Figure 4.9, the prediction matches the measurement data very well, with a deviation of less than 5%. The cross-section view of the gas-liquid volume fraction field obtained at the centre of the pipe length ($X = 7.5\,m$) is displayed in Figure 4.10. It is also confirmed here that different superficial liquid velocities yield different liquid levels. The lower liquid level was obtained for a lower superficial liquid velocity for a constant superficial gas velocity, while a higher superficial liquid velocity yielded a higher liquid level.

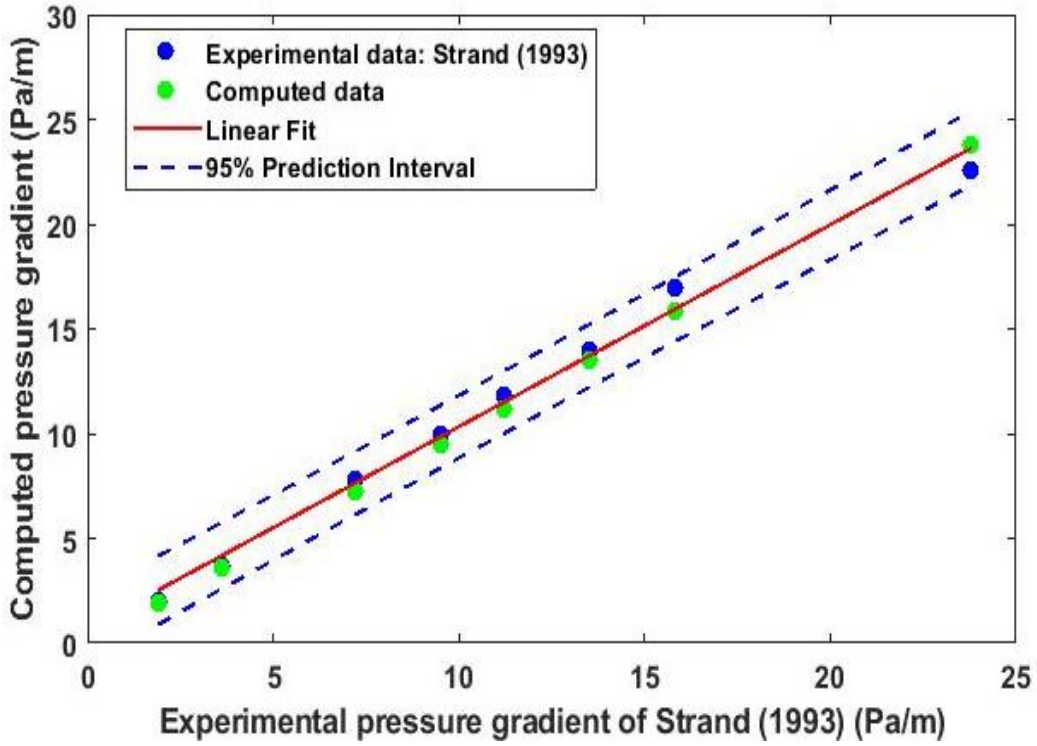


Figure 4-9: 5% linear fit plot comparison of computed pressure gradient with experiments data of Strand (1993).

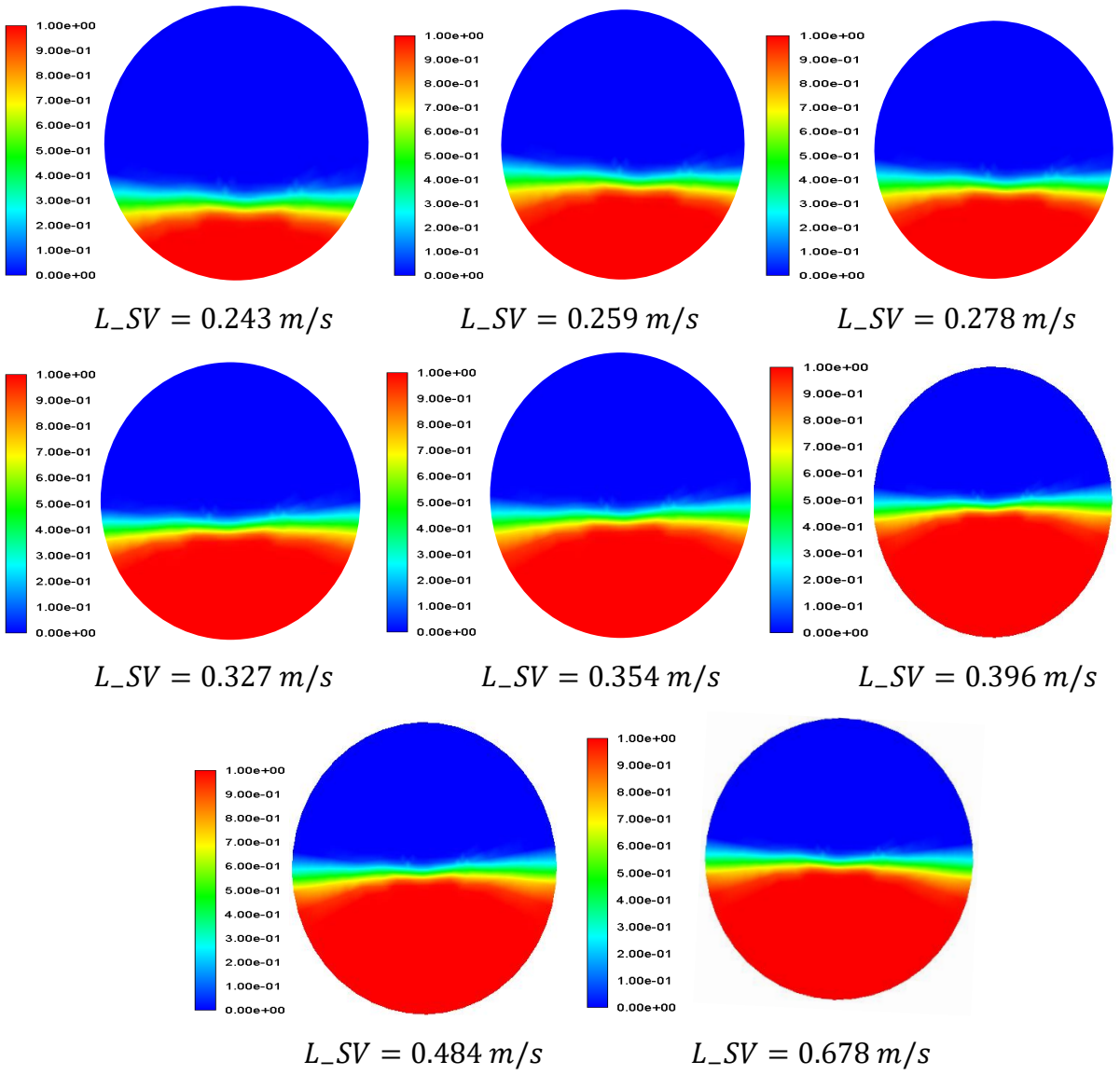


Figure 4-10: Contours of the gas-liquid volume fraction field in the cross-section plane at the centre of the pipe length ($X = 9\ m$) for the different superficial liquid velocities. The L_SV represent superficial liquid velocity, the blue colour represents the gas phase and the red colour represents the liquid phase.

4.5 Pipeline leaks comparison against experimental data

Experimental data focused on the multiphase pipeline with the leak is seldomly reported, and it is not easy to set up a flow rig similar to the one reported by Molina-Espinosa et al. (2013), to test the sgas-liquid two-phase pipeline leakage. The experimental data obtained from the same geometric

model and simulation conditions in monophase systems by Molina-Espinosa et al. (2013) is employed to verify the current s model boundary conditions. Molina-Espinosa *et al.*(2013) carried out numerical modelling backed up by physical experiments for pipe leakage. This study measured pressure drop on incompressible flow in a pipe with and without leakage. The experiments were performed using a 2.33 m long horizontal pipe of 0.0127 m internal diameter. The fluid considered for the experiment is water, while the static pressure tapping points ($P_1 - P_6$) distributed along the main measurement section. A discharged valve was used to creats leakages.

Figure 4.11 shows the comparison of the current numerical results and experimental data reported in (Molina-Espinosa et al., 2013). The pressure distribution proved effective and scientific to characterise stratified flow behaviours in this study. The leak effect on stratified flow behaviours induced by leaks has previously been reported similar to the monophase pipeline leakage in the previous study (Figueiredo *et al.*(2017). They concluded that the leak localisation strategy based on the upstream and downstream pressure profiles commonly employed in monophase flow pipeline leakage could be extended to the stratified-flow system. However, all the data reported in that study was based on the 1-D pipeline.

The present stratified flow model carried out in a 3-D pipeline is compared with the monophase flow system and validated with the experimental data reported by Molina-Espinosa *et al.*(2013). Molina-Espinosa *et al.*(2013) measured pressure distribution for the leak-free and leak diameters of 0.0033, 0.0052 and 0.0074 m, which form the leak sizes considered for the validation in the present study. The pipeline could be hundreds or thousands of meters long in reality; however, irrespective of the length of the pipeline, the pressure gradient would remain the same under normal flow conditions. Therefore, a comparison between the simulation results obtained from the pipeline length considered in the present study and the experimental data presented in (Molina-Espinosa et al., 2013) is scientifically sound.

The comparison of the pressure profile between experimental data and monophase results is shown in Figure 4.11. The pressure profile without leak is illustrated in Figure 4.11(a), and the resulting pressure profile with

leak sizes 0.0033, 0.0052 and 0.0074 m are shown in Figure 4.11(b), Figure 4.11(c), and Figure 4.11(d), respectively. Figure 4.12 compares stratified flow against monophase results in Figure 4.11. The monophase and stratified flow models are set up based on the experimental configuration for validation (Molina-Espinosa et al., 2013). As shown in Figure 4.11, the monophase simulation results agree with the experimental data conducted on a single-phase scenario at a higher degree. The pressure profile correlation in Figure 4.12 reveals a slight divergence. The reason is that the stratified model is made up of gas-liquid phases, leading to the gas release rate probably being higher than the liquid quantities under the same leak size. Statistical tests are applied to verify the consistency among pressure data obtained from the monophase simulation, stratified flow simulation and experiments reported in (Molina-Espinosa et al., 2013).

Furthermore, statistical tests are applied to verify the consistency among pressure data obtained from the monophase simulation, stratified flow simulation and experiments reported in the literature. The analysis was computed in MATLAB 2018b using one-way Analysis of Variance (ANOVA) to compare the pressure gradient before and after the leak. The summary of the hypothesis test results for the monophase simulations, experimental data and stratified model is presented in Table 4.3. The p-values measure how much the means difference of the three data disagrees with the null hypothesis (the sample means of data taken from the three groups are equal). As is clearly shown, the p-values for all the cases are range from 0.131 to 0.734, using 0.05 significance (α) level. These indicate that the mean difference between the three data are not statistically significant and demonstrates strong evidence for the null hypothesis. We fail to reject the null hypothesis at the significant level of 0.05.

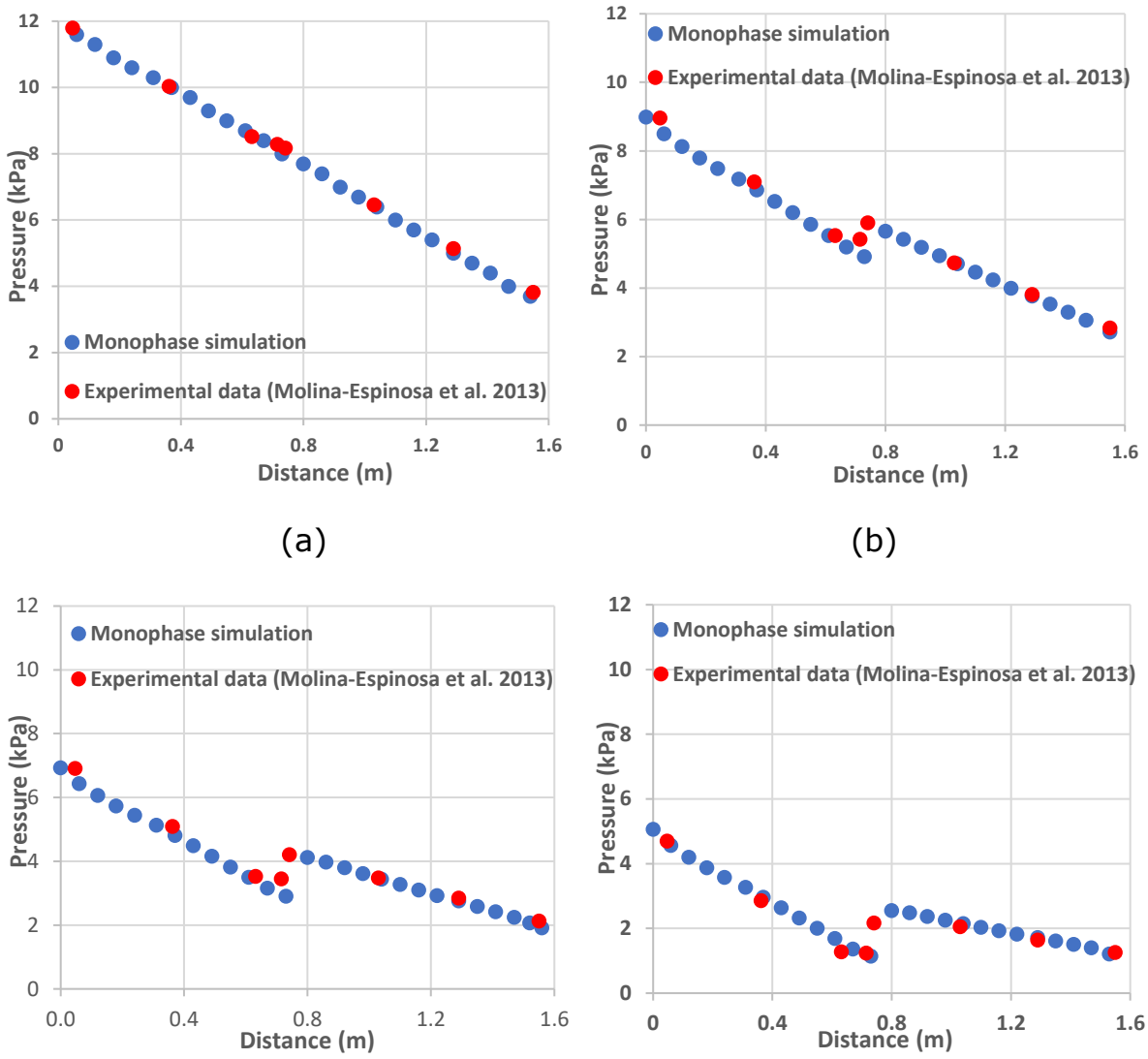


Figure 4-11: Comparison of the computed monophase pressure profile against experimental data reported by Molina-Espinosa et al. (2013); (a) leak free, (b) 0.0033 m leak, (c) 0.0052 m leak, (d) 0.0074 m leak.

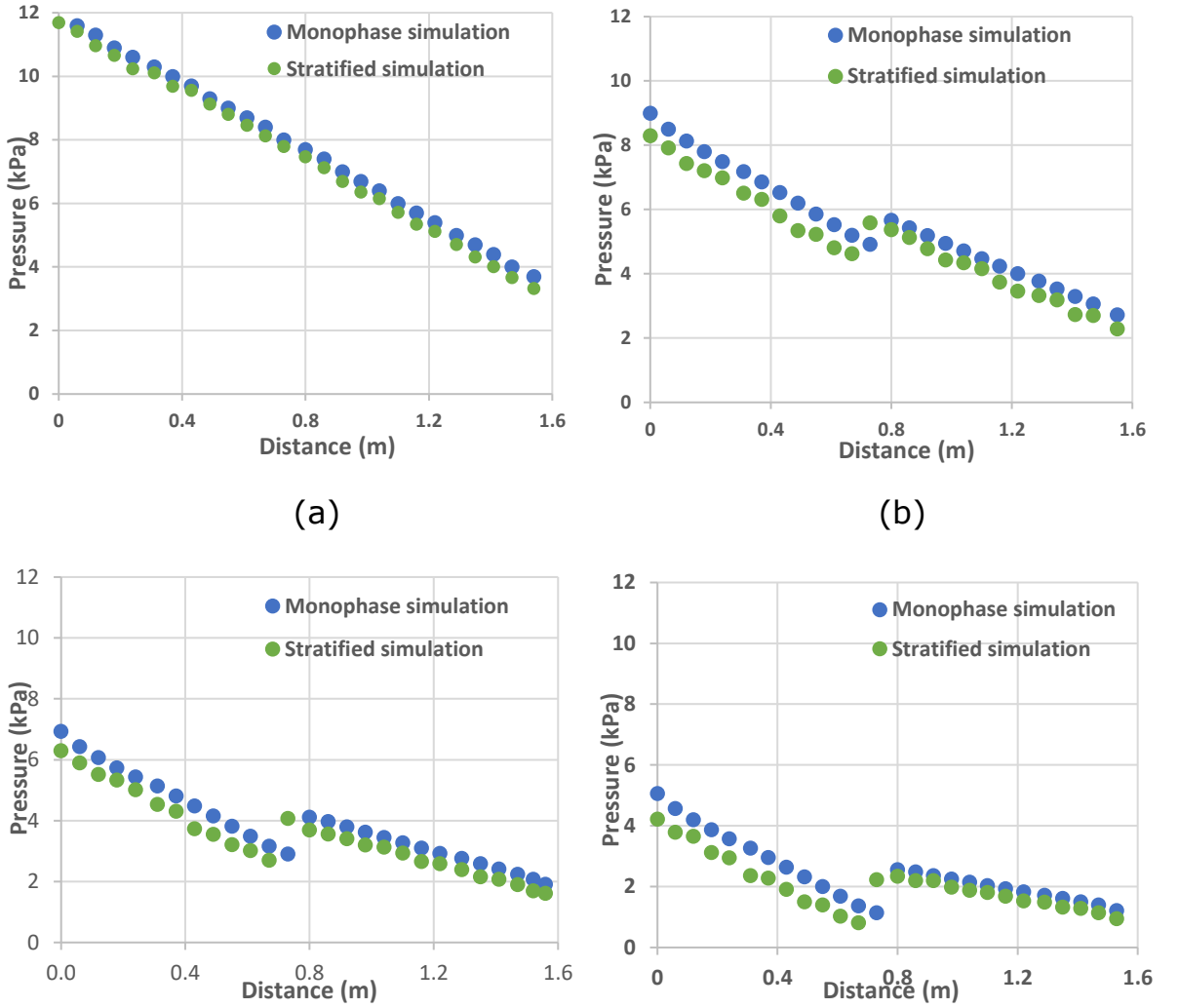
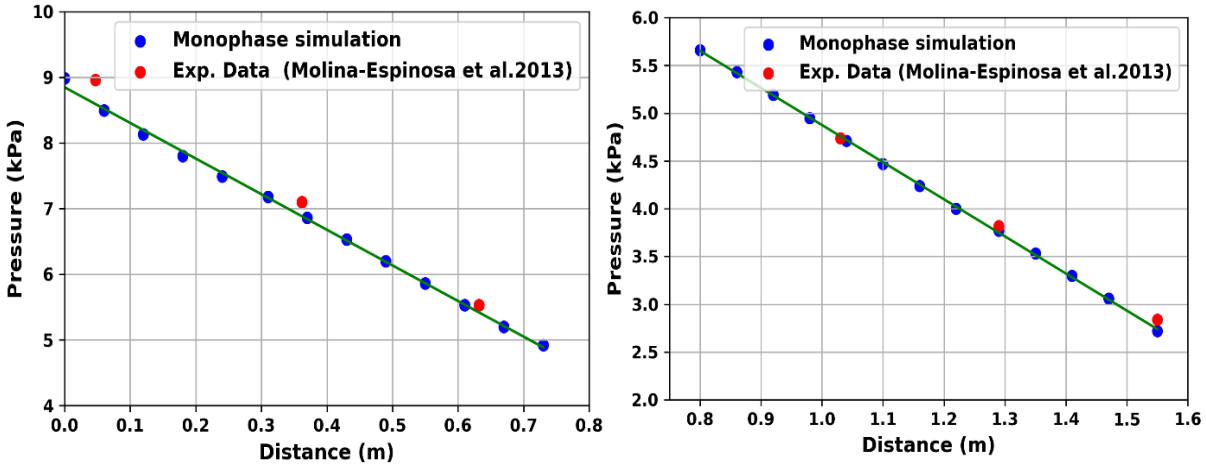


Figure 4-12: Comparison of the pressure profile between the monophase flow and the stratified flow model; (a) leak free, (b) 0.0033 m leak, (c) 0.0052 m leak, (d) 0.0074 m leak.

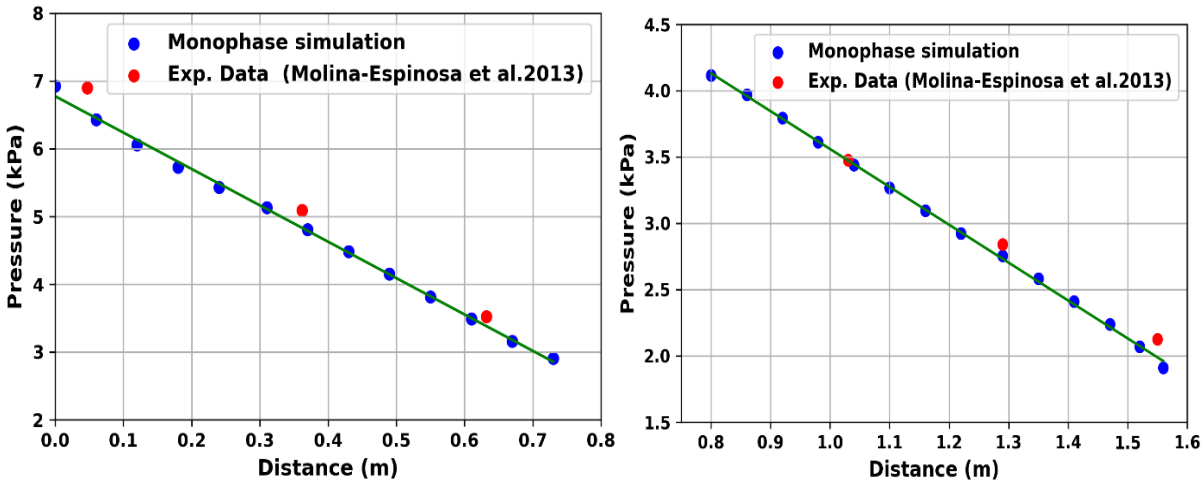
Table 4-3: Numerical (monophase and stratified) simulations and experimental data comparison using one-way ANOVA; 0.05 significance (α) level

Leak scenario	**Pressure gradient**	**p-values**
Leak-free	Upstream pressure	0.734
	Downstream pressure	0.747
Leak 1	Upstream pressure	0.382
	Downstream pressure	0.365
Leak 2	Upstream pressure	0.473
	Downstream pressure	0.354
Leak 3	Upstream pressure	0.365
	Downstream pressure	0.131

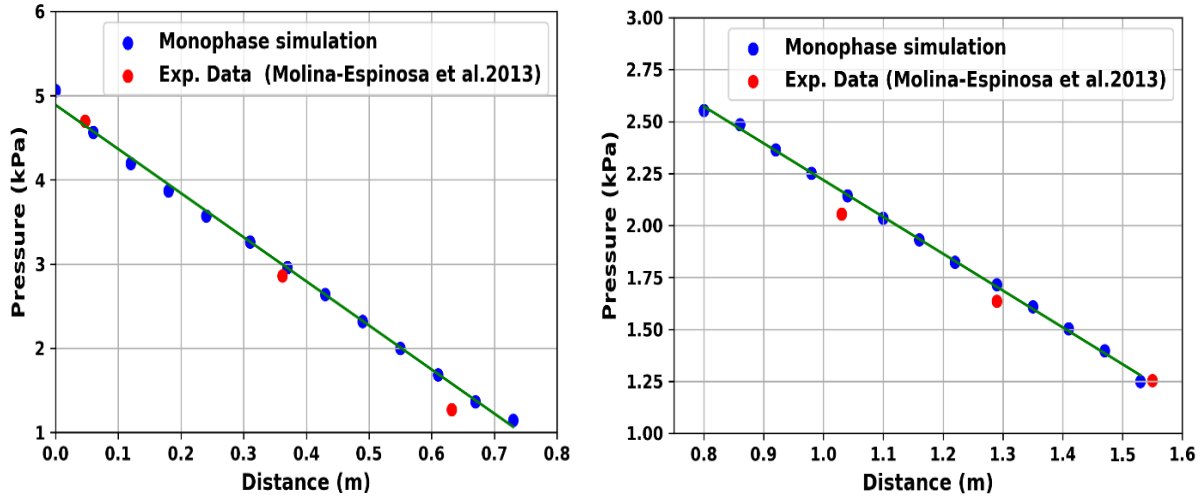
The linear regression plot shown in Figure 4.13 demonstrates the adequate closeness of the experimental and monophase simulation data points to the regression model. The average variance of the experimental data from the fitness model is calculated using Mean Absolute Deviation (MAD). The obtained results are presented in Table 4.4. From these results, the highest MAD value is 0.263, which shows good agreement between the two data.



