Pipeline leakage detection and characterisation with adaptive surrogate modelling using particle swarm optimisation.

ADEGBOYE, M.A., KARNIK, A., FUNG, W.-K. and PRABHU, R.

2022

Pipeline Leakage Detection and Characterisation with Adaptive Surrogate Modelling using Particle Swarm Optimisation

Mutiu Adesina Adegboye
School of Engineering
Robert Gordon University
Aberdeen, United Kingdom
m.adegboye@rgu.ac.uk

Aditya Karnik
School of Engineering
Robert Gordon University
Aberdeen, United Kingdom
a.karnik@rgu.ac.uk

Wai-keung Fung
Cardiff School of Technologies
Cardiff Metropolitan University
Cardiff, United Kingdom
WFung@cardiffnet.ac.uk

Radhakrishna Prabhu
School of Engineering
Robert Gordon University
Aberdeen, United Kingdom
r.prabhu@rgu.ac.uk

***Abstract*—Pipelines are often subject to leakage due to ageing, corrosion, and weld defects, and it is difficult to avoid as the sources of leakages are diverse. Several studies have demonstrated the applicability of the machine learning model for the timely prediction of pipeline leakage. However, most of these studies rely on a large training data set for training accurate models. The cost of collecting experimental data for model training is huge, while simulation data is computationally expensive and time-consuming. To tackle this problem, the present study proposes a novel data sampling optimisation method, named adaptive particle swarm optimisation (PSO) assisted surrogate model, which was used to train the machine learning models with a limited dataset and achieved good accuracy. The proposed model incorporates the population density of training data samples and model prediction fitness to determine new data samples for improved model fitting accuracy. The proposed method is applied to 3-D pipeline leakage detection and characterisation. The result shows that the predicted leak sizes and location match the actual leakage. The significance of this study is two-fold: the practical application allows for pipeline leak prediction with limited training samples and provides a general framework for computational efficiency improvement using adaptive surrogate modelling in various real-life applications.**

Keywords—adaptive surrogate model, data optimisation, machine learning, pipeline leak detection, particle swarm optimisation

I. INTRODUCTION

The demand for energy is increasing worldwide. Completely substituting the hydrocarbon power source with renewable technology is not yet mature enough. The 2020 annual energy report of the Energy Information Administration shows 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 [1]. Pipelines are an important component for oil and gas transportation in the industry, and their internal flows such as water and petroleum are important resources for national development. Pipeline is the most cost-effective means of conveying petroleum over a long distance. About 97% of all-natural gas and crude oil production is currently transported by pipeline in Canada, while 70% of petroleum is transported through the pipeline in the USA [2].

Despite pipelines being considered the cheapest and safer than other modes of transportation such as rail and road vehicles, they are still subject to leakage due to ageing, corrosion, and weld defects. A leak in the pipeline may lead to economic losses, environmental contamination, and human casualties in some cases [3][4]. Therefore, it is necessary to monitor the pipeline for timely detection as early detection of leaks will aid the quick response to stop oil discharge [5]. In this regard, several studies have taken advantage of machine learning algorithms to implement pipeline leak detection systems. Surrogate models that model the fluid flow behaviour involved in large pipeline systems help in understanding pipeline leakage mechanism and thus lead to the prediction of leakages. Caputo and Pelagagge [6] proposed a monitoring scheme to detect leak events in the pipeline network using a Multilayer Perceptron (MLP) neural network. The MLP is trained on the set of pressure and flow rate data, which characterise several states of the fluid network under normal and abnormal operating conditions. A pipeline monitoring method based on Discrete Incremental Clustering Fuzzy (DICF) Artificial Neural Network (ANN) was proposed in [7]. A set of raw sensor information consisting of pressure and flow measurements was used as input to the fuzzy ANN. The experimental results showed that the model can effectively detect leakage in a 14.117 km long pipeline with an inner diameter of 273 mm. Similar studies employing pressure and flow rate sensors positioned at specific locations on pipeline systems can be found in [8].

The application of machine learning models in pipeline leakage detection and characterisation has many strengths, such as low cost and quick leak detection without explicit programming. Similarly, the leak detection task can be automatically handled at high speed. However, a large dataset is required to develop accurate machine learning models that cover the parameter space thoroughly. It is difficult to obtain experimental data for accurate model training. Some of the challenges include huge costs of data collection, poor accessibility to the pipeline, time, human efforts etc. [1]. Moreover, datasets constructed from data acquired in laboratory or field tests are usually imbalanced as leakage data

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

samples are generated from artificial leaks, which require thorough measures to avoid pollution.

