Research Article

A novel hybrid predictive maintenance model based on clustering, smote and multi-layer perceptron neural network optimised with grey wolf algorithm

Albert Buabeng¹ · Anthony Simons² · Nana Kena Frempong³ · Yao Yevenyo Ziggah⁴

Received: 30 November 2020 / Accepted: 20 April 2021

Published online: 30 April 2021

© The Author(s) 2021

Abstract

Considering the complexities and challenges in the classification of multiclass and imbalanced fault conditions, this study explores the systematic combination of unsupervised and supervised learning by hybridising clustering (CLUST) and optimised multi-layer perceptron neural network with grey wolf algorithm (GWO-MLP). The hybrid technique was meticulously examined on a historical hydraulic system dataset by first, extracting and selecting the most significant statistical time-domain features. The selected features were then grouped into distinct clusters allowing for reduced computational complexity through a comparative study of four different and frequently used categories of unsupervised clustering algorithms in fault classification. The Synthetic Minority Over Sampling Technique (SMOTE) was then employed to balance the classes of the training samples from the various clusters which then served as inputs for training the supervised GWO-MLP. To validate the proposed hybrid technique (CLUST-SMOTE-GWO-MLP), it was compared with its distinct modifications (variants). The superiority of CLUST-SMOTE-GWO-MLP is demonstrated by outperforming all the distinct modifications in terms of test accuracy and seven other statistical performance evaluation metrics (error rate, sensitivity, specificity, precision, F score, Mathews Correlation Coefficient and geometric mean). The overall analysis indicates that the proposed CLUST-SMOTE-GWO-MLP is efficient and can be used to classify multiclass and imbalanced fault conditions.

Article Highlights

- The issue of multiclass and imbalanced class outputs is addressed for improving predictive maintenance.
- A multiclass fault classifier based on clustering and optimised multi-layer perceptron with grey wolf is proposed.
- The robustness and feasibility of the proposed technique is validated on a complex hydraulic system dataset.

Keywords Clustering · Classification · Multiclass · Imbalanced data

1 Introduction

In recent years, considerable amount of resources have been dedicated to the area of classification and its applications in fault diagnosis using machine learning techniques

[1, 2]. As frequently observed in many disciplines is that most fault classification tasks are often multiclass and imbalanced in nature [2–4]. As a result, building a robust model to effectively handle such conditions is much more complicated because the complexity in the selection and

✉ Albert Buabeng, abuabeng@umat.edu.gh | ¹Faculty of Engineering, Mathematical Sciences Department, University of Mines and Technology, Tarkwa, Ghana. ²Faculty of Engineering, Mechanical Engineering Department, University of Mines and Technology, Tarkwa, Ghana. ³College of Sciences, Statistics and Actuarial Science Department, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana. ⁴Faculty of Mineral Resources Technology, Geomatic Engineering Department, University of Mines and Technology, Tarkwa, Ghana.

fine-tuning of model hyperparameters increases. Hence, most existing and well-known machine learning techniques fail to produce the desired results when experimented on a multiclass or imbalanced dataset or both.

In literature, the independent implementation of supervised or unsupervised learning algorithms over the years in various fault classification task has been proven to yield some level of satisfactory results [5]. However, these two major forms of learning possess their strength and limitations. For instance, among the widely used supervised algorithms for fault classification like Artificial Neural Networks (ANNs) [6–8], Support Vector Machine (SVM) [2, 9, 10], Linear Discriminant Analysis (LDA) [11–13] and Bayes classifiers [3, 14, 15] are considered superior in producing labels, but assumes that the objects classified are drawn from an independent and identical distribution, and as such does not consider their interdependencies [16].

With regards to the categories of unsupervised clustering algorithms, methods such as the partition-based [3, 17, 18], distribution-based [19–21], hierarchy-based [1, 22, 23] and the model-based [24, 25] are frequently used for classifying fault conditions. For instance, Amruthnath and Gupta [17] in their attempt to implementing rapid predictive maintenance in fault prediction and class detection utilised Gaussian Mixture Model (GMM) and K-Means. Lu et al. [21] used GMM for extracting sub-patterns for improving the predictive accuracy of heating load patterns. Also, Raptodimos and Lazakis [24] used the neural network-based Self Organising Map (SOM) for clustering marine engine data for condition monitoring purposes. Liu and Ge [1] proposed a fault classification scheme based on hierarchical clustering selection for complex industrial processes. Similarly, other studies such as Chen et al. [3], Park et al. [18] used KMeans, Skowron et al. [25] used SOM, Zhu et al. [19] and Hong et al. [20] used GMM whereas Blue et al. [22] and Barmada et al. [23] used hierarchy-based algorithms for classifying various fault conditions. Although these clustering techniques are deficient in producing labels, they consider the interdependencies within objects. This interdependency assumption provides supplementary constraints that can be leveraged upon for improving the generalisability of classifiers.

For these reasons, researchers from various disciplines have directed their focus into building a robust synergistic classifier by leveraging on the strength and weakness of the supervised and unsupervised learning algorithms [2, 5, 16]. On the contrary, little or no work to the best of our knowledge has reported on the systematic combination of unsupervised and supervised learning in fault classification which is predominantly multiclass and imbalanced by design. As a result, this study proposes a systematic synergistic framework for classifying fault conditions based on unsupervised clustering and supervised classification

techniques. Here, the supplementary constraints provided by the unsupervised clustering is fully utilised, thus, improving the generalisability and accuracy of the supervised classifiers in classifying fault conditions. Despite the promising results obtained with the proposed hybrid technique, especially when dealing with multiclass (where an input data vector has to be assigned one of multiple output classes), the problem of class imbalance needs to be fully addressed. As a result, the resampling strategy known as Synthetic Minority Over Sampling Technique (SMOTE) [26] is implemented in this study to create new instances of the minority class(es) before training the supervised ANN classifier.

In the context of supervised ANN classifiers, the Multi-Layer Perceptron (MLP) neural network is among the most versatile algorithms used in various fault classification tasks [27–29]. Compared with other well-known classifiers, MLP has the unique ability to dynamically create complex predictive functions and emulate human thinking and has proven its superiority in learning and modelling non-linear and complex relationships [28, 30, 31]. In addition to that, MLP is known to be less affected by data imbalance than other classifiers [32].

However, as observed in the MLP computational procedure, the initialisation of learning parameters and the choice of the learning optimisation algorithm is very crucial in arriving at the optimum solution to a given problem. It is widely known that MLP uses deterministic optimisation methods of backpropagation [33] and gradient descent for the assignment and updating of weights and biases in the network. Conversely, such optimisation methods are highly dependent on the initial solution and are most often prone to local optimal entrapment [30]. Thus, making them unreliable in practical situations. In view of this, many scholars have resorted to the use of stochastic optimisation methods (e.g. Genetic Algorithm (GA) [34], Particle Swarm Optimisation (PSO) [35], Differential Evolution (DE) [36], Ant Colony Optimization (ACO) [37], Gravitational Search Algorithm (GSA) [38] and Artificial Bee Colony (ABC) [39]) for training MLP due to their initialisation of the learning process with random solution(s) which are then evolved for improvement, thus, possessing high local optima avoidance capability.

However, it has been found that obtaining an optimal balance between the exploration and exploitation search phases of stochastic optimisation methods is a challenging problem due to their random nature [40]. As a result, physics-based algorithms like GSA achieves high performance in the exploration phase but slow in the exploitation phase [40, 41] thereby increasing its computational complexity. Swarm-based algorithms like ACO, PSO and ABC achieves low performance during the exploration

phase but improves gradually during the exploitation phase. Thus, the challenge of local optima still persists.

