

A fault prognosis strategy for an external gear pump using Machine Learning algorithms and synthetic data generation methods

Kayal Lakshmanan^{a,b,*}, Fabrizio Tessicini^c, Antonio J. Gil^a, Ferdinando Auricchio^b

^a Faculty of Science and Engineering Swansea University, Bay Campus, SA1 8EN, Swansea, United Kingdom

^b Department of Civil Engineering and Architecture, University of Pavia, Via Adolfo Ferrata, 3, Pavia PV 27100, Italy

^c Fluid-o-Tech, Via Leonardo Da Vinci, 40, Corsico MI 20094, Italy

ARTICLE INFO

Article history:

Received 9 December 2022

Revised 4 May 2023

Accepted 3 July 2023

Available online 7 July 2023

Keywords:

Fault prognosis

Machine learning

High-fidelity model

Synthetic data generation

ABSTRACT

Fault prognosis is an important area of research that aims to predict and diagnose faults in complex systems. The sudden failure of industrial components can have adverse consequences for an organisation in terms of time, cost and workflow. It is, therefore, critical to ensure the maintenance of equipment components in optimal condition in order to avoid downtime that may cause significant disruption. Due to this reason, in recent years, there has been an increasing interest in creating innovative methods for fault prognosis that can increase system reliability and reduce maintenance costs. Gear pumps are widely utilised in a variety of industrial applications, and their reliability and effectiveness are crucial for achieving optimal system performance. Gear pumps, on the other hand, are prone to malfunctions and failures, which can result in substantial downtime and maintenance costs. The challenge is to develop a fault prognosis approach that is reliable and accurate enough to detect and diagnose defects in a gear pump before they cause system failures. This paper proposes a novel computational strategy for the fault prognosis of an external gear pump using Machine Learning (ML) approaches. Due to the unavailability of sufficient experimental data in the vicinity of failure mechanisms, a novel approach to generating a high-fidelity in-silico dataset via a Computational Fluid Dynamic (CFD) model of the gear pump in healthy and faulty working conditions is presented. However, considering the computational demand for rerunning the same CFD simulations, novel synthetic data generation techniques are implemented by perturbing the frequency content of the time series to recreate other working conditions and constructing degradation behaviour using linear and cubic interpolation methods for run-to-failure scenarios. The synthetically created datasets are used to train the underlying ML metamodel for fault prognosis. Two types of ML algorithms are employed for fault prognosis: Multilayer Perceptron (MLP) and Support Vector Machine (SVM) algorithms. A series of numerical examples are shown, allowing us to infer that the proposed modelling technique is feasible in an industrial setting and that employing the MLP algorithm delivers superior fault prognosis results when compared to SVM. Furthermore, datasets generated using the cubic interpolation method have

* Corresponding author.

E-mail addresses: k.lakshmanan@swansea.ac.uk (K. Lakshmanan), f.tessicini@f-lab.org (F. Tessicini), a.j.gil@swansea.ac.uk (A.J. Gil), auricchi@unipv.it (F. Auricchio).

lower prediction errors than datasets generated using the linear interpolation method due to the smoother degradation behaviour in the data.

© 2023 The Author(s). Published by Elsevier Inc.
This is an open access article under the CC BY license
(<http://creativecommons.org/licenses/by/4.0/>)

1. Introduction

Gear pumps are widely used in many industrial applications due to their reliability, simplicity, and efficiency. Due to a sophisticated and compact design, external gear pumps can easily fit inside complex industrial equipment, reducing manufacturing time and associated costs. However, gear pumps, like any mechanical system, are prone to faults and failures, which can negatively affect the industry in terms of time, cost, and workflow. To ensure optimal system performance and avoid costly failures, reliable and accurate fault prognosis methods that can detect and diagnose defects in gear pumps before they cause system failures are required. Fault prognosis is a branch of condition monitoring that entails forecasting and diagnosing failures in a system using sensor data and other pertinent data. As a result, it is critical to regularly examine the mechanical condition of the machine's health in order to avoid unexpected failure [1]. In general, there are two types of fault prognosis methods: model-based approaches, which utilise mathematical models of the system to anticipate and diagnose defects, and data-driven methods, which employ statistical analysis and machine learning algorithms to recognise patterns in data and identify faults [2]. Both approaches have been widely studied and applied in various industries, including aerospace, automotive, manufacturing, and power systems.

In equipment's prognosis, the most commonly used type is the prediction of time remaining before a failure occurs, given the equipment's current condition and its past performance profile [3]. The time left before observing a failure is designated as Remaining Useful Life (RUL). In some situations, especially when a failure is potentially catastrophic, it is more than desirable to predict the probability that equipment operates without fault or failure up to some future point in time (the next inspection interval), given the current equipment condition and its past operational profile [4]. In any case, the probability that equipment operates without a fault until the next inspection interval is a good reference point for a maintenance engineer to determine the appropriateness of the inspection interval [5,6]. However, this is not a popular method due to the limitation in gathering the required data for prediction. To perform prognosis, in addition to the condition monitoring data of equipment, obtaining the fault propagation process and information on the failure mechanism is necessary. However, gaining knowledge of the physical processes in the vicinity of failure is not always possible or straightforward [3]. The fault propagation process is usually tracked by a trending or forecasting model for certain condition variables. There are two methods to describe the failure mechanism. The first one assumes that failure depends only on the condition variables, which reflect the actual fault level and the predetermined boundary condition (i.e. the pre-established level at which failure will occur). The second method constructs a model for the failure mechanism using historically available data [3].

ML approaches are widely documented for performing fault prognoses in literature. Methodologies employed to predict the RUL of rolling bearings using the gated recurrent unit method and the Support Vector Machine (SVM) regression method are discussed in [7] and [8], respectively. SVM regression algorithm is extensively used for prognosis within a variety of applications [9–11]. Multilayer Perceptrons (MLP) with backpropagation is another method widely applied in the field of prognosis. Reference [12] proposed the application of an echo state network and a recurrent multilayer perceptron. Reference [13] took the approach of comparing the results of applying recurrent neural networks and neural-fuzzy inference systems to predict the failure of the machinery. Recent developments in prognosis have also used the deep neural network, recurrent neural network, or convolutional neural network [3,7,14–16] and have been applied within various industrial sectors.

There has been an increasing interest in creating innovative and improved fault prognosis methods for gear pumps in recent years. This is due to rising demand for reliable and effective pump systems, as well as advancements in sensor technology and data analytic techniques. Some of the recent research works in this area include: deep learning-based intelligent fault diagnosis approaches for rotating machinery [17], a fault diagnosis of an external gear pump using Machine Learning classification algorithms [18], a physics-constraint variational neural network for wear state assessment of the external gear pump proposed in [19], a sensor acquired vibration, flow rate, and pressure signals to establish a reliable fault prognostics framework of a gear pump subjected to different fuel contamination levels are studied in [20].

In order to perform fault prognosis using ML approaches, the gear pump's actual history/run-to-failure history needs to be monitored. In addition, ML predictive capability is strongly dependent upon the availability of large sets of high-fidelity data [21]. In case of absence (or limited availability) of experimental data, these can be superseded using high-fidelity in-silico data [22–24]. Due to the high computational demand in simulating high-fidelity data for similar events, synthetic data generation methods can be utilised to enhance the data available for fault prognosis. Moreover, synthetic data generation methods can be employed where data privacy can be of significant concern [25].

This Industry-Academia collaboration has sought to address one specific bottleneck: finding an efficient strategy to perform fault prognosis of an external gear pump with insufficient availability of experimental data. As a result, this work proposes a fault prognosis methodology using supervised ML algorithms (i.e. MLP and SVM) and synthetic data generation

methods to enhance high-fidelity data obtained from an accurate CFD model resembling the external gear pump operations in healthy and various fault conditions (speed variations, radial gap variations, viscosity variations, etc.). Specifically, pressure, flow rate, and torque are monitored for several working conditions of the pump to develop a fault-prognostic framework of a gear pump. Additionally, synthetic data generation techniques are implemented by perturbing the frequency content of the time series to recreate other working conditions and constructing degradation behaviour using linear and cubic interpolation methods for run-to-failure scenarios. Following an extensive series of results, it is shown that the proposed method is very useful in situations where there is limited or insufficient experimental data.

Following this introductory section, the paper is organised as follows: [Section 2](#) introduces the generation of high-fidelity in silico data using a Computational Fluid Dynamics (CFD) model of the gear pump. [Section 3](#) describes two efficient ML algorithms, namely MLP and SVM. [Section 4](#) presents the implementation of a synthetic data generation method and data labelling. [Section 5](#) presents the complete framework and a comprehensive set of numerical examples. [Section 6](#) summarises some concluding remarks. [Appendix A](#) presents the mathematical model and governing equations used for performing the CFD Simulations.

2. High-fidelity data generation using computational fluid dynamics simulation of an external gear pump

External gear pumps are extensively utilised for moving fluids such as oil, fuel, and lubricants in a variety of industrial applications. Simulating these pumps is critical in building efficient and reliable systems, and numerous approaches are used to achieve this. The analytical approach involves solving mathematically simplified equations to simulate the pump's behaviour. This approach is useful for simple systems and can yield fast solutions. However, it is not suitable or generalisable to more complex systems that involve multiple parameters [26]. Another approach involves Computational Fluid Dynamics (CFD), which predicts the pump's behaviour using high-fidelity numerical simulations. CFD can provide detailed information about the flow dynamics and pressure distribution within the pump, making it useful for complex systems. CFD simulations, on the other hand, can be computationally expensive and time-consuming [27]. The experimental approach is a third strategy that involves physically testing the pump and measuring its performance characteristics. This procedure can yield accurate and trustworthy data, but it can be expensive, time-consuming and, in some cases, unfeasible [27].

In this study, the CFD model has been built using the commercial code 'Simerics MP+' that numerically solves the fundamental conservation equations of mass, momentum, and energy and includes accurate physical models for turbulence and cavitation [28,29]. The mathematical model and governing equations are detailed in [Appendix A](#).

2.1. Geometry of the external gear pump

Gear pumps are rotating positive displacement pumps, where the fluid is displaced from the inlet side of the pump to the outlet side of the pump due to the intermeshing of the drive and the slave gear's teeth. The geometry of the external gear pump, which is of main interest to this project, is shown in [Fig. 1](#). Some specific features of the geometry have been hidden due to commercial sensitivity.

The gears are located inside a tight, leak-free casing and connected to the lubrication channel through the top channel with rotating shafts. Similarly, the backside of the gear is connected to the magnetic drive mechanism via a bottom channel with rotating shafts, and the magnet is connected to the rest of the structure of the pump. The inlet and outlet ports are connected to the sides of the gear. The inlet and outlet sensors are placed in the inlet and outlet ports to monitor pressure from the CFD simulation. The outlet sensor is placed away from the end to avoid reversing the flow and accurately measure the pressure distribution in the outlet port. During the operation, these types of pumps are noiseless, pulsation-free, and capable of handling liquids up to 95 °C with pressure up to 12 bar and flow rate up to 210 l/h [30]. Due to commercial sensitivity, exact inner geometry and dimensions cannot be reported. However, simulations can be replicated to any desired geometry of the external gear pump.

2.2. Model attributes

The fluid domain of the external gear pump is extracted from the Computer-Aided Design (CAD) of the pump, which is provided by the industrial partner F-Lab and the geometry is imported into Simerics MP+. The mesh of the entire fluid domain is generated using high-quality structured and unstructured hexahedral cells in Simerics MP+, shown in [Fig. 2](#).

The binary tree unstructured mesh is utilised to create hexahedral cells for highly curved regions or narrow-cut regions. In addition, Simerics MP+ has a specialised tool for the meshing of the internal geometry of the gear using a rotor gear template mesh, producing high-quality structured hexahedral cells [29].

In order to perform the CFD analysis of this gear pump, the conservation of mass and momentum equations are solved along with turbulence and cavitation model equations [28]. For completeness, the equations for the conservation of mass, momentum, turbulence, and cavitation model are summarised in [Appendix A](#). For pump flows, cavitation is often encountered and is usually considered by the industry as an important design factor that must be accounted for [29], as it can potentially damage blades and, in this context, gear teeth. The turbulence model is essential to accurately predict the flow characteristics within the pump. Turbulent flow is characterized by fluctuations in velocity and pressure, where these fluctuations can have a significant impact on pump performance, such as increasing the pressure drop and decreasing the pump's

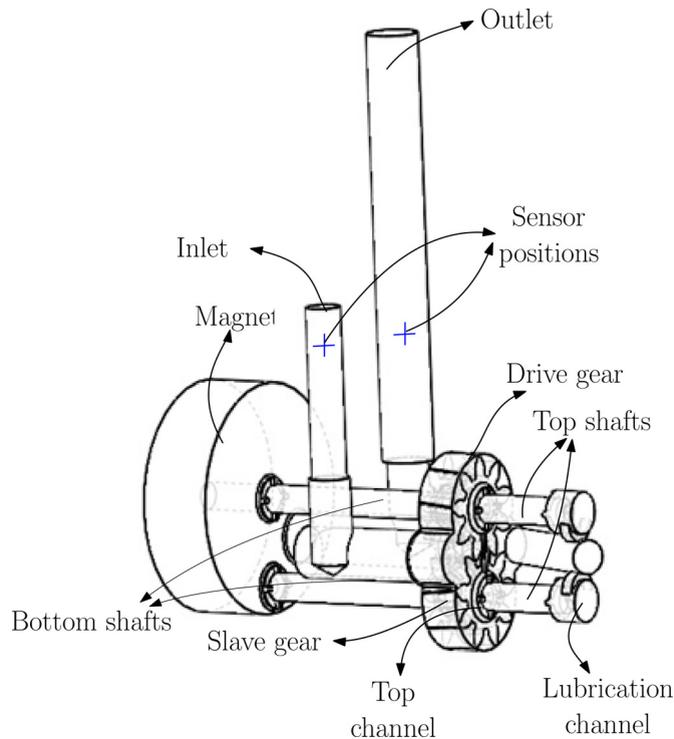


Fig. 1. Geometry of the external gear pump. Specific features of the geometry and the measurements have been hidden due to commercial sensitivity.