To overcome these challenges, many studies proposed dynamic modelling as a good alternative and have been widely used in the industry and by the research community. Dynamic 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, external environment and inspect the complex relationship connecting the flow parameters such as pressure, flow rate, and temperature in the presence and absence of leakage through computationally intensive simulations.

With advances in computing technologies, the dynamic modelling approach has been attracting more and more attention from industry and academia. A comprehensive review of the dynamic modelling method is presented in [9], [10]. Zhu et al. [11] analysed the flow effect on the damaged submarine pipeline with different leak sizes. Doshmanziari et al. [12] proposed a framework to detect leakage on a 50 km high-pressure operational gas pipeline using OLGA multiphase flow simulation tool. Analysis of pressure distribution through the leak using 3-D computational fluid dynamics (CFD) model was presented in [13] and compared with pressure distribution from experimental data. Identical pressure profile was reported for both numerical and experimental data (pressures and flow rates) recorded at the pipeline inlet and outlet. Yang et al. [14] modelled the 2012 disastrous Gumi hydrogen fluoride gas leakage using CFD simulation and compared the simulation results with the post-accidental vegetation fluoride concentration. The authors reported mean relative error of 6.8%. Numerical modelling of under-ground pipeline leakages was reported in [15]. The model was performed on a two- and three-dimensional pipeline considering surrounding soil as a porous medium. The authors reported 7% error between the simulation and experimental results for all the leak cases considered such as pipe and leak diameters variation, and the gas flow pressure.

Kim et al. [1] developed a machine learning model that trained on numerical data obtained on a 13,000 m long subsea pipeline and validated the model using a set of raw sensors data obtained in the gas platform. The training was performed on 3200 data samples using 90% for training and validation and 10% for the testing. The authors reported model performance of 0.987 and 0.01 for the R^2 and Mean Absolute Error (MAE) respectively, when tested for various leak size scenarios. A similar study reported in [16] proposed a two-stage neural network to detect and estimate the leak position in a pipeline. The first stage involves friction factor calculation using inlet and outlet flow rates. Moreover, the second stage considered the friction factor together with the inlet and outlet pressures to locate the position of leakage. The neural network was trained on the validated numerical simulator of the pilot plant, while the developed model was validated experimentally in a pilot plant with several leak sizes and locations. It was reported that the designed neural network provides a robust predictor for leak size and location estimation.

As supported by recent literature, machine learning or surrogate modelling has applicability for pipeline leakage detection and characterisation. However, existing studies typically depend on a large training dataset, which may not be possible to acquire in a physical pipeline due to the damages impact and costs associated with experimentations and surrounding environments. Numerical simulation is also suggested as an alternative for data set generation but it is computationally expensive. A realistic CFD simulation can take days or even weeks to complete despite the advances in high-performance computing [17]. Therefore, in the present study, a new data sample generation methodology called adaptive PSO-assisted surrogate model is proposed to optimally select training samples in machine learning applications involving computationally expensive problems, like CFD simulation in pipelines. The number of data samples and thus the number of CFD simulation trials is minimised by the proposed method without the sacrifice of model accuracy through the following features of the proposed method:

- Starting the model with limited data samples and iteratively refining the model by finding the optimal data points that capture the topography of the response surface of a CFD model in an adaptive manner using PSO.
- Combining system fitness value information and existing sample distance to determine the best candidate data samples for evaluation.
- Numerical simulation is performed for the individual with the best fitness value or greatest distance to lessen the consumed fitness evaluations.

The rest of the paper is structured as follows: Section II briefly describes the theoretical overview of PSO and minmax distance technique. The details of the proposed framework is presented in Section III. The evaluation of the proposed model in the context of a real-world case study of pipeline leakage characterisation and benchmark functions is presented in Section IV. Section V concludes the paper.