Consequently, this has led to the development of new and advanced meta-heuristic algorithms for training MLP such as the hybrid PSO-GSA [42], PSO with Autonomous Groups (PSOAG) [43], Invasive Weed Optimiser (IWO) [44], Chemical Reaction Optimiser (CRO) [45], Stochastic Fractal Search (SFS) [46], Biogeography-Based Optimizer (BBO) [47], Adaptive Best-Mass GSA (ABMGSA) [48], Chimp Optimisation Algorithm (COA) [49], Dragonfly Optimisation Algorithm (DOA) [50], Salp Swarm Optimiser (SSO) [51], Social Spider Optimisation Algorithm (SSOA), Grey Wolf Optimisation (GWO) [41], Equilibrium Optimiser (EO) [52], Sine Cosine Algorithm (SCA) [53], Modified Sine Cosine Algorithm (MSCA) [54], Whale Optimisation Algorithm (WOA) [55], Improved WOA [56], Modified WOA [57] among others.

However, considering the theory of No Free Lunch [58] which implies that no single optimiser can boast of being superior to the others for all optimisation tasks, and as such meta-heuristics are task-specific, inspires the selection of GWO as the choice for training the MLP classifier in this study. GWO is a recently proposed Swarm-based algorithm which emulates the social leadership hierarchy and unique hunting mechanism of Grey Wolves [59]. When compared to other metaheuristics, the unique search mechanism of the GWO significantly enhances the algorithm's ability to optimally achieve a balance between the exploration and exploitation search phases. Also, the novelty in the mathematical framework of GWO makes it possible to dynamically relocate solutions in an n -dimensional space, thus, mimic the hunting strategy of grey wolves in nature (i.e. chasing and encircling of prey) [60]. Moreover, GWO reserves the three best solutions for enhancing exploration and utilises only one position vector which simplifies the algorithm and makes it computationally less expensive when compared to other metaheuristics. Furthermore, the GWO is noted to outperform most swarm-based algorithms in balancing between the two search phases [40, 41]. Due to GWO's unique capabilities, its usage in solving real-life problems has been on the rise [60–62] which has resulted in different variants of the GWO (e.g. hybrid GWO with mutation operator, GWO with opposition-based learning and chaotic local search for integer and mixed-integer optimisation problems, random walk GWO, opposition-based chaotic GWO, sine cosine GWO, etc.) in literature.

This study proposes a systematic hybrid technique for the efficient classification of multiclass and imbalanced fault conditions. This was achieved via the combination of unsupervised clustering and supervised MLP neural network optimised with GWO. The imbalanced fault conditions were handled using SMOTE. The proposed

hybrid technique is found to be versatile and effective in improving the classification of fault conditions in the presence of multiclass and imbalanced data distribution while reducing the computational complexity of the classifier. The efficiency of the proposed hybrid technique is verified using the University of California, Irvine (UCI) established machine learning repository dataset of multi-sensors [11, 63, 64].

1.1 Contributions of the study

The contributions of this study are to:

- (a) propose a hybrid synergistic predictive maintenance framework capable of effectively handling multiclass and imbalanced datasets; and
- (b) evaluate and compare the performance of the proposed hybrid approach with its distinct modification (variants).

The stated contributions will advance the field of predictive maintenance by improving the accuracy of fault diagnosis as the issue of multiclass and imbalance impedes the ability of machine learning algorithms to producing desired results.

The remaining sections are organised as follows: Sect. 2 briefly describes the experimental condition monitoring dataset. Section 3 discusses the methodology of the proposed hybrid technique; feature extraction and selection, comparative study of some unsupervised clustering algorithms, handling of class imbalance by SMOTE, and MLP optimised by GWO. The application of the proposed technique to fault classification as well as its superiority to other distinct modifications of the proposed technique is demonstrated in Sect. 4. Section 5 concludes the findings.

2 Hydraulic system dataset

The hydraulic system dataset is obtained from the University of California Irvine (UCI) machine learning repository (<http://archive.ics.uci.edu/ml/datasets/Condition+monitoring+of+hydraulic+systems>). The data consist of 2205 instances and 43,680 attributes recorded from 17 process sensors with varying sampling rates. The 17 processes comprised of six pressure sensors, four temperature sensors, two volume flow sensors and five different sensors for recording motor power, vibration, cooling efficiency, cooling power and system efficiency. The dataset also contains fault scenarios depicting the variations in fault condition of four major components including hydraulic accumulator, cooler, internal pump leakage and valve. The details of the four major components are shown in Table 1.

Table 1 Details of fault components

Fault type	Unit	No. of classes	Class output	Remark
Accumulator	bar	4	90	Close to total failure
			100	Severely reduced pressure
			115	Slightly reduce pressure
			130	Optimal pressure
Cooler condition	%	3	3	Close to total failure
			20	Reduced efficiency
			100	Full efficiency
Internal pump leakage	–	3	0	No leakage
			1	Week leakage
			2	Severe leakage
Valve condition	%	4	73	Close to total failure
			80	Severe lag
			90	Small lag
			100	Optimal switching behaviour

Fig. 1 Schematic of the proposed hybrid technique

3 Methodology

Figure 1 shows the schematic of the proposed hybrid classifier from feature extraction stage through to the validation of the classification model stage. The details of the various stages are discussed in the subsequent subsections.

3.1 Feature extraction and selection

Different statistical time-domain features were extracted from the historical hydraulic system dataset. The statistical time-domain functions such as the mean, median, variance, standard deviation, skewness, kurtosis and position of maximum value were considered. The estimates for the statistical time-domain features were achieved by first partitioning each sensor data into various time intervals via

determining the average of each feature within individual sensors, thus, ensuring uniformity across all cycles. Based on the characteristics of the 17 process sensors as indicated in Sect. 2, the sensors were partitioned as follows: the six Pressure Sensor (PS): PS1 (13), PS2 (14), PS3 (18), PS4 (25), PS5 (19), PS6 (19); the four Temperature Sensor (TS): TS1 (7), TS2 (7), TS3 (8), TS4 (15); volume Flow Sensor (FS): FS1 (18), FS2 (18); motor Efficiency Power Sensor (EPS): EPS1 (13), Vibration Sensor (VS) 15; Cooling Efficiency (CE) sensor 13; Cooling Power (CP) sensor 13; System Efficiency (SE) sensor 23. The statistical time-domain features extracted from the various segments of the dataset resulted in a complete feature vector of 1806 features from the original 43,680.

However, using the entire 1806 extracted features for training a classifier will be computationally expensive and can lead to overfitting as well as increasing uncertainty in the modelling process which can negatively influence the ability of the classifier to efficiently classify target outputs [65]. This is due to the involvement of redundant and irrelevant features. Hence, after transforming the historical dataset into statistical time-domain features, the extracted features that captured relevant fault characteristics were identified using correlation-based feature subset selection. A better correlation of selected features with a fault

process is required in producing the context needed for discriminating between the relevant and irrelevant of the extracted features [66, 67]. From the obtained 1806 features, 20 most significant continuous features were selected based on their Pearson's correlation coefficient. The 20 most significant features were found to be adequate as implemented by Helwig et al. [11].

Table 2 shows the selected time-domain features for each fault type and their corresponding Pearson correlation coefficient $|r|$. As observed, the cooler features mainly comprised of various statistical time-domain features (all with correlation coefficients ≥ 0.9913) from the Cooling Efficiency (CE) sensor. Similarly, the valve features which comprised of features from all the pressure sensors (PS1, 2 and 3), the volume of flow (FS) and the motor power (EPS) sensor obtained correlation coefficients ranging from 0.8839 to 0.9866. The pump features were highly influenced by features from the System's Efficiency (SE), volume flow (FS), pressure (PS) and the motor power (EPS) sensors. Their $|r|$ values ranged from as low as 0.4236 to as high as 0.9458. With regards to the accumulator features, their $|r|$ values were significantly lower (from 0.4664 to 0.5721). The obtained accumulator features composed of features from all the pressure sensors (PS1, 2 and 3), FS and EPS. This confirms the results obtained by Helwig