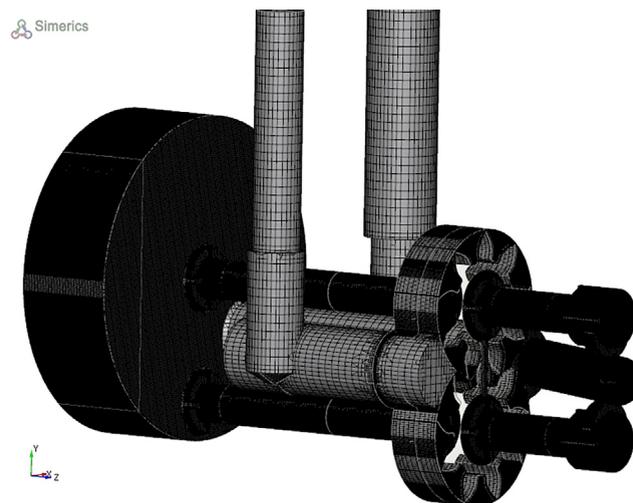


Fig. 2. Mesh of the external gear pump (the entire computational domain) created using Simerics MP+.

efficiency. The simulation of the external gear pump running in healthy conditions applies to a high viscous Newtonian ink-type fluid with properties detailed in Table 1. Boundary conditions are provided for flow, turbulence and cavitation variables in Table 2. Notice that although the energy equation is not included as part of the governing equations for simulation, that is, the simulation is run in isothermal mode, some materials parameters, such as the gas density within the cavitation model, are temperature dependent [28], which is the reason of the inclusion of the temperature in Table 1. Following advice from the industrial partner, the consideration of temperature gradients is not applicable for the pump under study, and isothermal conditions hold.

Table 1

Properties of a healthy (normal working condition) gear pump.

Operating conditions of the gear pump	
Speed	1450 rpm
Radial gap	0.03 mm
Number of drive gear teeth	9
Number of slave gear teeth	9
Material properties of the gear pump	
Viscosity	0.0383 Pa-s
Density	800 kg/m ³
Temperature	300 K

Table 2

Boundary conditions of flow variables, turbulence, and cavitation models.

	Inlet	Outlet	Wall
<i>Flow:</i>			
Pressure	1.01325 bar	3.15 bar	Zero gradient
Velocity	Zero gradient	Zero gradient	No-slip condition
<i>Turbulence:</i>			
Kinetic energy	0.01 m ² /s ²	0.01 m ² /s ²	Standard wall function
Dissipation rate	1 m ² /s ²	1 m ² /s ²	Standard wall function
<i>Cavitation:</i>			
Vapour mass fraction	0	0	Zero gradient

Table 3Mesh sensitivity analysis: execution time is in a dimensionless quantity, where h_1 is used as reference.

	h_1	h_2	h_3	h_4
No of elements	44,518	247,495	446,393	4,605,699
Max cell size	0.1	0.07	0.03	0.018
Min cell size	0.005	0.005	0.002	0.001
Execution time	1	3.24	7.57	182.76

2.3. Mesh sensitivity analysis

A mesh sensitivity analysis is carried out, whereby finer discretisations are used until a threshold of accuracy is passed as moving from one model to the next [31]. For simplicity and to optimise the number of cells, a simpler geometry is considered with the inlet port, the outlet port and the external gear in Fig. 1. Simulations are performed for fixed time step¹ with four different grid sizes (h_1, h_2, h_3, h_4) ranging from coarsest to finest mesh. The properties of the different grid sizes and their computational time are displayed in Table 3.

Figure 3 a displays the overall flow rate time history for the four discretisations. It can be clearly seen that the solution converges as the grid is refined, with h_3 and h_4 grid solutions showing hardly any distinguishable difference. Figure 3b depicts the absolute relative error of grids h_1 to h_3 with respect to the reference grid h_4 . This error is computed by calculating the difference for a given time step between dimensionless quantities of interest, specifically, overall torque, overall flow rate and pressure at the sensor spatial location. It can be clearly seen that the error decreases drastically as the mesh is refined. All in all, h_3 results provide the best balance between accuracy and computational time. As an outcome of this investigation, all CFD analyses are carried out using mesh size h_3 .

2.4. CFD model validation against experimental data

To validate the CFD results, the industrial partner provided the experimental results for the exact external gear pump (Fig. 1) with the same specifications and material descriptions as the CFD numerical model described in Section 2.2. Due to commercial sensitivity, the experimental activities cannot be fully disclosed, but relevant information is provided next. The experiments were conducted by fixing speed and viscosity values for different outlet pressure values (1, 2, . . . , 9 bar). In total, 21 experiments were conducted for the same gear pump. The inlet pressure is assumed to be atmospheric pressure (1.01325 bar) for all the experiments and the outlet pressure is varied for each experiment. In particular, the experiments were performed for two different rotational speed values of the gear, 1450 rpm and 2900 rpm. For the 1450 rpm speed, two different viscosity values were used, 0.001 Pa-s and 0.075 Pa-s, and for the 2900 rpm speed, three different viscosity

¹ The size of the time step is sufficiently small to remove possible error sources stemming from the time discretisation.

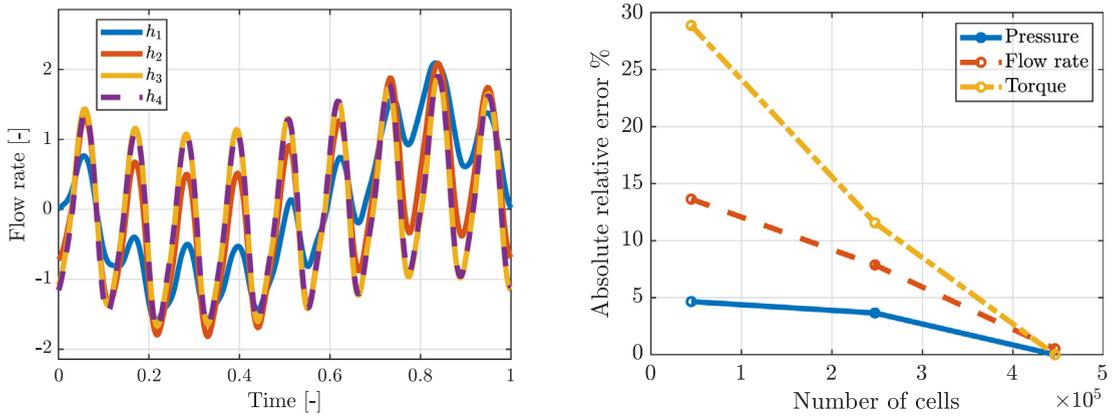


Fig. 3. Overall flow rate time history for the four discretisations (a), and the absolute relative error (%) of grids h_1 to h_3 with respect to the reference grid h_4 (b).

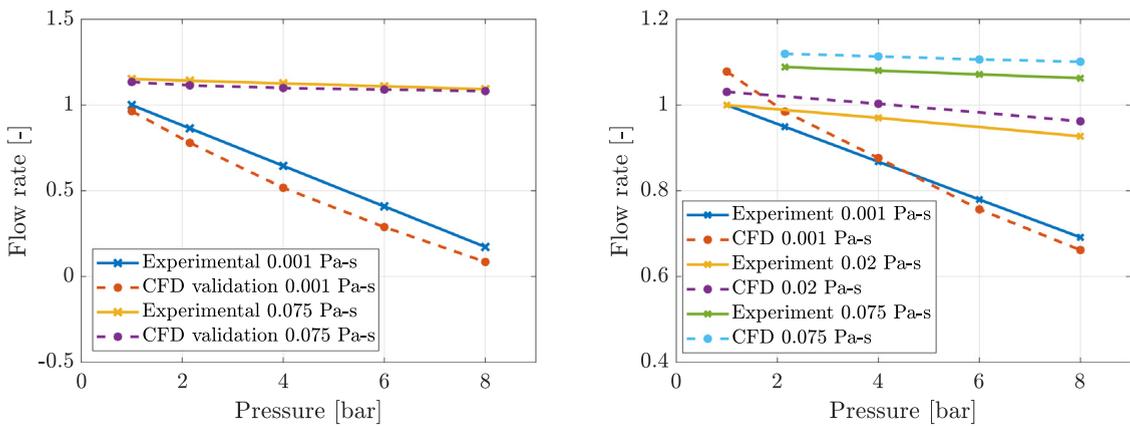


Fig. 4. Standardised mean discharge flow rate of experimental results against CFD model of an external gear pump with the speed of 1450 rpm (a) and 2900 rpm (b) for different relative pressure values. In the x-axis, the values are relative pressure calculated using the difference between outlet and inlet pressure.

values, 0.001 Pa-s, 0.02 Pa-s and 0.075 Pa-s. CFD simulations match experimental values very closely, providing confidence in subsequent developments.

The discharge flow rate, defined as the time-varying surface integral of the outflow flux, is computed at the far end of the pump simulation model. Subsequently, this magnitude is time-averaged and displayed in Fig. 4a and b. Figure 4a shows the mean experimental discharge flow rate values compared with the CFD simulation model for different pressure values and different viscosity values (0.001 Pa-s and 0.075 Pa-s) of a gear pump with the rotating speed of 1450 rpm. Figure 4b shows the mean experimental discharge flow rate values compared with the CFD simulation model for different pressure values and different viscosity values (0.001 Pa-s, 0.02 Pa-s and 0.075 Pa-s) of the gear pump with a rotating speed of 2900 rpm, respectively. From Fig. 4a and b, it can be seen that the simulation results are in very good agreement with the experiments. Also, it can be noted that an increase in the viscosity value provides a higher discharge flow rate. After validation for various viscosity values and two different speed values in the CFD simulations with experimental values, we now move to build potential fault scenarios which can cause the pump to degrade or even to fail, and this is discussed in the following section.

2.5. Implementation of fault conditions in the external gear pump

First, the results of an external gear pump's healthy condition are shown, followed by the implementation of all fault scenarios. A CFD simulation is performed following the operational conditions and material properties in Section 2.2. The gear pump is run for five gear revolutions (1450 rpm) with an atmospheric (1.01325 bar) inlet pressure and outlet pressure of 3.15 bar. The first revolution is for the stabilisation of the flow, and the second to fifth revolutions are considered for results. For post-processing and data acquisition, pressure at the outlet sensor (sensor is shown in Fig. 1), the discharge flow

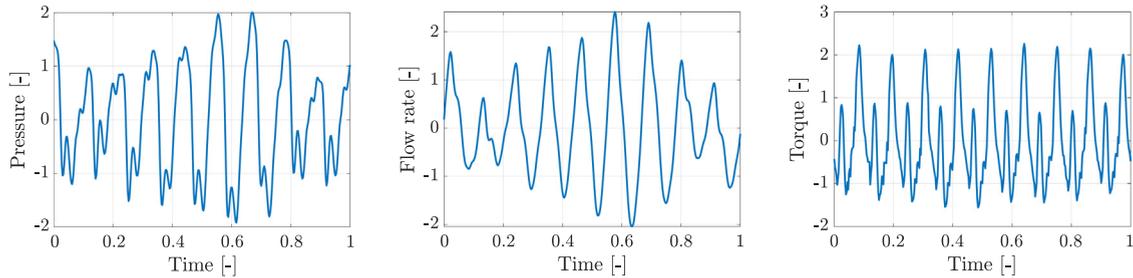


Fig. 5. External gear pump running in a healthy condition; standardised and fully developed pressure (a), flow rate (b) and torque (c) with respect to time for one revolution.

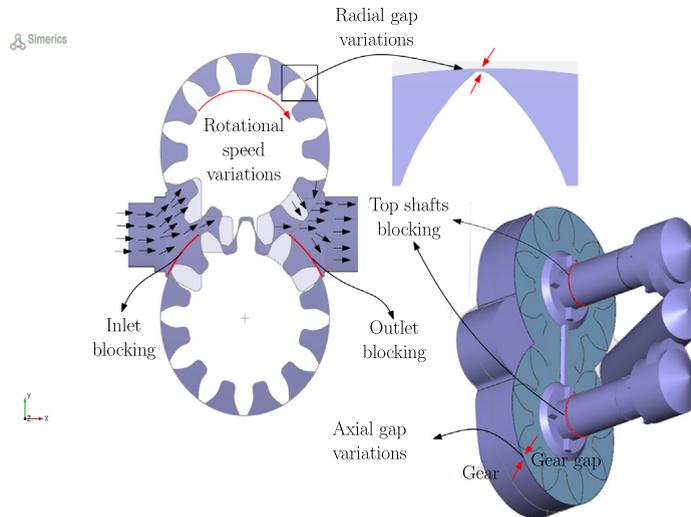


Fig. 6. Implemented fault scenarios considering the healthy working conditions of the gear pump as a reference.

rate at the far end of the outlet, and the sum of the drive and slave gear's torque have been considered as primary sensor variables monitored from the CFD simulation. In this study, the variables are monitored in the same way for all simulations.