(a) Leak 1



(b) Leak 2



(c) Leak 3

Figure 4-13: Linear regression plot for monophase simulation against experimental data. Pressure gradient before leak (left) and pressure gradient after leak (right).

Table 4-4: The results of computed Mean Absolute Deviation (MAD) of experimental data from monophase simulation regression model.

Leak scenario	**Pressure gradient**	**MAD**
Leak free	Upstream pressure	0.060
	Downstream pressure	0.123
Leak 1	Upstream pressure	0.234
	Downstream pressure	0.060
Leak 2	Upstream pressure	0.263
	Downstream pressure	0.089
Leak 3	Upstream pressure	0.149
	Downstream pressure	0.061

Table 4.5 also presents the results of the hypothesis tests performed to determine whether the constants and coefficients of linear regression models of the monophase and stratified pressure gradients variation before and after the leak are statistically significant. As demonstrated in the results shown in Table 4.5, the high R-square values indicate that the fitted linear regression models approximate the process which generates the data well. It is important to notice that the least R-squared value is 0.997 despite the multiphase coefficients p-value higher than 0.05. This indicates the possible disband among the stratified data due to the transient state of the multiphase model.

Table 4-5: Regression hypothesis results for monophase and stratified simulations comparison

Leak scenario		**R-Square**	**RSME**	**Constant p-value**	**Mono. Coef. p-value**	**Multiphase Coef. p-value**
Leak free	Upstream pressure	0.998	0.033	1.0295×10^{-13}	0.043353	0.28861
	Downstream pressure	**1.000**	0.005	1.7711×10^{-20}	0.0005064	0.054394
Leak 1	Upstream pressure	0.998	0.011	1.902×10^{-12}	0.0020	0.2820
	Downstream pressure	**1.000**	0.004	4.4253×10^{-20}	3.7577×10^{-09}	0.57519
Leak 2	Upstream pressure	0.998	0.009	4.774×10^{-13}	0.0020	0.0690
	Downstream pressure	0.998	0.014	7.8827×10^{-19}	1.2721×10^{-06}	0.75957
Leak 3	Upstream pressure	0.998	0.012	1.305×10^{-11}	0.0010	0.1890
	Downstream pressure	**0.997**	0.021	3.1492×10^{-14}	0.0008683	0.84597

4.6 Summary

In this chapter, the potential of using CFD tools to model pipeline leakage as a function of multiphase flow is explored. The VOF model and SST k-ω turbulence modelling scheme were applied to simulate the two-phase flow in a horizontal pipeline. The superficial inlet velocities were chosen such that the stratified flow regime was formed. The results of the simulations were validated against the latest experimental and numerical data reported in the literature, and a good agreement was obtained. Statistical tests were also applied to verify the consistency among pressure data obtained from the monophase simulation, stratified flow simulation and experiments reported in the literature. In particular, p-values were employed to measure the level of disagreement between the three data with the null hypothesis. It was observed that the p-values for all the cases are range from 0.131 to 0.734, using a 0.05 significance (α) level. These indicate that the mean difference between the three data are not statistically significant and demonstrates strong evidence for the null hypothesis. The numerical pipeline leak detection results are presented in Chapter 5.

Chapter 5 Results of Numerical Pipeline Leakage Modelling

5.1 Introduction

This chapter presents and discusses numerical pipeline leakage detection results conducted on a 3-D pipe as a multiphase flow system. The objective is to study the effect of leak sizes, longitudinal leak locations, axial leak positions and multiple leakages. The results are presented for the flow rate, pressure gradient and volume fractions. Numerical simulations are performed on a horizontal pipe with different leak scenarios. Holes on the pipe, which are sources of leaks, are assumed to be circular. Its distribution sizes are determined based on the International Association of Oil and Gas Producers (IOGP) recommended hole sizes for subsea pipelines (Li et al., 2018). According to the pipeline opening sizes description specified in (Li et al., 2018), for a standard subsea pipeline with an average diameter of 0.334 m, a leak diameter of less than 0.02 m is regarded as a low leak. Moreover, a leak size between 0.02 to 0.08 m is classified as medium leakage, while a leak diameter higher than 0.08 m is regarded as a large leak. The computed pipe opening dimensions for the 60 mm diameter pipe employed in this study follow the recommended values in IOGP and they are listed in Table 5.1. The superficial gas and liquid velocities used for pipeline leak modelling are 4.5 m/s and 0.5 m/s, respectively, while the pipeline length is 50 times the diameter. These values are determined using the horizontal gas-liquid flow regime map to observe a stratified flow pattern (Kanin et al., 2019). The effect of leak sizes, longitudinal leak locations, axial leak positions and multiple leakages are investigated, and results are presented for the flow rate, pressure gradient and volume fractions. In this thesis, the pressure and flow rate were measured at thirteen different points along the pipe and used to analyse pipe leakage conditions. The pressure drop was computed from the obtained numerical solution by recording the time-averaged static pressure at different successive points on the pipe wall (planes cut at different pipe locations) after the flow was ensured to reach

a steady state. This approach followed the common practice reported in the literature, particularly for the stratified flows (Ali et al., 2022; Newton & Behnia, 1996).

Table 5-1: Hole diameters used for the simulations. These values are determined by rescaling the leak sizes in the 60 mm pipe to match the ratios by IOGP.

Hole size classes	Values (mm)	Leak size (percentage of pipe diameter)
Low	1.5	2.5%
Medium	9	15%
Large	14.5	24.2%
Rupture	18	30%

5.2 Leak Magnitudes Effect Analysis

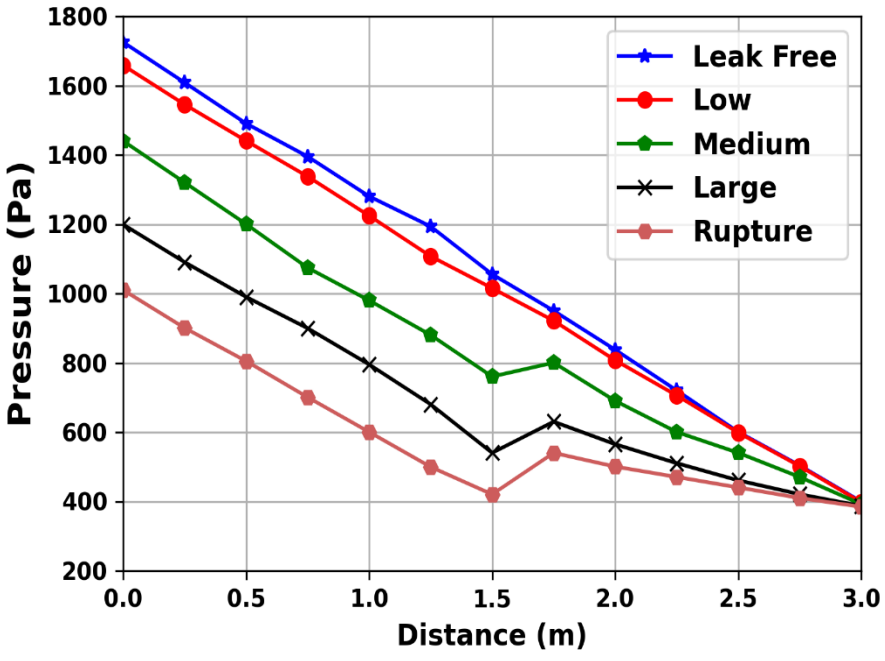
Leak size has a significant impact on the behaviour of fluids flow in the pipeline. In order to study the effect of leak magnitude on the multiphase flow behaviour induced by the leak, simulations of pipeline leakages for the different leak scenarios corresponding to the low, medium, large and rupture scenarios are conducted and analysed. The leak is placed at the top-middle part of the pipe, as shown in Figure 4.2. Table 5.1 presents the values of the leak sizes considered and their corresponding categories. The effects of leak size on the pressure gradient, the flow rate and the volume fraction (gas void fraction and liquid holdup) at selected planes along the pipeline are presented. The pressure profiles for different leak sizes investigated are shown in Figure 5.1(a). The pressure gradient remains identical for the leak-free scenario. The occurrence of the pipe resulted in pressure distribution profile changes. The increase in the leak size causes an increase in pressure drop resulting from an increase in friction due to the impact of fluid discharging from the leak point. The pressure drop decreased after reaching its maximum drop at the leak location and

gradually increasing downstream of the pipe. This phenomenon becomes comparatively pronounced as the leak size increases.

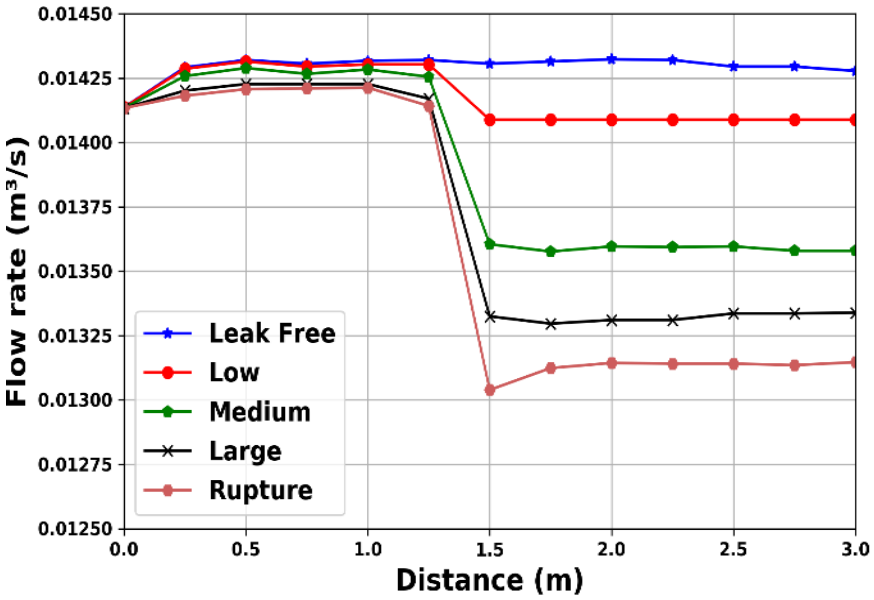
Although the existence of a small leak leads to a decrease in pressure upstream of the pipe, the effect of the small leak is not significant at the leak location. This agrees with the analytical calculation in (Kam, 2010), which affirmed that the presence of a small leak is not visible at the location of the leakage. However, as the pipe leak opening size increases, more fluids tend to discharge through the orifice region. A similar pressure response was also reported in physical experiment data reported by Molina-Espinosa et al. (2013) conducted on single-phase leakages.

The pressure at the pipe upstream drops with respect to the leak size due to the local pressure drop at the location of the leakage. As exemplified in Figure 5.1(a), the magnitude of the pipeline opening size affects the rate of fluids discharge in the leak neighbourhood. The increase in fluids escaping from the leak medium leads to a rise in pressure drop, particularly within the vicinity of the leakage. This implies that the pressure profile around the neighbourhood of the leak can aid the accurate identification of the leak location, particularly when the leak is medium size or large. The presence of a large leak reveals that the larger the leak, the more the fluids tend to discharge from the pipeline until it reaches the rupture stage. The effect of leak sizes on total flow rate characteristics based on various leak diameters is depicted in Figure 5.1(b). The maximum decrease in flow rate suddenly occurs immediately after the leak position. There is not much significant variation in flow rate before the occurrence of leakage, but as the size of the leak increases, the fluids flow rate also reduces dramatically starting from the leak location. Therefore, the increases in pipe opening size result in the decrement of the total flow rate downstream of the leak. This implies that the flow rate decreases with increasing leak size. This phenomenon is interpret as the cause for the pressure drop in the case of pipeline leakage. From the flow responses depicted in Figure 5.1, it can be concluded that upstream pressure serves as a pertinent indicator to detection of leakage as it appears to be the most sensitive indicator even if the size of the leak is small. At the same time, downstream flow rate response will be more

favourable for leak detection if the flow transducer is deployed downstream. Figure 5.2 presents the volume fraction contours at 2.5 m along the pipe under the same leak scenarios shown in Figure 5.1. The blue colour denotes the air void fraction, while the red indicates the liquid holdup. The air void fraction and the liquid holdup are distributed equally in the absence of leakage in Figure 5.2(a). The occurrence of the leak leads to the reduction in air void fraction downstream of the pipe, which causes an increase in the liquid holdup. Comparing the fluids volume fraction under different leak sizes shown in Figure 5.2 shows that leak size significantly influences the saturation of fluids flow. Overall, the larger the leak size, the more the relative amount of gas discharged from the pipeline if the leak is located at the top wall of the pipe. Therefore, the gas void fraction downstream of the leak becomes lower, which eventually increases the liquid holdup. This occurs because the gas is less dense and more mobile than the liquid, leading to the liquid replacing the escaped gas in the pipeline. The interface contour depicted in Figure 5.2 agrees with the observation of Figueiredo et al. (2017) on liquid holdup. The liquid holdup increases as the leak size increases due to the loss of mass at the leak location. Thus, the gas and liquid flow rates drop at the leak region to satisfy the continuity equations (Figueiredo et al., 2017).

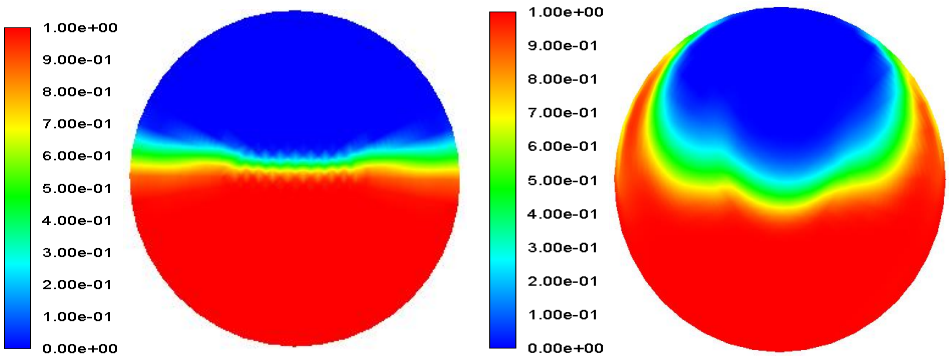


(a)

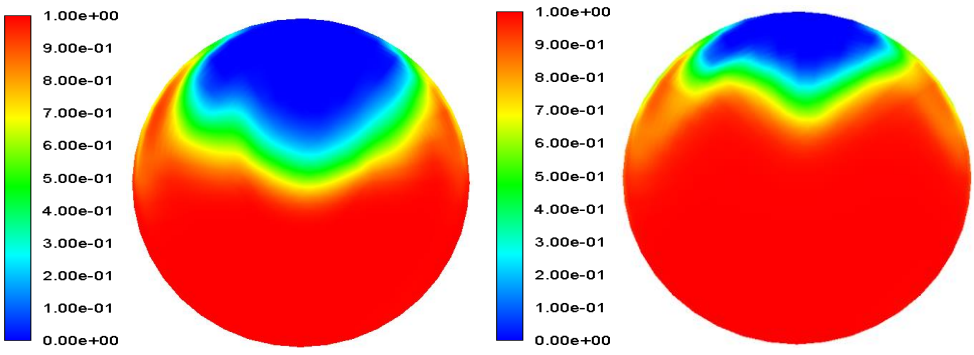


(b)

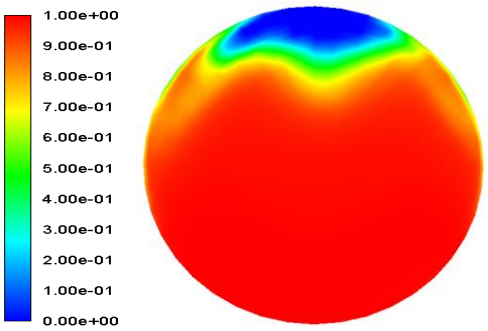
Figure 5-1: Leak sizes variation simulations response; (a) pressure distributions, (b) flow rate. Note that the flow rate represents the total flow rate for the two phases. Note that the leak is located at $x/2$, where x is the pipe length.



(a) leak-free (b) small



(c) medium (d) large



(e) rupture

Figure 5-2: Liquid volume fraction contour plots at 2.5 m for different leak opening sizes (Red and blue colours indicate water and air, respectively)

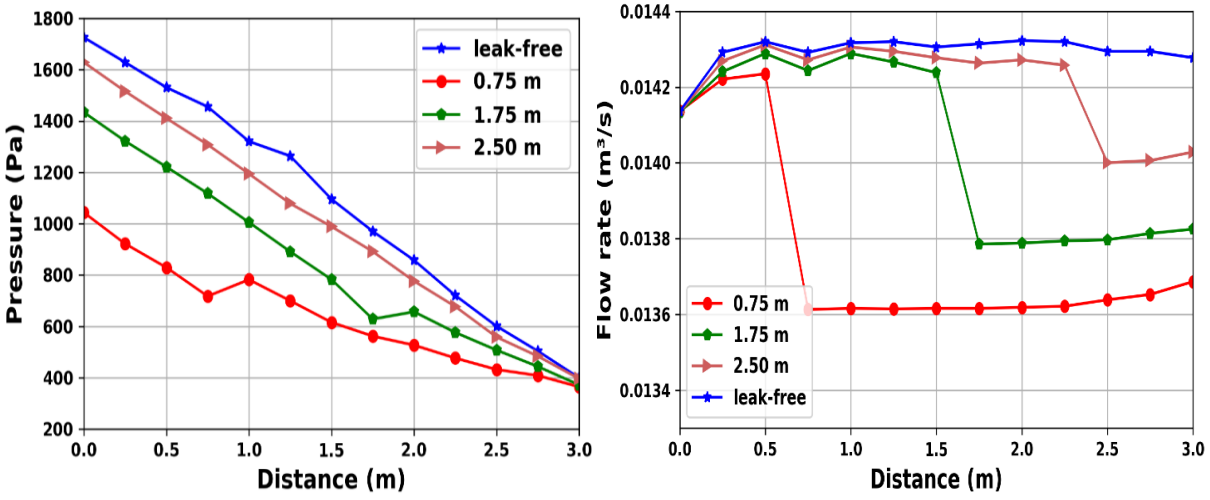
5.3 Longitudinal Leak Location Effect Analysis

Various challenges may be experienced in the process of identifying the position of leakage along a pipe, especially if the pipeline is installed underground or in a subsea environment. Therefore, it is important to investigate the effect of leaks on different locations along the pipe length to enhance leak assessment and emergency planning. In this study, the effect of leaks on different longitudinal locations is investigated and analysed. The leak location 1, location 2 and location 3 are set at 0.75 m, 1.75 m and 2.5 m, respectively away from the pipe upstream. Figure 5.3 presents the effect of longitudinal leak detection on the medium pipeline opening size for the pressure and flow rate responses. Figure 5.3(a) shows the effect of different longitudinal leak locations on the pressure profile. As seen in Figure 5.3(a), the occurrence of leakage toward the downstream of the pipe (at 2.5 m) led to little pressure drop. However, as the leak is positioned more towards the upstream section of the pipe, the leak effect becomes pronounced. Similar responses have also been observed in the analytical solution in multiphase pipeline leakage reported by Kam (2010).

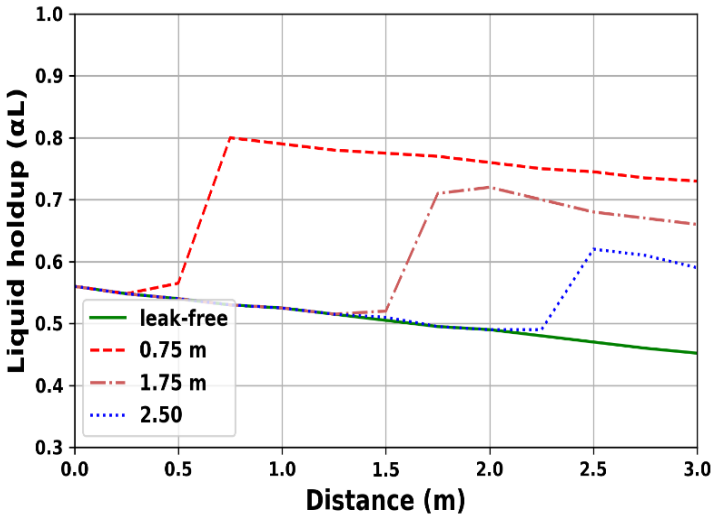
As observed in Figure 5.3(b), the leak occurrence leads to the flow rate decrement starting from the leak position down to the pipeline outlet. The leak occurred at 2.50 m away from the upstream pipeline cause about 0.00024 m^3/s flow rate reduction. By positioning a leak further upstream of the pipeline, the effect of a leak becomes more pronounced. This agrees

with the analytical solution reported in (Kam, 2010). If a leak occurs closer to the pipeline upstream, it is more favourable to detect the leak using inlet pressure monitoring. The result of the liquid holdup is illustrated in Figure 5.3(c). As it is clearly shown, the loss of pressure as the leak location closer to the upstream of the pipe reveals increases in liquid holdup accordingly. Figure 5.3(d) shows a comparison of the published liquid holdup of Figueiredo et al. (2017) described in Section 2.4.2 against the result in Figure 5.3(c). The relative rise in liquid holdup is determined by dividing the obtained liquid holdup for leak cases by the leak-free liquid level. The figure reveals reductions in relative jump, particularly as the leak closer to the pipeline downstream. Note that the simulation conditions in (Figueiredo et al., 2017) were performed using a 1D pipeline. However, the comparison in Figure 5.3(d) is made to correlate how leak location changes affect the liquid holdup.