II. THEORETICAL OVERVIEW

A. 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 comprises of 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 $X = \{x_1, x_2, ..., x_m\}^T$ 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 the sacrifice of 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: $f : \mathbb{R}^N \rightarrow \mathbb{R}$

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

For the target function f defined in $X \in \mathbb{R}^N$, the initial sampling process begins with a set of sample pairs $(X_i, f_i), i = 1, 2, 3, ..., I$ 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 obtains denote $I_{\max}$; $g(X), i = 1, 2, 3, ..., I_{\max}$. In this study, the approximation model $g(X)$ is implemented using Multilayer Perceptron using training procedure presented in [18]. The population density of the sample points was computed using the MinMax distance technique, while PSO was employed for sample points placement optimisation. The description of PSO and MinMax distance techniques are given as follows:

B. PSO Algorithm

PSO is a metaheuristic algorithm that is used for the optimisation of problems. It was originally proposed by Kennedy and Eberhart [19] to simulate the behaviour of birds flock flying together in the search space in search of some optimal solutions. PSO has been successfully applied to a series of problems due to its simplicity and attractive search efficiency. In the design space, the position of each particle is described as the vector $x_1 \in R^n$ and its movement by the velocity of the particle $x_1 \in R^n, i = 1, 2, ..., n.$ At every iteration, the velocity and position of a particle is updated as in (1) and (3), respectively [20]:

$$v_{i,d}(t+1) = w_t v_{i,d}(t) + c_1 r_1 (P_{i,d}(t) - x_{i,d}(t)) + c_2 r_2 (Pg(t) - x_{i,d}(t)) \quad (1)$$

$$\text{with } w_t = w_{max} - \left(\frac{w_{max} - w_{min}}{iter_{max}} \right) iter_{cur} \quad (2)$$

$$x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1) \quad (3)$$

where $v_i(t) = (v_{i,1}(t), v_{i,2}(t), ..., v_{i,D}(t))$ is the velocity of particle i, $x_i(t) = (x_{i,1}(t), x_{i,2}(t), ..., x_{i,D}(t))$ is the position of particle i, $P_i(t) = (P_{i,1}(t), P_{i,2}(t), ..., P_{i,D}(t))$ is the historical best position found by particle i, which also known as personal best, $P_g(t) = (P_{g,1}(t), P_{g,2}(t), ..., P_{g,D}(t))$ is the historical best position found by swarm (also known as global best), r_1 and r_2 are random generated numbers in the range [0,1], c_1 and c_2 are acceleration coefficients, $iter_{max}$ is the maximum number of iteration, $iter_{cur}$ is the current iteration number, w_{max}, and w_{min} are selected as 0.9 and 0.4, respectively.

C. MinMax Distance Technique

The essence of minmax distance technique is to assign a new sample set X_C that minimise the maximum distance between the sample points. Then, refined the sample set as $X_{N+1} = X_C \cup X_N$ where the existing sample set denotes X_N. The mathematical representation of minimax is given as:

$$\min \left[\max_{X_C} \quad d(X_{C_i}, X_{N_j}) \right] \quad X_{C_i} \neq X_{N_j} \quad (4)$$

$$1 \leq i \leq m, 1 \leq j \leq l + m$$

where $X_{C_i} \in X_C (i = 1, ..., m)$ and $X_{N_j} \in X_N (j = 1, ..., l + m)$, d denote distance between the sample points. Consider sample points in the parameter space $X = \{x_1, x_2, .., x_m\}$. The Crowding Distance (CD) of an arbitrary point X_C to its neighbouring solutions can be estimated by [21]:

$$CD(X_C) = \sum_{i=1}^{m} || X_C - X_{C_i} ||^2 \quad (5)$$

The large crowding distance value $CD(X_C)$ of a sample point reflects low sample density (few sample points in its neighbourhood). The surrogate model accuracy is likely relatively lower around that region. Therefore, more sample points will probably be needed in such a region to enhance the surrogate model accuracy.

III. PROPOSED METHODS

The adaptive surrogate model constructed in this study involves three steps, namely, generation of initial training data samples, sample placement optimisation, and model quality assessment. These procedures are explained as follows:

A. Initial Data Samples for Training

The sample size and location of the samples governs 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 Latin Hypercube Design (LHD) method [22]. LHD has been widely used by many researchers 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.

B. Sample Placement optimisation

The sample placement optimisation is described as follows:

The surrogate model algorithm begins with a set of initial training samples $X = \{x_1, x_2, ..., x_m\}$ generated using the LHD technique, and the corresponding outputs $y = f(X)$ are estimated using the computational fluid dynamics simulator performed on 3-D pipeline leakage. The data obtained from the simulator are used for the initial training of the MLP, which is the coarse surrogate model constructed. The quality of the developed surrogate model is then assessed using Mean Square Error (MSE). The adaptive training process begins if the prediction quality of the initial surrogate model is not acceptable. In the adaptive sampling process, the model was designed to add new data samples to the region, which will enhance the surrogate model accuracy via the following steps:

Step 1: Initialise the PSO for optimisation with the initial generated points and determine the initial velocities particles v and initial positions of each particle.