Figure 5 shows the standardised² and fully developed outlet pressure, discharge flow rate and torque of the rotating gear for one revolution (x -axis is normalised between 0 and 1). The low-frequency contribution is due to the pump rotation and is very consistent with the pump's rotational frequency. High-frequency contributions are due to gear engagement and disengagement. The number of oscillations shown in Fig. 5a–c corresponds to the number of teeth in the gear.

Utilising the validated CFD simulation models of a gear pump as a reference, a variety of working conditions, including several fault scenarios, are recreated based on the experience of the industrial partner. Several fault scenarios are considered: radial gap degradation, speed variations (validated experimentally), viscosity variations (validated experimentally) and temperature variations. Although simulations are run in isothermal mode, some of the fluid material properties are temperature dependent, and that is the reason why the study of the influence of different temperature conditions is relevant. These scenarios are considered to understand the pump's complete behaviour when running in an abnormal condition. Figure 6 shows the fault scenarios implemented, and Table 4 shows the fault variables and the different cases considered (in percentage). The percentage is calculated considering the healthy working conditions parameters of the gear pump as a reference. For ease of simplicity, different cases are considered in increasing trend (i.e. higher values represent higher deviation with respect to the healthy scenario). For example, the temperature variation is increased by 2.5%, 5% and 7.5%. So three different simulations are simulated by only changing the temperature, and every other parameter is considered as in the healthy case simulation. For all fault scenarios, 15 simulations were run in all cases, and they were not conducted in parallel. Each fault scenario is simulated using the boundary conditions and material properties described in Section 2.2, and they are presented below.

For instance, fault scenarios due to radial gap degradation are created based on the concept that mass particles can enter the pump, causing the gear's teeth to erode. The radial gap faults are generated by grinding all the gears teeth and re-

² Standardised refers to z-score normalisation and is expressed as $(x_i - \mu)/\sigma$, where x_i is the i th entry in \mathbf{x} , and μ and σ are mean and standard deviation of \mathbf{x} [32].

Table 4

Implemented fault scenarios; the percentage is calculated considering the healthy working conditions parameters of the gear pump as a reference.

Fault scenarios	Different cases
Radial gap degradation (G)	66.66%, 133.33%, 200%
Axial gap degradation (A)	2.5%, 5%, 7.5%
Speed variations (S)	1.78%, 3.57%, 5%
Viscosity variations (V)	0.8%, 1.6%, 2.4%
Temperature variations (T)	2.5%, 5%, 7.5%

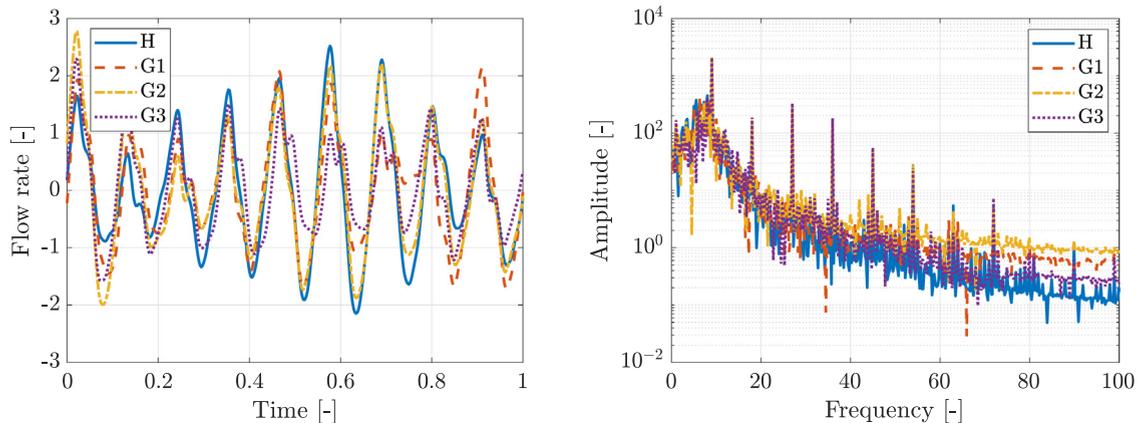


Fig. 7. Discharge flow rate with respect to the time (a) and the frequency (b) for fault scenarios due to radial gap degradation.

running appropriate CFD simulations after modifying the geometry of the model. This step-by-step degradation behaviour is generated, namely G1, G2 and G3, to understand the erosion behaviour on the gear teeth. The standardised discharge flow rate of the scenarios (G1, G2, G3) compared with the healthy scenario (H) in the time domain and frequency domain is shown in Fig. 7. By increasing the gap size, the mean flow rate decreases. Since the gap between the gear and gear gap increases, i.e., gear teeth size decreases, the gear pumps a lower amount of fluid comparatively, reducing the flow rate. With respect to the healthy scenario, the mean flow rate of G1, G2 and G3 faulty scenarios is decreased by 21%, 40% and 56%, respectively.

The fault scenarios due to axial gap degradation are created in order to analyse the effect of flow leakage in the fluid volume, and they are generated by increasing the gap between the gears and the gear casing (side leakage). Similar to radial gap fault scenarios, the discharge flow rate decreases with the incremental increase of axial gap size. Concerning the healthy scenario, the mean flow rate of axial gap variations (A1, A2 and A3) is decreased by 2%, 4% and 6.3%, respectively.

The fault scenarios due to speed variations are generated by increasing the rotational speed (in RPM) of the gear pump to comprehend the flow behaviour. The speed variations are created to validate that for positive displacement pumps, as the rotational speed increases, the discharge flow rate increases in a linear trend, which is proven in [29] for an external gear pump. Considering the healthy scenario, the mean flow rate of speed variations (S1, S2 and S3) is increased by 2%, 4.5% and 6.3%, respectively.

The fault scenarios due to viscosity variations are generated to understand the pump performance with viscosity change. Gear pumps are best suitable for high-viscosity applications. When the fluid viscosity decreases, slip increases. The slip takes place when the pump recirculates from the outlet to the inlet side as it escapes through machined clearances while the pump tries to progress forward. The amount of slip is determined by the fluids viscosity and discharge pressure. A thin fluid with a low viscosity can squeeze through the clearances more easily than thicker fluids. High discharge pressure can force thinner fluids back through the pump, also causing more slip. Therefore, because thin fluids slip, the pumps efficiency decreases as less product progresses with the forward movement, hence the lower discharge flow rate. As a result, the mean flow rate of viscosity variations (V1, V2 and V3) is increased by 0.4%, 1% and 1.5%, respectively, to the healthy scenario.

The fault scenarios due to temperature variations are generated to understand the changes in the gear pump's performance with respect to temperature variations. With the pump operation, fluid viscosity drops in nature. With time, any working fluid would lose part of its viscous effect due to natural degradation, but the temperature would remain the same, and some of the fluid material properties related to cavitation are temperature dependent [28]. To replicate that, separate viscosity and temperature variations were conducted. Considering the healthy scenario, the mean flow rate of temperature variations (T1, T2 and T3) is decreased by 0.16%, 0.25% and 0.33%, respectively. Other fault scenarios mentioned in Table 4 are simulated, and the results in the form of time series are extracted.

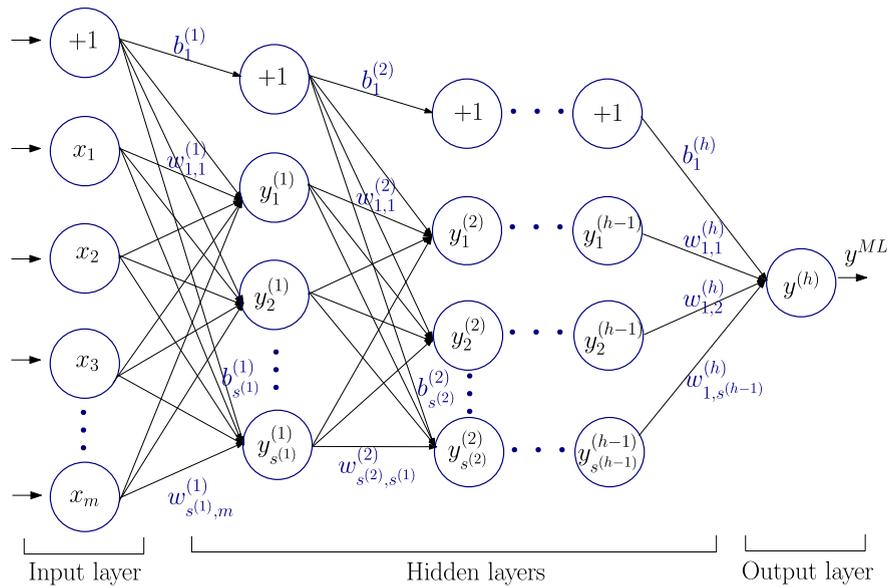


Fig. 8. Fully connected MLP architecture.

3. Machine learning algorithms for fault prognosis

Supervised Machine Learning (ML) algorithms are typically divided according to their input and output data. Supervised learning algorithms usually include classification and regression, with the latter used for predicting continuous quantities and the former to predict discrete values. An extensive study of supervised ML algorithms is given in [33]. In the present work, two types of ML regression algorithms will be utilised and compared, namely Multilayer Perceptron (MLP) and Support Vector Machine (SVM). For regression problems, the metric used for training (or testing) during the offline phase is the R2 score (measure the proportion of variance in the dependent variables that is predictable from the independent variables in a regression model) and Mean Squared Error (MSE) of the predicted output values \hat{y}_i against the true output values y_i ,

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2. \tag{1}$$

3.1. Multilayer perceptron

Multilayer Perceptron (MLP) is a class of feedforward Artificial Neural Networks (ANN). The study of ANNs has been inspired by observing that biological learning systems are built of very complex networks of interconnected neurons. The average human brain consists of nearly 10^{11} neurons of various types, with each neuron connected on average to 10^4 others [34]. However, the fastest neuron switching times are in the order of 10^{-3} s, which is relatively slow compared to the computer switching speed of 10^{-10} s. Yet, humans can make complex decisions surprisingly fast. This observation has hypothesised that the information-processing abilities of biological neural systems must follow highly parallel processes operating on representations that are distributed over many neurons. A motivation for ANN is to capture this kind of highly parallel computation based on distributed representations [35].

The MLP architecture is presented in Fig. 8, where there is an input layer (first layer), an output layer (last layer), and the layers between them are called hidden layers. A hidden layer is comprised of neurons, and the outputs of these neurons are the input to the next layer until the final output is achieved. There are $h - 1$ hidden layers shown in Fig. 8.

For the forward propagation of MLP, the value related to each neuron can be calculated by utilizing the weights of the connections \mathbf{W} (vector represented), activation functions ϕ and the values associated with the output of the previous layer. The output of the first layer in Fig. 8 can be computed as

$$\mathbf{y}^{(1)} = \phi^{(1)}(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}), \tag{2}$$

where \mathbf{x} is an input, \mathbf{y} is an output, \mathbf{b} is referred to as bias and $\star^{(1)}$ denotes the first layer of the variable \star . Likewise, for the h th layer

$$\hat{\mathbf{y}} = \mathbf{y}^{(h)} = \phi^{(h)}(\mathbf{W}^{(h)}\mathbf{y}^{(h-1)} + \mathbf{b}^{(h)}), \tag{3}$$

where $\hat{\mathbf{y}}$ is the output of the last layer. When the output layer has only one neuron, like in Fig. 8

$$y^{ML} = y^{(h)} = \phi^{(h)}(\mathbf{W}^{(h)}\mathbf{y}^{(h-1)} + b_1^{(h)}). \tag{4}$$

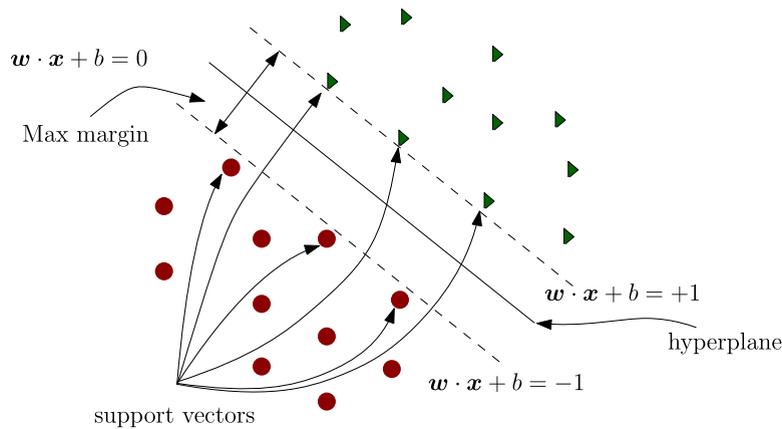


Fig. 9. SVM architecture for the linearly separable case.

The activation function $\phi^{(h)}$ determines the output of the neural network. The function takes the net input signal $\mathbf{W}^{(h)}\mathbf{y}^{(h-1)} + \mathbf{b}^{(h)}$, then compresses the values between a certain range and outputs the signal. There are different types of activation functions that are used in ANN, depending on the application. Some of the common activation functions are the sigmoid function, hyperbolic tangent sigmoid function or tanh function, and Rectified Linear Unit function (ReLU) [34]. When MLP is used for multiclass classification, another activation function called softmax function is used in the MLP network [36].