Table 2 Highly correlated features selected for specific fault types

Sn	Cooler features				Pump features				Valve features				Accumulator features			
	Sns	Fxn	Int	$ r $	Sns	Fxn	Int	$ r $	Sns	Fxn	Int	$ r $	Sns	Fxn	Int	$ r $
1	CE	MD	12	0.9934	SE	MD	7	0.9458	PS2	SKW	4	0.9866	PS3	STD	3	0.5721
2	CE	ME	12	0.9934	SE	ME	7	0.9449	PS2	ME	4	0.9766	FS1	SKW	17	0.5376
3	CE	MAX	12	0.9934	SE	ME	6	0.9193	EPS1	SKW	4	0.9705	PS1	STD	13	0.5302
4	CE	ME	11	0.9933	SE	MD	6	0.9193	PS1	SKW	4	0.9657	PS3	MAX	3	0.5247
5	CE	MD	11	0.9932	SE	MAX	6	0.9193	EPS1	KUR	4	0.9621	PS1	MD	2	0.5170
6	CE	ME	13	0.9932	SE	MAX	7	0.9182	PS1	KUR	4	0.9576	FS1	STD	7	0.5117
7	CE	MD	13	0.9932	FS1	ME	6	0.8650	EPS1	VAR	4	0.9513	PS1	KUR	6	0.5092
8	CE	MAX	11	0.9932	FS1	MD	6	0.8648	EPS1	STD	4	0.9488	PS3	MD	7	0.5028
9	CE	MAX	13	0.9932	FS1	MAX	6	0.6672	PS1	VAR	4	0.9482	FS1	ME	7	0.5014
10	CE	MD	10	0.9929	SE	VAR	2	0.5834	PS1	STD	4	0.9452	FS1	MD	7	0.5014
11	CE	ME	10	0.9929	SE	ME	1	0.4692	PS3	ME	4	0.9433	FS1	MAX	2	0.4976
12	CE	MAX	10	0.9927	PS3	VAR	10	0.4370	PS2	MD	4	0.9415	PS1	KUR	12	0.4852
13	CE	MAX	9	0.9922	SE	ME	12	0.4306	FS1	MAX	5	0.9325	FS1	STD	2	0.4847
14	CE	ME	9	0.9922	SE	ME	13	0.4306	PS3	MAX	5	0.9325	PS3	VAR	3	0.4830
15	CE	MD	9	0.9921	SE	MD	12	0.4306	PS3	MD	4	0.9297	FS1	ME	2	0.4821
16	CE	ME	1	0.9919	SE	MD	13	0.4306	FS1	STD	5	0.9296	EPS1	SKW	2	0.4753
17	CE	MD	1	0.9919	SE	MAX	12	0.4306	PS3	STD	5	0.9253	EPS1	KUR	12	0.4730
18	CE	ME	8	0.9917	SE	MAX	13	0.4306	FS1	VAR	5	0.8992	EPS1	SKW	12	0.4707
19	CE	MD	8	0.9917	EPS1	ME	1	0.4263	PS2	STD	5	0.8839	PS1	STD	11	0.4672
20	CE	MAX	8	0.9913	EPS1	MAX	9	0.4236	PS1	ME	4	0.8763	PS3	ME	3	0.4664

NB Sensor (Sns), Statistical Function (Fxn), Time Interval within Cycles (Int), correlation Coefficient ($|r|$); Statistical Functions: Mean (ME), Median (MD), Variance (VAR), Standard Deviation (SD), Skewness (SKW), Kurtosis (KUR) and Position of Maximum Value (MAX)

et al. [11]. These 20 most significant features selected for each fault type served as inputs to the proposed hybrid GWO-MLP classifier.

However, the robustness and diversity of the proposed GWO-MLP classifier will be greatly affected by the inherent multiclass and imbalanced class distribution of those extracted features. As a means of improving the diversity and performance of the GWO-MLP technique in classifying fault conditions, unsupervised clustering was performed to group the features into distinct classes. This will simultaneously increase classification performance and significantly reduce the complexity in fault classification.

3.2 Unsupervised clustering techniques

In literature, clustering of high-dimensional data has proven to be an effective technique of grouping data into distinct features and also as a pre-processing phase for most machine learning frameworks, especially in predictive maintenance. Among the categories of clustering techniques, the partition-based, the distribution-based, hierarchy-based and the neural network-based are widely used for classifying fault conditions [1, 17, 21, 25].

However, these clustering techniques frequently used in developing models for predictive maintenance involves several parameters, mostly operating in high dimensional spaces, while managing the influences from noisy, incomplete or sampled data [68]. Thus, selecting the right clustering technique for a given task is paramount for the optimal separation of a dataset into distinct clusters since their performance can vary significantly for different applications and data types (i.e. not exhaustive). Generally, K-Means have widely been used in clustering fault conditions, however, several clustering algorithms such as SOM, GMM, hierarchical algorithms, among others are known to produce desirable outputs [69, 70]. This raises the question, which clustering technique should be chosen for pre-processing the fault features for further analysis. Nonetheless, a comparative approach is adopted in this study as a means to select a suitable clustering technique. That is, the performance of four different categories of clustering algorithms (partition-based K-Means, model-based SOM, distribution-based GMM and hierarchy-based) known to consistently produce high-quality clusters and widely used for fault classification are compared.

3.2.1 Partition-based K-means

K-Means [71] is a widely used iterative technique for partitioning data into a given number of clusters. Suppose that $X = \{x_1, x_2, \dots, x_n\}$ represents the time-domain features extracted from the historical sensor dataset, where each x_i denotes an instance in X . The objective of the

K-Means algorithm is to partition X into k clusters ($k \leq n$), $\{C_1, C_2, \dots, C_k\}$. Denoting the number of instances in cluster C_j as n_j , the respective centroid c_j of cluster C_j is defined as Eq. (1).

$$c_j = \frac{1}{n_j} \sum_{x_i \in C_j} x_i \quad (1)$$

Using Eq. (1), the initial composition of clusters is obtained by assigning each element x_i in X to the nearest centroid. After the first assignment, the centroids are updated and further reallocation is made. The procedure is repeated until the centroids do not change anymore (converges). That is, K-Means attempts to partition data such that it minimises the sum of distance (similarity) between the cluster components and the centroids using the Sum of Squares Error (SSE) criterion expressed as Eq. (2).

$$SSE = \sum_{j=1}^k \sum_{x_i \in C_j} \|x_i - c_j\|^2 \quad (2)$$

K-Means algorithm is sensitive to the initially selected cluster centroids. As such, the optimal number of clusters k must be specified in advance. The time complexity of K-Means is expressed as Big O notation is $O(nkt)$, where n is the number of data points in X , k is the number of clusters and t , the number of iterations [72].

3.2.2 Neural network-based self-organising map (SOM)

SOM [73] is an effective neural network approach for clustering high dimensional data by mapping the dataset to a lower-dimensional grid (usually 2D). Thus, the method possesses the ability to transform complex and nonlinear characteristics from a higher dimensional space to a lower dimension while preserving the topology of the data [74]. Unlike the traditional ANN, the implementation of the SOM network is generally two-layered, an input layer and the output layer. The features are fed as inputs which in turn produces clusters as output. This is achieved by allocating a neuron for each input and assigning weights by training the network.

The training is iteratively implemented by an initial assignment of weights randomly. Once the set of weights w , has been assigned to the neurons, the distance between the weight vector w and its corresponding n -dimensional input vector x are estimated using the Euclidean distance shown in Eq. (3).

$$\|x - w\| = \sqrt{\sum_{i=1}^n (x_i - w_i)^2} \quad (3)$$

The weight close to the corresponding input is denoted as the Best Matching Unit (BMU). The weight vector of the BMU and its neighbouring neurons are updated so as to further increase the proximity of the weights to the input vector. The update is implemented using the rule expressed as Eq. (4).

$$w(t+1) = w(t) + \alpha(t) \cdot h_{lm}(x - w(t)) \quad (4)$$

where $0 < \alpha(t) < 1$ is the learning rate, h_{lm} is the neighbourhood function expressed as Eq. (5).

$$h_{lm} = \exp\left(-\frac{\|l - m\|^2}{2\sigma(t)^2}\right) \quad (5)$$

where h_{lm} monotonically decreases with respect to the distance between the best matching neuron l and its neighbouring output nodes m on the grid. σ is the width of the topological neighbourhood. The training (Eq. (4)) is repeated iteratively until a convergence criterion is met. The time complexity of SOM is expressed as $O(mdt)$, where m is the number of units in the network grid, d is the dimension of vectors and t , the number of iterations which depends on the layer specified in the algorithm [72, 75].