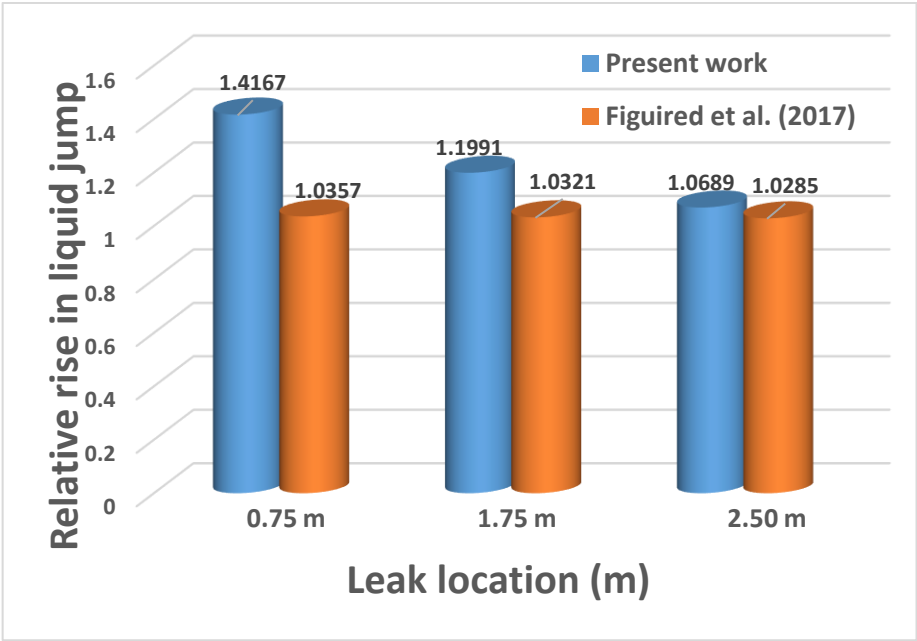
The volume fraction contour plots at 2.75 m for the longitudinal locations are illustrated in Figure 5.4. By comparison, a significant difference can be found in volume fraction as the location of leakage changes from the pipe upstream to the outlet. In the absence of leakage, the fraction of each phase distributes equally. However, the variation in leak position increases liquid accumulation as the leak location changes toward the upstream of the pipeline.



(a) (b)

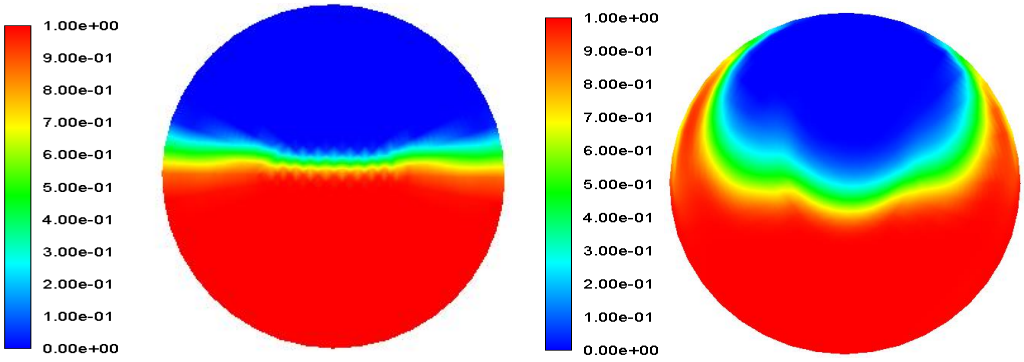


(c)

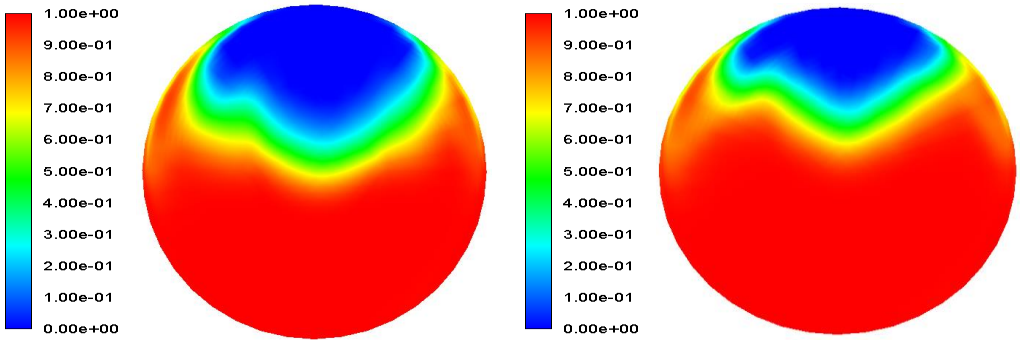


(d)

Figure 5-3: Effect of longitudinal leak locations; (a) pressure distributions, (b) flow rate, (c) liquid holdup, (d) liquid holdup comparison with published data. The legend shows different locations of leakage from the pipe upstream to the downstream. Note that the flow rate represents the total flow rate for the two phases.



(a) leak-free (b) 2.50 m



(a) 1.75 m (b) 0.75 m

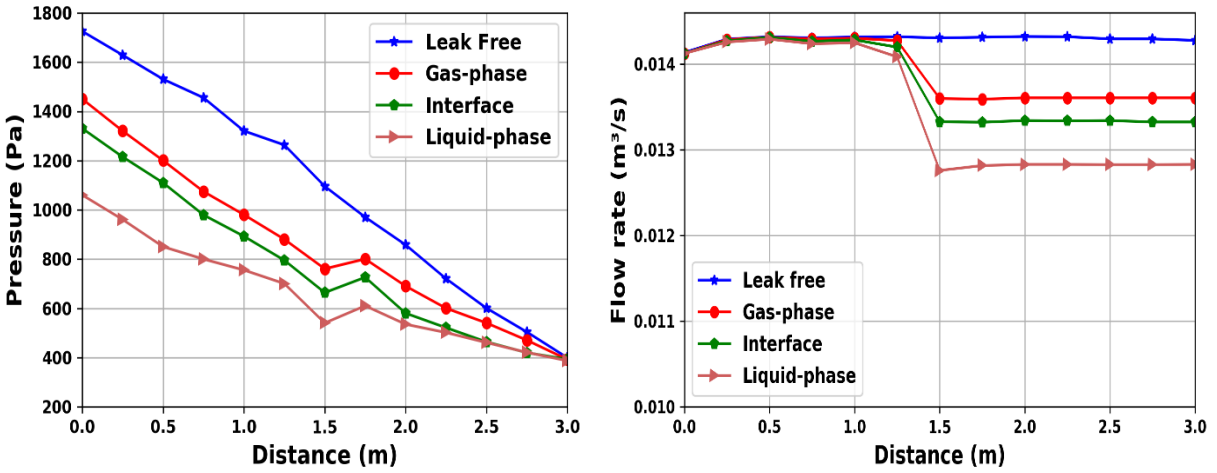
Figure 5-4: Volume fraction contour plots at 2.75 m for different longitudinal leak locations. (Red and blue colours indicate water and air, respectively

5.4 Circumferential Leak Positions Effect Analysis

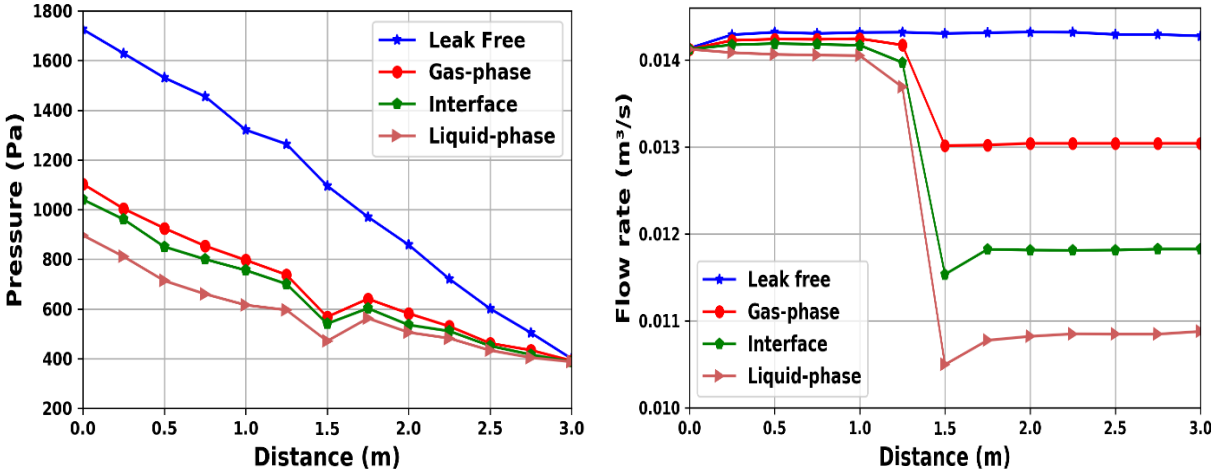
In sections 5.2 and 5.3, the leak was set to locate in the gas phase. Knowledge about pipeline leak position, namely gas-phase, liquid-phase or interface of the two phases, is important for enhancing the understanding of leak effect on a multiphase pipeline system. The leak scenarios for the medium and large sizes are considered to study hydraulic behaviours induced by leak at different fluid phases. The leak is located in the middle of the pipe ($X = \frac{L}{2}$, where L represents the pipe length), as shown in Figure 4.2. The legend indicates the fluid phases where the leak occurred. The flow parameters that are investigated include the pressure gradient, the total flow rate and the volume fraction of the fluids within the pipeline. The flow parameters variation for the medium leak size under different leak positions is presented in Figure 5.5(a). The legend indicates the fluid phases where the leak occurred. As seen in these figures, it is apparent that the location of leakage on the multiphase pipeline affects the flow pressure profile in the pipeline. A significant effect exists when the leak is situated on the liquid-

phase side. Similarly, the flow rate responses in Figure 5.5 (a) imply that the maximum total flow rate drop occurs at the liquid-phase axis, while the least drop is observed at the gas-phase position. Similar behaviour for the case of large leak can also be observed in Figure 5.5(b).

By comparison, we can find that the influence of pipeline leakage is more pronounced on the liquid phase than gas or gas-liquid interface, and the reasons are two-fold. Firstly, the leak at the bottom of the pipeline (liquid phase) favours the pipeline's fluid discharge quantity. The pressure drop in the liquid phase is higher due to the force of gravity that is much stronger than the surface tension that holds the fluids together inside the pipe (Cheah et al., 2013). Thus, work to pull the fluid (liquid) at the pipe's wall where leakage occurred. Secondly, the fluids' physical properties could also be another reason for the higher-pressure drop in the liquid phase. The liquid-gas physical properties such as viscosity, density and surface tension influence the two-phase flow condition, which then influences the pressure drop (Choi et al., 2008). For instance, the high density of the liquid may be one of the factors contributing to the higher pressure drop when the leak is situated in the liquid phase. The gas-liquid volume fraction distribution for the leak at the gas-phase, liquid-phase and interface of the two phases are examined using contour plots at 2.5 m away from the pipe upstream. Figure 5.6 shows the responses of fluids fraction for the same leak scenarios as in Figure 5.5(b). The absence of leak shows that the void fraction and liquid holdup is nearly uniform with the clear interface between the liquid and gas phase as previously observed in Figure 5.5(a) and (b) for the pressure profile and flow rate responses, respectively. However, Figure 5.6(b) shows that the occurrence of a leak at the gas phase attracts liquid moving from the bottom of the pipeline toward the leak region. Figure 5.6(c) and (d) present the fluids saturation for the leak event at the gas-liquid interface and liquid phase. The occurrence of a leak at the gas-liquid interface allows air to diffuse into the water as both phases discharge simultaneously from the pipeline.



(a)



(b)

Figure 5-5: Effect of axial leak positions; (a) medium size, (b) large size. (Pressure distributions (left) and flow rate (right). Note that the flow rate represents the total flow rate for the two-phases.

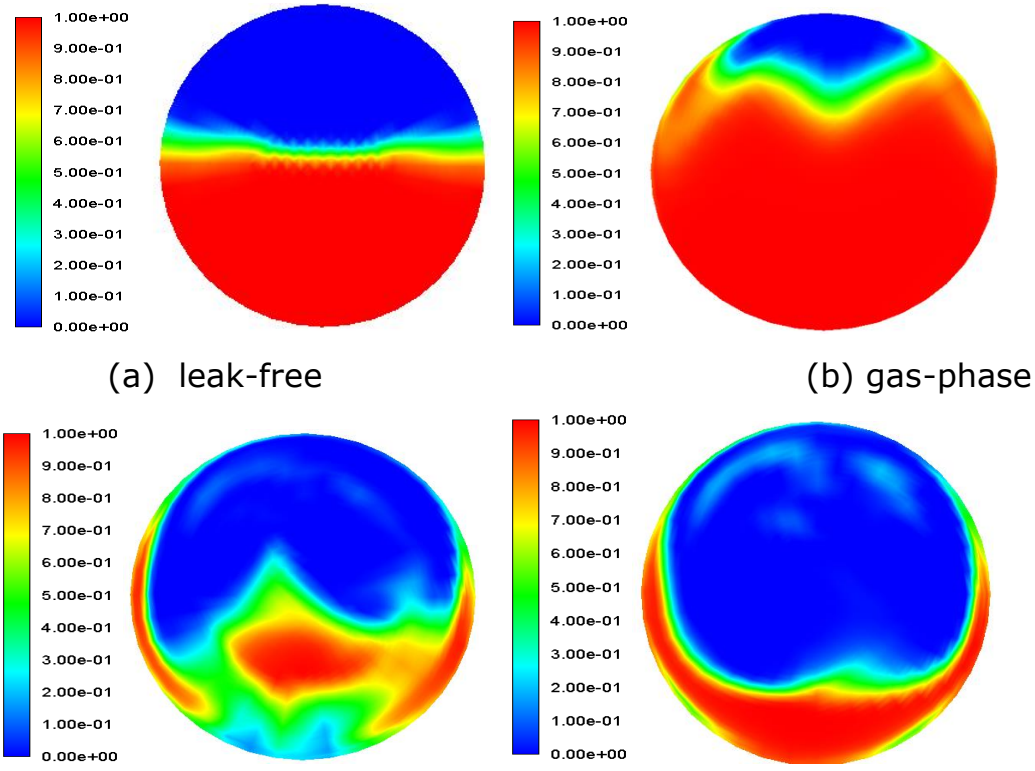


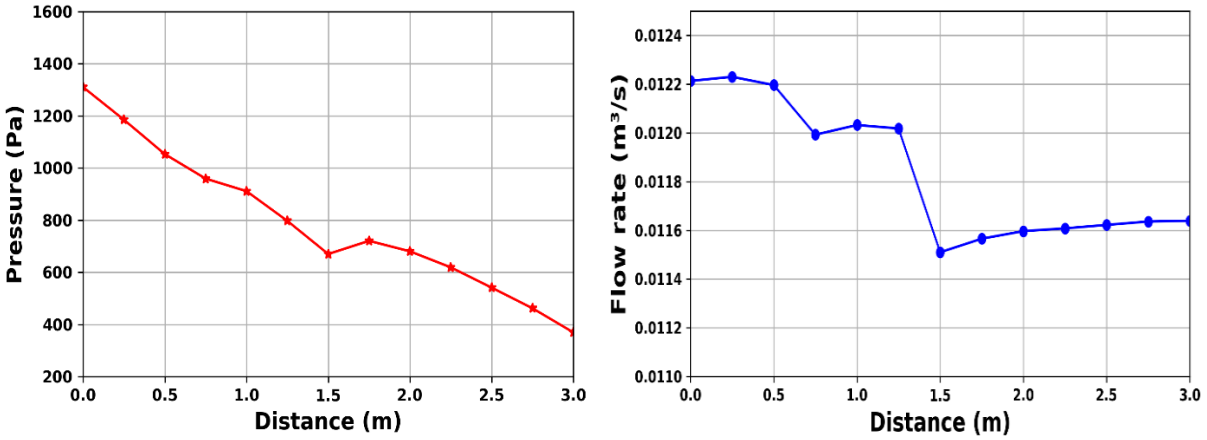
Figure 5-6: Volume fraction contour plots at 2.5 m for the leak at different axial positions. (Red and blue colours indicate water and air, respectively. The leak is located in the middle of the pipeline).

5.5 Multiple Leakages Effect Analysis

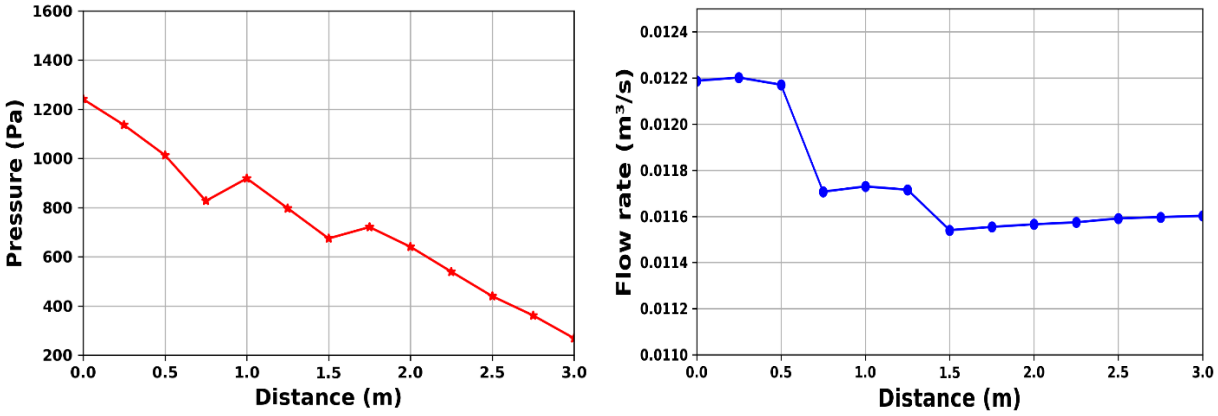
The emergence of double leaks on a single pipeline can easily affect the accuracy of detecting pipeline leakage. Therefore, the investigation of multiphase flow in the pipe with multiple leaks plays a crucial role in accurately determining the leaks' size and identifying the pipeline leakage location. The impact of double leaks on pipeline leak detection and localisation has been considered and analysed in this study. Figure 5.7 illustrates the pressure gradients and the flow rates in various multiple leak scenarios. The first leak location is set at 0.75 m away from the pipe upstream, while the second leak is located at 1.5 m, which is the mid-point of the pipeline. The two leak sizes are chosen among small, medium and large. The second hole is chosen to have a medium size in all scenarios. Figure 5.7(a) shows the double leak scenario where the first leak has a small size. The flow responses behave significantly differently with different leak sizes. The pressure drop for the medium leak size is more significant

than that of the small size. It is observed that a small leak position at 0.75 m is difficult to locate if the pressure profile is employed as an indicator for detecting or locating the leak position.

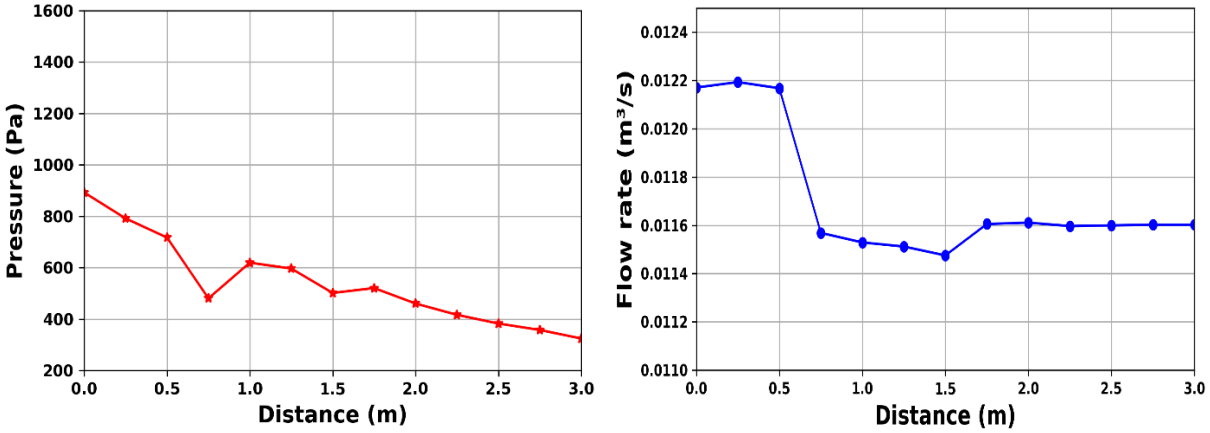
Figure 5.7(b) illustrates low-medium leak scenarios with equal (medium-medium) leak sizes. The system responses show that the emergence of the second leak does not cause significant effects on the pressure drop compared to the leak closer to the upstream of the pipeline. A leak closer to the pipe upstream always results in higher drop in pressure and flow rate than the second leak. Similar responses are also observed in Figure 5.7(c) for the leak scenario with the large-medium leak located at 0.75 m and 1.5 m away from the upstream of the pipe, respectively. There are two major observations from the double leak scenarios: Firstly, when there are two leaks with different leak sizes, the large leak easily masks out the small one. This is because more fluid tends to escape through the large opening size. Therefore, it causes an increase in pressure drops around the large leak region. Secondly, in the event of double leaks with equal size, a leak closer to the pipe upstream has a dominant effect on the flow. This could be linked to higher pressure in the upstream section of the pipe, leading to a more significant loss on the leak closer to the upstream of the pipe.



(a) low - medium sizes



(b) medium - medium sizes



(c) large - medium sizes

Figure 5-7: Effect of double leaks with different leak sizes. Pressure distributions (left) and flow rate (right).

5.6 Summary of Findings

A comprehensive assessment of multiphase pipeline leakage on a three-dimensional pipeline for different leak sizes, longitudinal leak locations, axial positions and simultaneous occurrence of leakages were performed using the CFD model. The simulation results showed that numerical simulation could help compile a set of guidelines for conducting prior leak assessment and contingency planning for accidental leakage of the pipeline. It was found that when a pipeline leakage occurs, the fluids flow parameters experienced a fluctuation, particularly within the vicinity of the leak regions, which makes it possible to detect and locate the leak position. Leak size significantly impacts the amount of fluids discharged through the leak region, which increases with the leak size. The flow parameters investigated as possible leak detection and localisation indicators are pressure drop, flow rate and volume fractions. In all cases studied, it was observed that the outlet flow rate is better for leak detection if the flow transducer is considered as an indicator for pipeline leak detection. However, upstream pressure is preferred if the pressure transducer is used as a pipeline leak detection sensor. The volume fractions are believed to be effective for quantifying the leak sizes in the multiphase flow system.

Chapter 6 Application of Surrogate Model for Pipeline Leakage Detection

6.1 Introduction

As discussed in Chapter 1, machine learning algorithms have demonstrated a suitable approach to predicting pipeline leakage. However, the training phase of the machine learning algorithms typically requires a large dataset. The cost of collecting physical experiment data is enormous. In contrast, the numerical model, which served as an alternative approach is computationally expensive. One simulation can take days or weeks despite the advances in high-performance computing. Therefore, the developed surrogate model is implemented to optimise the dataset for pipeline leakage detection and characterisation.

This chapter extends the application of the surrogate model constructed in Chapter 3 to engineering problems by applying it to pipeline the leakage prediction model. Adaptive surrogate model and numerical modelling of pipeline leakage discussed in Chapter 3 and Chapter 4, respectively are combined to optimise the training dataset by generating essential sample points in the most prominent locations in the parameterised design space. The algorithm is implemented on single-phase and multiphase models, and the performance is compared with the conventional space-filling techniques and experimental data reported in the literature.