For the last or output layer, the error between the predicted and the ground truth value can be calculated using a loss function (also known as a cost function) depending on the specific type of application. In the case of classification, the cross-entropy loss function is preferred and it is given by

$$J(\mathbf{W}) = \frac{1}{n} \sum_{j=1}^n [y_j \log(\hat{y}_j) + (1 - y_j) \log(1 - \hat{y}_j)] + \frac{\lambda}{2n} \sum_{l=1}^h \|\mathbf{W}^{(l)}\|^2, \tag{5}$$

where λ is a regularisation parameter. The first part of the right-hand side of Eq. (5) is a cross-entropy loss function, and the second part is a regularisation term that addresses overfitting. This loss function is trained by a backpropagation learning algorithm [37]. Backpropagation is a method to optimise the connection weights starting from the output layer and propagating backwards by adapting the weights, layer by layer, compensating for error during the training, and efficiently dividing the error among the connections. Technically, backpropagation efficiently computes the gradient of the loss function associated with a given state with respect to the weights. Backpropagation employs the gradient descent method to minimise the squared error between the network output values and desired values for those outputs. This procedure is repeated for MSE in the case of regression.

3.2. Support vector machine

Support Vector Machine (SVM) is one of the most widely used classification algorithms. SVM provides a globally optimal solution for any binary/multi-class classification problems. For the case of a linear binary classification problem, a decision hypothesis \mathcal{H} is defined as

$$\mathcal{H} = \{\mathbf{x} \mapsto \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) : \mathbf{w} \in \mathbb{R}^m, b \in \mathbb{R}\}. \tag{6}$$

where \mathbf{w} is the weight vector, b is the bias and the equation of the hyperplane is defined as

$$\mathbf{w} \cdot \mathbf{x} + b = 0. \tag{7}$$

The hypothesis \mathcal{H} labels all the data points either positively (on one side of the hyperplane) or negatively (on the other side of the hyperplane). When a given sample dataset is linearly separable, then there exists a hyperplane that perfectly separates the training samples into two classes. This is equivalent to the existence of $(\mathbf{w}, b) \in (\mathbb{R}^m - \{0\}^m \times \mathbb{R})$ such that

$$y_j(\mathbf{w} \cdot \mathbf{x}_j + b) \geq 0, \quad \forall j = 1, \dots, n, \tag{8}$$

where the parenthesis indicates a multiplication. In Fig. 9, there can be infinitely many separating hyperplanes for the given problem. The SVM algorithm selects the optimal hyperplane based on the Maximum margin and then it is converted into a convex optimisation problem, given as

$$\begin{aligned} &\underset{\mathbf{w}, b}{\text{minimise}} && \frac{1}{2} \|\mathbf{w}\|^2, \\ &\text{subject to} && y_j(\mathbf{w} \cdot \mathbf{x}_j + b) \geq 1, \quad \forall j = 1, \dots, n. \end{aligned} \tag{9}$$

This optimisation problem admits a unique solution, an important property that does not hold for other learning algorithms [38]. Figure 9 illustrates the maximum-margin hyperplane returned by the SVM solution for the separable case. It also shows the marginal hyperplanes, which are the hyperplanes parallel to the separating hyperplane and passing through the closest points on the negative and positive sides, which are also called support vectors.

When a given sample dataset is non-separable, which implies that for any hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0, \exists \mathbf{x}_j$ such that

$$y_j(\mathbf{w} \cdot \mathbf{x}_j + b) \not\geq 1, \tag{10}$$

then constraints for linear separable cases cannot hold. $\forall j = 1, \dots, n, \exists \zeta_j \geq 0$ such that

$$y_j(\mathbf{w} \cdot \mathbf{x}_j + b) \geq 1 - \zeta_j. \tag{11}$$

The slack variable ζ_j measures the distance by which \mathbf{x}_j violates the inequality. Similar to the separable case, the optimisation problem is now defined as

$$\begin{aligned} & \underset{\mathbf{w}, b, \zeta}{\text{minimise}} && \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{j=1}^n \zeta_j^p, \\ & \text{subject to} && y_j(\mathbf{w} \cdot \mathbf{x}_j + b) \geq 1 - \zeta_j \wedge \zeta_j \geq 0, \quad \forall j = 1, \dots, n, \end{aligned} \tag{12}$$

where the regularisation parameter $C \geq 0$ determines minimisation of $\|\mathbf{w}\|^2$ and the minimisation of the slack penalty $\sum_{j=1}^n \zeta_j^p$, for some $p \geq 1$ [38].

For a more generic non-linear classification problem, kernel functions \mathcal{K} are used to map the training sample data into a higher dimensional space, where the classification problem transforms into a linear one. The kernel function defined over \mathcal{X} is $\mathcal{K} = \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$.

When a given sample dataset is non-linear or high dimensional, then kernel functions \mathcal{K} are used to non-linearly map the input space to high-dimensional space, where the linear separation is possible. The kernel function defined over \mathcal{X} is $\mathcal{K} = \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ [38] and

$$\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}, \mathcal{K}(\mathbf{x}_1, \mathbf{x}_2) = \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2) \rangle, \tag{13}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. Different kernel functions such as linear, polynomial, Gaussian and sigmoid, which are widely used in SVM for non-linear classification and formulations of these kernel functions, can be found in [38].

Similar to the SVM classification, SVM regression has the same properties of the margin maximisation and kernel for non-linear mapping with only a minor difference [39,40]. The SVM regression function approximates all pairs of input and output (\mathbf{x}_j, y_j) while maintaining the differences between estimated values and real values under *epsilon* (ϵ) precision. ϵ is called a margin of tolerance. That is, for every input \mathbf{x}_j in S ,

$$y_j - (\mathbf{w} \cdot \mathbf{x}_j + b) \leq \epsilon. \tag{14}$$

Recalling Eq. (9), by minimising $\|\mathbf{w}\|^2$ to maximise the margin, the training in SVM regression becomes a constrained optimisation problem.

$$\begin{aligned} & \underset{\mathbf{w}, b}{\text{minimise}} && \frac{1}{2} \|\mathbf{w}\|^2, \\ & \text{subject to} && y_j - (\mathbf{w} \cdot \mathbf{x}_j + b) \leq \epsilon, \quad \forall j = 1, \dots, n. \end{aligned} \tag{15}$$

The solution to this problem follows similar steps to solving an SVM classification problem.

3.3. Optimisation of hyper-parameters

Most ML and deep learning algorithms have some parameters that need to be adjusted, called hyper-parameters. The hyper-parameters are optimised during the training of ML algorithms as they are crucial and directly control the performance of the algorithm. A good choice of hyper-parameters is essential in order to build a robust and accurate model, thus, preventing the model from overfitting or underfitting [41]. Although several automatic optimisation techniques exist, they have different strengths and drawbacks when applied to different types of problems [42].

The main parameters for MLP regression are weights, activation functions, number of hidden layers and neurons. The weights have been adjusted by the backpropagation algorithm according to the loss function specified when compiling the model. The backpropagation algorithm requires that the network is trained for a specified number of epochs of the training dataset. Each epoch can be partitioned into groups called batches. The batch size defines the number of patterns that the network is exposed to before the weights are updated within an epoch. It is common practice to show the model loss with respect to the number of epochs. This plot can detect if the model is overfitting, underfitting or suitably fitting the training dataset [43]. In addition, during training, an *Early Stopping* argument is used in order to stop the training when the chosen performance measure stops improving [44]. Hidden layers and hidden layer neurons play a vital role in the performance of the backpropagation neural network [45]. For the MLP regression algorithm, the activation functions (tanh, elu or Relu) and the number of hidden neurons are optimised using a random search algorithm [46]. Table 5 shows the optimised hyperparameters for the MLP model. The hidden layer is approximated by the model’s performance.

Considering SVM regression, the most important hyper-parameters are the type of kernel functions, the kernel scale, the regularisation parameter C in Eq. (12) and the tolerance margin ϵ . This is performed using a random search algorithm [46]. Table 6 shows the optimised hyperparameters for the SVM regression model.

Table 5
Hyperparameters of the MLP model.

Hyperparameters	Value
Number of neurons	8, 16, 32, 664, 128, 256, 512
Activation function	Relu, elu, tanh
Learning rate	$1e^{-3}$, $1e^{-4}$, $1e^{-5}$
Batch size	8, 16, 32, 64, 128
Optimiser	Adam, SGD, RMSprop

Table 6
Hyperparameters of the SVM model.

Hyperparameters	Value
Kernel function	linear, polynomial, sigmoid, Gaussian
Kernel scale	1, 2, 3
Margin of tolerance	0.1, 0.2, 0.3

4. Data generation methods

In order to improve the learning phase of any ML algorithm, an essential requirement is to have a large quantity of data. Due to the insufficient amount of experimental data in the current work context, the CFD model has been used to generate high-fidelity data by recreating a variety of working conditions for the gear pump. However, generating a CFD simulation of one identical case several times can drastically increase the computational time, so synthetic data must be generated. In this work, 16 different CFD simulations were used, consisting of one healthy scenario and 15 fault scenarios, as mentioned in Table 4. Time-varying values of pressure, flow rate, and torque monitored from the CFD simulations are used as primary high-fidelity data for fault prognosis. This high-fidelity data is perturbed using a noise perturbation method to recreate possible environmental variations of the working pump conditions; this is only generated to increase the data availability for ML algorithms. The noise perturbation method is presented below.

4.1. Synthetic data generation method using a noise perturbation method

Consider $x(t)$ a continuous time-varying field (i.e. pressure, flow rate, etc.) in the time interval $[0, T]$

$$x : [0, T] \subset \mathbb{R} \rightarrow \mathbb{R} \\ t \mapsto x(t), \tag{16}$$

where $x(t)$ is the value of the field at time t . Let us define a discrete set (\mathcal{D}) of time values being chosen for the sampling evaluation of the field x in time, namely

$$\mathcal{D} = \{t_1, t_2, \dots, t_m\}; \quad t_i = (i - 1)\Delta t, \quad i = 1, \dots, m; \quad \Delta t = \frac{T}{m - 1}. \tag{17}$$

Thus, the discrete set \mathbf{x} is given by,

$$\mathbf{x} = [x(t_1), x(t_2) \dots x(t_m)]^T, \quad \mathbf{x} \in \mathbb{R}^m. \tag{18}$$

The discrete set \mathbf{x} is converted to the frequency domain using a Discrete Fourier Transform (DFT) ($\tilde{\mathcal{F}}_{\Delta t}$), and it is expressed as

$$\hat{\mathbf{x}} = \tilde{\mathcal{F}}_{\Delta t}(\mathbf{x}) = [\hat{x}(\omega_1), \hat{x}(\omega_2), \dots, \hat{x}(\omega_m)]^T, \quad \hat{\mathbf{x}} \in \mathbb{R}^m, \tag{19}$$

where $\hat{\mathbf{x}}$ represents the discrete approximation of the Fourier transform for a set of discrete frequencies $\mathcal{W} = \{\omega_1, \omega_2, \dots, \omega_m\}$. The objective is to perturb the DFT $\hat{\mathbf{x}}$ series by adding noise ϵ in a selected group of frequencies contained in a set $\tilde{\omega} \subset \mathcal{W}$. Specifically,

$$\epsilon : \tilde{\omega} \subset \mathcal{W} \rightarrow \mathbb{R} \\ \tilde{\omega}_j \mapsto \epsilon(\tilde{\omega}_j). \tag{20}$$

As an example, the set $\tilde{\omega}$ is defined on the basis of local maxima observed frequencies contributing to the frequency spectrum. Figure 10 shows pressure as a function of frequency (DFT) in normal working conditions of the gear pump from the CFD simulation. The main frequency contribution represents the rotational speed, and the small fluctuations are related to gear engaging and disengaging. This figure is shown as an example to demonstrate where the noise is added. For a given monitored field x , where the set of selected frequencies $\tilde{\omega} = \{\tilde{\omega}_1, \tilde{\omega}_2, \dots, \tilde{\omega}_{10}\}$ can be observed.