3.2.3 Distribution-based gaussian mixture model (GMM)

GMM [76] is a parametric probability-based algorithm which assumes that the dataset is generated from a mixture of K components density functions, where each component k is characterised as a weighted sum of Gaussian probability densities as shown in Eq. (6).

$$p(x) = \sum_{k=1}^K w_k f\left(X|\mu_k, \sum_k\right) \quad (6)$$

where X is the dataset, and w_k is the mixture weight function, such that $\sum_{k=1}^K w_k = 1$. The Gaussian density function $f(X|\mu_k, \sum_k)$ with μ_k and $\sum_k$ as the mean vector and covariance matrix respectively is given by Eq. (7).

$$f\left(X|\mu_k, \sum_k\right) = \frac{1}{\sqrt{(2\pi)^k |\sum_k|}} \exp\left[\frac{-\frac{1}{2}(X - \mu_k)^T}{\sum_k (X - \mu_k)}\right] \quad (7)$$

GMM estimate the parameters for each component k using the iterative Expectation Maximisation (EM) algorithm [77], given that the log-likelihood of X with N observations is expressed as Eq. (8).

$$L = \sum_{i=1}^N \ln \left[\sum_{k=1}^K w_k f\left(X|\mu_k, \sum_k\right) \right] \quad (8)$$

EM is an efficient algorithm for finding the maximum likelihood estimate of parameters for X with time complexity expressed as $O(n^2kt)$, where n is the number of data points in X , k is the number of clusters and t , the number of iterations [72].

3.2.4 Hierarchy-based hierarchical agglomerative clustering (HAC)

HAC [78] is a widely used clustering procedure of grouping objects into a hierarchy of clusters. HAC algorithm begins by initially assuming each object to be a single cluster. Based on a defined similarity measure (distance), two clusters in closest proximity are aggregated or merged as one cluster. The merging procedure is repeated until all the clusters have been agglomerated as one big cluster.

In literature, there exist five linkage criteria for determining the distance between two clusters; single, complete, average, centroid and Ward. These linkage criteria are defined based on how they estimate the distance between two clusters [79]. Suppose that cluster C_i and another cluster aggregated from clusters C_j and C_k , the distance $d(C_i, C_j + C_k)$ is estimated using Eq. (9).

$$d(C_i, C_j + C_k) = w_1 d(C_i, C_j) + w_2 d(C_i, C_k) + w_3 d(C_j, C_k) + w_4 \left| \frac{d(C_i, C_j)}{-d(C_i, C_k)} \right| \quad (9)$$

where w_i are the weighting factors determined by a linkage criteria.

In this study, the Ward linkage [80] is adopted due to its ability to produce better clustering results. Moreover, most of the other linkage criteria tend to form chains or "globular-shaped" clusters [81].

Hence, $w_1 = \frac{n_{C_i} + n_{C_j}}{n_{C_i} + n_{C_j} + n_{C_k}}$, $w_2 = \frac{n_{C_i} + n_{C_k}}{n_{C_i} + n_{C_j} + n_{C_k}}$, $w_3 = -\frac{n_{C_j}}{n_{C_i} + n_{C_j} + n_{C_k}}$ and $w_4 = 0$. n_{C_i} , n_{C_j} and n_{C_k} are the number of instances in clusters C_i , C_j and C_k respectively. The time complexity of HAC is expressed as $O(n^2k)$, where n is the number of data points in X and k is the number of clusters [72].

3.2.5 Optimal number of clusters estimation

The clustering algorithms (K-Means, SOM, GMM and HAC) utilised in this study requires the initialisation of some key parameter(s) like the number of clusters, that have great implication on the clustering partitions [82]. However, the determination of an optimal number of clusters k requires either prior knowledge of the dataset or time-consuming sequential trial and error computation. The determination further becomes difficult when dealing with high-dimensional datasets as observed in this study. Consequently,

to ascertain the veracity of the clustering results for varying number of clusters produced by each approach, this study compares five effective validation measures namely Calinski-Harabasz Index (CHI), Silhouette Index (SI), Davies-Bouldin Index (DBI), Gap Index (GI), Krzanowski and Lai Index (KLI).

3.3 Calinski–harabasz index (CHI)

The CHI [83] is a goodness of clustering measure that evaluates the validity of clusters based on the within- and between-cluster distances as shown in Eq. (10).

$$CHI = \frac{\frac{\text{trace}\left(\sum_{i=1}^k n_k (c_k - \bar{x})(c_k - \bar{x})^T\right)}{(k-1)}}{\frac{\text{trace}\left(\sum_{i=1}^k \sum_{j \in C_k} (x_j - c_k)(x_j - c_k)^T\right)}{(n-k)}} \quad (10)$$

where c_k and $\bar{x}$ are the centroids of cluster C_k and the dataset X respectively, n and k are the number of instances and clusters respectively. The optimal number of clusters is obtained when the index is maximum.

3.4 Silhouette index (SI)

The SI [84] assesses the average distance between each point in cluster C_i and other points in neighbouring clusters C_k where $k \neq i$. The SI is shown as Eq. (11).

$$SI = \frac{\sum_{i=1}^n \min_{k \neq i} (d_{iC_k}) - A(x)}{\max [A(x), \sum_{i=1}^n \min_{k \neq i} (d_{iC_k})]}, \quad SI \in [-1, 1] \quad (11)$$

where $A(x) = \frac{\sum_{j \in C_i} d_{ij}}{n_i - 1}$ is the average dissimilarity of point i to all points of cluster C_i and $d_{iC_k} = \frac{\sum_{j \in C_k} d_{ij}}{n_k}$ is the average dissimilarity of point i to all points of cluster C_k . The optimal number of clusters is obtained when the index is maximum.

3.5 Davies–bouldin index (DBI)

The DBI [85] evaluates the validity of clusters based on the average distance between each cluster C_i and all other neighbouring clusters C . The maximum value is then assigned to C_i based on the similarity between clusters as shown in Eq. (12).

$$DBI(k) = \frac{1}{k} \sum_{i=1}^q \max_{i \neq j} \left(\frac{\delta_i + \delta_j}{d_{ij}} \right) \quad (12)$$

where $d_{ij} = \sqrt{\sum_{l=1}^p |c_{il} - c_{jl}|^2}$ is the distance between centroids of cluster C_i and C_j for $i, j = 1, \dots, k$, $\delta_i = \sqrt{\frac{1}{n_i} \sum_{m \in C_i} \sum_{l=1}^p |x_{ml} - c_{il}|^2}$ is the standard deviation

of the distance of points to the centroid in cluster C_i . The optimal number of clusters is obtained when the index is minimum.

3.6 Gap index (GI)

The GI proposed by Tibshirani et al. [86] assesses the results of clustering algorithms by comparing the within-cluster dissimilarities with that expected under an appropriate reference null distribution. The GI is estimated using Eq. (13).

$$GI(k) = \frac{1}{B} \sum_{b=1}^B \log W_{kb} - \log W_k \quad (13)$$

where the number of reference dataset uniformly generated is denoted as B , W_{kb} is the within-cluster dissimilarity matrix defined as $\sum_{i=1}^k \sum_{j \in C_k} (x_i - c_k)(x_j - c_k)^T$. The optimal number of clusters is obtained when the index is maximum.

3.7 Krzanowski and lai index (KLI)

The KLI [87] is an evaluation criterion for obtaining the optimal value of clusters for a given dataset. The optimal number of clusters is obtained when the stopping criteria (Eq. (14)) is maximum.

$$KLI(k) = \left| \frac{DIFF_k}{DIFF_{k+1}} \right| \quad (14)$$

where $DIFF_k = (k-1)^{\frac{2}{p}} \cdot \text{trace}(W_{k-1}) - k^{\frac{2}{p}} \cdot \text{trace}(W_k)$ is the difference in the function when the number of groups in the cluster is increased from $(k-1)$ to k .