6.2 Methodology

The developed surrogate model (PSOASM) is applied to the three-dimensional pipeline leak detection to evaluate its capability for the engineering test case. The objective is to develop a leak prediction model $f^*(X)$ that approximate a function $f(X)$ for the parameter space $X \in \{x_1, x_2, \ldots\ldots, x_m\}$, based on a limited number N_s of function evaluations $\{f(X_i)\}, i = 1,2, \ldots\ldots N_s$. The computational domain is a 3-D pipeline, as shown in Figure 4.2(a). When creating the surrogate model, the geometry of the leakage is allowed to vary, such that the pipe leak size and location define each simulation condition. At first, a set of initial training conditions $\{X_i\}, i \in$

$[1, N_s]$ was generated using the LHS strategy, and the corresponding outputs are simulated using fluent in ANSYS 18.1 tool. A coarse surrogate model is constructed using data obtained from the simulator. The surrogate model quality is then assessed using MSE approach. The generated sample points and their corresponding simulated data are archived in the database. An adaptive training process begins when the prediction quality of the initial constructed surrogate model is not acceptable.

The surrogate model is used to update the existing sample in parameter space, while PSO is employed to find the optimal data point for model accuracy maximisation. The initialisation of PSO is performed using the initially generated samples to form the PSO population and then calculate the personal best for each particle and swarm global best position. The point with the maximum distance to the other particles is selected as a new sample point. The CFD simulation is performed for the new point sampled, and the simulated data is added to the existing training sets, which are then used to retrain the MLP. The fitness of the added sample is assessed by comparing the accuracy of the newly trained model with the previous one using MSE. The updated surrogate and swarm are archived in the database. If the newly trained model's accuracy is higher than the previously trained model, further exploit the newly added sample region. The surrogate training process is designed to terminate if the MSE value is smaller than or equal to 0.04 MSE or the maximum number of iterations is equal to 100. The overall framework of the PSOASM is presented in Figure 3.1.

6.2.1 Problem formulation

The problem formulation of the surrogate model constructed for the pipeline leakage detection and characterisation is described by considered $\mathbb{R}^N$ as parameter space comprises different pipeline leakages spanned by pipe leak sizes and longitudinal leak locations. N denotes the dimensionality of the parameter space, which is two in this study. Let a vector X represents the leak scenarios within the parameter space $X = \{x_1, x_2, \ldots, x_m\}^T$ such that $x_m \in \mathbb{R}^N$. The numerical simulation of X that is computationally expensive is denoted as $y = f(X); f: \mathbb{R}^N \to \mathbb{R}$. To develop a pipeline leak detection model

using machine learning algorithm, a large data set with thorough data space coverage $y = f(X)$ is required, which is computationally costly. Therefore, an approximated surrogate model is needed to minimise the number of simulation trials without sacrifices model fitting accuracy. The surrogate model $y = g(X)$ which is approximation of $y = f(X)$ is developed based on the selected input-output data samples. This problem can be formulated as follows:

Given function:

$$f: \mathbb{R}^N \rightarrow \mathbb{R}^{\cdot} \tag{6.1}$$

find

$$X \in \mathbb{R}^N s.t. g(X) \leq f(X), \forall X \in \mathbb{R}^N \tag{6.2}$$

For the target function f defined in $X \in \mathbb{R}^N$, the initial sampling process begins with a set of sample pairs $(X_i, f_i), i = 1,2,3,\ldots\ldots,I$ with valid bounds $X^L \leq X \leq X^U$ where X^L and X^U represents the lower and upper bounds of parameter space, respectively. The maximum number of dataset sizes for which the desired accuracy of the developed surrogate model obtains is denoting $g(X), i = 1,2,3,\ldots\ldots,I_{max}$. In this study, the approximation model $g(X)$ is implemented using Multilayer Perceptron using the training procedure presented in (Golzari et al., 2015). The summary of the algorithm methodology is presented in Algorithm 6.1.

Algorithm 6.1: The main steps of the algorithm methodology:

Step1: Generate a set of initial sample points $\{X_i\}, i \in [1, N_s]$ in the design space using (LHS) to fill the entire domain evenly.

Step2: Evaluate the generated points using the pertinent objective function defined in equation (3.3), and run CFD simulation to obtain the response values $Y(X_i)$. Archive the generated samples and their corresponding simulated values into the database.

Step3: Construct the coarse surrogate model $\{f(X_i)\}, i = 1,2,\ldots\ldots N_s$ for the initial dataset store in the database.

Step4: Initialise the PSO for optimisation. Use the initial generated data points to form initial PSO population $pop(t)$ and determine the initial velocities particles v and initial positions of each particle.

Step5: Generate a new population $pop(t+1)$, using the basic behavioural learning process described as follows:

Assuming the sample set (particles) $X_N = \{x_1, x_2, \ldots\ldots, x_N\}$ generated from the previous iterations. Find the region with less population density and add a new sample X_C using crowding distance measures defined in equation 3.3.

Step6: Perform CFD simulation for the new point X_C and obtains its corresponding values Y_{new}.

Step7: Update the training dataset by refining the sample set as $X_{N+1} = X_C \cup X_N$ and the corresponding response values $Y_{N+1} = Y_{new} \cup Y_N$.

Step8: Retrain the surrogate model use updated dataset X_{N+1} and Y_{N+1} and archive the updated surrogate and swarm into the database.

Step9: Evaluate the fitness of the added sample by comparing the updated surrogate model with the previous surrogate response using MSE.

Step10: If new global position improved surrogate model accuracy, update the global best of the swarm.

Step11: Terminate the algorithm if the stopping conditions are met and output the surrogate model. Otherwise, increment the iteration and go to Step 12. The termination conditions are the maximum MSE is smaller than 0.04 or maximum number of iterations is equal to 100; there were chosen based on results from trial runs.

Step12: Update the velocities and positions of the particles by using equations (3.10) and (3.12), respectively. If the fitness

evaluation performed in Step 9 shows surrogate model improvement, go to Step 6. Otherwise, go to Step 5.

6.2.2 Physical model and numerical approach

The basic physical pipe used in this chapter is the same as that of the geometry in Figure 4.2(a). The geometry was scripted using ANSYS SpaceClaim. The pipe diameter is 0.06 m, while the pipe length is 50 times the diameter. The pipe leakage conditions were made to be circular opening size based on the IOGP recommended hole size distribution for subsea pipelines (Li et al., 2018). The leak diameter varies up to two-thirds diameter of the pipe. The ANSYS 18.0 is employed to generate the grids and perform numerical simulations. The details of the numerical simulation method are presented in Chapter 4. These include the boundary conditions setting, the incoming flow conditions, the setting of the turbulent model, and the grid-independent verification. Both single-phase and multiphase flow are considered for the test cases. The single-phase operational fluid is water, while the multiphase fluids are water and air. The physical properties of the fluid phases are presented in Table 4.2.

6.3 Result and discussion

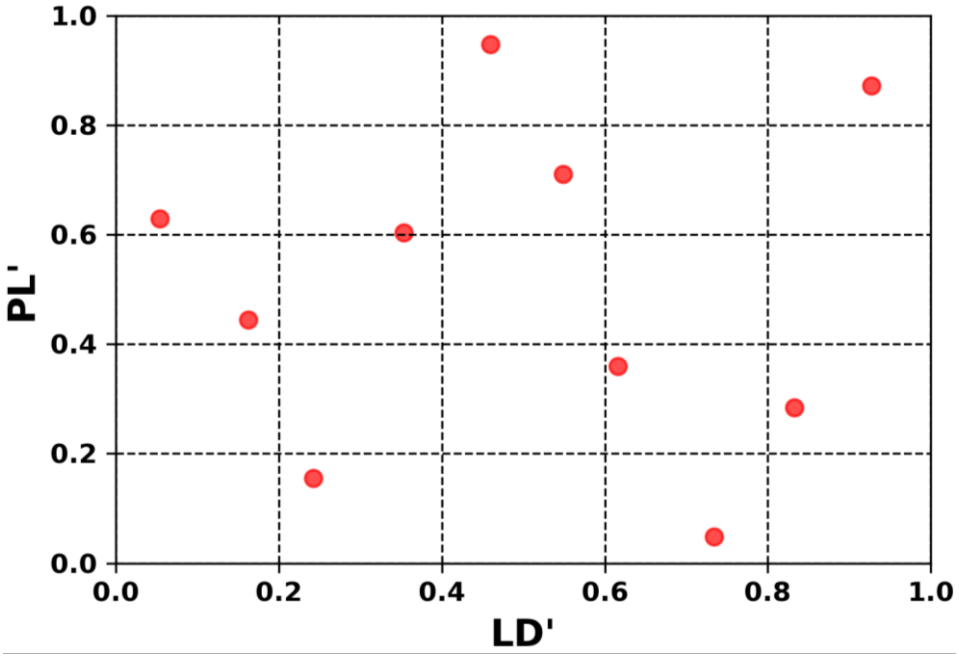
For the analysis conducted in this chapter, the initial value of the surrogate model is considered as 10, based on the numerical test conducted in Section 3.8.2. The parameters of the PSO algorithm were the same as that of the parameters setting presented in Section 3.8.1. MLP is used to construct the surrogate model, and the parameters of MLP are the same as those in table 3.1.

6.3.1 Single-phase Results

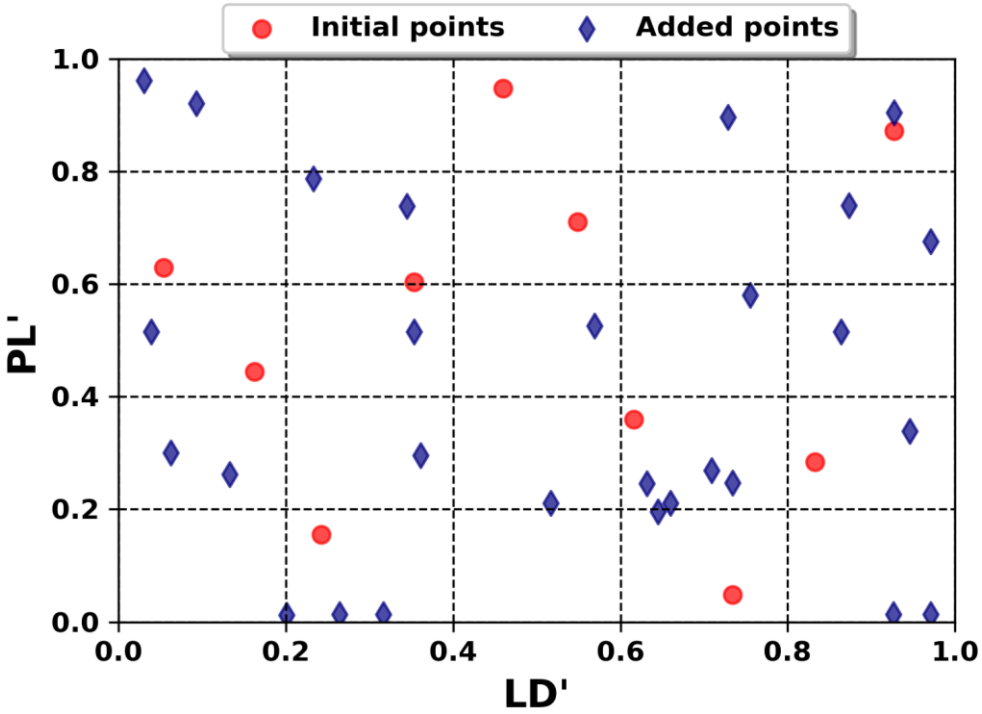
Figure 6.1 illustrates the initial and adaptive added training sample points methods on a domain of two-dimension defined as Leak Diameters (LD) and Leak Positions (LP). Please note that the LP' and LD' are normalised leak position and diameter, respectively. Figure 6.1(a) shows the initially generated sample points using LHS, and Figure 6.1(b) is the combination

of initial sample points and adaptively added points. The surrogate model is initially constructed using the sample in Figure 6.1(a). Then, the new data points are generated using the surrogate fitness value and population density of the existing sample points described in Algorithm 6.1. The fluid flow parameters, namely pressure and flow rate commonly used in the open literature to describe the pipeline leak location and sizes are computed using the CFD simulator. The data from the simulator (pressure and flow rate) and sample locations are input and output data used to construct the surrogate model. The statistical summary of the dataset used for training the single-phase surrogate model is presented in Appendix C.

The convergence profiles of the constructed surrogate model is shown in Figure 6.2, where the y-axis shows the fitness value of the model, the x-axis provides the number of iterations, also known as training sample sizes. The number of training points varies from 0 to 100. However, the algorithm converged at 40 training data points (i.e. 10 initial points plus 30 points added iteratively). It is important to highlight that surrogate accuracy (MSE) at the zero data trained size indicates model performance for the initial sample sets, which is 10 in all experiments carried out in this study. This value was selected based on the numerical test conducted in Section 3.8.2. The model fitness value improvement increase as the training sizes increase up to 35 (25 additional) sample sizes. Further adding more training points after the 35-training point appeared to contribute insignificantly to the surrogate model's prediction accuracy.



(a)



(b)

Figure 6-1: Illustration of sample points: (a) initial generated sample points using LHS, (b) initial and adaptively added points. (The red dots is initial samples, adaptive added samples in rhombus dark blue)

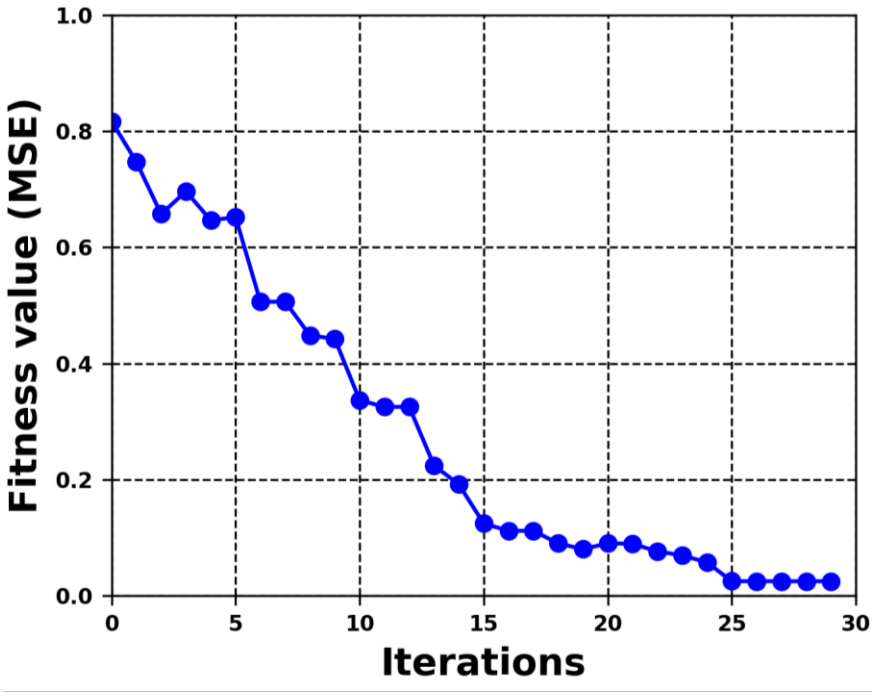


Figure 6-2: Convergence profiles of the constructed surrogate model (PSOASM) for single-phase pipeline leak detection.

6.3.1.1 PSOASM performance evaluations on the unknown dataset

One of the surrogate model's major features, developed for predicting the event's occurrence, is its robustness in predicting the event that it has never been exposed to. The performance of the developed surrogate model is evaluated using sets of data other than those used for the training phase. In this case, four new sets of uniform data were generated, named samples A, B, C, and D. The sample size for the A, B, C, and D samples are 25, 36, 64, and 121, respectively. The CFD simulations were performed for these samples, and the simulated data were used for the PSOASM evaluations. Four surrogate accuracy metrics, namely MSE, RMSE, R^2, and MAE described in Section 3.7, are employed for the evaluations. These accuracy metrics are used to measure how good the model is on new data. Table 6.1 gives the results of the model evaluation on new data. The bold entries indicate the minimum and maximum values of the error for each metric. Overall, the model performs well in all the data tested. Up to 98%, 90%,

93% 91% accuracies are obtained for MSE, RMSE, R^2, and MAE, respectively shown in Table 6.1.

Table 6-1: Results of performance evaluation on external data, for the 25, 36, 64 and 121 sample points.

Metrics	Sample A	Sample B	Sample C	Sample D
MSE	0.088	**0.125**	**0.012**	0.082
RMSE	0.298	**0.354**	**0.104**	0.285
R^2	**0.895**	0.932	0.913	**0.933**
MAE	0.171	**0.293**	**0.091**	0.218

Moreover, individual leak size and location were tested and analysed for the robustness and accuracy of the developed PSOASM model. Random leak conditions are simulated in ANSYS fluent, and the pressure and flow rate values obtained from the simulation tool are used to predict the true leak size and location using the PSOASM model. The true leak conditions are then compared with the computed value of the PSOASM, as shown in Table 6.2 and Table 6.3 for the leak locations and sizes, respectively. The bold entries in the tables denote the minimum and maximum percentage error computed for the true leaks and predicted values of the PSOASM. As shown in Table 6.2, the minimum and maximum percentage errors of 0.18% and 8.09%, respectively, are obtained for the leak location. While that of leak sizes presented in Table 6.3 gives the minimum and maximum percentage errors of 0.11% and 5.59%, respectively.

Table 6-2: Correlation between the leak locations of the transient model and predicted values by PSOASM

True leak location (m)	Predicted leak location (m)	Percentage error (%)
1.1011	1.0701	2.8154
1.7753	1.7612	0.7942
0.6515	0.6639	1.9033
1.3258	**1.2184**	**8.0932**
2.0002	1.9766	1.1799
1.8503	1.8163	1.8375
0.8263	0.8402	1.6822
2.0751	2.0476	1.3252
1.4007	1.4546	3.8481
1.2509	**1.2487**	**0.1759**
0.1520	0.1572	3.4211
1.5756	1.6029	1.7327
0.2269	0.2177	4.0547
0.4642	0.4403	5.1486
2.1500	2.1318	0.8465
1.4757	1.4972	1.4569
1.1385	1.1517	1.1595
0.8013	0.7861	1.8969
1.7854	1.7402	2.5316
1.3708	1.3588	0.8754

Table 6-3: Correlation between the leak sizes of transient model and predicted values by PSOASM

True leak size (mm)	Predicted leak size (mm)	Percentage error (%)
6.6830	6.6101	1.0908
1.1000	1.0590	3.7273
4.4500	4.3070	3.2135
3.3330	3.2610	2.1602
5.5670	5.4460	2.1735
7.8000	7.5240	3.5385
2.9940	3.0280	1.1356
7.1720	6.9420	3.2069
2.9387	**3.1030**	**5.5909**
1.6230	1.6680	2.7726
6.9460	6.8160	1.8750
5.1200	4.9390	3.5352
5.9500	**5.9571**	**0.1193**
2.3560	2.3830	1.1460
1.9130	1.9620	2.5614
3.8310	3.7120	3.1062
5.5820	5.7010	2.1319
7.7470	7.3490	5.1375
4.0100	3.9560	1.3466
5.9500	5.8200	2.1849

6.3.1.2 PSOASM evaluation using experimental data

The developed PSOASN model was evaluated using experimental data obtained from Molina-Espinosa *et al.* (2013), van der Walt *et al.*(2021) and Noguera-Polania *et al*. (2020). These data contain the flow rate and pressure measurements obtained from the pipes that are made up of different leak sizes. The experimental pipe of van der Walt *et al.* had a length of 65 m with a diameter of 25 mm, while the Molina-Espinosa *et al.* had a length of 2.23 m with a diameter of 12.7 mm. The Noguera-Polania *et al.* (2020) flow loop was equipped with a steel pipe of 76.2 mm diameter and 54 m length. Pressure measurements were taken at five intermediate points (P1 to P5) and a mass flow sensor was installed at the inlet of the pipeline. The fluid used for the three experiments was water. Figures 6.3, 6.4 and 6.5 shows the regression plots of the true leak sizes versus predicted leak sizes using PSOASM for the Molina-Espinosa *et al.* (2013), van der Walt *et al.*(2021) and Noguera-Polania *et al*. (2020), respectively. The correlations and linearities of the true leak sizes and predicted leak sizes (R^2) are 0.998, 0.986 and 0.975, respectively for Figure 6.3, Figure 6.4 and Figure 6.5. In all three cases, the slopes are close to 1 and the y-intercepts are close to 0. This shows the discrepancy between predicted values and true leak size is small. The PSOASM model predicted the leak sizes in Figure 6.3 with the error of 6.9% and 10.2%, respectively for the MSE and standard deviation of prediction error while the correlation between the predicted leak values and experimental data of van der Walt *et al.* shown in Figure 6.4 had 1.3% and 2.5%, respectively for the MSE and standard deviation of prediction error. Similarly, as shown in Figure 6.5, the PSOASM predicted results are in good agreement with the experimental data of Noguera-Polania *et al.* with the error of 2.1% and 2.5%, respective for the MSE and standard deviation of prediction error. This indicates that most of the predicted leak values are agree with the experimental data values satisfactorily. Therefore, one can conclude that overfitting does not happen in the developed PSOASM model.