The objective is to add noise in the response of the DFT with larger noise content for higher frequencies. The noise is added to one amplitude at a time. One of the ways to achieve this is by considering a monotonically increasing quadratic function,

$$\epsilon(\tilde{\omega}_j) = p_1\tilde{\omega}_j^2 + p_2\tilde{\omega}_j + p_3, \quad j = 1, 2, \dots, n, \tag{21}$$

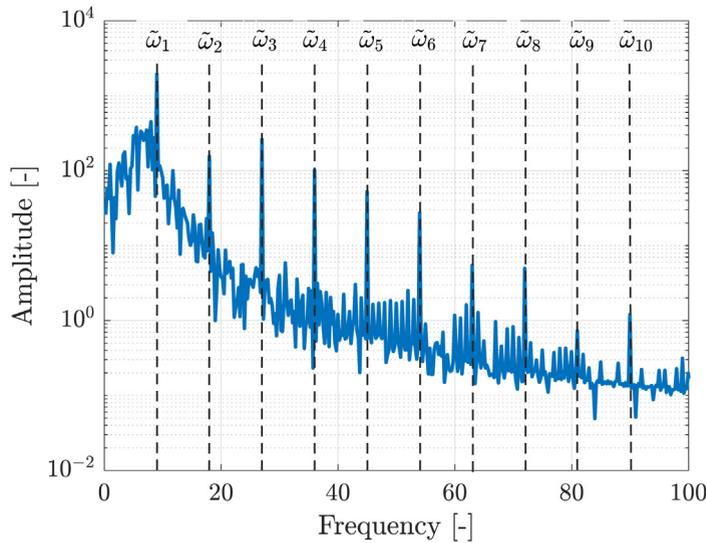


Fig. 10. Pressure as a function of frequency in normal working condition of gear pump where all the high frequencies of the frequency spectrum are chosen to add noise.

with unknown coefficients p_1, p_2 and p_3 . In order to find these coefficients, a system of three equations is created and solved. These three equations are generated by imposing constraints. The constraints are selected to have a balance between the generated time history and the original time history. A small noise content will not produce any difference in the generated time history, but, on the other hand, a more significant noise content can create an unreasonable time history. For example, ϵ is assumed to be zero for $\tilde{\omega}_1$ since adding noise in the first peak can create a very high-level difference from the original data. Similarly, ϵ is assumed to be 1 for $\tilde{\omega}_3$ for the same reason. The third equation is created by adopting the last peak value to be 100 because when the noise is added to the last peak, the effect generated by the noise is minimal. Thus, we can construct a matrix $\hat{\mathbf{X}} \in \mathbb{R}^{m \times (n+1)}$ by using Eqs. (19) and (21),

$$\hat{\mathbf{X}} = \begin{bmatrix} \hat{x}(\omega_1) & \hat{x}(\omega_1) & \hat{x}(\omega_1) & \dots & \hat{x}(\omega_1) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \vdots & \hat{x}(\tilde{\omega}_1)(1 + \epsilon(\tilde{\omega}_1)) & \vdots & \dots & \vdots \\ \vdots & \vdots & \hat{x}(\tilde{\omega}_2)(1 + \epsilon(\tilde{\omega}_2)) & \dots & \vdots \\ \vdots & \vdots & \vdots & \dots & \hat{x}(\tilde{\omega}_n)(1 + \epsilon(\tilde{\omega}_n)) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \dots & \dots & \dots & \dots & \dots \\ \hat{x}(\omega_m) & \hat{x}(\omega_m) & \hat{x}(\omega_m) & \dots & \hat{x}(\omega_m) \end{bmatrix}, \tag{22}$$

where $[\hat{\mathbf{X}}]_j$ represents a modified frequency series obtained after introducing noise perturbation for frequency content $\tilde{\omega}_j$. Hence,

$$\hat{\mathbf{X}}\mathbf{E}_j = [\hat{\mathbf{X}}]_j, \quad \text{with } \mathbf{E}_j = [0, \dots, 0, 1, 0, \dots, 0]^T, \tag{23}$$

where $[\hat{\mathbf{X}}]_j$ is the j th column of $\hat{\mathbf{X}}$, and in \mathbf{E}_j , 1 is the j th entry. Inverse Fourier transform ($\tilde{\mathcal{F}}_{\Delta t}^{-1}$) has been used to convert $[\hat{\mathbf{X}}]_j$ into time history $[\mathbf{X}]_j$, and it is expressed as,

$$[\mathbf{X}]_j = \tilde{\mathcal{F}}_{\Delta t}^{-1}([\hat{\mathbf{X}}]_j), \tag{24}$$

where in Eq. (24), the first column is equal to the original time-varying discrete set \mathbf{x} , and the second and the following columns are the noise-perturbed datasets. Figure 11 shows the time history of the original pressure dataset and two synthetically generated datasets. The original dataset, synthetic dataset 1 and synthetic dataset 2 are the inverse Fourier transform of the first, second, and third columns in Eq. (22).

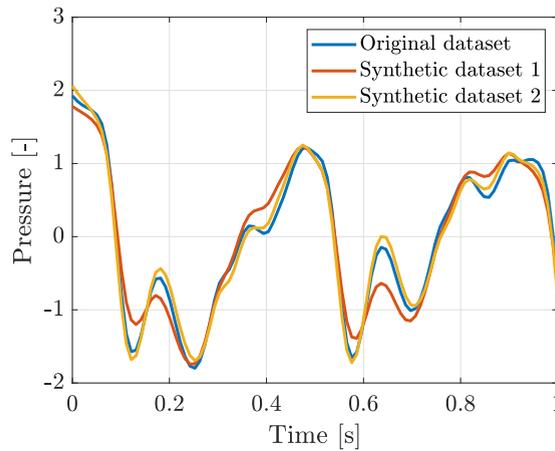


Fig. 11. Time history of the original pressure dataset and two synthetic datasets. The original dataset, synthetic dataset 1 and synthetic dataset 2 are the inverse Fourier transform of the first, second, and third columns in Eq. (22).

4.2. Synthetic degradation data generation method for fault prognosis

In this section, the data generation method for fault prognosis will be presented. As discussed in the introduction, fault prognosis requires degradation behaviour in the dataset. Two interpolation methods are considered in order to achieve degradation behaviour, where interpolation between high-fidelity simulations is used as a method for the generation of new data set points. This facilitates computational efficiency without excessively sacrificing the accuracy of results. The first approach uses a linear interpolation method to deteriorate from a healthy dataset (h) to a faulty dataset (f1), and the subsequent approach uses a spline function to interpolate between healthy data (h) to a series of faulty data (f1, f2, f3). f1, f2 and f3 are generic names for all 15 different fault scenarios mentioned in Table 4.

4.2.1. Linear interpolation method

Consider x_h a time-varying continuous function representing a monitored field of a gear pump in a healthy working condition. Similarly, x_{f1} is the time-varying continuous function representing the same monitored field under faulty working conditions. In order to obtain time-varying conditions for the situation in between these two scenarios, healthy and faulty, the degradation function is presented as

$$x(t, s) = x_{f1}(t) + g(s)[x_h(t) - x_{f1}(t)], \quad s \in [0, 1], \tag{25}$$

where

$$x(t, s = 0) = x_h(t), \tag{26}$$

$$x(t, s = 1) = x_{f1}(t), \tag{27}$$

where $g(s)$ is a degradation function and defined as

$$g(s) = 1 - s. \tag{28}$$

In order to generate a discrete set, the time continuous function Eq. (25) is evaluated in a discrete set of time values, rendering

$$[\mathbf{x}]_j = \mathbf{x}_{f1} + g(s_j)[\mathbf{x}_h - \mathbf{x}_{f1}], \quad j = 1, \dots, n, \tag{29}$$

Figure 12 shows the interpolated dataset between healthy dataset \mathbf{x}_h and faulty dataset \mathbf{x}_{f1} using the linear degradation function. Moreover, if a series of faulty scenarios ($\mathbf{x}_{f1}, \mathbf{x}_{f2}, \mathbf{x}_{f3}$) are available, then the linear interpolation method is employed intervals, i.e., \mathbf{x}_h to \mathbf{x}_{f1} , \mathbf{x}_{f1} to \mathbf{x}_{f2} and \mathbf{x}_{f2} to \mathbf{x}_{f3} . These interpolated datasets are employed as the input dataset for fault prognosis.

4.2.2. Cubic interpolation method

Consider x_h a time-varying continuous function representing a monitored field of the gear pump in a healthy working condition. Similarly, x_{f1}, x_{f2} and x_{f3} are the time-varying continuous functions representing the same monitored field under various faulty working conditions. To obtain the time-varying conditions for the situation in between these four scenarios, a spline (cubic) interpolation method can be used. A Lagrange interpolation polynomial can be used to yield,

$$x(m, t) = \sum_{i=1}^4 x_i(t)g_i(m), \quad g_i(m_j) = \delta_{ij}, \quad \delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j, \end{cases} \tag{30}$$

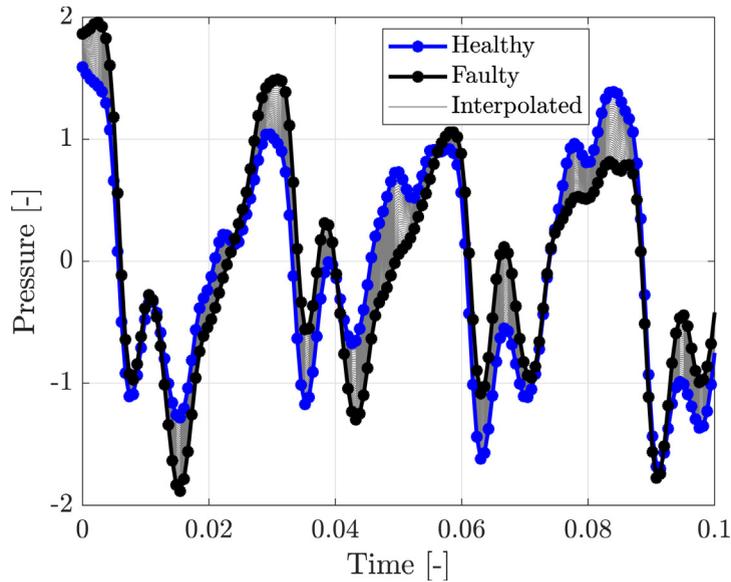


Fig. 12. Interpolated time-varying pressure values between healthy working conditions and faulty working conditions using linear degradation function presented in Eq. (29).

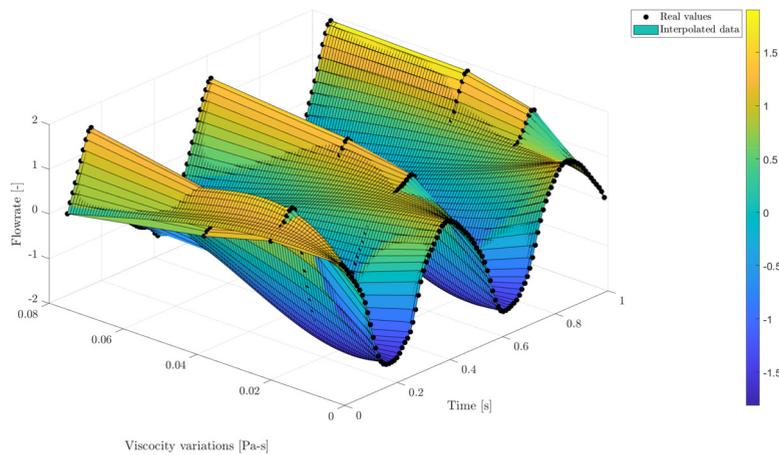


Fig. 13. Time-varying flow rate values for viscosity variations faulty scenarios for 1 s (100-time steps) created using the cubic interpolation method.

where x represents the time-varying condition of four working scenarios $x_h, x_{f1}, x_{f2},$ and x_{f3} of the gear pump, δ_{ij} is Kronecker delta symbol and m is the model parameter, which is considered to be experimentally validated for viscosity variations $m = [m_1, m_2, m_3, m_4] = [0.001, 0.02, 0.05, 0.075]$ Pa-s. The viscosity variations faulty scenarios are presented in Section 2.5. Four available model parameters can be used to form a system of equations that can be solved to find the unknown coefficients. Figure 13 shows the interpolated time-varying flow rate values for viscosity variations faulty scenarios for 1 s (100-time steps).

To validate the cubically interpolated value, the CFD simulation is performed for another viscosity value of 0.0384 Pa-s. Since the CFD simulations for different viscosity values have been validated experimentally, comparing interpolated values with the results of CFD simulation from the same properties provides sufficient reliability. Figure 14 shows the Validation of cubically interpolated value with the high-fidelity CFD results for the viscosity value of 0.0384 Pa-s. It can be observed that the interpolated degradation behaviour has good agreement with the CFD simulated value, especially in identifying amplitude peaks and time periods.

4.3. Data description and labelling

As mentioned earlier, ML-supervised algorithms are utilised to perform fault prognosis, which requires the input and output dataset to train the algorithm. In the context of fault prognosis, data generated using the linear and cubic interpolation methods are utilised as the input dataset for ML algorithms. The synthetically generated dataset consists of a total

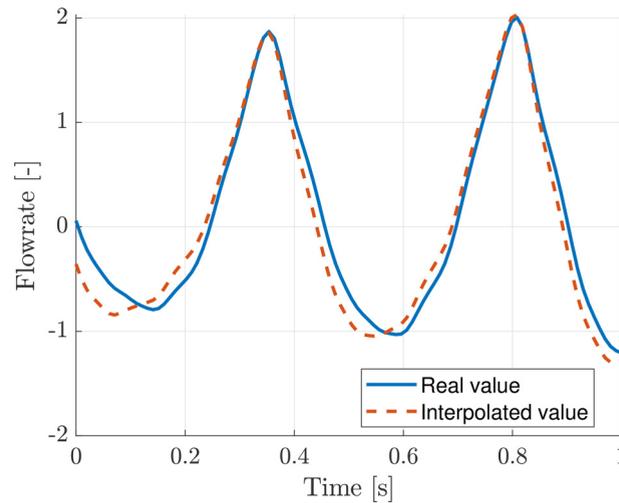


Fig. 14. Validation of cubically interpolated value with the high-fidelity CFD results for the viscosity value of 0.0384 Pa-s.