3.8 Handling class imbalance

The effect of class imbalance in building robust machine learning models, especially, in classification task has thoroughly been studied [2–4]. Class imbalance does not only lead to biased predictions in favour of the majority class but also impedes the performance of trained models in predicting the minority class when exposed to a new set of features. Thus, failing to produce the desired results. The effects worsen when dealing with multiclass datasets—which is the domain of most fault classification tasks.

Review of previous studies has shown that the frequently used metrics in measuring the extent of class imbalance are the Empirical Distribution (ED) [88], Class Frequencies (CF) [89], Imbalance Ratio (IR) [90], Imbalance Degree (ID) [91] and the Likelihood-Ratio based Imbalance Degree (LRID) [92]. Although the ED and CF metrics are both informative in class imbalance, they are deficient in

measuring the extent of imbalance of highly multiclass dataset [91]. The IR only utilises information from both the smallest minority and the largest majority class, hence, unable to capture the class distribution completely. This makes ED, CF and the IR suitable for imbalance learning on binary dataset not multiclass dataset. The ID and LRID on the other hand address the challenges in ED, CF and the IR by providing reliable estimates for multiclass data and thus, are adopted in this study for measuring the extent of class imbalance in the considered hydraulic dataset.

3.8.1 Imbalance problem formulation

Suppose that $x \in \mathbb{R}^{p \times 1}$ with its label y is a generative classification model learning the joint distribution expressed as Eq. (15).

$$p(x, y) = p(y)p(x|y) \quad (15)$$

where the prior knowledge on the probability of y is given as $p(y)$. Assuming that there are C possible outcomes for y , then $y = \{y_1, y_2, \dots, y_C\}$. Then each outcome y_i is associated with the probability p_i such that $\sum_{i=1}^C p_i = 1$. The frequencies of the possible labels $n = \{n_1, n_2, \dots, n_C\}$ can be modelled as a multinomial distribution (N, p) with parameters N and $p = \{p_1, p_2, \dots, p_C\}$. The estimate of p_i is given as $\hat{p}_i = \frac{n_i}{N}$. Thus, $\hat{p} = \{\hat{p}_1, \hat{p}_2, \dots, \hat{p}_C\}$.

For a perfectly balanced dataset, $p_i = \frac{1}{C}$ for $i = 1, 2, \dots, C$. Let $b = \{\frac{1}{C}, \frac{1}{C}, \dots, \frac{1}{C}\}$ denote the class distribution vector of the perfectly balanced dataset. An imbalanced dataset is defined by $\hat{p}_i \geq \frac{1}{C}$ as the majority class and $\hat{p}_i \leq \frac{1}{C}$ is the minority class. Thus, imbalance learning provides a metric to estimate the difference between $\hat{p}$ and b .

3.8.2 Imbalance degree (ID)

ID [91] is a high-resolution metric that considers information from all classes (both minority and majority) for imbalance learning which summarises the difference between $\hat{p}$ and b using Eq. (16).

$$ID = \frac{d(\hat{p}, b)}{d(p_m, b)} + m - 1 \quad (16)$$

where m is the number of minority classes, p_m denotes the distribution showing exactly m minority classes the highest distance. The maximum distance between b and all possible p_m is given as $d(p_m, b)$. Thus, for a balanced dataset, $ID = 1$, whilst $ID > 1$ signifies an imbalanced dataset.

In this study, the Hellinger Distance (HD) [93] is utilised in instantiating ID as it produces significant and robust

summaries of the extent of class imbalance [91]. The HD is expressed as Eq. (17).

$$HD = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^C (\sqrt{\hat{p}_i} - \sqrt{b_i})^2} \quad (17)$$

3.8.3 Likelihood-ratio imbalance degree (LRID)

Although the ID metric is informative in measuring the extent of imbalance in a multiclass dataset, the choice of distance function d , chosen can impact negatively on the results as there is no rule in choosing an appropriate distance function in literature. As a result, Zhu et al. [92] proposed another high-resolution metric, the LRID which addresses the challenge of ID. The simplicity of LRID makes it convenient in practice as it does not require testing with different distance or similarity functions.

LRID exploits the Likelihood-Ratio (LR) test in estimating a numeric value that can adequately summarise the difference between $\hat{p}$ and b . The LR test (Eq. (18)) for the multinomial distribution (N, p) test the hypothesis $H_0 : p = b$ against $H_1 : p = \hat{p}$ [94].

$$LR = -2 \ln \frac{L(b|n)}{L(\hat{p}|n)} \quad (18)$$

where $L(\cdot)$ is the likelihood function. Thus, for a balanced dataset, $L(b|n) = L(\hat{p}|n)$ leading to $LR = 0$, whilst $L(b|n) < L(\hat{p}|n)$ leads to $LR > 0$ for imbalanced dataset. This implies that the value of the test statistic becomes larger as the difference between $\hat{p}$ and b grows larger. Based on the LR statistic in Eq. (18), LRID is expressed as shown in Eq. (19).

$$LRID = -2 \sum_{i=1}^C n_i \ln \frac{b_i}{\hat{p}_i} \quad (19)$$

When $\hat{p}_i = \frac{n_i}{N}$, Eq. (19) can be simplified as Eq. (20).

$$LRID = -2 \sum_{i=1}^C n_i \ln \frac{N}{C n_i} \quad (20)$$

Once the extent of class imbalance has been estimated using Eqs. (16, 20), SMOTE [26] was then applied as a resampling strategy for creating new synthetic instances of the minority class(es). This will help deal with the unequal class representation and mitigate the biases introduced by the imbalanced data distribution.

3.9 Classification and parameter optimisation

3.9.1 Multi-layer perceptron (MLP) neural network

The Multi-Layer Perceptron (MLP) is one of the most versatile unidirectional Feed-Forward Neural Networks (FNNs) with one hidden layer. Its unique ability to dynamically create complex predictive functions proves its superiority in learning and modelling non-linear and complex relationships [28, 30, 31]. As a result, the MLP algorithm is being used in various research disciplines including fault classification [27–29].

In a Multi-Layer Perceptron (MLP), the network is generally structured in three layers. Starting with an input layer, through a middle-hidden layer containing neuron(s) and finally ending with an output layer. The procedure starts by feeding the algorithm with input features through which weights and biases are estimated for calculating the final output.

The weighted sums of the input features are estimated using Eq. (21).

$$s_j = \sum_{i=1}^n (W_{ij} \cdot X_i) - \theta_j, j = 1, 2, \dots, h \quad (21)$$

where X_i is the i^{th} input feature with n number of input nodes, W_{ij} is the weight connecting the i^{th} node of the input layer and the j^{th} node of the hidden layer of neuron(s). The bias (threshold) of the j^{th} hidden node is indicated as θ_j .

The output of each hidden node is then estimated using Eq. (22).

$$S_j = \text{sigmoid}(s_j) = \frac{1}{(1 + \exp(-s_j))}, j = 1, 2, \dots, h \quad (22)$$

Based on the outputs from the hidden nodes S_j , the final outputs are then estimated using Eqs. (23) and (24).

$$o_k = \sum_{j=1}^h (w_{jk} \cdot S_j) - \theta'_k, k = 1, 2, \dots, m \quad (23)$$

$$O_k = \text{sigmoid}(o_k) = \frac{1}{(1 + \exp(-o_k))}, k = 1, 2, \dots, m \quad (24)$$

where w_{jk} is the weight connecting the j^{th} node of the hidden layer and the k^{th} node of the output layer. The bias (threshold) of the k^{th} hidden node is indicated as θ'_k . The time complexity of the MLP algorithm is expressed as $O(s(h + o))$, where s is the population size, h is the number of hidden neurons and o is the outputs [57].

Observing Eqs. (21–24) critically, it can be inferred that, for a given number of input features, the final output of the MLP neural network depends highly on the estimates of the weights and biases. Thus, finding optimal estimates

for the weights and biases is imperative for obtaining a desirable relation between the input features and the final output.