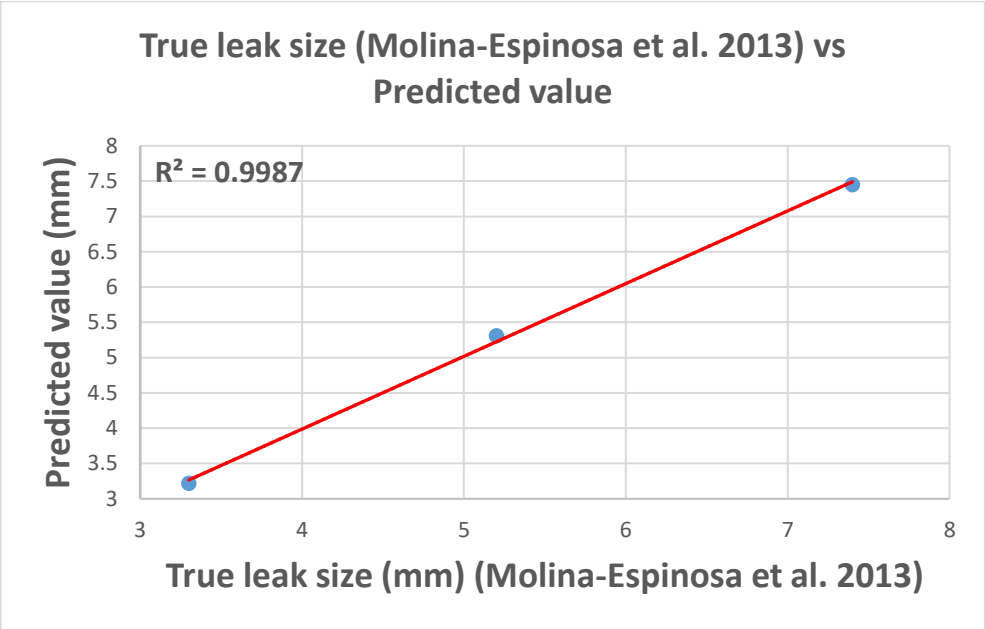


Figure 6-3: Correlation between the experimental data of Molina-Espinosa *et al.* (2013) and predicted values using PSOASM

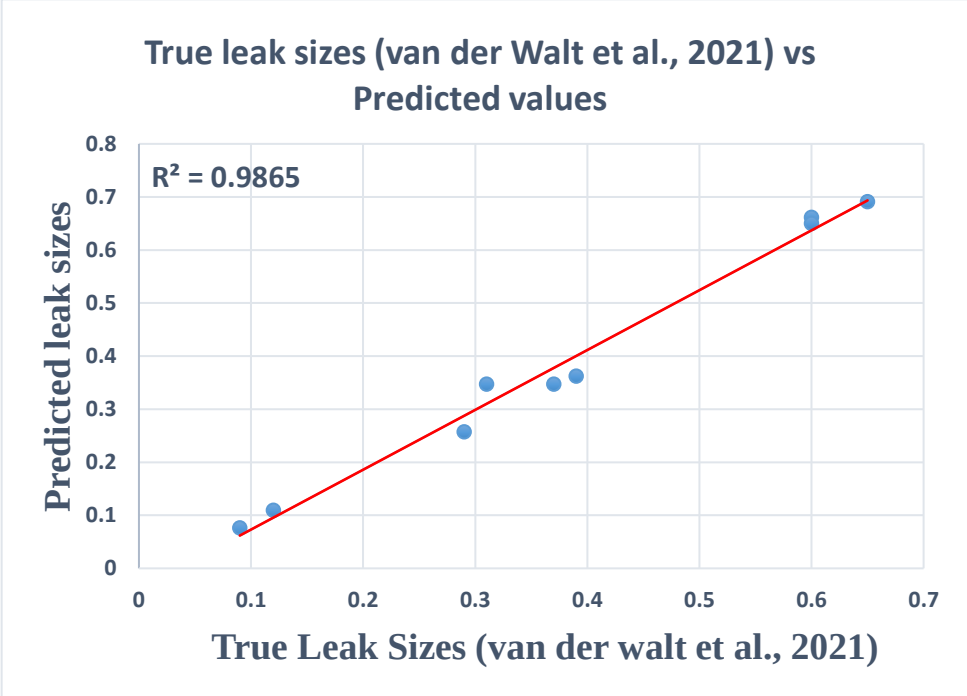


Figure 6-4: Correlation between the experimental of van der Walt *et al.* (2021) and predicted values using PSOASM.

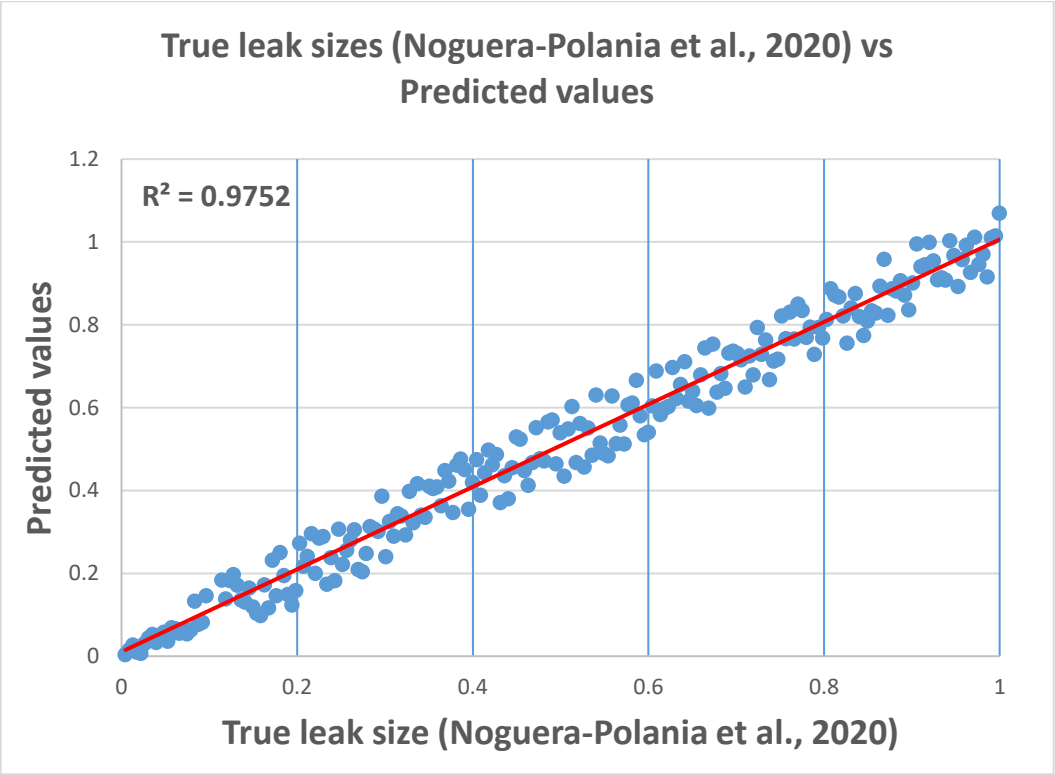
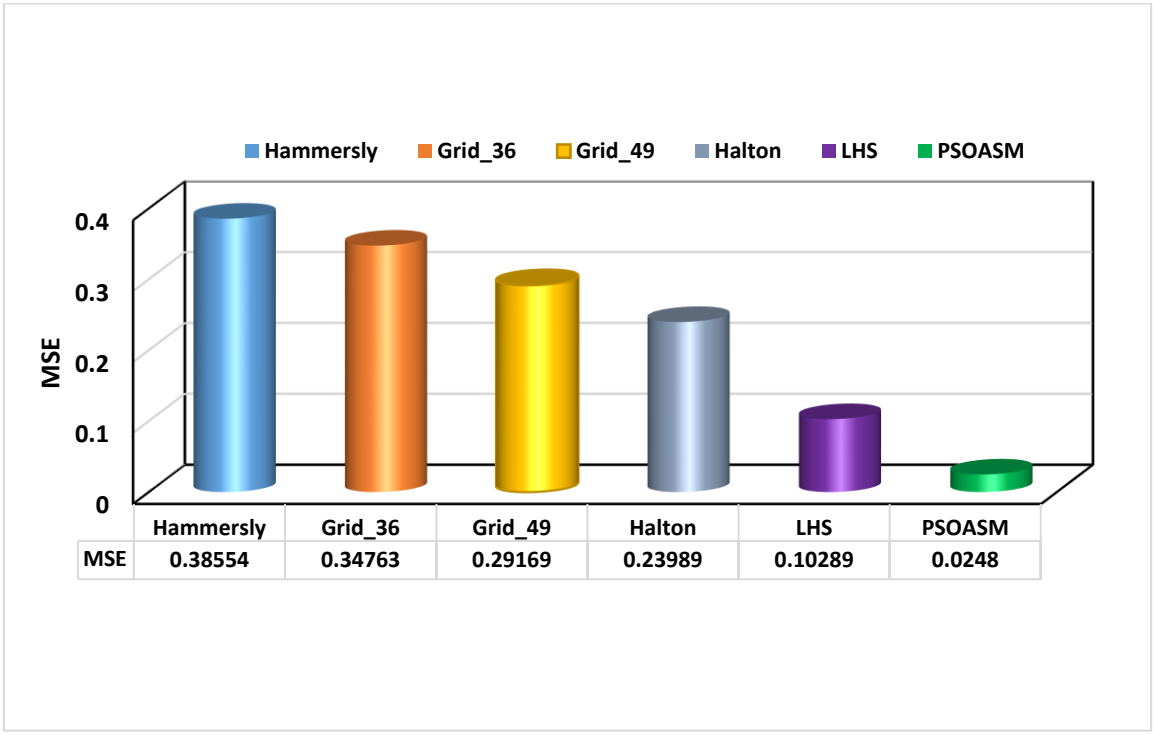


Figure 6-5: Correlation between the experimental of Noguera-Polania *et al.* (2020) and predicted values using PSOASM.

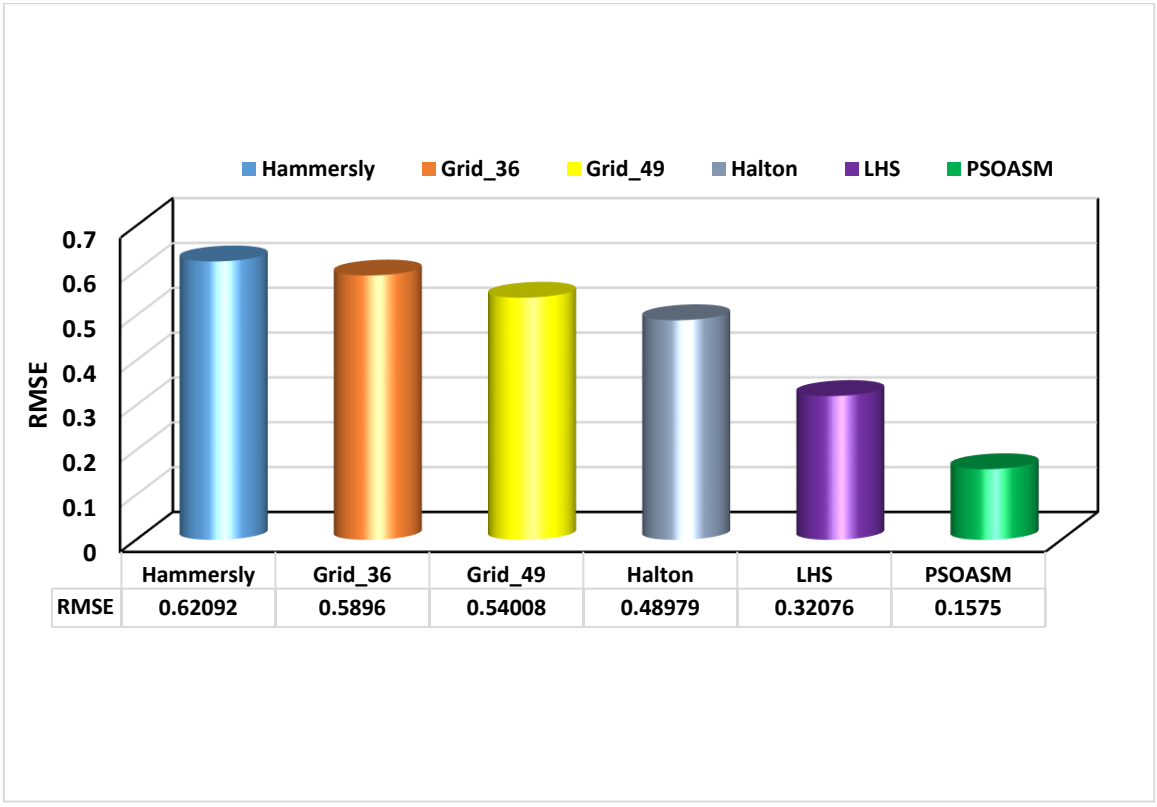
6.3.1.3 Performance comparison of PSOASM with conventional sequential sampling methods

To further examine the performance of the developed PSOASM on pipeline leakage detection and characterisation, four conventional sequential sampling approaches, including Hammersley, Grid, Halton, and LHS, were taken in the literature and compared with PSOASM using four performance metrics, namely MSE, RMSE, R^2 and MAE. The same training data points used for developing the PSOASM are generated using these sequential sampling methods for all the comparisons except the grid method, where 36 and 49 sample sizes are employed to guarantee uniform distribution. Results in Figure 6.6 illustrate the comparison results of the developed model with the different sampling methods compared. The MSE, RMSE, $R^2$and MAE obtained are presented in Figures 6.6(a), (b), (c), and (d), respectively. These results clearly show the robustness of the developed model to efficiently sample design domain with performance outperforming

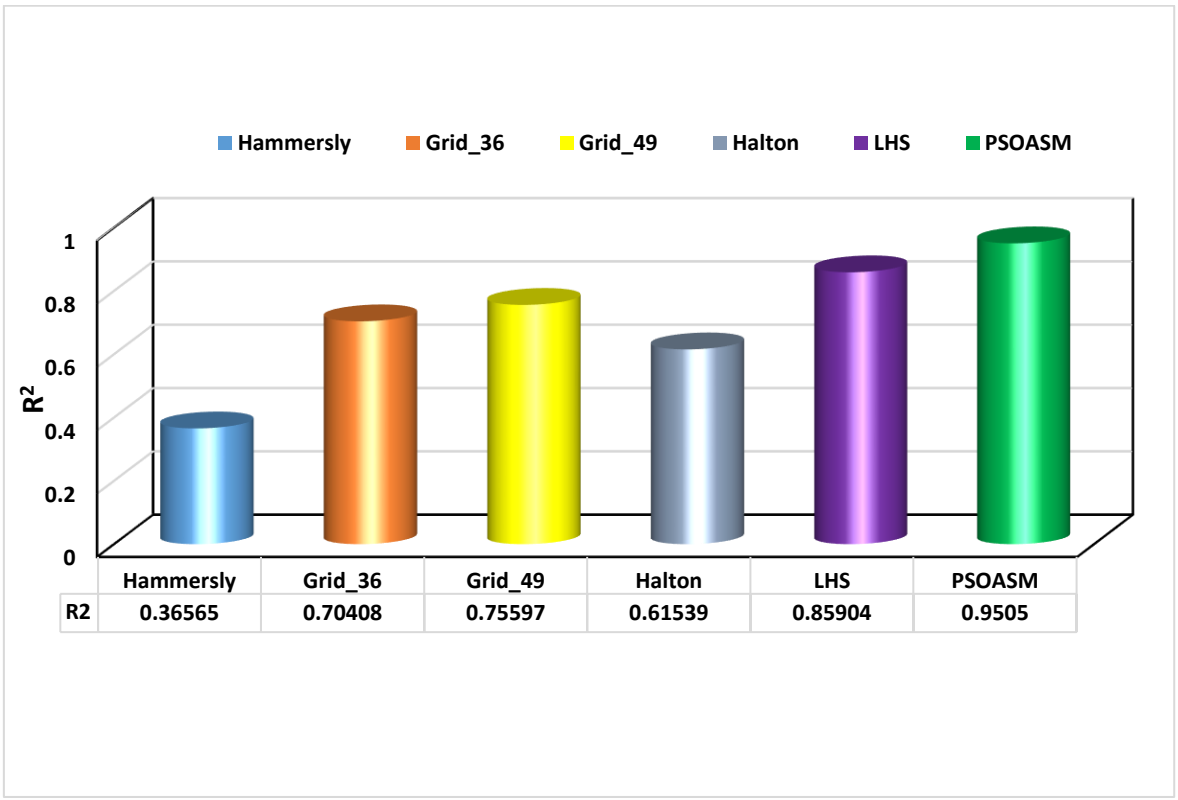
the conventional sampling methods by providing higher R^2 and lower MSE, RMSE and MAE. Thereby, it reduces the computational cost of the simulation model by reducing the number of surrogate training datasets and at the same time providing better prediction accuracy.



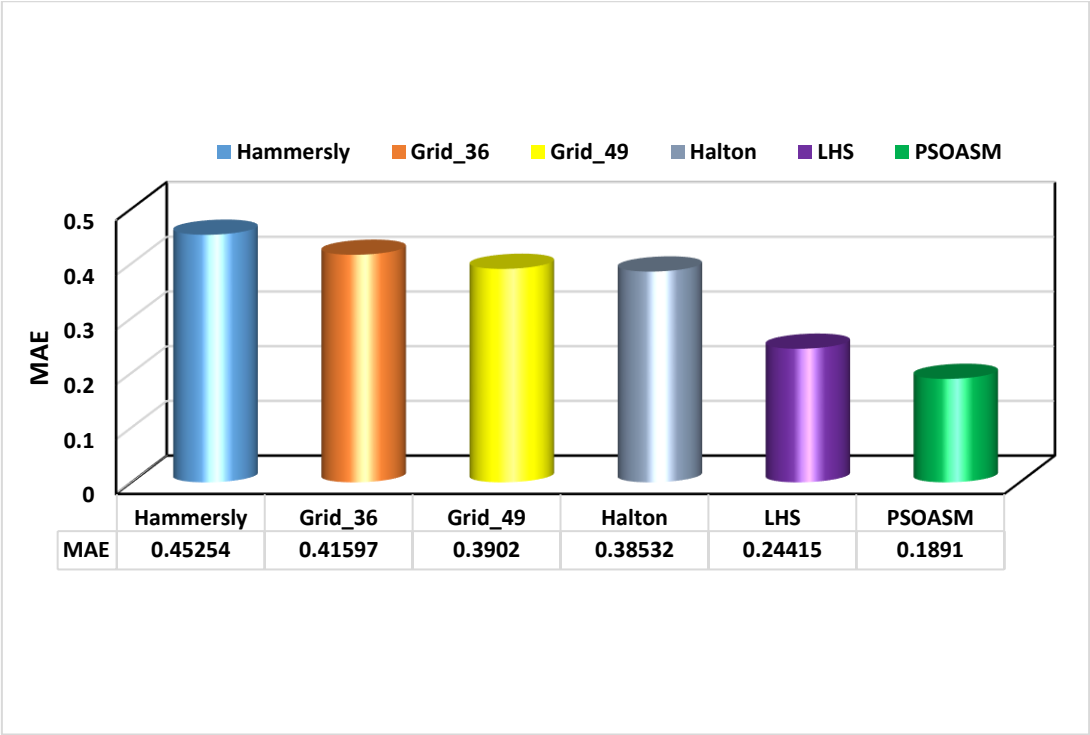
(a)


Hammersly
Grid_36
Grid_49
Halton
LHS
PSOASM
RMSE
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Hammersly Grid_36 Grid_49 Halton LHS PSOASM
RMSE 0.62092 0.5896 0.54008 0.48979 0.32076 0.1575

(b)


Hammersly
Grid_36
Grid_49
Halton
LHS
PSOASM
R2
1
0.8
0.6
0.4
0.2
0
Hammersly Grid_36 Grid_49 Halton LHS PSOASM
R2 0.36565 0.70408 0.75597 0.61539 0.85904 0.9505

(c)



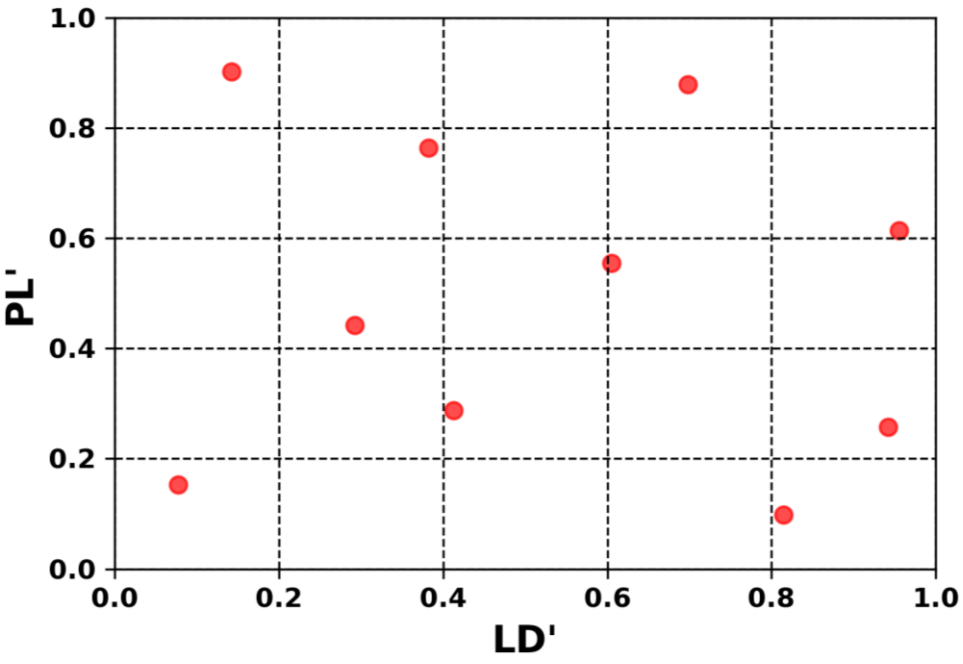
(d)

Figure 6-6: Comparison of PSOASM with the conventional sequential sampling approaches: (a) MSE, (b) RMSE, (c) R^2 and MAE results

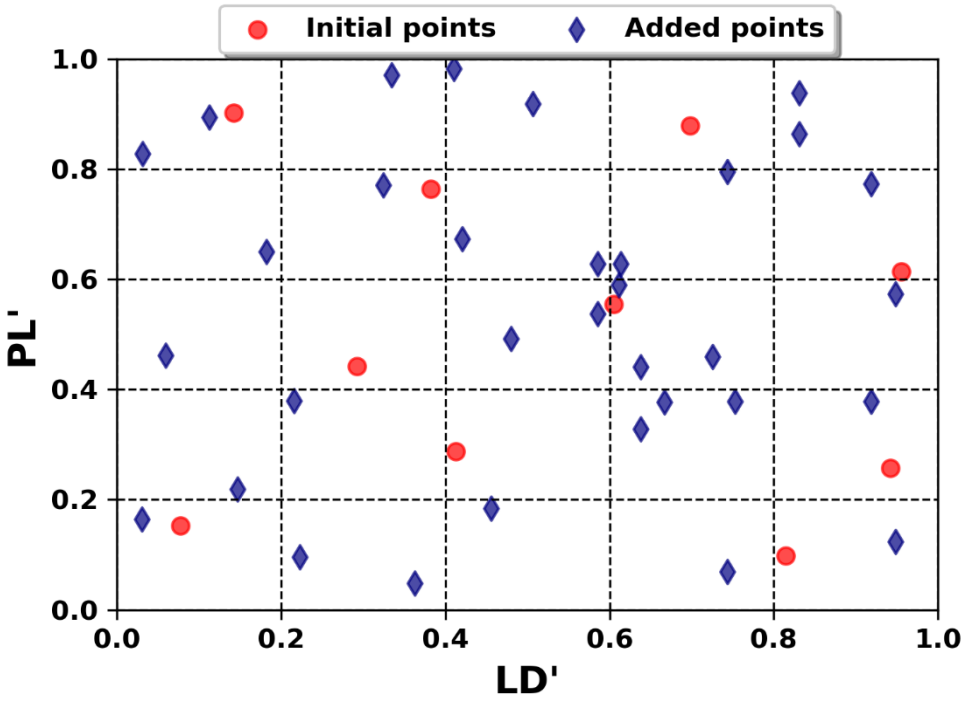
6.3.2 Multiphase Results

The results obtained using the developed surrogate model for multiphase pipeline leak detection and characterisation are presented in this section. Although the effectiveness of the proposed surrogate model has been demonstrated using a single-phase model. To further test the performance of the PSOASM on higher computational cost problems, two-phase gas-liquid pipeline leakage detection and characterisation described in Chapter 5 are used for the experimental study. The illustrations of the initial points and adaptive added training points defined as Leak Diameters (LD) and Leak Positions (LP) are shown in Figure 6.7, with the LP' and LD' representing normalised leak position and diameter, respectively. Figure 6.7(a) shows the initial training points generated using LHD, while Figure 6.7(b) illustrate the combination of the initial training points and the samples added in an adaptive manner as the algorithm grows. The CFD analysis of the data points generated was simulated using the CFD simulation. The output of the CFD simulations, which are pressure, flow rate, liquid holdup and void fraction, are considered surrogate input

parameters, while leaks size and locations were defined as output parameters. The statistical summary of the dataset used for training the multiphase surrogate model is presented in Appendix D. The sample points distribution shown around 0.6 LP' and 0.6 LD' in Figure 6.7(b) can be attributed to parameter space region that exhibits rapid changes or nonlinearity in the response surface due to the non-proportional changes in input to the output responses. The slower convergence profiles of the constructed surrogate model from 5 to 15 iterations in Figure 6.8 revealed the evidence of non-linearity around 0.6 LP' and 0.6 LD' in Figure 6.7(b). This indicates that the proposed method can identify a region requiring more simulation trials in the parameter space. The model was able to add more data points around 0.6 LP' and 0.6 LD' in the parameter space. The plot of the surrogate convergence profiles as the sample size increases is shown in Figure 6.8. The developed surrogate model converged after 43 training data points (i.e. 10 initial points plus 33 points added iteratively), and the fitness value (MSE) converged to 0.0344, satisfying the termination criteria.