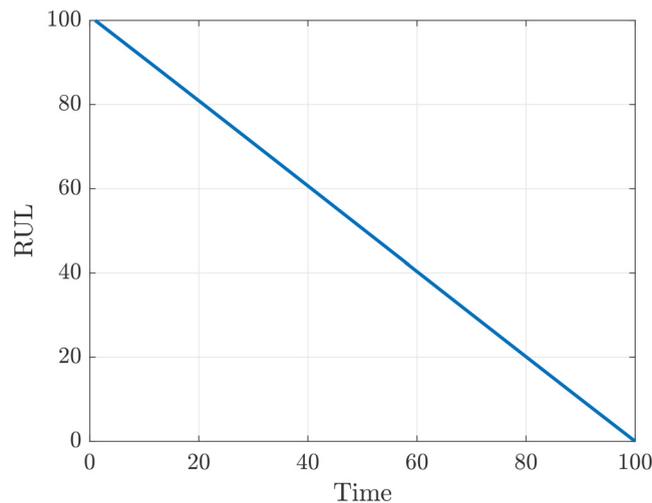


Fig. 15. Possible functions for RUL, linear decay function.

of 2000 samples containing the degradation behaviour of each of the specified fault categories. The output variable of fault prognosis is defined as RUL, which is a real number, quantifying the remaining life of the gear pump. In the absence of a full experimental dataset, where the degradation model is unavailable, the linear model has been reported as the most natural choice due to its simplicity [47]. Figure 15 shows a linear function that can be used as the RUL of a gear pump, with 100-time units being assumed as the maximum lifetime of the gear pump. It represents that with the increase in life, RUL decreases.

5. Numerical results for fault prognosis

The objective of this section is the numerical simulation of fault prognosis (prediction of the RUL) of the external gear pump using two alternative ML algorithms, namely MLP and SVM. The overview of the designed process for fault prognosis is presented in Fig. 16.

The high-fidelity data has been generated using an accurate CFD model resembling the operations of an external gear pump in healthy and various fault conditions (speed variations, radial gap variations, etc.). In order to enhance the predicted response of any ML algorithm, large amounts of data are usually necessary. To greatly facilitate the gathering of data, the synthetic data generation method is employed. The noise perturbation technique is used to recreate more data from the in-silico data, and the process of generating data is presented in Section 4.1. The generated datasets are, in turn, used for generating degradation datasets for fault prognosis.

In the absence of run-to-failure data or the lifetime history of the equipment, the deteriorating data of the equipment component is necessary to carry out fault prognosis [48]. In order to generate deterioration data, in this work, a linear

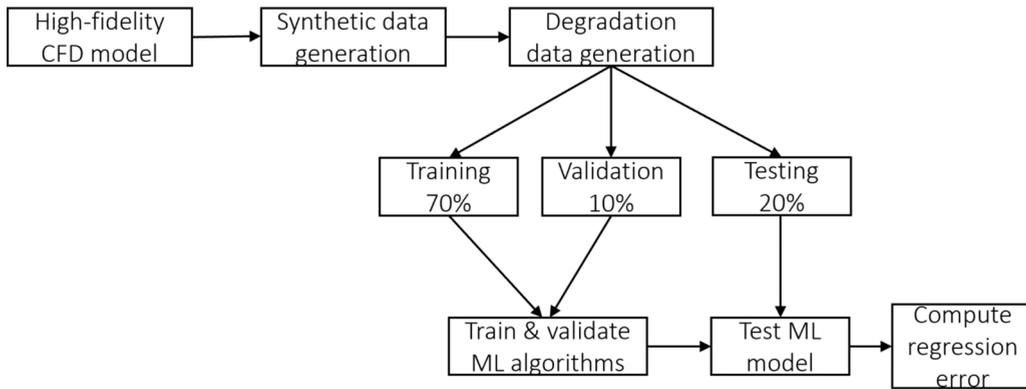


Fig. 16. Overview process of fault prognosis using ML regression algorithms.

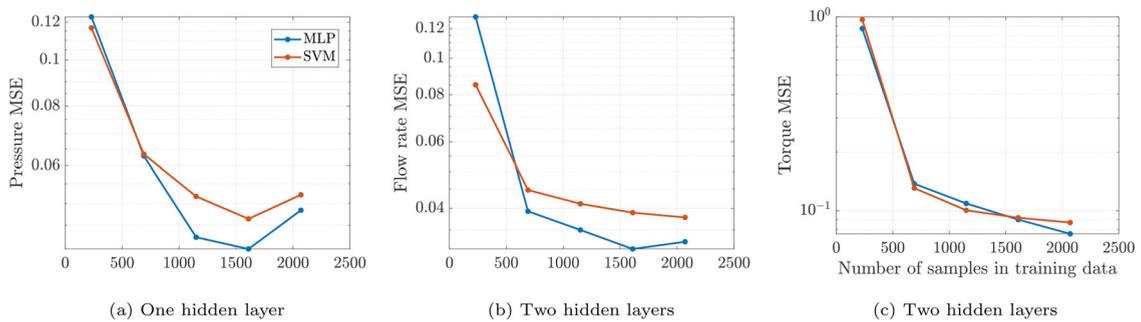


Fig. 17. Quantification analysis of prognosis of fault scenarios due to radial gap degradation using the linear interpolation method.

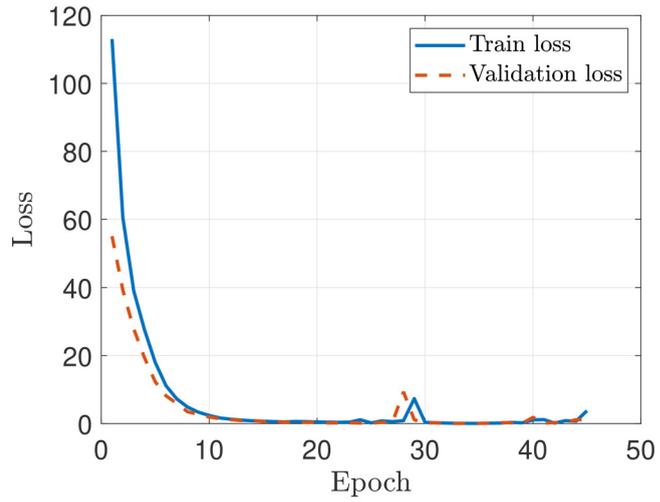
and a cubic interpolation method are employed, as discussed in Section 4.2. Datasets generated from the linear and cubic interpolation methods are standardised and used as input datasets for the training, validation and testing of ML regression algorithms. The output datasets consist of RUL, derived from linear function (Section 4.3). Also, RUL is normalised between 0 and 1.

Once the dataset is gathered, it is divided into a training dataset (used to fit the model) comprised of 80% of the dataset and a test dataset (used to provide an unbiased evaluation of a final model fit on the training dataset) comprised of the remaining 20% [49]. The training dataset is used to train each of the ML algorithms, which are then tested against the test dataset, and the Mean Squared Error (MSE) of the model is computed using Eq. (1).

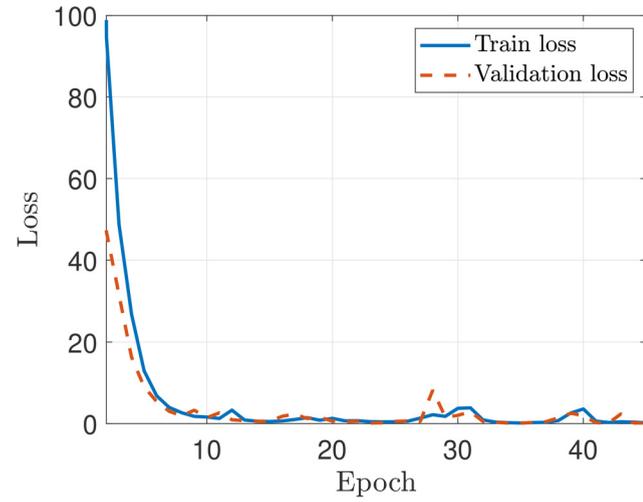
The quantification analysis is employed to understand the behaviour and the performance of ML algorithms with respect to the number of samples used for training. For the analysis, 10%, 30%, 50%, 70%, and 90% of the dataset are used to train the ML algorithms, MLP and SVM. Figure 17 show the quantification analysis of pressure, flow rate, and torque for fault scenarios due to radial gap degradation using ML algorithms, MLP and SVM, utilising the dataset generated from the linear interpolation method. In most cases, both ML algorithms follow a linear trend, showing that, as the number of samples increases, the machine learning algorithms provide better results (i.e., lower prediction error). However, there is inconsistency in the case of pressure using both ML algorithms.

The hyperparameters for both models are discussed in Section 3.3. The hyperparameter optimisation is performed while training the model, i.e., at the Train & validate ML algorithms stage in Fig. 16. As mentioned earlier, overfitting or underfitting can be identified by model loss with respect to the number of epochs, which is shown in Fig. 18a. The train and validation loss curves are slightly drifted compared to Fig. 18b, yet it does not show overfitting, hence, one layer is sufficient for the prediction, but having two hidden layers provides lower MSE. The number of epochs can be stopped during the training, using the early stopping option, when the chosen performance stops improving. The early stopping criteria minimise the computational time up to 50% when the optimisation is carried out without early stopping.

After hyperparameters are optimised, the best parameters can be extracted for subsequent prediction. For the MLP algorithm, the optimised hyperparameters are shown in Table 5; the best-optimised parameters are as follows: The number of neurons for hidden layer 1 is 64, hidden layer 2 is 32, the activation function is Relu, the learning rate is $1e^{-3}$, the batch size is 32, and the optimiser is Adam type. Similarly, for the SVM algorithm, the optimised hyperparameters are shown in Table 6; the best-optimised parameters are as follows: The kernel function is polynomial with degree 3, the kernel scale is 1, and the tolerance margin is 0.1.



(a) One hidden layer



(b) Two hidden layers

Fig. 18. Model loss on training and validation datasets. The model trained with one layer of neurons (a) and two layers of neurons (b) using MLP.

Table 7

MSE of pressure, flow rate and torque using MLP and SVM for fault scenarios due to speed variations using the linear interpolation method (a) and cubic interpolation method (b).

	MLP	SVM
Pressure	0.03857	0.0472
Flow rate	0.03746	0.0490
Torque	0.1401	0.1211
(a) Linear interpolation method		
	MLP	SVM
Pressure	0.000752	0.0026
Flow rate	0.00023	0.0084
Torque	0.00062	0.00052
(b) Cubic interpolation method		

5.1. Fault scenarios due to speed variations

In the present example, the RUL prediction of fault scenarios due to speed variations is discussed. Pressure, flow rate, and torque extracted from the fault scenarios due to speed variations are considered for this analysis. Following the methodology discussed earlier, a fault prognosis analysis is performed, and the R2 score and MSE are calculated for each scenario. The model R2 score is 0.9999 with minor differences for different scenarios, so MSE is considered to measure the difference between the results of all cases.

Table 7 shows the MSE of pressure, flow rate, and torque for fault scenarios due to speed variations using ML algorithms, MLP and SVM utilising the synthetic data generation approaches, the linear interpolation method (a) and the cubic interpolation method (b). For MLP and SVM, the dataset created using the cubic interpolation method provides lower error compared to the dataset created using the linear interpolation method. This occurs because the linear interpolation method is created between intervals of two-time series, whereas the cubic interpolation method is constructed between four-time series, enriching the prediction pattern both in terms of accuracy and smoothness. So, the dataset created using the cubic interpolation method gives a degradation behaviour that is smoother than the dataset created using the linear interpolation method. When the variables pressure, flow rate, and torque are considered using both ML approaches, torque has one magnitude higher error than pressure and flow rate for data obtained using the linear interpolation method. As can be observed in Table 7, MLP performs better than SVM due to its higher non-linear nature as a result of an increased number of hidden layers and thus hyper-parameters. Having more than one hidden layer helps with a better prediction of the output without over-fitting. Even though SVM has an in-built non-linear character due to the *kernel trick uplift*, training on the multiple layers enhances the robust prediction for MLP.

5.2. Fault scenarios due to viscosity variations

In this example, the RUL prediction of fault scenarios due to viscosity variations using ML algorithms MLP and SVM is described. The variables pressure, flow rate and torque extracted from the fault scenarios due to viscosity variations are considered for this analysis.

Following the fault prognosis procedure discussed earlier, the fault prognosis is performed using the ML algorithms, MLP and SVM, with the prediction error computed with MSE. Table 8 shows the prediction MSE of pressure, flow rate and torque for fault scenarios due to viscosity variations using ML algorithms, MLP and SVM utilising the dataset generated from the linear interpolation method (a) and the cubic interpolation method (b). Once again, the dataset created by the cubic interpolation method provides lower error compared to the dataset created by the linear interpolation method. When the variables pressure, flow rate, and torque are considered using both ML approaches, torque has a higher error than pressure and flow rate for data obtained using the linear interpolation method. As can be observed in Table 8, MLP performs better than SVM. Similarly, as it was explained above, the higher non-linearity of MLP helps obtaining a better prediction with respect to SVM.

5.3. Fault scenarios due to radial gap degradation

In the present example, the RUL prediction of fault scenarios due to radial gap degradation using ML algorithms, MLP and SVM is described. Pressure, flow rate, and torque data from the fault scenario due to radial gap degradation are considered for the analysis. Table 9 shows the MSE of pressure, flow rate, and torque of fault scenario due to radial gap degradation using ML algorithms, MLP and SVM utilising the linear interpolation method (a) and cubic interpolation method (b). When

Table 8

MSE of pressure, flow rate and torque using MLP and SVM for fault scenarios due to viscosity variations using the linear interpolation method (a) and cubic interpolation method (b).