Step 2: Find the particle X_C with the greatest crowding distance value that satisfy the minmax condition defined in eq. (4) using eq. (5) and assign it to the swarm global best.

Step 3: Perform CFD simulation for the new point X_C and obtains its corresponding values Y_{new}.

Step 4: Update the training dataset by refining the sample set as $X_{N+1} = X_C \cup X_N$ and the corresponding response values $Y_{N+1} = Y_{new} \cup Y_N$.

Step 5: Use updated dataset X_{N+1} and Y_{N+1} to retrain the surrogate model and archive the updated surrogate and swarm into the database.

Step 6: Evaluate the fitness of the added point by comparing the accuracy of the newly trained model with the previous one using MSE.

Step 7: If new global position improved surrogate model accuracy, update the global best of the swarm.

Step 8: Terminate the algorithm if the stopping condition is met and output the surrogate model. Otherwise, increment the iteration and go to Step 9. The termination condition are the maximum MSE is smaller than 0.04 or the maximum number of iterations is equal to 100, which was chosen based on results from trial runs.

Step 9: Update velocities and positions of the particles by using eqs. (1) and (3), respectively. If the fitness evaluation performed in Step 6 shows surrogate model improvement, go to Step 3. Otherwise, go to Step 2.

C. Surrogate model performance assessment

The surrogate model constructed using minimum training data points may possibly result in misleading predictions or optimal solutions located in an unfeasible region [23]. Therefore, it is essential to verify the quality of a surrogate model before using it for the prediction. The constructed surrogate model's was evaluated using a set of data points other than those used during the training phase. Four performance metrics include Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2) are employed to evaluate the performance of the developed surrogate model. It is important to highlight that the first three metrics (MSE, RMSE and MAE) are selected because they provides global error measure over the entire parameter space [21], [24], while the last metric is good to measure how good the best fit model compared to the baseline model [23]. For further details on performance metrics used in this study, readers are referred to surrogate model verification methods presented in [23]. To further assess the performance of our proposed surrogate model compared to existing sampling methodologies, three conventional space-filling sequential designs, namely uniform sampling, which is also called Grid sampling, Halton and LHD are employed for comparison. The details of these sampling algorithms can be found in [21].

IV. Evaluation and Analysis

The performance of the proposed algorithm was evaluated using two case studies. The first case presented in Section IV(A) employed two widely used benchmark problems, while the second case is performed on a 3-D pipeline leakage simulation model. For the PSO algorithm used in this study, c_1 and c_2 are 1.2 and 1.5, respectively, which fall to the range of the value commonly used in the literature [25]. The number of initial samples, which is also the size of the initial population is 10. MLP was used as a surrogate model for function approximation. The input nodes corresponding to the input variables, one hidden layer ranging from 8 to 20 neurons, proved suitable in this study, while the momentum and learning rate are 0.094 and 0.017, respectively.

A. Test case 1: Benchmark Problems

In this test case, the dataset generated from the benchmark functions was used to train the proposed surrogate model. Two widely used benchmark functions, namely Rosenbrock function and Ackley function, for optimisation are employed to illustrate the effectiveness of the constructed surrogate model and they are depicted, respectively as follows:

$$f(x,y) = (1-x)^2 + 10(y-x^2)^2 \qquad x \in [-1.5, 2.0],\ y \in [-1.5, 3.0] \tag{6}$$

$$f(x,y) = -20\exp\left[-0.2\sqrt{0.5(x^2+y^2)}\right] - \exp\left[0.5(\cos 2\pi x + \cos 2\pi y)\right] + e + 20 \qquad x \in [-5.0, 5.0],\ y \in [-5.0, 5.0] \tag{7}$$

We used benchmark functions to evaluate the influence of the training sample size on the constructed surrogate model performance. MSE is calculated to measure the model accuracy as the model learning progresses. Ten independent runs were performed during the experiments for each function, and the average learning accuracies were calculated. Note that the learning accuracy is referred to as the accuracy of the surrogate model as the number of training datasets increases and is calculated as a percentage of one minus MSE. The optimum training sample size was studied using the Rosenbrock and Ackley testing functions. The learning accuracy for the two functions is plotted in Fig. 1. 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 learning accuracy 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 arbitrarily, and the number of samples will grow incrementally with improving model accuracy. The learning accuracies of the model increase as the training sizes increases up to 30 (20 additional) sample sizes. Further adding more training points after the 30-training point appeared to contribute insignificantly to the surrogate model's prediction accuracy.