(a)

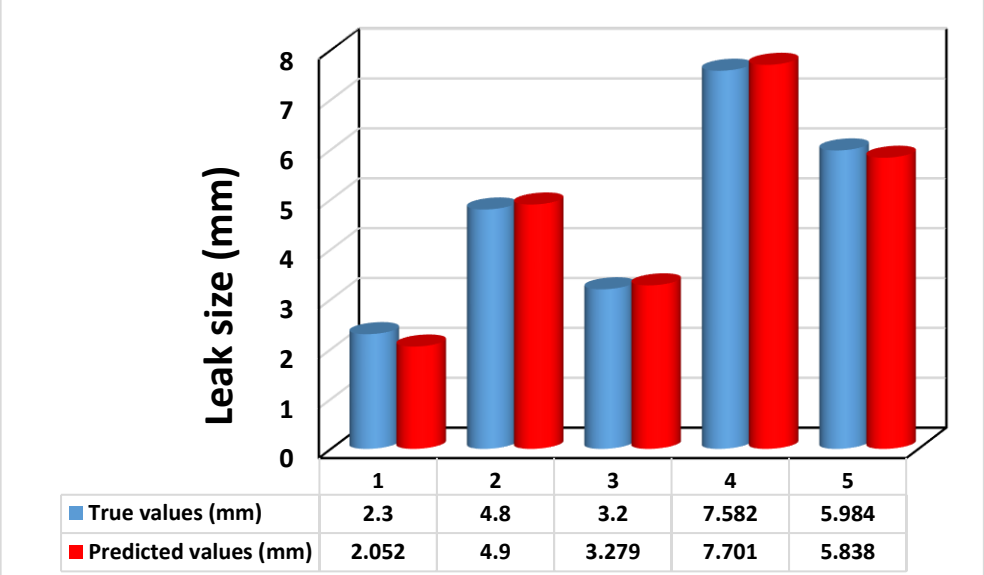


(b)

Figure 6-7: Illustration of sample points: (a) initial generated sample points using LHS, (b) initial and adaptively added points. (The red dots is initial samples, adaptive added samples in rhombus dark blue)

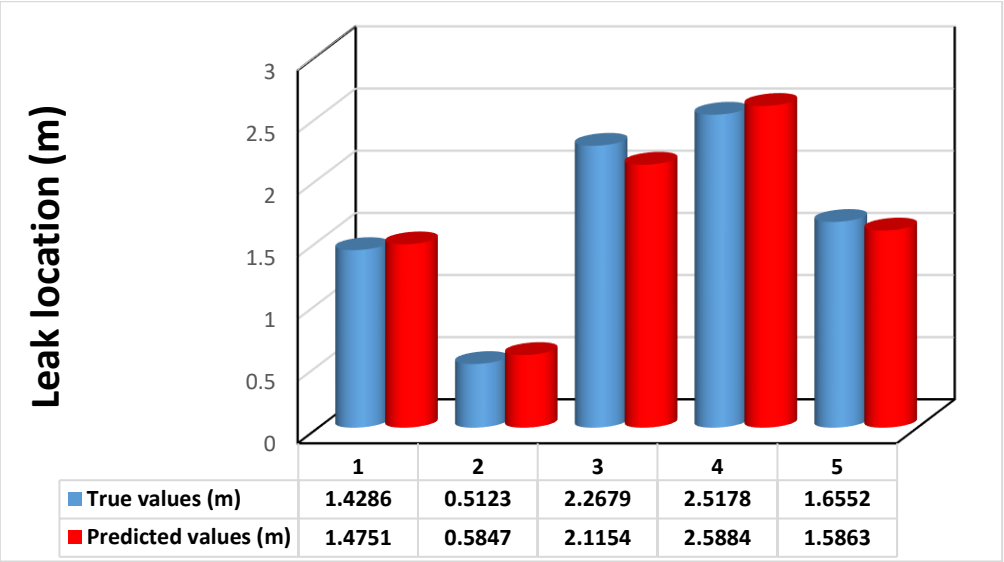
Figure 6-8: Convergence profiles of the constructed surrogate model (PSOASM) for multiphase pipeline leak detection.

In order to examine the prediction potential of the constructed surrogate model, new sets of data were randomly generated and then simulated using the CFD simulation. The output of the CFD simulation, which are pressure, flow rate, liquid holdup and void fraction, are fed into the developed surrogate model to predict the size and location of the leak. The comparison between the model prediction value and the ground truth is shown in Figure 6.9. Figure 6.9(a) illustrates the comparison between the predicted leak sizes using PSOASM and the true leak values simulated, while the comparison between the true leak locations and the predicted leak locations is presented in Figure 6.9(b). By comparing the true leak sizes and predicted leak values in Figure 6.9(a), it is possible to conclude that the developed surrogate model is not overfitting because the difference is not significant. The good performance of the developed surrogate model is also confirmed in Figure 6.9(b), where the predicted leak locations and the true values are much closer.



	1	2	3	4	5
True values (mm)	2.3	4.8	3.2	7.582	5.984
Predicted values (mm)	2.052	4.9	3.279	7.701	5.838

(a)



(b)

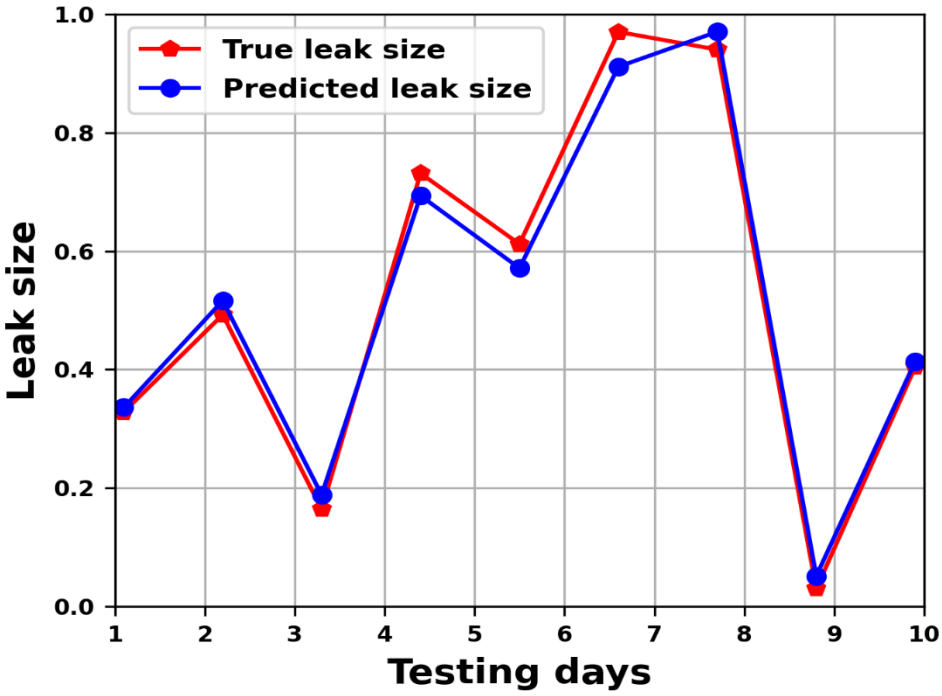
Figure 6-9: Correlation between the multiphase leak sizes of transient model and calculated values by PSOASM: (a) leak size, (b) leak locations

6.3.3 Near real-time pipeline leak detection and characterisation implementation

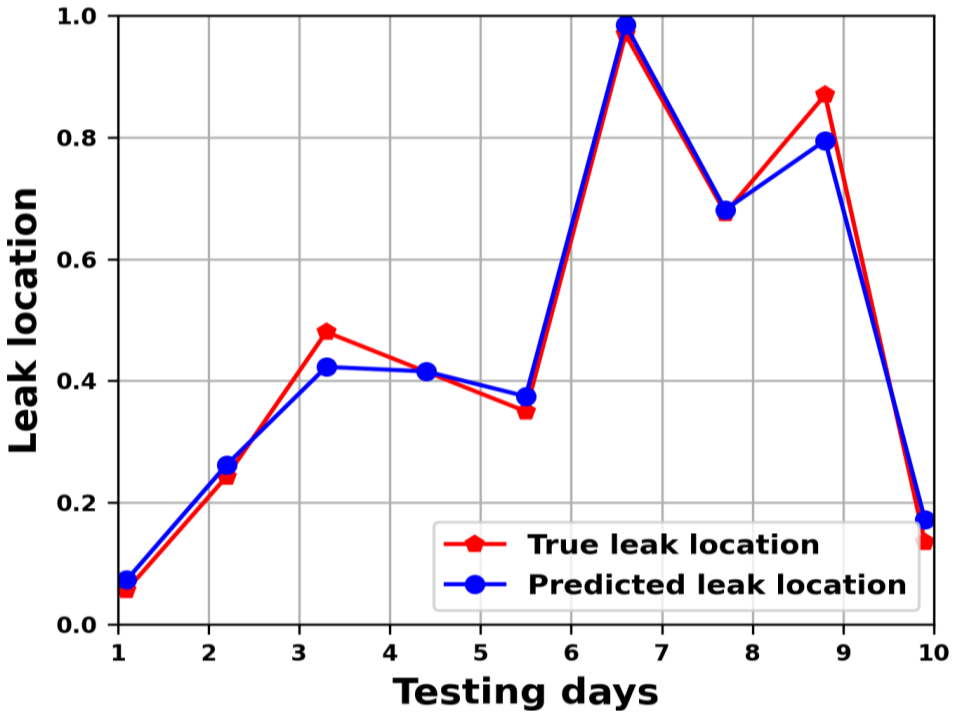
After testing the developed surrogate model on the data other than the dataset employed for the training process (section 6.3.1.1), the established connection between the surrogate model and pipeline leakage modelling developed in ANSYS 18.1 was allowed to continue monitoring the pipeline by inspecting the in-pipe flow parameters (pressure and flow rate). The simulation model extracts the pressure and flow rate at the interval of 2000 iterations, which implies 2 seconds for a time step of 0.001. The surrogate model was then applied to the extracted flow parameters (pressure and flow rate) to predict pipeline leak size and characterisation and display the predicted result on the spyder console depending on whether leakage detected or absence of leakage. It also displays the leak size and location in the event of leakage. The display function would be useful for the users to view the pipeline status vividly. The nine random leak sizes and characterisations generated using the random function in python were

modelled using the CFD model. The extracted pressure and flow rate from CFD were used for detecting pipeline leak size and location.

The comparison of the predictions and true leak sizes and locations are shown in Figure 6.10 (a) and (b), respectively. A good agreement between the predicted and true values was observed. This try-out emphasis is on the application of the surrogate model to real-time pipeline leakage detection and characterisation. In particular, pipeline engineers or users can use the surrogate model for detecting pipeline leakage. The surrogate model was tested in real-time in pipe fluid flow parameters serving as input features for predicting pipeline leakage and characterisation. Instead of simulation data, it is possible to use in-pipe flow data measure in the field, but it requires additional infrastructure for acquiring data and establishing communication leading to extra time and resources.



(a)



(b)

Figure 6-10: Near real-time leakage prediction comparison curves (a) leak size, (b) leak locations.

6.4 Summary

This chapter extends the application of the surrogate model constructed in Chapter 3 to engineering problems. The developed PSOASM is applied to single-phase and multiphase pipeline leak detections. The model was implemented to optimise the dataset for pipeline leakage detection and characterisation. The results obtained in this chapter have shown that the developed surrogate model can be applied to computationally expensive problems to reduce the large number of training datasets required to train machine learning algorithms without sacrificing the model accuracy. The trained algorithm for the single-phase mode converged to 0.0248 MSE after 40 training data points (i.e. 10 initial points plus 30 points added iteratively). Compared to the single-phase model, the multiphase model converged to 0.0344 MSE after 43 training data points (i.e. 10 initial points plus 33 points added iteratively). The developed surrogate model was tested on the new data using fluid flow parameters obtained from the CFD simulation. The minimum and maximum errors recorded for the surrogate

model developed for the single-phase pipeline are 0.012 and 0.125, respectively, using the MSE metric, while the minimum and maximum errors obtained using R^2 metric are 0.895 and 0.933, respectively. The performance of PSOASM is compared with the conventional sampling approaches. The comparisons show that the developed PSOASM outperformed conventional sequential sampling methods by providing the higher R^2 and lower MSE, RMSE and MAE under the same training data sizes.

Chapter 7 Conclusion and Future Work

This chapter brings together the main conclusions of the thesis and recommendations to further improve the proposed surrogate model. The model optimises training datasets in machine learning applications involving computationally expensive problems, like CFD simulation in pipelines.

This thesis contributes to the advancement of the state-of-the-art in adaptive surrogate modelling approaches for computationally expensive problems. In the proposed model, a novel data point selection strategy was introduced to place a new sample point in a region of high interest in parameter space. The proposed surrogate model was then optimised with swarm optimisation algorithm to find the optimal data points for model accuracy optimisation.

7.1 Conclusion

In this thesis, a novel data sampling optimisation method called adaptive particle swarm optimisation assisted surrogate model was proposed to train machine-learning applications involving computationally expensive problems like CFD simulation in a pipeline with a limited dataset. The proposed model incorporates the population density of training data samples and model prediction fitness to determine new data points to improving model fitness accuracy. The introduced model prediction fitness criteria can aid the proposed algorithm to exploit the parameter space, while the population density can reduce the chance of the model being stuck into the local optimum. The model was developed to automate the process without prior knowledge of the total number of training datasets required for developing machine-learning models. Moreover, it can maintain a good trade-off between exploration and exploitation by combining the global and local search strategies introduced in this thesis.

The proposed PSOASM are evaluated on five different machine learning algorithms and four sampling schemes using six benchmark problems. The results of the proposed method through extensive evaluations demonstrate

the use of PSOASM to reduce the large number of datasets required to train machine learning algorithms without sacrificing the model accuracy. The developed surrogate model performed well in all the machine learning algorithms employed for evaluation with up to 98% accuracy with an average 40 training data points in the benchmark problems tested. The comparison results and analysis demonstrate that the proposed PSOASM outperforms the conventional sequential sampling approaches by providing lower prediction error than conventional sampling methods employed for comparisons.

Systematic analysis of leak effect on multiphase pipeline system was performed to study fluids flow parameters, characterise multiphase pipeline leakage and develop pipeline leak model (simulator) for building surrogate model. CFD model was established to simulate different scenarios in which leak(s) may occur in a pipeline conveying more than one phase at a time. The VOF model and SST k-ω turbulence modelling scheme were applied to simulate the gas-liquid stratified flow in a horizontal multiphase pipeline. The effect of leak sizes, longitudinal leak locations, multiple leakages and axial leak position was investigated on the pressure gradient, flow rate and volume fractions in the multiphase pipeline. The simulation results showed that numerical simulation could help compile a set of guidelines for conducting prior leak assessment and contingency planning of accidental leakage of the multiphase pipeline. It was found that when a pipeline leakage occurs, the fluids flow parameters experience a fluctuation, particularly within the vicinity of the leak regions, which makes it possible to detect and locate the leak position. Leak size has a significant impact on the amount of fluids discharged through the leak region, which increases with the leak size. The results also reveal that when two leaks with different sizes co-occur in a single pipe, detecting the small leak becomes difficult if its size is below 25% of the large leak size. However, in the event of a double leak with equal sizes, the leak closer to the pipe upstream is easier to detect.

This thesis also considered the practical application of the proposed PSOASM. The proposed PSOASM was applied to 3-D pipeline leakage detection and characterisation. The implementation of PSOASM was performed on single-phase and multiphase pipeline leakages, and the performance was compared with the conventional sequential sampling approaches and experimental data reported in the literature. Based on the results obtained, the following conclusions can be drawn on the practical application of the proposed surrogate model:

- The proposed surrogate model is capable of finding optimal training data points for machine learning accuracy maximisation. It provides good prediction values for pipeline leak sizes and characterisation both in experimental and simulation data.
- The PSOASM outperformed the conventional sequential sampling approaches. This is due to the fact that in the developed surrogate model, a set of small dataset initially generated is growing through the selection of additional sample points in a region, which enhance the accuracy of the surrogate model.
- The proposed PSOASM provides a global representation of the search space by combining the global and local search strategies to maintain a good trade-off between exploration and exploitation in the simulation scenario parameter space.
- The proposed PSOASM not only allows for pipeline leak prediction with limited training samples but also provides a general framework for computational efficiency improvement using adaptive surrogate modelling in various real-time applications.

7.2 Recommendation for Future Research

This work has developed an adaptive surrogate modelling method that provides a framework for reducing computationally expensive problems such as CFD simulation in pipelines. The research study can be further extended in the following directions:

- In Chapter 3, numerical experiments conducted reveal that the developed PSOASM significantly performs better than the conventional sequential sampling approaches in all the analyses performed. Therefore, PSOASM proved to be a promising algorithm for addressing computationally expensive problems. The optimisation algorithm used in this study is PSO. Further study should compare the PSO with other meta-heuristic algorithms such as GA, SA and compare their performance. The developed PSOASM only consider parameter space as two-dimension in this study. In order to ensure the diversity of the PSOASM, it is an interesting study to evaluate PSOASM performance on the higher dimensional optimisation problems.

- In Chapter 4, the CFD model was developed to study the leak effect on multiphase pipeline systems and compared the modelled results with the experimental data reported in (Molina-Espinosa et al., 2013) to verify the boundary conditions of the computational field used for the study. Although, setting up a multiphase flow rig similar to the one presented in (Molina-Espinosa et al., 2013) is quite challenging in terms of cost, difficulty and time-consuming process. More research should be conducted on the actual multiphase rig to improve understanding of the leak effect on a practical scale. The model developed in this study is assumed to be isothermal and adiabatic. New possible research for multiphase pipeline leakage could be to look at the temperature effect on fluids flow parameters as a function of leakage.

- In Chapter 5, comprehensive simulations and assessment of multiphase flow behaviours induced by leaks were investigated, and the results agree with the previous study of Figueiredo *et al.*(2017) that concluded that a leak localisation strategy based on the upstream and downstream pressure profiles commonly employed in monophase flow pipeline leakage could be extended to the stratified-flow model. However, since the multiphase flow system spans beyond stratified flow patterns to better understand the leak effect in all the multiphase systems, comparisons of other multiphase flow regimes, such as bubble, slug, annular, etc., should be considered in future. The method of injecting gas-liquid in the computational domain in this study is a separate approach where gas is injected from the upper half-section of the pipe, while the liquid is injected from the bottom half cross-section of the pipe. Another method that involves the injection of the liquid into the pipe peripherally using power law velocity and gas with a uniform velocity profile in the centre region of the pipe should also be considered in future.

- In Chapter 6, the engineering application consider is 3-D pipeline leak detection and characterisation. In future work, other real-world optimisation problems, such as aerodynamic design optimisation, reliability optimisation of complex systems, streamline optimisation of the vehicles, etc., will be considered. The proposed method can be easily adapted to various engineering applications.

- The surrogate model developed in this thesis is a data-driven method that considers the simulator (CFD) as a black box. The simulation model used in the backend of ANSYS Fluent is physics-based. The simulation engine applies the finite volume method to discretise the PDE for numerical solution. A model-driven method, also called physics-based, should be considered in future work. However, this method requires access to the CFD simulation source code, which is generally difficult when using commercial CFD software. Still, it is

important, as it would indicate whether it is more beneficial in terms of computational time and accuracy.

- A semi-automated surrogate model development platform was constructed for building surrogate models for pipeline leakage detection and characterisation using ANSYS Fluent for CFD simulation. This platform can be modified to become automated with the help of advanced API provided by proprietary CFD simulator, ANSYS Fluent or COMSOL or open source CFD simulator like OpenFOAM.

- Collaborate with pipeline engineers to develop pipeline leakage detection and characterisation app using the proposed technology as a backend.

References

Ab Wahab, M. N., Nefti-Meziani, S., & Atyabi, A. (2015). A comprehensive review of swarm optimization algorithms. *PloS One*, *10*(5), e0122827.

Adegboye, M. A., Fung, W.-K., & Karnik, A. (2019). Recent advances in pipeline monitoring and oil leakage detection technologies: Principles and approaches. *Sensors (Switzerland)*, *19*(11). https://doi.org/10.3390/s19112548

Ahmadi, M. A. (2015). Developing a robust surrogate model of chemical flooding based on the artificial neural network for enhanced oil recovery implications. *Mathematical Problems in Engineering*, *2015*.

Akhlaghi, M., Mohammadi, V., Nouri, N. M., Taherkhani, M., & Karimi, M. (2019). Multi-fluid VoF model assessment to simulate the horizontal air–water intermittent flow. *Chemical Engineering Research and Design*, *152*, 48–59.

Al-Tameemi, W. T. M. (2018). *Two-phase flow in straight pipes and across 90 degrees sharp-angled mitre elbows*. University of Sheffield.

Alghurabi, A., Mohyaldinn, M., Jufar, S., Younis, O., Abduljabbar, A., & Azuwan, M. (2021). CFD numerical simulation of standalone sand screen erosion due to gas-sand flow. *Journal of Natural Gas Science and Engineering*, *85*, 103706.

Ali, I. T. (2017). *CFD Prediction of Stratified and Intermittent Gas-Liquid Two-Phase Turbulent Pipe Flow Using RANS Models*. The University of Manchester (United Kingdom).

Ameryan, A., Ghalehnovi, M., & Rashki, M. (2022). AK-SESC: a novel reliability procedure based on the integration of active learning kriging and sequential space conversion method. *Reliability Engineering & System Safety*, *217*, 108036.

Aute, V., Saleh, K., Abdelaziz, O., Azarm, S., & Radermacher, R. (2013). Cross-validation based single response adaptive design of experiments for Kriging metamodeling of deterministic computer simulations. *Structural and Multidisciplinary Optimization*, *48*(3), 581–605.

Baker, O. (1954). Design of Pipe Lines for the Simultaneous Flows of a Two-

Phase Mixture in a Horizontal Pipe.'. *The Oil and Gas Journal*.

Barmak, I., Gelfgat, A., Vitoshkin, H., Ullmann, A., & Brauner, N. (2016). Stability of stratified two-phase flows in horizontal channels. *Physics of Fluids*, *28*(4), 44101.

Barnea, D. (1987). A unified model for predicting flow-pattern transitions for the whole range of pipe inclinations. *International Journal of Multiphase Flow*, *13*(1), 1–12.

Bartz-Beielstein, T., Naujoks, B., Stork, J., & Zaefferer, M. (2016). Tutorial on surrogate-assisted modelling. *Tech. Rep. D. 12, Synergy for Smart Multi-Objective Optimisation*.

Bates, S., Sienz, J., & Toropov, V. (2004). Formulation of the optimal Latin hypercube design of experiments using a permutation genetic algorithm. *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, 2011.