In classical FNNs, deterministic methods such as back-propagation [33] and gradient-based optimisation methods are often used in estimating the weights and biases. However, as discussed in the introductory section, these deterministic methods are prone to local optimal entrapment due to their dependence on the initial solution [30, 40], thus, making them unreliable in practical situations.

In this study, this deficiency is corrected by the utilisation of a stochastic-based method, the Grey Wolf Optimisation (GWO) which initialises the learning process with random solution(s). The selection of GWO over other stochastic optimisation methods such as the GA, PSO, ACO, ABC, DE, among others is basically due to GWO's ability to optimally achieve a search balance between the exploration and exploitation phases [40, 41, 62].

3.9.2 Grey wolf optimisation (GWO) algorithm

The GWO is a recently proposed stochastic and swarm-based algorithm which mimics the social leadership hierarchy and hunting behaviour of Grey Wolves [41]. This unique search mechanism based on the leadership hierarchy and hunting strategy of grey wolves ensures a global exploration and prevents the local entrapment of the search agents. For these reasons, the GWO has gained significant attention in diverse disciplines due to its potential in solving problems in machine learning, engineering, medicine, environmental modelling, among others [60, 61]. Also, the impressive characteristics of the GWO have inspired the interest of researchers to further explore the algorithm by modifying the population structure and hierarchy, update the mechanism or adding a new parameter to the classical GWO.

Notable among such recent developments include the modified GWO [95], the modified augmented Lagrangian GWO [96], the grouped GWO [97], the weighted distance GWO [98], the GWO with dynamic adaptation of parameters [99], the hybrid GWO with mutation operator [100], the opposition-based chaotic GWO [101], the random walk GWO [102], the memory-based GWO [103], the sine cosine GWO [104], the modified sine cosine GWO [54] among others. However, just like any metaheuristic optimiser, the GWO variants are task-specific and thus possess some limitations in practice. These limitations are primarily due to the No Free Lunch theorem [58], which states that no single optimiser can optimally solve all optimisation problems.

In the GWO algorithm, the population consist of the alpha (α), beta (β), delta (δ) and omega (ω). The α , β

and δ are respectively regarded as the fittest within the population and as such leads the other wolves, ω , in the hunt (optimisation) for potential preys in a given search space.

In the course of the hunting (optimisation), the encircling mechanism of the wolves is expressed mathematically as shown in Eqs. (25) and (26) respectively.

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (25)$$

$$\vec{X}(t+1) = |\vec{X}_p(t) - \vec{A} \cdot \vec{D}| \quad (26)$$

where $\vec{X}(t)$ and $\vec{X}_p(t)$ are the position vectors of the grey wolf and prey at the t^{th} iteration. $\vec{D}$ is a position re-adjustment parameter, $\vec{A}$ and $\vec{C}$ are coefficient vectors which are estimated using Eqs. (27) and (28) respectively.

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (27)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (28)$$

where $\vec{a}$ linearly decreases from 2 to 0, $\vec{r}_1$ and $\vec{r}_2$ are random vector generated within 0 and 1. Detailed mathematical illustration of the concept of prey encircling can be found in the work of Mirjalili et al. [41] and Prajindra et al. [59].

In the GWO algorithm, the α , β and δ are always assumed to be in closest proximity to the prey (optimum). As such, the first three best solutions are assigned as α , β and δ respectively. Based on the best three solutions obtained, the other search agents (ω inclusive) are required to re-position with respect to the best search agents (α , β and δ) using the mathematical expressions shown in Eqs. (29–31).

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \quad (29)$$

$$\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \quad (30)$$

$$\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (31)$$

where $\vec{D}_\alpha$, $\vec{D}_\beta$ and $\vec{D}_\delta$ are the approximate distances (step size) of the current solution (ω) towards the α , β and δ respectively, with $\vec{X}_\alpha$, $\vec{X}_\beta$ and $\vec{X}_\delta$ being their respective positions. $\vec{C}_1$, $\vec{C}_2$ and $\vec{C}_3$ are random vectors, $\vec{X}$ denotes the initial position of the current solution.

The final position $\vec{X}(t+1)$ of the current solution (ω) is then estimated by averaging Eqs. (32–34) as shown in Eq. (35).

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1(\vec{D}_\alpha) \quad (32)$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2(\vec{D}_\beta) \quad (33)$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3(\vec{D}_\delta) \quad (34)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (35)$$

The adoption of GWO in this paper is based not only on its simplicity and ability to achieve global minima but also outperforms most stochastic algorithms in obtaining a desirable balance between the exploration and exploitation phases [40, 41]. This is made possible due to the random and adaptive vectors $\vec{A}$ and $\vec{C}$ in Eqs. (27) and (28).

As the position of the α , β and δ always determines the search strategy of grey wolves, the random vector $\vec{A} = [-2a, 2a]$ initiates the exploration of the search space. That is, candidate solutions diverge from a prey in search of another and fitter prey when $|\vec{A}| > 1$. Likewise, they converge towards the prey when $|\vec{A}| < 1$. This enhances GWO's ability to search globally within the given search space. In addition to enhancing the exploration phase, the random vector $\vec{C} = [0, 2][0, 2]$ randomly provides weights to the prey based on its position from a wolf. The random weights are generated throughout the optimisation process in order to stochastically emphasise ($C < 1$) or deemphasise ($C > 1$) the effect of prey in defining the distance in Eq. (25). Thus, making $\vec{C}$ useful in avoiding the issue of local optima entrapment. The time complexity of the GWO algorithm is estimated using the expression $O(snt)$, where s is the population size, n is the dimension of problem or features and t , the maximum number of iterations [105]. The pseudo-code for the GWO is presented in Fig. 2

3.9.3 MLP NN weights and biases estimation using GWO

In literature, there are generally three ways for using stochastic algorithms such as GWO in training the MLP NNs – either to determine the optimal weights and biases, suitable structure or parameters for a gradient-based learner [40]. In this paper, GWO is employed for determining the optimal combination of weights and biases with the objective of improving the classification performance of MLP NN. This was necessary because finding optimal estimates for the weights and biases is imperative for obtaining a relation that fits the desired output. However, in designing the GWO for training the MLP NN, a vector $\vec{V}$, containing

Fig. 2 GWO Pseudo Code

```

Specify the maximum number of iterations  $T$ 
Initialise the search agents  $X_i (i = 1, 2, \dots, n)$ 
Initialise  $a$ ,  $A$  and  $C$ 
Fitness estimation of each search agent
 $X_\alpha$  = the best search agent
 $X_\beta$  = the second-best search agent
 $X_\delta$  = the third-best search agent
Initialise  $t = 1$ 
while ( $t < T$ ) do
    for search agent  $i$  do
        Update the position of the search agent  $i$  using Equation (35)
    end for
    Update  $a$ ,  $A$  and  $C$ 
    Estimate the fitness of all search agents
    Update  $X_\alpha$ ,  $X_\beta$  and  $X_\delta$ 
     $t = t + 1$ 
end while
return  $X_\alpha$ 

```

the weights and the biases have to be expressed as shown in Eq. (36).

$$\vec{V} = \{\vec{W}, \vec{\theta}\} = \{w_{1,1}, w_{1,2}, \dots, w_{n,n}, \theta_1, \theta_2, \dots, \theta_n\} \quad (36)$$

where n is the number of the input nodes, W_{ij} is the weight connecting the i^{th} node to the j^{th} node, θ_j is the bias (threshold) of the j^{th} hidden node.

Since the objective of training MLP NN with GWO is to produce the least possible error estimates, the average Mean Square Error (MSE_{avg}) (a widely used metric) is utilised in evaluating the performance of the trained MLP NN. The average MSE is expressed as shown in Eq. (37).

$$MSE_{avg} = \sum_{k=1}^s \left[\frac{\sum_{i=1}^m (Y_i^k - O_i^k)}{s} \right]^2 \quad (37)$$

where s is the number of training samples, m is the number of outputs. Y_i^k and O_i^k are the desired and the observed outputs respectively.