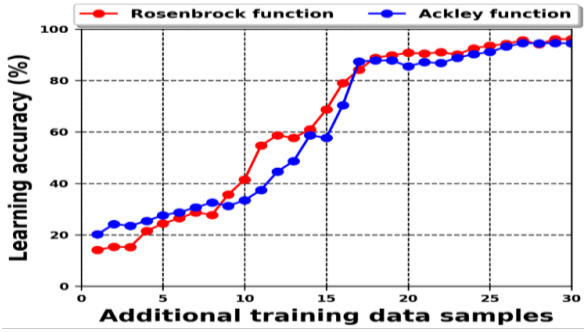

Fig. 1. Learning accuracies as a function of 30 additional training sample sizes

Furthermore, the performance of the surrogate model was assessed using sets of data other than those used in the training phase. The testing data varied from 5 to 100. Fig. 2 illustrates the model accuracy on the testing data. The model was able to predict new data well with the overall accuracy of 88.07% and 89.87% for the Rosenbrock and Ackley functions, respectively. The comparison of ground truth and predicted values for 100 testing sample sizes is illustrated in Fig.3. It can be observed that the figures demonstrate a good match between the ground truth and the predicted value.

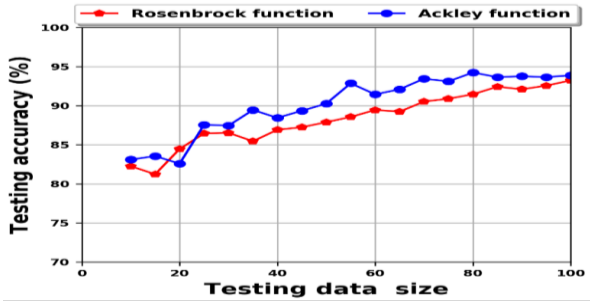

Fig. 2. Tesing accuracy on new sets of testing data

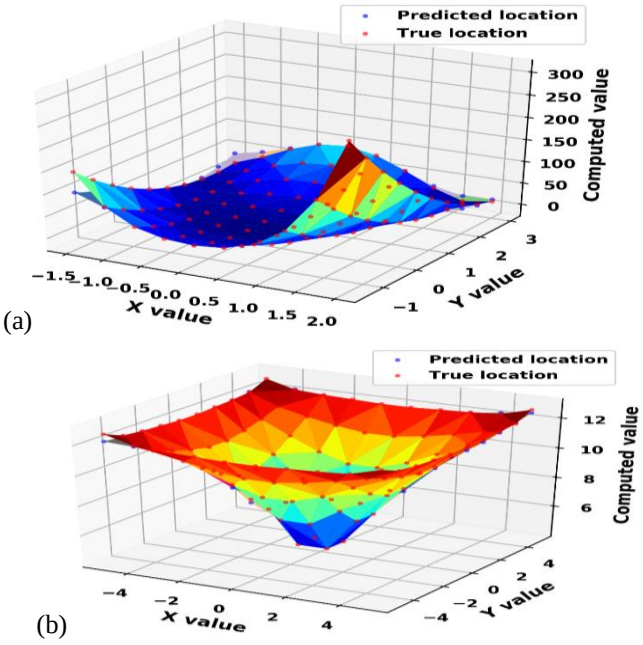

Fig. 3. PSOASM predicted values against ground truth: (a) Rosenbrock function, (b) Ackley function