Beggs, D. H., & Brill, J. P. (1973). A study of two-phase flow in inclined pipes. *Journal of Petroleum Technology*, *25*(05), 607–617.

Behari, N., Sheriff, M. Z., Rahman, M. A., Nounou, M., Hassan, I., & Nounou, H. (2020). Chronic leak detection for single and multiphase flow: A critical review on onshore and offshore subsea and arctic conditions. *Journal of Natural Gas Science and Engineering*, *81*, 103460.

Bhosekar, A., & Ierapetritou, M. (2018). Advances in surrogate based modeling, feasibility analysis, and optimization: A review. *Computers & Chemical Engineering*, *108*, 250–267.

Birkeland, S. A. (2014). *A Numerical Study of Transient Friction and Transient Friction Modelling in Ramp-up and Ramp-down Flow Conditions Similar to Pump Ramp-up and Valve Closure in Gas Transport Pipelines*. NTNU.

Bolotina, I., Borikov, V., Ivanova, V., Mertins, K., & Uchaikin, S. (2018). Application of phased antenna arrays for pipeline leak detection. *Journal of Petroleum Science and Engineering*, *161*, 497–505.

Bouhlel, M. A., & Martins, J. R. R. A. (2019). Gradient-enhanced kriging for high-dimensional problems. *Engineering with Computers*, *35*(1), 157–173.

Braconnier, T., Ferrier, M., Jouhaud, J.-C., Montagnac, M., & Sagaut, P. (2011). Towards an adaptive POD/SVD surrogate model for aeronautic design. *Computers & Fluids*, *40*(1), 195–209.

Bratland, O. (2010). *The Flow Assurance. Bergen Area. Norway. Oil and Energy*. http://www.drbratland.com/PipeFlow2/chapter1.html

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Busby, D., Farmer, C. L., & Iske, A. (2007). Hierarchical nonlinear approximation for experimental design and statistical data fitting. *SIAM Journal on Scientific Computing*, *29*(1), 49–69.

Chaibub Neto, E., Bare, J. C., & Margolin, A. A. (2014). Simulation studies as designed experiments: the comparison of penalized regression models in the "large p, small n" setting. *PloS One*, *9*(10), e107957.

Cheah, M. J., Kevrekidis, I. G., & Benziger, J. B. (2013). Water slug formation and motion in gas flow channels: the effects of geometry, surface wettability, and gravity. *Langmuir*, *29*(31), 9918–9934.

Chen, Y., Yang, K.-S., Chang, Y.-J., & Wang, C.-C. (2001). Two-phase pressure drop of air–water and R-410A in small horizontal tubes. *International Journal of Multiphase Flow*, *27*(7), 1293–1299.

Cheng, K., Lu, Z., Ling, C., & Zhou, S. (2020). Surrogate-assisted global sensitivity analysis: an overview. *Structural and Multidisciplinary Optimization*, *61*(3), 1187–1213. https://doi.org/10.1007/s00158-019-02413-5

Cheng, K., Lu, Z., Wei, Y., Shi, Y., & Zhou, Y. (2017). Mixed kernel function support vector regression for global sensitivity analysis. *Mechanical Systems and Signal Processing*, *96*, 201–214.

Chinello, G., Ayati, A. A., McGlinchey, D., Ooms, G., & Henkes, R. (2019). Comparison of computational fluid dynamics simulations and experiments for stratified air-water flows in pipes. *Journal of Fluids Engineering*, *141*(5).

Chisholm, D. (1967). A theoretical basis for the Lockhart-Martinelli correlation for two-phase flow. *International Journal of Heat and Mass Transfer*, *10*(12), 1767–1778.

Choi, K.-I., Pamitran, A. S., Oh, C.-Y., & Oh, J.-T. (2008). Two-phase

pressure drop of R-410A in horizontal smooth minichannels. *International Journal of Refrigeration*, *31*(1), 119–129.

Chu, L., De Cursi, E. S., El Hami, A., & Eid, M. (2015). Reliability based optimization with metaheuristic algorithms and Latin hypercube sampling based surrogate models. *Appl. Comput. Math*, *4*, 462–468.

Chu, S.-C., Du, Z.-G., Peng, Y.-J., & Pan, J.-S. (2021). Fuzzy Hierarchical Surrogate Assists Probabilistic Particle Swarm Optimization for expensive high dimensional problem. *Knowledge-Based Systems*, *220*, 106939.

Chua, P. C., Moon, S. K., Ng, Y. T., & Ng, H. Y. (2021). Prediction of production performance in smart manufacturing using multivariate adaptive regression spline. *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, *85376*, V002T02A064.

Clifford, D., Payne, J. E., Pringle, M. J., Searle, R., & Butler, N. (2014). Pragmatic soil survey design using flexible Latin hypercube sampling. *Computers & Geosciences*, *67*, 62–68.

Crombecq, K. (2011). *Surrogate modeling of computer experiments with sequential experimental design*. Ghent University.

Crombecq, K., Laermans, E., & Dhaene, T. (2011). Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling. *European Journal of Operational Research*, *214*(3), 683–696.

Davis, S. E., Cremaschi, S., & Eden, M. R. (2018). Efficient surrogate model development: Impact of sample size and underlying model dimensions. In *Computer Aided Chemical Engineering* (Vol. 44, pp. 979–984). Elsevier.

de Vasconcellos Araújo, M., de Farias Neto, S. R., & de Lima, A. G. B. (2013). Theoretical evaluation of two-phase flow in a horizontal duct with leaks. *Advances in Chemical Engineering and Science*, *2013*.

de Vasconcellos Araújo, M., de Farias Neto, S. R., de Lima, A. G. B., & de Luna, F. D. T. (2014). Hydrodynamic study of oil leakage in pipeline via CFD. *Advances in Mechanical Engineering*, *6*, 170178.

Deb, K., & Myburgh, C. (2016). Breaking the billion-variable barrier in real-

world optimization using a customized evolutionary algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, 653–660.

Demirel, G., Acar, E., Celebioglu, K., & Aradag, S. (2017). CFD-driven surrogate-based multi-objective shape optimization of an elbow type draft tube. *International Journal of Hydrogen Energy*, *42*(28), 17601–17610.

Denk, K. (2007). *Development of a pressure-based solver for both incompressible and compressible flows* [Middle east Technical University]. https://open.metu.edu.tr/

Devabhaktuni, V. K., & Zhang, Q.-J. (2000). Neural network training-driven adaptive sampling algorithm for microwave modeling. *2000 30th European Microwave Conference*, 1–4.

Ding, S., Shi, Z., Chen, K., & Azar, A. T. (2015). Mathematical modeling and analysis of soft computing. In *Mathematical Problems in Engineering* (Vol. 2015). Hindawi.

Dong, H., Song, B., Wang, P., & Dong, Z. (2018). Hybrid surrogate-based optimization using space reduction (HSOSR) for expensive black-box functions. *Applied Soft Computing*, *64*, 641–655.

Doyle, T., & Defoe, J. (2020). Effects of Exhaust Positioning and Vehicle Operating Conditions on Rear Fascia Temperature. *SAE International Journal of Passenger Cars-Mechanical Systems*, *13*(1), 55–78.

Dukler, A. E., & Hubbard, M. G. (1975). A model for gas-liquid slug flow in horizontal and near horizontal tubes. *Industrial & Engineering Chemistry Fundamentals*, *14*(4), 337–347.

Eason, J., & Cremaschi, S. (2014). Adaptive sequential sampling for surrogate model generation with artificial neural networks. *Computers & Chemical Engineering*, *68*, 220–232.

Ebrahimi-Moghadam, A., Farzaneh-Gord, M., Arabkoohsar, A., & Moghadam, A. J. (2018). CFD analysis of natural gas emission from damaged pipelines: Correlation development for leakage estimation. *Journal of Cleaner Production*, *199*, 257–271.

Emamzadeh, M. (2012). *Modelling of annular two-phase flow in horizontal*

and vertical pipes including the transition from the stratified flow regime.

Espedal, M. (1998). *An experimental investigation of stratified two-phase pipe flow at small inclinations*. Norwegian University of Science and Technology (NTNU), Norway.

Felkner, J., Chatzi, E., & Kotnik, T. (2013). Interactive particle swarm optimization for the architectural design of truss structures. *2013 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*, 15–22.

Figueiredo, A. B., Sondermann, C. N., Patricio, R. A. C., Bodstein, G. C. R., & Rachid, F. B. F. (2017). A Leak Localization Model for Gas-Liquid Two-Phase Flows in Nearly Horizontal Pipelines. *ASME International Mechanical Engineering Congress and Exposition*, *58424*, V007T09A006.

Filip, A., Bălţăreţu, F., & Damian, R.-M. (2014). COMPARISON OF TWO-PHASE PRESSURE DROP MODELS FOR CONDENSING FLOWS IN HORIZONTAL TUBES. *Mathematical Modeling in Civil Engineering*, *4*.

Fonseca, L. G., Barbosa, H. J. C., & Lemonge, A. C. C. (2009). A similarity-based surrogate model for enhanced performance in genetic algorithms. *Opsearch*, *46*(1), 89–107.

Freitas, D., Lopes, L. G., & Morgado-Dias, F. (2020). Particle swarm optimisation: a historical review up to the current developments. *Entropy*, *22*(3), 362.

Freund, Y., Seung, H. S., Shamir, E., & Tishby, N. (1993). Information, prediction, and query by committee. *Advances in Neural Information Processing Systems*, 483.

Friedel, L. (1979). Improved friction pressure drop correlation for horizontal and vertical two-phase pipe flow. *Proc. of European Two-Phase Flow Group Meet., Ispra, Italy, 1979*.

Fu, H., Yang, L., Liang, H., Wang, S., & Ling, K. (2020). Diagnosis of the single leakage in the fluid pipeline through experimental study and CFD simulation. *Journal of Petroleum Science and Engineering*, *193*, 107437.

Fuhg, J. N., Fau, A., & Nackenhorst, U. (2020). State-of-the-art and

comparative review of adaptive sampling methods for kriging. *Archives of Computational Methods in Engineering*, 1–59.

Fushiki, T. (2011). Estimation of prediction error by using K-fold cross-validation. *Statistics and Computing*, *21*(2), 137–146.

Gad, A. G. (2022). Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. *Archives of Computational Methods in Engineering*, 1–31.

Garbai, L., & Sánta, R. (2012). Flow pattern map for in tube evaporation and condensation. *4th International Symposium on Exploitation of Renewable Energy Sources, EXPRESS*, 125–130.

Garud, Sushant S, Karimi, I. A., & Kraft, M. (2016). Smart adaptive sampling for surrogate modelling. In *Computer Aided Chemical Engineering* (Vol. 38, pp. 631–636). Elsevier.

Garud, Sushant S, Karimi, I. A., & Kraft, M. (2017). Design of computer experiments: A review. *Computers & Chemical Engineering*, *106*, 71–95.

Garud, Sushant Suhas, Karimi, I. A., & Kraft, M. (2017). Smart sampling algorithm for surrogate model development. *Computers & Chemical Engineering*, *96*, 103–114.

Gaspar, B., Teixeira, A. P., & Soares, C. G. (2014). Assessment of the efficiency of Kriging surrogate models for structural reliability analysis. *Probabilistic Engineering Mechanics*, *37*, 24–34.

Gershenson, C. (2003). Artificial neural networks for beginners. *ArXiv Preprint Cs/0308031*.

Ghojogh, B., Nekoei, H., Ghojogh, A., Karray, F., & Crowley, M. (2020). Sampling algorithms, from survey sampling to Monte Carlo methods: Tutorial and literature review. *ArXiv Preprint ArXiv:2011.00901*.

Gholizadeh, M., Jamei, M., Ahmadianfar, I., & Pourrajab, R. (2020). Prediction of nanofluids viscosity using random forest (RF) approach. *Chemometrics and Intelligent Laboratory Systems*, *201*, 104010.

Giunta, A., Wojtkiewicz, S., & Eldred, M. (2003). Overview of modern design of experiments methods for computational simulations. *41st Aerospace Sciences Meeting and Exhibit*, 649.

Golzari, A., Sefat, M. H., & Jamshidi, S. (2015). Development of an adaptive surrogate model for production optimization. *Journal of Petroleum Science and Engineering*, *133*, 677–688.

Grömping, U. (2009). Variable importance assessment in regression: linear regression versus random forest. *The American Statistician*, *63*(4), 308–319.

Gutmann, H.-M. (2001). A radial basis function method for global optimization. *Journal of Global Optimization*, *19*(3), 201–227.

Haeri, A., & Fadaee, M. J. (2016). Efficient reliability analysis of laminated composites using advanced Kriging surrogate model. *Composite Structures*, *149*, 26–32.

Hale, C. P. (2001). *Slug formation, growth and decay in gas-liquid flows*. University of London.

Halton, J. H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, *2*(1), 84–90.

Han, Z.-H., & Zhang, K.-S. (2012). Surrogate-based optimization. *Real-World Applications of Genetic Algorithms*, *343*.

Haykin, S. (2009). *Neural networks and learning machines, 3/E*. Pearson Education India.

Hess, S., Train, K. E., & Polak, J. W. (2006). On the use of a Modified Latin Hypercube Sampling (MLHS) method in the estimation of a Mixed Logit model for vehicle choice. *Transportation Research Part B: Methodological*, *40*(2), 147–163.

Hoarau, Q., Ginolhac, G., Atto, A. M., & Nicolas, J.-M. (2017). Robust adaptive detection of buried pipes using GPR. *Signal Processing*, *132*, 293–305.

Holmås, K., Nossen, J., Mortensen, D., Schulkes, R., & Langtangen, H. P. (2005). Simulation of wavy stratified two-phase flow using Computational Fluid Dynamics (CFD). *12th International Conference on Multiphase Production Technology*.

Huang, K., Krügener, M., Brown, A., Menhorn, F., Bungartz, H.-J., & Hartmann, D. (2021). Machine learning-based optimal mesh generation

in computational fluid dynamics. *ArXiv Preprint ArXiv:2102.12923*.

Iturriaga, S., & Nesmachnow, S. (2012). Solving very large optimization problems (up to one billion variables) with a parallel evolutionary algorithm in CPU and GPU. *2012 Seventh International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 267–272.

Jia, L., Alizadeh, R., Hao, J., Wang, G., Allen, J. K., & Mistree, F. (2020). A rule-based method for automated surrogate model selection. *Advanced Engineering Informatics*, *45*, 101123.

Jiang, C., Cai, X., Qiu, H., Gao, L., & Li, P. (2018). A two-stage support vector regression assisted sequential sampling approach for global metamodeling. *Structural and Multidisciplinary Optimization*, *58*(4), 1657–1672.

Jiang, P., Shu, L., Zhou, Q., Zhou, H., Shao, X., & Xu, J. (2015). A novel sequential exploration-exploitation sampling strategy for global metamodeling. *IFAC-PapersOnLine*, *48*(28), 532–537.

Jiang, P., Zhou, Q., & Shao, X. (2020). Verification methods for surrogate models. In *Surrogate Model-Based Engineering Design and Optimization* (pp. 89–113). Springer.

Jin, R., Chen, W., & Sudjianto, A. (2002). On sequential sampling for global metamodeling in engineering design. *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, *36223*, 539–548.

Jin, S.-S. (2021). Accelerating Gaussian Process surrogate modeling using Compositional Kernel Learning and multi-stage sampling framework. *Applied Soft Computing*, *104*, 106909.

Jin, Y. (2011). Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, *1*(2), 61–70.

Johnson, M. E., Moore, L. M., & Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, *26*(2), 131–148.

Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global*

Optimization, *13*(4), 455–492.

Junior, J. R. B., Batista Filho, S. F., Funcia, M. A., da Silva, L. O. M., Santos, A. A. S., & Schiozer, D. J. (2022). A comparison of machine learning surrogate models for net present value prediction from well placement binary data. *Journal of Petroleum Science and Engineering*, *208*, 109208.

Kalagnanam, J. R., & Diwekar, U. M. (1997). An efficient sampling technique for off-line quality control. *Technometrics*, *39*(3), 308–319.

Kam, S. I. (2010). Mechanistic modeling of pipeline leak detection at fixed inlet rate. *Journal of Petroleum Science and Engineering*, *70*(3–4), 145–156.

Kang, F., Xu, Q., & Li, J. (2016). Slope reliability analysis using surrogate models via new support vector machines with swarm intelligence. *Applied Mathematical Modelling*, *40*(11–12), 6105–6120.

Kanin, E. A., Osiptsov, A. A., Vainshtein, A. L., & Burnaev, E. V. (2019). A predictive model for steady-state multiphase pipe flow: Machine learning on lab data. *Journal of Petroleum Science and Engineering*, *180*, 727–746.

Kaveh, A., & Dadras, A. (2017). A novel meta-heuristic optimization algorithm: thermal exchange optimization. *Advances in Engineering Software*, *110*, 69–84.

Kim, J., Chae, M., Han, J., Park, S., & Lee, Y. (2021). The development of leak detection model in subsea gas pipeline using machine learning. *Journal of Natural Gas Science and Engineering*, *94*, 104134.

Knotek, S., Schmelter, S., & Olbrich, M. (2021). Assessment of different parameters used in mesh independence studies in two-phase slug flow simulations. *Measurement: Sensors*, *18*, 100317.

Le Gratiet, L., & Cannamela, C. (2015). Cokriging-based sequential design strategies using fast cross-validation techniques for multi-fidelity computer codes. *Technometrics*, *57*(3), 418–427.

Li, F., Cai, X., & Gao, L. (2019). Ensemble of surrogates assisted particle swarm optimization of medium scale expensive problems. *Applied Soft Computing*, *74*, 291–305.

Li, F., Cai, X., Gao, L., & Shen, W. (2020). A surrogate-assisted multiswarm optimization algorithm for high-dimensional computationally expensive problems. *IEEE Transactions on Cybernetics*, *51*(3), 1390–1402.

Li, F., Shen, W., Cai, X., Gao, L., & Wang, G. G. (2020). A fast surrogate-assisted particle swarm optimization algorithm for computationally expensive problems. *Applied Soft Computing*, *92*, 106303.

Li, G., Aute, V., & Azarm, S. (2010). An accumulative error based adaptive design of experiments for offline metamodeling. *Structural and Multidisciplinary Optimization*, *40*(1–6), 137.

Li, S., Fan, S., Gu, J., Li, X., & Huang, Z. (2022). Blind-Kriging based natural frequency modeling of industrial Robot. *Precision Engineering*, *74*, 126–139.

Li, X., Chen, G., Khan, F., & Xu, C. (2019). Dynamic risk assessment of subsea pipelines leak using precursor data. *Ocean Engineering*, *178*, 156–169.

Li, X., Chen, G., Zhang, R., Zhu, H., & Xu, C. (2019). Simulation and assessment of gas dispersion above sea from a subsea release: A CFD-based approach. *International Journal of Naval Architecture and Ocean Engineering*, *11*(1), 353–363.

Li, X., Chen, G., & Zhu, H. (2017). Modelling and assessment of accidental oil release from damaged subsea pipelines. *Marine Pollution Bulletin*, *123*(1–2), 133–141.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*, *2*(3), 18–22.

Lin, D. K. J., Simpson, T. W., & Chen, W. (2001). Sampling strategies for computer experiments: design and analysis. *International Journal of Reliability and Applications*, *2*(3), 209–240.

Liu, C., Li, Y., & Xu, M. (2019). An integrated detection and location model for leakages in liquid pipelines. *Journal of Petroleum Science and Engineering*, *175*, 852–867.

Liu, H., Ong, Y.-S., & Cai, J. (2018). A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. *Structural and Multidisciplinary Optimization*,

57(1), 393–416.

Liu, H., Xu, S., Ma, Y., Chen, X., & Wang, X. (2016). An adaptive Bayesian sequential sampling approach for global metamodeling. *Journal of Mechanical Design*, *138*(1), 11404.

Lo, S., & Tomasello, A. (2010). Recent progress in CFD modelling of multiphase flow in horizontal and near-horizontal pipes. *7th North American Conference on Multiphase Technology*.

Lockhart, R. W., & Martinelli, R. C. (1949). Proposed correlation of data for isothermal two-phase, two-component flow in pipes. *Chemical Engineering Progress*, *45*(1), 39–48.

Loeppky, J. L., Sacks, J., & Welch, W. J. (2009). Choosing the sample size of a computer experiment: A practical guide. *Technometrics*, *51*(4), 366–376.

Luo, W., Guo, X., Dai, J., & Rao, T. (2021). Hull optimization of an underwater vehicle based on dynamic surrogate model. *Ocean Engineering*, *230*, 109050.

Maakala, V., Järvinen, M., & Vuorinen, V. (2018). Optimizing the heat transfer performance of the recovery boiler superheaters using simulated annealing, surrogate modeling, and computational fluid dynamics. *Energy*, *160*, 361–377.

Mackman, T., & Allen, C. (2010). Aerodynamic data modeling using multi-criteria adaptive sampling. *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, 9194.

Mackman, T. J., Allen, C. B., Ghoreyshi, M., & Badcock, K. J. (2013). Comparison of adaptive sampling methods for generation of surrogate aerodynamic models. *AIAA Journal*, *51*(4), 797–808.

Mahulja, S., Larsen, G. C., & Elham, A. (2018). Engineering an optimal wind farm using surrogate models. *Wind Energy*, *21*(12), 1296–1308.

Mandhane, J. M., Gregory, G. A., & Aziz, K. (1974). A flow pattern map for gas—liquid flow in horizontal pipes. *International Journal of Multiphase Flow*, *1*(4), 537–553.

Manoochehri, M., & Kolahan, F. (2014). Integration of artificial neural network and simulated annealing algorithm to optimize deep drawing

process. *The International Journal of Advanced Manufacturing Technology*, *73*(1–4), 241–249.

Martins, J. C., & Seleghim, P. (2010). Assessment of the performance of acoustic and mass balance methods for leak detection in pipelines for transporting liquids. *Journal of Fluids Engineering*, *132*(1).

McKay, M. D., Beckman, R. J., & Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, *42*(1), 55–61.

Meckesheimer, M., Barton, R. R., Simpson, T. W., & Booker, A. J. (2001). Computationally inexpensive metamodel assessment strategies. *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, *80227*, 191–201.

Mengistu, T., & Ghaly, W. (2008). Aerodynamic optimization of turbomachinery blades using evolutionary methods and ANN-based surrogate models. *Optimization and Engineering*, *9*(3), 239–255.

Menz, M., Dubreuil, S., Morio, J., Gogu, C., Bartoli, N., & Chiron, M. (2021). Variance based sensitivity analysis for Monte Carlo and importance sampling reliability assessment with Gaussian processes. *Structural Safety*, *93*, 102116.

Metropolis, N., & Ulam, S. (1949). The monte carlo method. *Journal of the American Statistical Association*, *44*(247), 335–341.

Molina-Espinosa, L., Cazarez-Candia, O., & Verde-Rodarte, C. (2013). Modeling of incompressible flow in short pipes with leaks. *Journal of Petroleum Science and Engineering*, *109*, 38–44.

Moore, T. (1999). *Alaska Department of Environmental Conservation. Technical Review of Leak Detection Technologies*. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjcy6P8oPnoAhWUTsAKHZD1AQMQFjAAegQIARAB&url=https%3A%2F%2Fdec.alaska.gov%2Fmedia%2F8148%2Fvol2-ast.pdf&usg=AOvVaw01UkATVbkYCCVTDa5y_yHn (access on 15 January 2019)

Muggleton, J. M., Hunt, R., Rustighi, E., Lees, G., & Pearce, A. (2020). Gas pipeline leak noise measurements using optical fibre distributed

acoustic sensing. *Journal of Natural Gas Science and Engineering*, *78*, 103293.