	MLP	SVM
Pressure	0.0401	0.0406
Flow rate	0.0346	0.042
Torque	0.0961	0.1087
(a) Linear interpolation method		
	MLP	SVM
Pressure	0.00077	0.0036
Flow rate	0.00011	0.00064
Torque	0.00223	0.0059
(b) Cubic interpolation method		

Table 9

MSE of pressure, flow rate and torque using MLP and SVM for fault scenarios due to radial gap degradation using the linear interpolation method (a) and cubic interpolation method (b).

	MLP	SVM
Pressure	0.0563	0.0700
Flow rate	0.0367	0.0422
Torque	0.0711	0.0889
(a) Linear interpolation method		
	MLP	SVM
Pressure	0.00041	0.0048
Flow rate	0.00035	0.0059
Torque	0.0024	0.0056
(b) Cubic interpolation method		

the variables pressure, flow rate, and torque are considered using both ML approaches, torque has a higher error than pressure and flow rate for data obtained using both the linear and cubic interpolation methods. In all cases, MLP performs relatively better than SVM. The dataset created with the cubic interpolation method provides lower error when compared to the dataset created with the linear interpolation method. This occurs because the linear interpolation method is created between intervals of two-time signals, whereas the cubic interpolation method is used between four-time signals. So, the dataset created using the cubic interpolation method gives a degradation behaviour that is smoother than the dataset created using the linear interpolation method. As can be observed in Table 9, MLP performs better than SVM. Similarly, as it was explained above, the higher non-linearity of MLP helps obtaining a better prediction with respect to SVM.

5.4. Fault scenarios due to axial gap degradation

The present example describes the RUL prediction of fault scenarios due to axial gap degradation using ML algorithms, MLP and SVM. Variables such as pressure, flow rate, and torque data monitored from the fault scenario due to axial gap degradation are considered for this analysis. Table 10 shows the prediction MSE of pressure, flow rate and torque for fault scenarios due to axial gap degradation using ML algorithms, MLP and SVM utilising the dataset generated from the linear interpolation method (a) and the cubic interpolation method (b). When the variables pressure, flow rate, and torque are considered using both ML approaches, torque has a higher error than pressure and flow rate for data obtained using the linear and cubic interpolation methods. Once again, The dataset created by the cubic interpolation method provides lower error compared to the dataset created by the linear interpolation method. As can be observed in Table 10, MLP performs better than SVM. Similarly, as it was explained above, the higher non-linearity of MLP helps obtaining a better prediction with respect to SVM.

Table 10

MSE of pressure, flow rate and torque using MLP and SVM for fault scenarios due to axial gap degradation using the linear interpolation method (a) and cubic interpolation method (b).

	MLP	SVM
Pressure	0.0401	0.0406
Flow rate	0.0346	0.042
Torque	0.0961	0.1087
(a) Linear interpolation method		
	MLP	SVM
Pressure	0.00077	0.0036
Flow rate	0.00011	0.00064
Torque	0.00223	0.0059
(b) Cubic interpolation method		

Table 11

MSE of pressure, flow rate and torque using MLP and SVM for fault scenarios due to temperature variations using the linear interpolation method (a) and cubic interpolation method (b).

	MLP	SVM
Pressure	0.037	0.0445
Flow rate	0.0488	0.0416
Torque	0.0682	0.0689
(a) Linear interpolation method		
	MLP	SVM
Pressure	0.0011	0.0063
Flow rate	0.0002	0.0062
Torque	0.0006	0.0052
(b) Cubic interpolation method		

5.5. Fault scenarios due to temperature variations

In this example, the RUL prediction of fault scenarios due to temperature variations using ML algorithms MLP and SVM is described. The variables pressure, flow rate and torque extracted from the fault scenarios due to temperature variations are considered for this analysis.

Following the fault prognosis methodology discussed earlier, the fault prognosis is performed using the ML algorithms, MLP and SVM, with the prediction error computed with MSE. Table 11 shows the prediction MSE of pressure, flow rate and torque for fault scenarios due to temperature variations using ML algorithms, MLP and SVM utilising the dataset generated from the linear interpolation method (a) and the cubic interpolation method (b). When the variables pressure, flow rate, and torque are considered using both ML approaches, in most cases, torque has a higher error than pressure and flow rate for data obtained using the linear and cubic interpolation methods. Once again, the dataset created by the cubic interpolation method provides lower error compared to the dataset created by the linear interpolation method. Once again, MLP performs better than SVM for similar reasons to those argued above.

5.6. Gaussian noise robustness of the utilised regression algorithms

This section aims to study fault prognosis considering noisy measurements in order to understand the sensitivity of ML algorithms. The noise influence in various ML algorithms has been widely studied [50]. The noise can be added to the input, output, training, testing, or a combination of all of these. The noise generation can be characterised by its distribution and where to introduce it [25]. In general, noisy data may cause biases in the learning process, making it more difficult for learning algorithms to form accurate models from the data. Therefore, developing learning techniques that effectively and efficiently deal with these types of data is a key aspect in ML [50].

It is important to emphasize that the nature of the Gaussian noise described in this section is very different from the data perturbation technique used in previous sections in order to efficiently build a dense dataset. Whilst the Gaussian noise

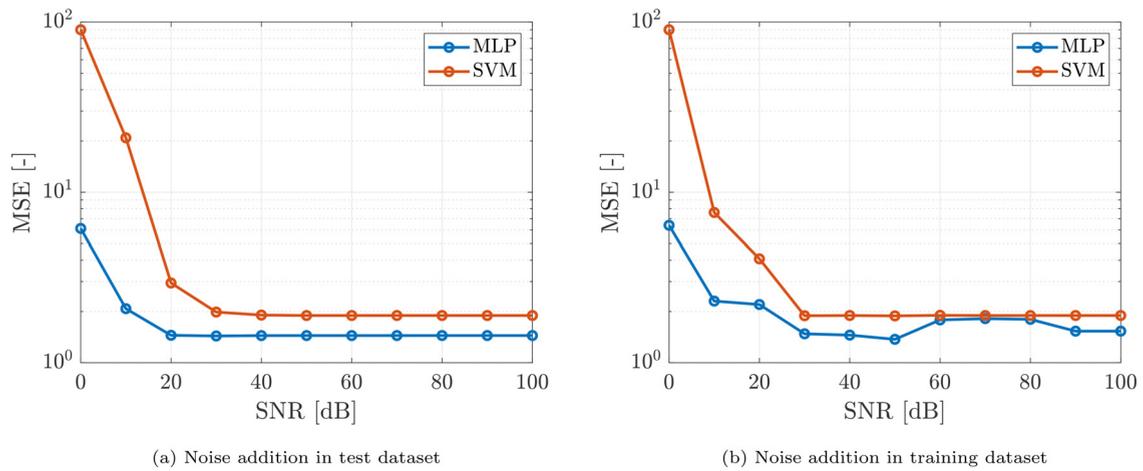


Fig. 19. MSE of torque using MLP and SVM for noise addition in test dataset (a) and noise addition in training dataset.

is constructed in a time-varying manner, similar to the noise that a real sensor would experience, the data perturbation technique was designed to minimally alter the fundamental frequencies of a given response in order to expand the dataset. These two very different approaches were chosen in order to clearly distinguish both perturbation mechanisms. In real industry applications, sensors are installed in the device to extract datasets. However, the collected datasets typically contain noise, so in order to replicate this scenario, Gaussian white noise is added to the input data of training or test datasets. The noise addition in the input data can be expressed as

$$\hat{\mathbf{x}} = \mathbf{x} + \boldsymbol{\epsilon}, \tag{31}$$

where \mathbf{x} is the time history and $\boldsymbol{\epsilon}$ is Gaussian white noise. In order to vary the level of noise added, the Signal Noise Ratio (SNR) in decibels (dB) [51] is utilised, and it is defined as

$$\text{SNR} = 10 \log_{10} \frac{P_{\mathbf{x}}}{P_{\boldsymbol{\epsilon}}}, \tag{32}$$

where $P_{\mathbf{x}}$ is the power of a signal, which is the sum of the absolute squares of its time history samples divided by the signal length,

$$P_{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n x_i^2. \tag{33}$$

Similarly, $P_{\boldsymbol{\epsilon}}$ is the power of a signal, and it can be written as

$$P_{\boldsymbol{\epsilon}} = \frac{1}{n} \sum_{i=1}^n \epsilon_i^2. \tag{34}$$

The dataset considered for this analysis is fault scenarios due to radial gap degradation. The variable torque extracted from the fault scenarios due to radial gap degradation is considered for this analysis. The degradation dataset generated using the cubic interpolation method is the input data, and the linear decay function is the output for ML algorithms. The input and output datasets are divided into 80% of the training and the remaining 20% of the testing dataset. The training and testing datasets are standardised and used for ML training. The training dataset is used to train each ML algorithm, MLP and SVM, which are then tested against the test dataset with noise. The model's prediction MSE is computed using Eq. (1) between ML predicted values and true output values. In our case, two studies of fault prognosis are performed to analyse the Gaussian noise effect. Firstly, in 20% of the testing dataset, the Gaussian noise is added using Eq. (31), which is tested against the ML model trained with a dataset without noise. Later, in the 80% training dataset, the Gaussian noise is added and tested against the test dataset without noise.

Figure 19 a and b show the MSE of torque using MLP and SVM to predict RUL for noise addition in the testing and training datasets, respectively. In both cases, the prediction error follows a decreasing trend in the beginning and then a constant trend when there is an increase in the SNR level, i.e., a decrease in noise. For overall prediction, MLP performs better than SVM. However, both algorithms perform poorly for SNR levels below 30, which generally indicates a relatively low signal strength relative to the background noise. The acceptable SNR level depends on the specific application and the required level of accuracy or reliability. For example, in mechanical sensing applications, a low SNR might result in inaccurate or unreliable measurements, affecting system performance. For accurate flow rate measurements in gear pumps, an SNR level of at least 30 dB is typically required. As a result, both of our algorithms work effectively at SNR levels of 30 or higher.

This above-discussed technique can be adapted to any fault scenario with a series of faults. Based on the observations from the above three cases, MLP and SVM provide excellent results based on the R2 score and MSE, so there is no necessity to assess other algorithms. Torque provides comparatively higher mean squared error compared to pressure and flow rate, resulting in lower error versus cost. Therefore, if the proffered choice is to use torque to perform fault prognosis, there is a trade between cost and error.

6. Conclusions

This paper has presented a novel methodology for fault prognosis using Machine Learning (ML) algorithms for an external gear pump. The main novelties of this work are the high-fidelity data generation for an external gear pump using CFD models, the introduction of synthetic data generation techniques and a computational framework for fault prognosis exploiting the use of ML algorithms. In-silico and synthetic data generation methods provide a long-term solution to insufficient or unavailable experimental data and enable further investigation via ML or other applicable data analysis techniques.

The fault prognosis of the gear pump has been performed using two ML regression algorithms: MLP and SVM. Synthetically generated degradation datasets are applied to perform this analysis using linear and cubic interpolation methods. The data generated using the cubic interpolation method provides a superior prediction of the life expectancy of the gear pump compared to the linear interpolation method. Furthermore, MLP yields better pressure and flow rate results than SVM, whereas MLP and SVM perform similarly for the torque analysis. Although MLP outperforms SVM in most cases, there are certain advantages to adopting SVM. SVM, for example, is resistant to outliers in the data, which can be a difficulty in many prediction tasks, and it has strong generalisation performance, which means it can predict new data accurately. Finally, notice that the fault prognosis methodology discussed in this paper can be adapted to any fault scenario with a combination of faults.

A primary interest for the industry is the performance of these analyses with minimal cost; this would suggest using torque as the magnitude as it can be efficiently monitored via a cost-effective sensor. However, torque provides comparatively higher mean squared error compared to pressure and flow rate, resulting in lower error versus cost. Therefore, if torque is to be chosen as the preferred measure by the industry, there is a trade between cost and error. Finally, it is important to emphasise that the computational framework developed can be straightforwardly used, without any modification, with additional experimental data if available.

Data availability

The data that has been used is confidential.

Acknowledgments

The authors would also like to acknowledge the financial support received through the European Commission EACEA Agency, Framework Partnership Agreement 2013-0043 Erasmus Mundus Action 1b, as a part of the EM Joint Doctorate “Simulation in Engineering and Entrepreneurship Development (SEED)”.