B. Test case 2: 3-D Pipeline Leakage Model

The dataset employed for pipeline leakage and localisation was generated from the CFD model. The in-pipe flow rate and pressure profile were used as input features, while the leak sizes and locations were used as a targeted feature. The basic physical pipeline model with the leakage employed for study in this work is the same as that in [3]. The pipe diameter is 60 mm, while the pipe length is 50 times of the diameter. The CFD analysis of the pipeline leakage was carried out using ANSYS-FLUENT 18.0 and validated with the experimental data reported in the literature. The SST k-omega turbulence model was employed to simulate the liquid flow in the horizontal pipeline. The detailed numerical simulation approach, including pipe boundary conditions, grid independence study, and incoming flow conditions used in this study are similar to that in [3]. The fluid flow parameters such as pressure profile and flow rate commonly used in the open literature to describe the pipeline leak location and sizes are calculated using the CFD simulator. The data from the simulator (pressure and flow rate) and sample locations are input and output data used to develop the surrogate model. Fig. 4 shows the learning accuracy of the surrogate model constructed for the pipeline leak detection. The algorithm attains 96.8% accuracy for the 40-training data size. Moreover, the developed surrogate model performance is compared with the conventional space-filling sampling methods - the same number of training points used for the constructed surrogate model employed for the conventional approaches except for the Grid-49 and Grid-36 where 7x7 and 6x6 sample points are used in order to generate uniform grids. In the adaptive surrogate model, the initial model was designed using 10 data points. Then, new points are added in an adaptive manner until additional 30 points are added (i.e. 10 initial points plus 30 points added adaptively). The training data points are generated in a space-filling manner for the conventional methods. Table 1

summarises predictive errors for all the methods tested. It can be seen that the developed model outperforms the conventional sampling techniques by providing lower MSE, RMSE, MAE, and higher R^2 values than all the sampling methods employed for comparison.

Fig. 4. Learning accuracy for the pipeline leak detection

TABLE I: PERFORMANCE COMPARISON OF PSOASM WITH CONVENTIONAL SEQUATIAL SAMPLING METHODS

Sampling methods	Grid-49	Halton	Grid-36	LHD	PSOASM
MSE	0.2916	0.23989	0.34763	0.10289	0.0248
RMSE	0.5400	0.48979	0.5896	0.32076	0.1575
R2	0.7559	0.61539	0.70408	0.85904	0.9505
MAE	0.3902	0.38532	0.41597	0.24415	0.1891

V. CONCLUSIONS

This study presents an efficient method to optimise training data for the machine learning algorithm involving computationally expensive problems like 3D pipeline leakage characterisation. The model fitness value information and population distance criteria are incorporated to select candidate solutions for exact fitness evaluation. The developed surrogate model was implemented to optimise pipeline leakage detection and the performance compared with the space-filling sequential sampling approaches. The results show that the developed surrogate model outperformed conventional space-filling methods. The results achieved from the tested benchmark functions indicate that the developed surrogate model can efficiently explore and exploit the design parameter space and generate adequate sample points in an area that enhances surrogate model accuracy.

REFERENCES

[1] J. Kim, M. Chae, J. Han, S. Park, and Y. Lee, "The development of leak detection model in subsea gas pipeline using machine learning," J. Nat. Gas Sci. Eng., vol. 94, p. 104134, 2021.

[2] M. Layouni, M. S. Hamdi, and S. Tahar, "Detection and sizing of metal-loss defects in oil and gas pipelines using pattern-adapted wavelets and machine learning," Appl. Soft Comput., vol. 52, pp. 247–261, 2017.

[3] M. A. Adegboye, A. Karnik, and W.-K. Fung, "Numerical study of pipeline leak detection for gas-liquid stratified flow," J. Nat. gas Sci. Eng., vol. 94, p. 104054, 2021.

[4] A. Lukman, A. EMMANUEL, N. Chinonso, A. Mutiu, A. James, and K. Jonathan, "An anti-theft oil pipeline vandalism detection: embedded system development," Int. J. Eng. Sci. Appl., vol. 2, no. 2, pp. 55–64, 2018.

[5] M. A. Adegboye, W.-K. Fung, and A. Karnik, "Recent advances in pipeline monitoring and oil leakage detection technologies: Principles and approaches," Sensors (Switzerland), vol. 19, no. 11, 2019, doi: 10.3390/s19112548.

[6] A. C. Caputo and P. M. Pelagagge, "Using neural networks to monitor piping systems," Process Saf. Prog., vol. 22, no. 2, pp. 119–127, 2003.

[7] J. Feng and H. Zhang, "Algorithm of pipeline leak detection based on discrete incremental clustering method," in International Conference on Intelligent Computing, 2006, pp. 602–607.