Hence, from the concept behind Eqs. (36, 37), the optimisation problem of training an MLP is formulated as shown in Eq. (38).

$$\min f(\vec{V}) = MSE_{avg} \quad (38)$$

The MLP training with GWO phase in Fig. 3 shows the process in which the weights and biases from GWO algorithm are fed to MLP for estimating the average MSE for

all the training samples. In order to minimise average MSE, the GWO algorithm is performed iteratively with varying weights and biases. Thus, the probability of obtaining improved MLP classification results after each iteration is high.

The total time complexity (Eq. (39)) of the proposed hybrid technique based on clustering, SMOTE and GWO-MLP is the summation of the individual time complexities from each modelling phase.

$$O(CLUST + GWO + MLP) = O(CLUST) + O(GWO) + O(MLP) \quad (39)$$

The complexity of each phase has been discussed in previous sections. The estimation of $O(CLUST)$ is dependent on the clustering algorithm under consideration. Literature has shown that, the complexity of hybridised techniques similar to the proposed will increase linearly as the number of samples increases whereas it will increase by n^2 when the number of features n is increased [57].

3.10 Classification performance evaluation

As a means to ensure the reliability of the proposed technique in classifying fault conditions, eight evaluation metrics were used to assess the performance of the various classifiers implemented in this study. The evaluation metrics were Accuracy, Error rate, Sensitivity, Specificity, Precision, F score, Mathews Correlation Coefficient (MCC), and the Geometric Mean.

Fig. 3 MLP training with GWO

3.10.1 Accuracy

Accuracy is a frequently used multiclass evaluation metrics for validating the performance of classifiers [106]. The metric is shown as Eq. (40).

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (40)$$

where TP is the counts of correct classification when there is a fault condition, TN is the counts of correct classification when there is no fault condition, FP is the number of misclassification counts when there is a fault condition and FN is the number of misclassification counts when there is no fault condition for a specific classification model.

3.10.2 Error rate

The error rate metric is also a predominantly used performance indicator for assessing the level of misclassifications sustained during a classification task [106]. The metric is given as Eq. (41).

$$\text{Error Rate} = \frac{FP + FN}{TP + FP + TN + FN} \quad (41)$$

3.10.3 Sensitivity

The ability of a classifier to determine positive instances is measured by the sensitivity metric [106] and is expressed as a fraction of positive instances that are correctly classified as shown in Eq. (42).

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (42)$$

3.10.4 Precision

The precision measures the exactness of a classifier to correctly classify targets after a given classification task [106]. The precision metric is shown as Eq. (43).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (43)$$

3.10.5 Specificity

The specificity measures the ability of a classifier to determine negative instances [107]. The metric is expressed as the fraction of negative instances that are correctly classified as shown in Eq. (44).

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (44)$$

3.10.6 Matthews correlation coefficient (MCC)

MCC is generally known to be a balanced metric for assessing classifiers with varying class sizes [108]. Compared to other evaluation metrics, MCC considers the balanced ratios from the four categories in the confusion matrix, thus, is considered more informative. MCC ranges from -1 to +1, where +1 denotes a perfect classification whilst -1 indicates a total misclassification. The metric is expressed as Eq. (45).

$$\text{MCC} = \frac{(TP * TN) - (FP * FN)}{\sqrt{[(TP + FP)(TP + FN)(TN + FP)] * (TN + FN)}} \quad (45)$$

3.10.7 F score

The F Score metric assesses the overall performance of a classifier as the harmonic mean of precision and sensitivity [107]. F Score ranges from zero to one, where high values indicate high classification performance and vice versa. The metric is given by Eq. (46).

$$FScore = \frac{2(Precision * Sensitivity)}{Precision + Sensitivity} \quad (46)$$

3.10.8 Geometric mean (GM)

GM is a classification metric that seeks to maximise the rate of true positives and negatives instances whilst maintaining a balance between both rates [109]. This makes the metric suitable for various classification tasks, even when the classes are imbalanced. The metric is expressed as an aggregate of specificity and sensitivity as shown in Eq. (47).

$$GM = \sqrt{\frac{TP}{TP + FN} * \frac{TN}{TN + FP}} \quad (47)$$

Table 3 Degree of class imbalance

Fault type	ID	LRID
Cooler condition	1.0031360	0.0000333
Internal pump leakage	1.2428725	0.2038595
Valve condition	2.2711664	0.3105193
Accumulator	1.1891476	0.0914137

Table 4 Optimal number of clusters

Fault type	No. of clusters	CHI	GI	DBI	SI	KLI
Accumulator	2	8793.3070	0.9565	0.4236	0.8085	18.1674
	3	6064.5490	0.5997	0.6252	0.7641	155.4982
	4	4172.6030	− 0.0549	1.1387	0.7386	0.1228
	5	3162.1250	− 0.4814	1.0273	0.7387	0.0274
Cooler	2	27,409.9300	0.7052	0.1834	0.8716	12.0375
	3	373,230.2600	3.3135	0.1410	0.9217	542.2288
	4	267,508.0100	2.7044	0.1195	0.9170	1.5424
	5	211,239.5700	2.3206	0.1100	0.9040	0.0992
Pump	2	1946.8760	1.4162	0.1834	0.9008	1.7146
	3	2768.6130	1.4177	0.6937	0.4694	68.5518
	4	1889.8650	0.7532	0.5235	0.4697	0.6328
	5	1498.0840	0.3581	0.5265	0.4392	0.0479
Valve	2	9497.7190	0.4429	0.5420	0.6642	8.0279
	3	7138.0680	0.3510	0.6455	0.5805	1.3985
	4	9574.2020	0.2992	0.6431	0.5393	4.9674
	5	8923.7690	0.0673	0.6837	0.5004	0.6246

4 Results and discussion

4.1 Class imbalance results

Table 3 shows the degree of class imbalance of the four major fault conditions of the hydraulic system dataset utilised in this study. As observed in Table 3, all the fault data-set types were imbalanced to a certain degree since all the estimated values for *ID* (Eq. (16)) and *LRID* (Eq. (20)) were greater than 1 and 0. That is, *ID* = 1 and *LRID* = 0 implies a balanced dataset whilst *ID* > 1 and *LRID* > 0 implies imbalanced dataset. However, among the fault types, the valve condition followed by the internal pump leakage exhibited a wider degree of class imbalance as depicted by the larger *ID* and *LRID* estimates.

4.2 Optimal cluster determination results

Table 4 shows the summary of the five validation measures for ensuring the goodness of clustering results based on the 20 features selected (refer to Table 2) for each of the fault types. The decision on the optimal number of clusters is based on the simple majority as suggested by the results produced by the various indices (bolded). As observed in Table 4, the majority of the indexes (Calinski-Harabasz Index (CHI), Gap Index (GI), Silhouette Index (SI), Krzanowski and Lai Index (KLI)) suggest that the cooler features are well partitioned into three clusters. Similarly, indexes CHI, GI and KLI recommended three clusters for the pump features. In contrast, four indexes (GI, DBI (Davies-Bouldin Index), SI and KLI) suggest that the valve

features be grouped into two clusters. The accumulator features were also grouped in two clusters based on the majority recommendation from CHI, GI, DBI and SI.

Based on the validation results (Table 4) from these indices, the optimal number of clusters initialised or extracted from each of the unsupervised clustering algorithms (K-Means, SOM, GMM and Hierarchical) are as follows: Cooler (3), Pump (3), Valve (2) and finally, Accumulator (2). However, before these clusters are fed to the MLP trained with GWO, each cluster was divided into training and testing dataset with a percentage ratio of 70:30. The training dataset was used for the model development whilst the testing dataset was reserved for final model validation. Due to the unbalanced nature of the training dataset, SMOTE was applied as a resampling technique to balance the training dataset. Using the balanced training dataset as input, MLP neural network classifier trained with GWO were then developed for classifying the various fault conditions.