Nezami, O. M., Bahrampour, A., & Jamshidlou, P. (2013). Dynamic Diversity Enhancement in Particle Swarm Optimization (DDEPSO) algorithm for preventing from premature convergence. *Procedia Computer Science*, *24*, 54–65.

Nickabadi, A., Ebadzadeh, M. M., & Safabakhsh, R. (2011). A novel particle swarm optimization algorithm with adaptive inertia weight. *Applied Soft Computing*, *11*(4), 3658–3670.

Noguera-Polania, J. F., Hernández-García, J., Galaviz-López, D. F., Torres, L., Guzmán, J. E. V, Sanjuán-Mejía, M. E., & Jiménez-Cabas, J. (2020). Dataset on water–glycerol flow in a horizontal pipeline with and without leaks. *Data in Brief*, *31*.

Ökten, G., Shah, M., & Goncharov, Y. (2012). Random and deterministic digit permutations of the Halton sequence. In *Monte Carlo and Quasi-Monte Carlo Methods 2010* (pp. 609–622). Springer.

Pathak, S., Mishra, I., & Swetapadma, A. (2018). An assessment of decision tree based classification and regression algorithms. *2018 3rd International Conference on Inventive Computation Technologies (ICICT)*, 92–95.

Pekel, E. (2020). Estimation of soil moisture using decision tree regression. *Theoretical and Applied Climatology*, *139*(3), 1111–1119.

Pérez-Pérez, E. J., López-Estrada, F. R., Valencia-Palomo, G., Torres, L., Puig, V., & Mina-Antonio, J. D. (2021). Leak diagnosis in pipelines using a combined artificial neural network approach. *Control Engineering Practice*, *107*, 104677.

Png, W. H., Lin, H. S., Pua, C. H., & Abd Rahman, F. (2018). Pipeline monitoring and leak detection using Loop integrated Mach Zehnder Interferometer optical fiber sensor. *Optical Fiber Technology*, *46*, 221–225.

Pourghasemi, H. R., & Kerle, N. (2016). Random forests and evidential belief function-based landslide susceptibility assessment in Western Mazandaran Province, Iran. *Environmental Earth Sciences*, *75*(3), 1–

17.

Quirante, N., & Caballero, J. A. (2016). Large scale optimization of a sour water stripping plant using surrogate models. *Computers & Chemical Engineering*, *92*, 143–162.

Rahmati, O., Pourghasemi, H. R., & Melesse, A. M. (2016). Application of GIS-based data driven random forest and maximum entropy models for groundwater potential mapping: a case study at Mehran Region, Iran. *Catena*, *137*, 360–372.

Raimondi, L. (2019). Stratified gas-liquid flow–An analysis of steady state and dynamic simulation for gas-condensate systems. *Petroleum*, *5*(2), 128–132.

Regis, R. G. (2014a). Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. *Engineering Optimization*, *46*(2), 218–243.

Regis, R. G. (2014b). Particle swarm with radial basis function surrogates for expensive black-box optimization. *Journal of Computational Science*, *5*(1), 12–23.

Regis, R. G., & Shoemaker, C. A. (2007). A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing*, *19*(4), 497–509.

Rumpfkeil, M., Yamazaki, W., & Dimitri, M. (2011). A dynamic sampling method for kriging and cokriging surrogate models. *49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 883.

Saeedipour, M., Vincent, S., & Pirker, S. (2019). Large eddy simulation of turbulent interfacial flows using approximate deconvolution model. *International Journal of Multiphase Flow*, *112*, 286–299.

Salehi, H., Das, S., Biswas, S., & Burgueño, R. (2019). Data mining methodology employing artificial intelligence and a probabilistic approach for energy-efficient structural health monitoring with noisy and delayed signals. *Expert Systems with Applications*, *135*, 259–272.

Salem, M. Ben, & Tomaso, L. (2018). Automatic selection for general surrogate models. *Structural and Multidisciplinary Optimization*, *58*(2),

719–734.

Scott, D. S. (1964). Properties of cocurrent gas-liquid flow. In *Advances in chemical engineering* (Vol. 4, pp. 199–277). Elsevier.

Seung, H. S., Opper, M., & Sompolinsky, H. (1992). Query by committee. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 287–294.

Severance, C. (2015). Guido van rossum: The early years of python. *Computer*, *48*(2), 7–9.

Sheikholeslami, R., & Razavi, S. (2017). Progressive Latin Hypercube Sampling: An efficient approach for robust sampling-based analysis of environmental models. *Environmental Modelling & Software*, *93*, 109–126.

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360)*, 69–73.

Silva, R. A., Buiatti, C. M., Cruz, S. L., & Pereira, J. A. F. R. (1996). Pressure wave behaviour and leak detection in pipelines. *Computers & Chemical Engineering*, *20*, S491–S496.

Simpson, T. W., Poplinski, J. D., Koch, P. N., & Allen, J. K. (2001). Metamodels for computer-based engineering design: survey and recommendations. *Engineering with Computers*, *17*(2), 129–150.

Song, Q., Chen, G., Yang, Z., Wang, H., & Gong, M. (2018). New adiabatic and condensation two-phase flow pattern maps of R14 in a horizontal tube. *International Journal of Heat and Mass Transfer*, *127*, 910–924.

Steinberg, H. A. (1963). Generalized quota sampling. *Nuclear Science and Engineering*, *15*(2), 142–145.

Steponavičė, I., Shirazi-Manesh, M., Hyndman, R. J., Smith-Miles, K., & Villanova, L. (2016). On sampling methods for costly multi-objective black-box optimization. In *Advances in Stochastic and Deterministic Global Optimization* (pp. 273–296). Springer.

Strand, Ø. (1993). *An experimental investigation of stratified two-phase flow in horizontal pipes*. PhD thesis, University of Oslo, Oslo, Norway.

Straus, J., & Skogestad, S. (2019). A new termination criterion for sampling for surrogate model generation using partial least squares regression. *Computers & Chemical Engineering*, *121*, 75–85.

SUMAN, C. (2014). *NPTEL : Aerospace Engineering - Principles of Fluid Dynamics*. https://nptel.ac.in/courses/101/103/101103004/

Sun, C., Jin, Y., Zeng, J., & Yu, Y. (2015). A two-layer surrogate-assisted particle swarm optimization algorithm. *Soft Computing*, *19*(6), 1461–1475.

Sun, Q., Zhu, H., Wang, G., & Fan, J. (2011). Effects of mesh resolution on hypersonic heating prediction. *Theoretical and Applied Mechanics Letters*, *1*(2), 22001.

Sun, X, Roberts, S., Croke, B., & Jakeman, A. (2017). A comparison of global sensitivity techniques and sampling method. *Proc. 22nd Int. Congr. Modelling Simulation*, 57–63.

Sun, Xifu, Croke, B., Roberts, S., & Jakeman, A. (2021). Comparing methods of randomizing Sobol' sequences for improving uncertainty of metrics in variance-based global sensitivity estimation. *Reliability Engineering & System Safety*, *210*, 107499.

Taalab, K., Cheng, T., & Zhang, Y. (2018). Mapping landslide susceptibility and types using Random Forest. *Big Earth Data*, *2*(2), 159–178.

Taitel, Y., & Dukler, A. E. (1976). A model for slug frequency during gas-liquid flow in horizontal and near horizontal pipes. *International Journal of Multiphase Flow*, *3*(6), 585–596.

Tarantola, S., Becker, W., & Zeitz, D. (2012). A comparison of two sampling methods for global sensitivity analysis. *Computer Physics Communications*, *183*(5), 1061–1072.

Terzuoli, F., Galassi, M. C., Mazzini, D., & D'Auria, F. (2008). CFD code validation against stratified air-water flow experimental data. *Science and Technology of Nuclear Installations*, *2008*.

Tian, J., Sun, C., Tan, Y., & Zeng, J. (2020). Granularity-based surrogate-assisted particle swarm optimization for high-dimensional expensive optimization. *Knowledge-Based Systems*, *187*, 104815.

Tzannetakis, N., & Van de Peer, J. (2002). Design optimization through

parallel-generated surrogate models, optimization methodologies and the utility of legacy simulation software. *Structural and Multidisciplinary Optimization*, *23*(2), 170–186.

Vakili, S., & Gadala, M. S. (2013). Low cost surrogate model based evolutionary optimization solvers for inverse heat conduction problem. *International Journal of Heat and Mass Transfer*, *56*(1–2), 263–273.

Van Beers, W. C. M., & Kleijnen, J. P. C. (2003). Kriging for interpolation in random simulation. *Journal of the Operational Research Society*, *54*(3), 255–262.

van der Walt, J. C., Heyns, P. S., & Wilke, D. N. (2021). Model calibration to find leaks in water networks by desensitizing measurements to the model parameters using Artificial Neural Networks. *Urban Water Journal*, *18*(5), 352–363.

Vardy, A., & Brown, J. (1996). On turbulent, unsteady, smooth-pipe friction. *BHR Group Conference Series Publication*, *19*, 289–312.

Viana, F. A. C. (2016). A tutorial on Latin hypercube design of experiments. *Quality and Reliability Engineering International*, *32*(5), 1975–1985.

Vítkovský, J. P., Bergant, A., Simpson, A. R., & Lambert, M. F. (2003). Frequency-domain transient pipe flow solution including unsteady friction. *Proc., Int. Conf. on Pumps, Electromechanical Devices and Systems Applied to Urban Water Management*, *2*, 773–780.

Vlachos, N. A., Paras, S. V, & Karabelas, A. J. (1999). Prediction of holdup, axial pressure gradient and wall shear stress in wavy stratified and stratified/atomization gas/liquid flow. *International Journal of Multiphase Flow*, *25*(2), 365–376.

Vu, K. K., d'Ambrosio, C., Hamadi, Y., & Liberti, L. (2017). Surrogate-based methods for black-box optimization. *International Transactions in Operational Research*, *24*(3), 393–424.

Wang, Z., & Sobey, A. (2020). A comparative review between Genetic Algorithm use in composite optimisation and the state-of-the-art in evolutionary computation. *Composite Structures*, *233*, 111739.

Wei, O. Y., & Masuri, S. U. (2019). Computational fluid dynamics analysis on single leak and double leaks subsea pipeline leakage. *CFD Letters*,

11(2), 95–107.

Wei, X., Wu, Y.-Z., & Chen, L.-P. (2012). A new sequential optimal sampling method for radial basis functions. *Applied Mathematics and Computation*, *218*(19), 9635–9646.

Weisman, J., Duncan, D., Gibson, J., & Crawford, T. (1979). Effects of fluid properties and pipe diameter on two-phase flow patterns in horizontal lines. *International Journal of Multiphase Flow*, *5*(6), 437–462.

Wesseling, P. (2009). *Principles of computational fluid dynamics* (Vol. 29). Springer Science & Business Media.

Wortmann, T., Costa, A., Nannicini, G., & Schroepfer, T. (2015). Advantages of surrogate models for architectural design optimization. *AI EDAM*, *29*(4), 471–481.

Xinhong, L., Guoming, C., Renren, Z., Hongwei, Z., & Jianmin, F. (2018). Simulation and assessment of underwater gas release and dispersion from subsea gas pipelines leak. *Process Safety and Environmental Protection*, *119*, 46–57.

Xu, C., Rangaiah, G. P., & Zhao, X. S. (2015). Application of artificial neural network and genetic programming in modeling and optimization of ultraviolet water disinfection reactors. *Chemical Engineering Communications*, *202*(11), 1415–1424.

Xu, J., Wu, Y., Shi, Z., Lao, L., & Li, D. (2007). Studies on two-phase co-current air/non-Newtonian shear-thinning fluid flows in inclined smooth pipes. *International Journal of Multiphase Flow*, *33*(9), 948–969.

Xu, S., Liu, H., Wang, X., & Jiang, X. (2014). A robust error-pursuing sequential sampling approach for global metamodeling based on voronoi diagram and cross validation. *Journal of Mechanical Design*, *136*(7), 71009.

Xu, X., & Karney, B. (2017). An overview of transient fault detection techniques. *Modeling and Monitoring of Pipelines and Networks*, 13–37.

Yan, S., Zou, X., Ilkhani, M., & Jones, A. (2020). An efficient multiscale surrogate modelling framework for composite materials considering progressive damage based on artificial neural networks. *Composites Part B: Engineering*, *194*, 108014.

Yang, S., Jeon, K., Kang, D., & Han, C. (2017). Accident analysis of the Gumi hydrogen fluoride gas leak using CFD and comparison with post-accidental environmental impacts. *Journal of Loss Prevention in the Process Industries*, *48*, 207–215.

Yao, W., Chen, X., & Luo, W. (2009). A gradient-based sequential radial basis function neural network modeling method. *Neural Computing and Applications*, *18*(5), 477–484.

Zhang, H.-Q., Wang, Q., Sarica, C., & Brill, J. P. (2003). Unified model for gas-liquid pipe flow via slug dynamics—part 1: model development. *J. Energy Resour. Technol.*, *125*(4), 266–273.

Zhang, J., Chowdhury, S., & Messac, A. (2012). An adaptive hybrid surrogate model. *Structural and Multidisciplinary Optimization*, *46*(2), 223–238.

Zhang, T., Zeng, P., Jimenez, R., Li, T., Feng, X., & Sun, X. (2022). System reliability analysis of soil slopes using shear strength reduction and active-learning surrogate models. *Arabian Journal of Geosciences*, *15*(6), 1–17.

Zhang, W., Goh, A. T. C., & Zhang, Y. (2016). Multivariate adaptive regression splines application for multivariate geotechnical problems with big data. *Geotechnical and Geological Engineering*, *34*(1), 193–204.

Zhou, C., Shi, Z., Kucherenko, S., & Zhao, H. (2022). A unified approach for global sensitivity analysis based on active subspace and Kriging. *Reliability Engineering & System Safety*, *217*, 108080.

Zhou, J., Turng, L. S., & Kramschuster, A. (2006). Single and multi objective optimization for injection molding using numerical simulation with surrogate models and genetic algorithms. *International Polymer Processing*, *21*(5), 509–520.

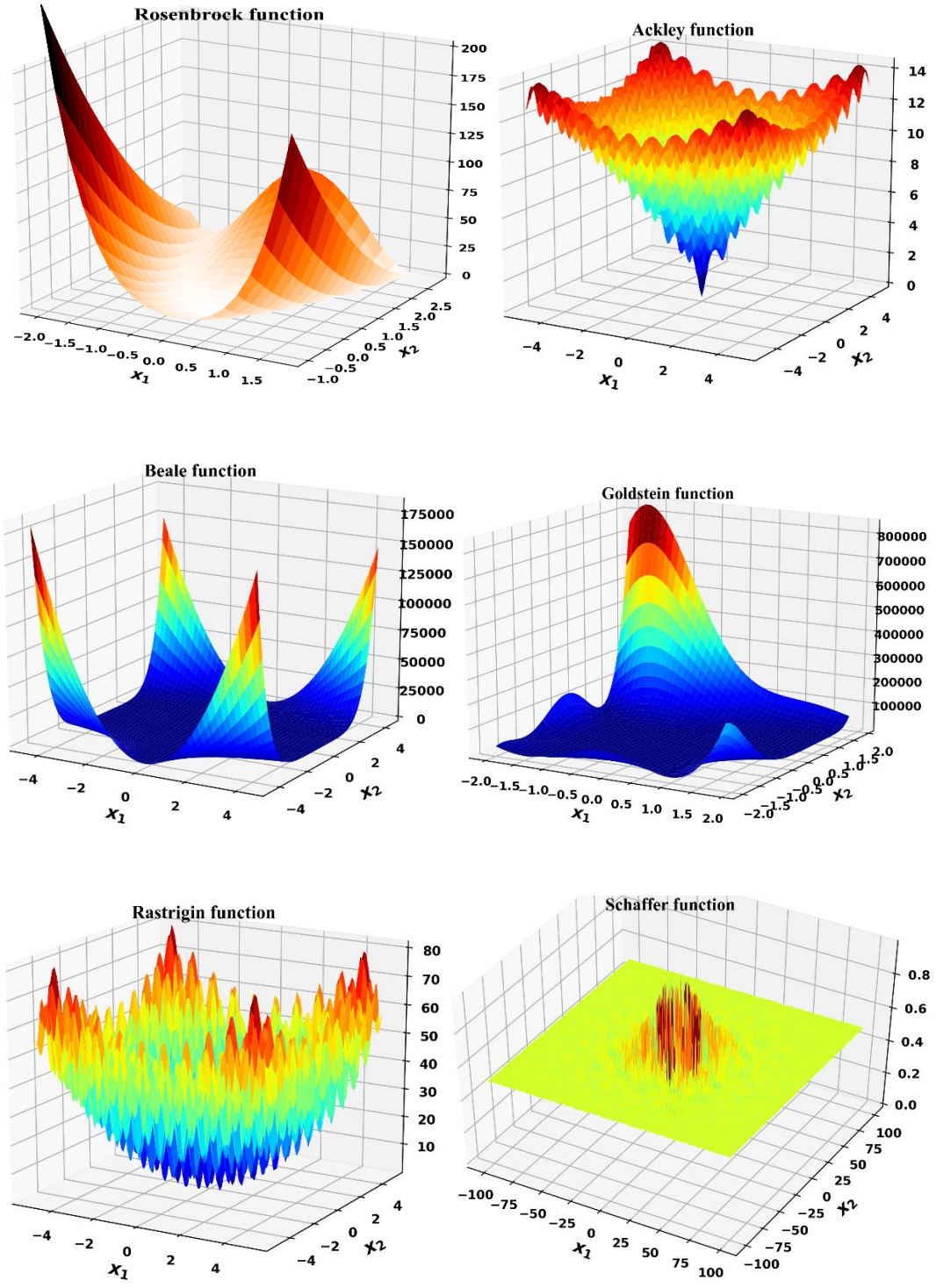
Zhu, H., Lin, P., & Pan, Q. (2014). A CFD (computational fluid dynamic) simulation for oil leakage from damaged submarine pipeline. *Energy*, *64*, 887–899.

Zhu, X., & Liu, B. (2019). A method to assess the reliability of casings in marine gas reservoirs based on Bayesian theory optimization. *Journal*

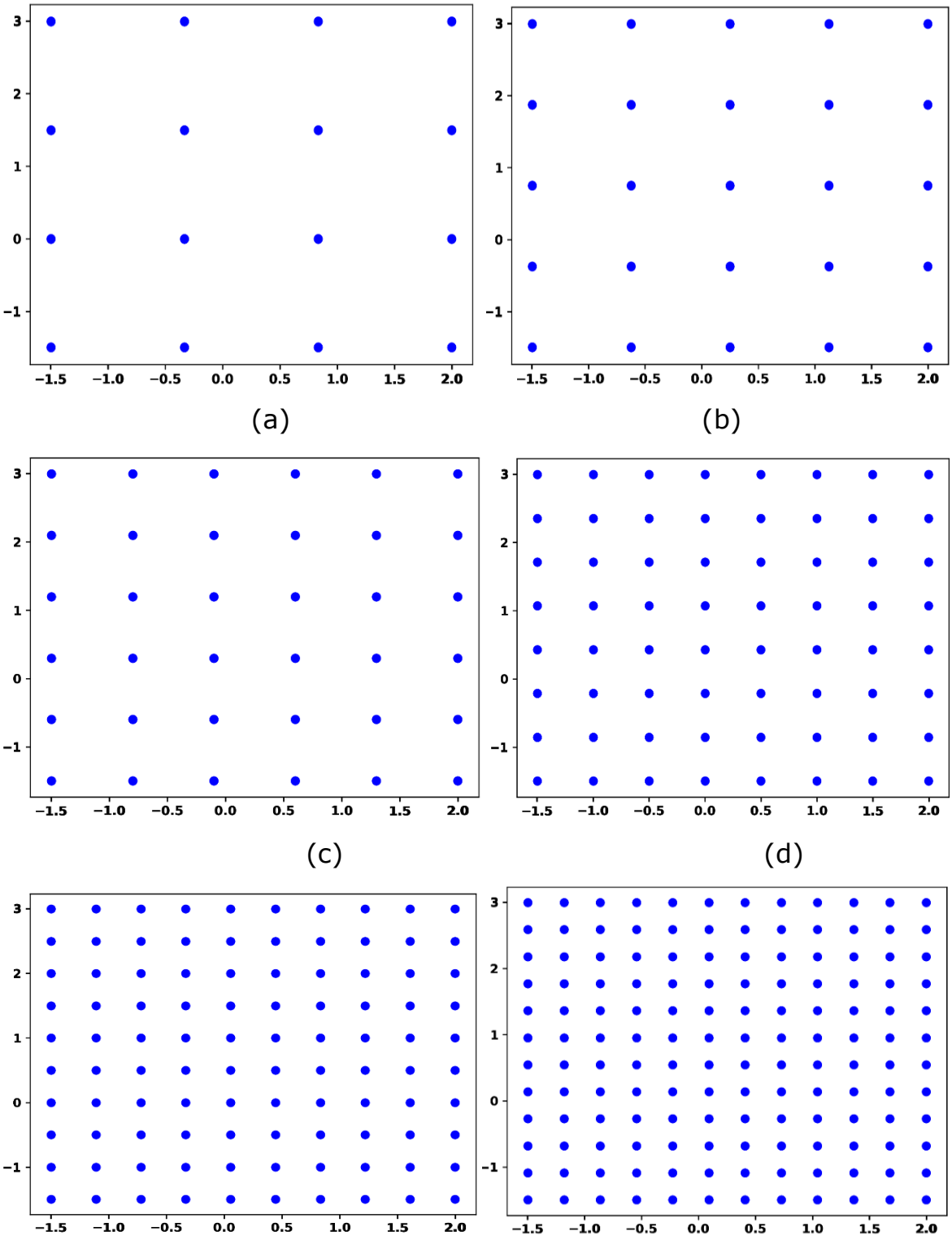
of Petroleum Science and Engineering, *172*, 1248–1256.

Appendix

Appendix A: Perspective view of three-dimensional benchmark functions used for the numerical experiment.



Appendix B: Illustration of sample points used for PSOASM evaluation: (a) 16 sampls, (b) 25 sampls, (c) 36 sampls, (d) 64 sampls, (e) 100 sampls, (f) 144 sampls.



Appendix C: Statistical summary of the dataset used for the single-phase surrogate modelling

	Pressure Profile (Pa/m)	**Flow rate (m^3/s)**
Count	40	40
Mean	4707.05	0.036789
Std	649.90	0.003964
Min	3125.00	0.030988
25%	4430.75	0.034290
50%	4734.50	0.036449
75%	5099.25	0.040132
Max	5816.50	0.046482

Appendix D: Statistical summary of the dataset used for the multiphase surrogate modelling

	Pressure Profile (Pa/m)	**Flow rate (m^3/s)**	**Liquid holdup**	**Volume fraction**
Count	43	43	43	43
Mean	362.2325	0.0138	0.5877	0.4123
Std	52.8329	0.0004	0.0847	0.0846
Min	207.0000	0.0129	0.4300	0.2100
25%	326.0000	0.0135	0.5250	0.3600
50%	381.0000	0.0138	0.5800	0.4200
75%	404.5000	0.0141	0.6400	0.4750
Max	440.0000	0.0147	0.7900	0.5700