[8] M. Shinozuka, J. Liang, and M. Q. Feng, "Use of supervisory control and data acquisition for damage location of water delivery systems," J. Eng. Mech., vol. 131, no. 3, pp. 225–230, 2005.

[9] N. Behari, M. Z. Sheriff, M. A. Rahman, M. Nounou, I. Hassan, and H. Nounou, "Chronic leak detection for single and multiphase flow: A critical review on onshore and offshore subsea and arctic conditions," J. Nat. Gas Sci. Eng., vol. 81, p. 103460, 2020, pp. 1-9.

[10] A. Abdulshaheed, F. Mustapha, and A. Ghavamian, "A pressure-based method for monitoring leaks in a pipe distribution system: A Review," Renew. Sustain. Energy Rev., vol. 69, pp. 902–911, 2017.

[11] H. Zhu, P. Lin, and Q. Pan, "A CFD (computational fluid dynamic) simulation for oil leakage from damaged submarine pipeline," Energy, vol. 64, pp. 887–899, 2014.

[12] R. Doshmanziari, H. Khaloozadeh, and A. Nikoofard, "Gas pipeline leakage detection based on sensor fusion under model-based fault detection framework," J. Pet. Sci. Eng., vol. 184, p. 106581, 2020.

[13] H. Fu, L. Yang, H. Liang, S. Wang, and K. Ling, "Diagnosis of the single leakage in the fluid pipeline through experimental study and CFD simulation," J. Pet. Sci. Eng., vol. 193, p. 107437, 2020.

[14] S. Yang, K. Jeon, D. Kang, and C. Han, "Accident analysis of the Gumi hydrogen fluoride gas leak using CFD and comparison with post-accidental environmental impacts," J. Loss Prev. Process Ind., vol. 48, pp. 207–215, 2017.

[15] A. Ebrahimi-Moghadam, M. Farzaneh-Gord, A. Arabkoohsar, and A. J. Moghadam, "CFD analysis of natural gas emission from damaged pipelines: Correlation development for leakage estimation," J. Clean. Prod., vol. 199, pp. 257–271, 2018.

[16] E. J. Pérez-Pérez, F. R. López-Estrada, G. Valencia-Palomo, L. Torres, V. Puig, and J. D. Mina-Antonio, "Leak diagnosis in pipelines using a combined artificial neural network approach," Control Eng. Pract., vol. 107, p. 104677, 2021.

[17] H. Ding, F. C. Visser, Y. Jiang, and M. Furmanczyk, "Demonstration and validation of a 3D CFD simulation tool predicting pump performance and cavitation for industrial applications," in Fluids Engineering Division Summer Meeting, 2009, vol. 43727, pp. 277–293.

[18] M. A. Adegboye, A. M. Aibinu, J. G. Kolo, I. Aliyu, T. A. Folorunso, and S.-H. Lee, "Incorporating intelligence in fish feeding system for dispensing feed based on fish feeding intensity," IEEE Access, vol. 8, pp. 91948–91960, 2020.

[19] S. Hess, K. E. Train, and J. W. Polak, "On the use of a Modified Latin Hypercube Sampling (MLHS) method in the estimation of a Mixed Logit model for vehicle choice," Transp. Res. Part B Methodol., vol. 40, no. 2, pp. 147–163, 2006.

[20] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360), 1998, pp. 69–73.

[21] J. Zhang, S. Chowdhury, and A. Messac, "An adaptive hybrid surrogate model," Struct. Multidiscip. Optim., vol. 46, no. 2, pp. 223–238, 2012.

[22] A. Golzari, M. H. Sefat, and S. Jamshidi, "Development of an adaptive surrogate model for production optimization," J. Pet. Sci. Eng., vol. 133, pp. 677–688, 2015.

[23] P. Jiang, Q. Zhou, and X. Shao, "Verification methods for surrogate models," in Surrogate Model-Based Engineering Design and Optimization, Springer, 2020, pp. 89–113.

[24] B. K. Nuhu, I. Aliyu, M. A. Adegboye, J. K. Ryu, O. M. Olaniyi, and C. G. Lim, "Distributed network-based structural health monitoring expert system," Build. Res. Inf., vol. 49, no. 1, pp. 144–159, 2021.

[25] F. Li, W. Shen, X. Cai, L. Gao, and G. G. Wang, "A fast surrogate-assisted particle swarm optimization algorithm for computationally expensive problems," Appl. Soft Comput., vol. 92, p. 106303, 2020.