Appendix A. The mathematical model and governing equations

In this appendix, the governing equations solved during the CFD simulations are presented. The conservation of mass principle can be written using an Arbitrary Lagrangian Eulerian (ALE) description as follows [28]

$$\frac{\partial}{\partial t} \int_{\Omega(t)} \rho d\Omega + \int_{\partial\Omega(t)} \rho(\mathbf{v} - \mathbf{v}_s) \cdot \mathbf{n} ds = 0, \tag{A.1}$$

where \mathbf{v} is the fluid velocity, ρ is the fluid density and \mathbf{v}_s is the mesh velocity. The conservation of linear momentum written in ALE description is given as,

$$\frac{\partial}{\partial t} \int_{\Omega(t)} \rho \mathbf{v} d\Omega + \int_{\partial\Omega(t)} [\rho \mathbf{v} \otimes (\mathbf{v} - \mathbf{v}_s)] \cdot \mathbf{n} ds = \int_{\partial\Omega(t)} \boldsymbol{\tau} \mathbf{n} ds - \int_{\partial\Omega(t)} p \mathbf{n} ds + \int_{\Omega(t)} \rho \mathbf{f} d\Omega, \tag{A.2}$$

where $\boldsymbol{\tau}$ is the Reynold’s average viscous shear stress tensor, p is the pressure, and \mathbf{f} is a body force. In this work, both turbulence and cavitation physical models are considered during the simulation. The standard $k - \varepsilon$ model is based on the following two equations. The first equation is

$$\frac{\partial}{\partial t} \int_{\Omega(t)} \rho k d\Omega + \int_{\partial\Omega(t)} \rho k(\mathbf{v} - \mathbf{v}_s) \cdot \mathbf{n} ds = \int_{\partial\Omega(t)} \left(\mu + \frac{\mu_t}{\sigma_k} \right) (\nabla k \cdot \mathbf{n}) ds + \int_{\Omega(t)} (G_t - \rho \varepsilon) d\Omega, \tag{A.3}$$

where μ_t is the turbulent viscosity, $\sigma_k = 1$ is the turbulent Prandtl number and G_t is the turbulent generation term. The second equation is

$$\frac{\partial}{\partial t} \int_{\Omega(t)} \rho \varepsilon d\Omega + \int_{\partial\Omega(t)} \rho \varepsilon(\mathbf{v} - \mathbf{v}_s) \cdot \mathbf{n} ds = \int_{\partial\Omega(t)} \left(\mu + \frac{\mu_t}{\sigma_\varepsilon} \right) (\nabla \varepsilon \cdot \mathbf{n}) ds + \int_{\Omega(t)} \left(c_1 G_t \frac{\varepsilon}{k} - c_2 \rho \frac{\varepsilon^2}{k} \right) d\Omega, \tag{A.4}$$

where $\sigma_\varepsilon = 1.3$ is the turbulent Prandtl number for the turbulent kinetic energy dissipation rate and $c_1 = 1.44$ and $c_2 = 1.92$ are constants. For pump flows, cavitation is often encountered and is an important factor for design considerations [29]. The vapour mass fraction coupled with the momentum equation via the pressure field is solved via

$$\frac{\partial}{\partial t} \int_{\Omega(t)} \rho f_v d\Omega + \int_{\partial\Omega(t)} \rho f_v (\mathbf{v} - \mathbf{v}_s) \cdot \mathbf{n} ds = \int_{\partial\Omega(t)} \left(D_{f_v} + \frac{\mu_t}{\sigma_{f_v}} \right) (\nabla f_v \cdot \mathbf{n}) ds + \int_{\Omega(t)} (R_e - R_c) d\Omega, \quad (\text{A.5})$$

where D_{f_v} is the diffusivity of the vapour mass fraction f_v , σ_{f_v} is the turbulent Schmidt number and μ_t is the turbulent viscosity. The source terms R_e and R_c are the vapour generation and the condensation. During simulation, these equations are discretised using the finite volume method and solved using a pressure-based solution algorithm. For further details, please refer to [52].

References

- [1] R.K. Mobley, An Introduction to Predictive Maintenance, Elsevier, 2002.
- [2] Z. Gao, X. Liu, An overview on fault diagnosis, prognosis and resilient control for wind turbine systems, *Processes* 9 (2) (2021) 300.
- [3] X. Li, Q. Ding, J.-Q. Sun, Remaining useful life estimation in prognostics using deep convolution neural networks, *Reliab. Eng. Syst. Saf.* 172 (2018) 1–11.
- [4] C.S. Byington, M.J. Roemer, T. Galie, Prognostic enhancements to diagnostic systems for improved condition-based maintenance [military aircraft], in: *Proceedings, IEEE Aerospace Conference*, Vol. 6, IEEE, 2002, pp. 6–6.
- [5] D. Lin, V. Makis, Recursive filters for a partially observable system subject to random failure, *Adv. Appl. Probab.* (2003) 207–227.
- [6] H. Sohn, C.R. Farrar, F.M. Hemez, G. Park, A.N. Robertson, T.O. Williams, A coupled approach to developing damage prognosis solutions, in: *Key Engineering Materials*, Vol. 245, Trans Tech Publ, 2003, pp. 289–306.
- [7] S. Haidong, C. Junsheng, J. Hongkai, Y. Yu, W. Zhantao, Enhanced deep gated recurrent unit and complex wavelet packet energy moment entropy for early fault prognosis of bearing, *Knowl. Based Syst.* 188 (2020) 105022.
- [8] A. Rai, S.H. Upadhyay, Intelligent bearing performance degradation assessment and remaining useful life prediction based on self-organising map and support vector regression, *Proc. Inst. Mech.Eng. Part C J. Mech. Eng. Sci.* 232 (6) (2018) 1118–1132.
- [9] B.-Y. Sun, Z.-H. Zhu, J. Li, B. Linghu, Combined feature selection and cancer prognosis using support vector machine regression, *IEEE/ACM Trans. Comput. Biol. Bioinf.* 8 (6) (2010) 1671–1677.
- [10] T. Benkedjouh, K. Medjaher, N. Zerhouni, S. Rechak, Health assessment and life prediction of cutting tools based on support vector regression, *J. Intell. Manuf.* 26 (2) (2015) 213–223.
- [11] C. Li, Z. Wang, S. Bu, H. Jiang, Z. Liu, A novel method based on least squares support vector regression combining with strong tracking particle filter for machinery condition prognosis, *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* 228 (6) (2014) 1048–1062.
- [12] Y. Xie, T. Zhang, The application of echo state network and recurrent multilayer perceptron in rotating machinery fault prognosis, in: *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, IEEE, 2016, pp. 2286–2291.
- [13] W.Q. Wang, M.F. Golnaraghi, F. Ismail, Prognosis of machine health condition using neuro-fuzzy systems, *Mech. Syst. Signal Process.* 18 (4) (2004) 813–831.
- [14] R. Yam, P.W. Tse, L. Li, P. Tu, Intelligent predictive decision support system for condition-based maintenance, *Int. J. Adv. Manuf. Technol.* 17 (5) (2001) 383–391.
- [15] K. Lakshmanan, E. Borghini, A. Beckmann, C. Pleydell-Pearce, C. Giannetti, Data modelling and remaining useful life estimation of rolls in a steel making cold rolling process, *Procedia Comput. Sci.* 207 (2022) 1057–1066.
- [16] J. Hong, Z. Wang, Y. Yao, Fault prognosis of battery system based on accurate voltage abnormality prognosis using long short-term memory neural networks, *Appl. Energy* 251 (2019) 113381.
- [17] S. Tang, S. Yuan, Y. Zhu, Deep learning-based intelligent fault diagnosis methods toward rotating machinery, *IEEE Access* 8 (2019) 9335–9346.
- [18] K. Lakshmanan, A.J. Gil, F. Auricchio, F. Tessicini, A Fault Diagnosis Methodology for An External Gear Pump with the use of Machine Learning Classification Algorithms: Support Vector Machine and Multilayer Perceptron, Loughborough University, 2020.
- [19] W. Xu, Z. Zhou, T. Li, C. Sun, X. Chen, R. Yan, Physics-constraint variational neural network for wear state assessment of external gear pump, *IEEE Trans. Neural Netw. Learn. Syst.* (2022).
- [20] M.-S. Lee, T.A. Shifat, J.-W. Hur, Kalman filter assisted deep feature learning for RUL prediction of hydraulic gear pump, *IEEE Sens. J.* 22 (11) (2022) 11088–11097.
- [21] Z. Obermeyer, E.J. Emanuel, Predicting the future—big data, machine learning, and clinical medicine, *N. Engl. J. Med.* 375 (13) (2016) 1216.
- [22] Y. Zhao, H.D. Akolekar, J. Weatheritt, V. Michelassi, R.D. Sandberg, RANS turbulence model development using CFD-driven machine learning, *J. Comput. Phys.* 411 (2020) 109413.
- [23] F.T. Zahura, J.L. Goodall, J.M. Sadler, Y. Shen, M.M. Morsy, M. Behl, Training machine learning surrogate models from a high-fidelity physics-based model: application for real-time street-scale flood prediction in an urban coastal community, *Water Resour. Res.* 56 (10) (2020). e2019WR027038
- [24] K. Balla, R. Sevilla, O. Hassan, K. Morgan, An application of neural networks to the prediction of aerodynamic coefficients of aerofoils and wings, *Appl. Math. Model.* 96 (2021) 456–479.
- [25] K. Mivule, Utilizing noise addition for data privacy, an overview, arXiv preprint arXiv:1309.3958(2013).
- [26] N.D. Manring, S.B. Kasaragadda, The theoretical flow ripple of an external gear pump, *J. Dyn. Sys., Meas., Control* 125 (3) (2003) 396–404.
- [27] A. Corvaglia, A. Ferrari, M. Rundo, O. Vento, Three-dimensional model of an external gear pump with an experimental evaluation of the flow ripple, *Proc. Inst. Mech.Eng. Part C J. Mech. Eng. Sci.* 235 (6) (2021) 1097–1105.
- [28] H. Ding, F.C. Visser, Y. Jiang, M. Furmanczyk, Demonstration and validation of a 3D CFD simulation tool predicting pump performance and cavitation for industrial applications, *J. Fluids Eng.* 133 (1) (2011).
- [29] F. Qi, S. Dhar, V.H. Nichani, C. Srinivasan, W. De Ming, L. Yang, Z. Bing, J.J. Yang, A CFD study of an electronic hydraulic power steering helical external gear pump: model development, validation and application, *SAE Int. J. Passenger Cars-Mech. Syst.* 9 (2016-01–1376) (2016) 346–352.
- [30] Fluid-o-tech, (2020). Accessed: 2020-09-03, url<https://www.fluidotech.it/en/products/technologies/external-gear-pumps/>.
- [31] W.-G. Jiang, R.-Z. Zhong, Q.H. Qin, Y.-G. Tong, Homogenized finite element analysis on effective elastoplastic mechanical behaviors of composite with imperfect interfaces, *Int. J. Mol. Sci.* 15 (12) (2014) 23389–23407.
- [32] C. Saranya, G. Manikandan, A study on normalization techniques for privacy preserving data mining, *Int. J. Eng. Technol.(IJET)* 5 (3) (2013) 2701–2704.
- [33] C. Crisci, B. Ghattas, G. Perera, A review of supervised machine learning algorithms and their applications to ecological data, *Ecol. Modell.* 240 (2012) 113–122.
- [34] K.-L. Du, M.N.S. Swamy, *Neural Networks and Statistical Learning*, Springer Science & Business Media, 2013.
- [35] T.M. Mitchell, et al., *Machine learning*, 1997.
- [36] A.S. Almryad, H. Kutucu, Automatic identification for field butterflies by convolutional neural networks, *Eng. Sci. Technol.Int. J.* 23 (1) (2020) 189–195.
- [37] I. Sutskever, J. Martens, G. Dahl, G. Hinton, On the importance of initialisation and momentum in deep learning, in: *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, ICML, 2013, pp. 1139–1147.
- [38] M. Mohri, A. Rostamizadeh, A. Talwalkar, *Foundations of Machine Learning*, MIT press, 2018.

- [39] H. Yu, S. Kim, SVM tutorial-classification, regression and ranking, *Handb. Natural Comput.* 1 (2012) 479–506.
- [40] G.F. Smits, E.M. Jordaán, Improved SVM regression using mixtures of kernels, in: *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, Vol. 3, IEEE, 2002, pp. 2785–2790.
- [41] A. Candelieri, I. Giordani, F. Archetti, K. Barkalov, I. Meyerov, A. Polovinkin, A. Sysoyev, N. Zolotykh, Tuning hyperparameters of a SVM-based water demand forecasting system through parallel global optimization, *Comput. Oper. Res.* 106 (2019) 202–209.
- [42] J.S. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for hyper-parameter optimization, in: *Advances in Neural Information Processing Systems*, 2011, pp. 2546–2554.
- [43] K. Kumar, B. Abhishek, *Artificial Neural Networks for Diagnosis of Kidney Stones Disease*, GRIN Verlag, 2012.
- [44] Keras, (2015). Accessed: 2020-09-06, url<https://keras.io>.
- [45] S. Karsoliya, Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture, *Int. J. Eng. Trends Technol.* 3 (6) (2012) 714–717.
- [46] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, *J. Mach. Learn. Res.* 13 (2) (2012).
- [47] G.S. Babu, P. Zhao, X.-L. Li, Deep convolutional neural network based regression approach for estimation of remaining useful life, in: *International Conference on Database Systems for Advanced Applications*, Springer, 2016, pp. 214–228.
- [48] A.K.S. Jardine, D. Lin, D. Banjevic, A review on machinery diagnostics and prognostics implementing condition-based maintenance, *Mech. Syst. Signal Process.* 20 (7) (2006) 1483–1510.
- [49] D.T. Ahneman, J.G. Estrada, S. Lin, S.D. Dreher, A.G. Doyle, Predicting reaction performance in C–N cross-coupling using machine learning, *Science* 360 (6385) (2018) 186–190.
- [50] D.F. Nettleton, A. Orriols-Puig, A. Fornells, A study of the effect of different types of noise on the precision of supervised learning techniques, *Artif. Intell. Rev.* 33 (4) (2010) 275–306.
- [51] D.R. Pauluzzi, N.C. Beaulieu, A comparison of SNR estimation techniques for the AWGN channel, *IEEE Trans. Commun.* 48 (10) (2000) 1681–1691.
- [52] K. Lakshmanan, Predictive maintenance of an external gear pump using machine learning algorithms, Swansea University, UK, 2021 Ph.D. thesis.