4.3 GWO-MLP classification results

The stochastic nature of GWO in providing weights and biases for MLP will yield slightly different output after each run. As a result, GWO-MLP classifiers were trained 10 times for each neuron in the hidden layer once the architecture for the MLP is defined. The final output is presented as an average of all the outputs from the 10 runs. This training procedure conforms to what has been widely utilised in Mirjalili [30] and thus was implemented in this study. However, the optimal number of neurons in the hidden layer for each MLP was determined using the well-known sequential trial-and-error (1 to 50 neurons) procedure. The basic parameters and values for the GWO that produced the best outputs are shown in Table 5. The classification experiment was simulated on MATLAB (2019a version) using an ASUS computer with a 2.3 GHz processor computer with 4 GB memory.

4.3.1 Effects of class imbalance on CLUST-SMOTE-GWO-MLP fault classification

In order to understand the effects of class imbalance on the proposed hybrid technique of clustering (CLUST), SMOTE and MLP optimised with grey wolf algorithm (CLUST-SMOTE-GWO-MLP) in classifying the various fault conditions, the classifications were done using two distinct forms of training datasets. That is, the unbalanced training set produced by the various clustering methods (CLUST-GWO-MLP) and the balanced training set using SMOTE (CLUST-SMOTE-GWO-MLP). Tables 6, 7, 8, 9 show the comparative training and testing performance of CLUST-GWO-MLP and CLUST-SMOTE-GWO-MLP in classifying

Table 5 Basic parameters and values of GWO

Parameter	Value
Number of search agents (wolves)	200
Maximum Iterations	500
Lower Bound	− 10
Upper Bound	10

the various fault types based on the clusters derived from the K-Means, SOM, GMM and HAC clustering algorithms respectively. The LRID reports on the degree of class imbalance of the training sets of each cluster with unbalanced classes (CLUST-GWO-MLP). The optimal number of neurons considered to achieve the highest average test accuracy as well as its corresponding average train MSE after the 10 runs are also reported for each cluster.

Focusing on the classification based on K-Means clustering, it was evident in Table 6 that training the classifier with balanced input data using the SMOTE (CLUST_(KMEANS)-SMOTE-GWO-MLP) yielded better test classification results, especially, for the pump and accumulator fault conditions. As observed, the test accuracy of pump conditions significantly increased from 95.56% to 98.21%. A similar increment was observed for the accumulator conditions where the test accuracy increased from 87.82% to 88.27%. In the case of the cooler and valve conditions, an aggregate performance of 100% test accuracy was obtained for both balanced (CLUST_(KMEANS)-SMOTE-GWO-MLP) and unbalanced training (CLUST_(KMEANS)-GWO-MLP). This phenomenon could be attributed to the feature selection technique (correlation-based) used since the selected features were generally correlated with the class outputs of cooler and valve conditions (refer to Table 2). This resulted in less complexity during the learning phase and subsequently allowed for easier classification during the testing phase. However, a critical inspection of the training MSEs (aggregate) for cooler and valve conditions showed a lower average train MSE of 1.03E-09 and 1.67E-20 respectively when SMOTE is applied and thus, have greater chances of better performance.

With regards to the classification based on clusters from SOM (Table 7), the clusters with balanced classes (CLUST_(SOM)-SMOTE-GWO-MLP) showed improvement for all the fault conditions except the valve conditions. That is, the CLUST_(SOM)-SMOTE-GWO-MLP fault classification performance of clusters (aggregate) from the cooler, pump and accumulator conditions yielded better test accuracy results of 100%, 98.34% and 91.65%. On the contrary, identical testing accuracy of 100% was achieved for both the balanced (CLUST_(SOM)-SMOTE-GWO-MLP) and unbalanced (CLUST_(SOM)-GWO-MLP) datasets for the valve condition.

Table 6 Fault classification based on $CLUST_{(KMEANS)}$ -GWO-MLP and $CLUST_{(KMEANS)}$ -SMOTE-GWO-MLP

Fault	Cluster	$CLUST_{(KMEANS)}$ -GWO-MLP (Unbalanced)				$CLUST_{(KMEANS)}$ -SMOTE-GWO-MLP (Balanced)		
		LRID	Neuron	Train MSE	Test accuracy	Neuron	Train MSE	Test accuracy
Accumulator	1	0.1308	8	1.26E-01	83.30	11	7.47E-02	84.05
	2	1.7432	9	3.31E-02	92.33	10	3.41E-04	92.50
	Aggregate			7.97E-02	87.82		3.75E-02	88.27
Cooler	1	1.8720	39	3.75E-14	100.00	49	1.92E-14	100.00
	2	1.8234	35	7.03E-17	100.00	45	1.43E-15	100.00
	3	1.7695	20	5.32E-08	100.00	38	3.08E-09	100.00
	Aggregate			1.77E-08	100.00		1.03E-09	100.00
Pump	1	0.7737	7	2.51E-02	96.20	11	2.31E-02	97.52
	2	1.9011	20	2.49E-15	99.18	48	9.18E-27	99.01
	3	0.0862	8	4.48E-02	91.32	15	4.43E-03	98.09
	Aggregate			2.33E-02	95.56		9.19E-03	98.21
Valve	1	0.2947	48	2.22E-47	100.00	42	2.55E-50	100.00
	2	0.5128	36	5.10E-14	100.00	42	3.34E-20	100.00
	Aggregate			2.55E-14	100.00		1.67E-20	100.00

Table 7 Fault classification based on $CLUST_{(SOM)}$ -GWO-MLP and $CLUST_{(SOM)}$ -SMOTE-GWO-MLP

Fault	Cluster	$CLUST_{(SOM)}$ -GWO-MLP (Unbalanced)				$CLUST_{(SOM)}$ -SMOTE-GWO-MLP (Balanced)		
		LRID	Neuron	Train MSE	Test accuracy	Neuron	Train MSE	Test accuracy
Accumulator	1	1.0614	2	4.08E-10	100.00	2	1.65E-12	100.00
	2	0.1391	7	1.14E-01	82.87	11	8.71E-02	83.30
	Aggregate			5.72E-02	91.43		4.35E-02	91.65
Cooler	1	1.8380	25	3.78E-04	99.74	41	1.89E-08	100.00
	2	1.8131	48	6.10E-18	100.00	40	3.39E-18	100.00
	3	1.7958	44	1.23E-20	100.00	42	2.13E-20	100.00
	Aggregate			1.26E-04	99.91		6.30E-09	100.00
Pump	1	0.4435	15	2.52E-02	95.56	4	6.99E-03	98.61
	2	0.0039	25	3.40E-03	91.75	39	2.85E-03	96.40
	3	1.2694	42	4.78E-57	100.00	45	2.22E-57	100.00
	Aggregate			9.52E-03	95.77		3.28E-03	98.34
Valve	1	0.0006	49	3.42E-99	100.00	47	1.37E-103	100.00
	2	0.2466	41	1.75E-57	100.00	44	9.19E-56	100.00
	Aggregate			8.73E-58	100.00		4.59E-56	100.00

The classification performance based on clusters from the GMM clustering algorithm is shown in Table 8. As observed in Table 8, all the fault conditions showed improvement in classification performance when trained with clusters with balanced classes ($CLUST_{(GMM)}$ -SMOTE-GWO-MLP). This clearly shows the significant role in balancing training dataset with SMOTE before learning, plays in enhancing the performance of classifiers. Similar to the results obtained with $CLUST_{(SOM)}$ -SMOTE-GWO-MLP in Table 7, the classification based on clusters derived from HAC ($CLUST_{(HAC)}$ -SMOTE-GWO-MLP) in Table 9 showed

improved classification performance in cooler, pump and accumulator conditions when trained with clusters with balanced classes. Competitive aggregate performance of 100% test accuracy was obtained for both balanced ($CLUST_{(HAC)}$ -SMOTE-GWO-MLP) and unbalanced ($CLUST_{(HAC)}$ -GWO-MLP) data for the valve conditions.

A closer look at the training and testing performance for both cooler and valve conditions (Tables 6, 7, 8, 9) show an interesting scenario where the classification accuracies during testing are higher than the training. That is, while some instances of misclassification were recorded during training (i.e. $MSE_s \neq 0$, implying $< 100\%$

Table 8 Fault classification based on CLUST_(GMM)-GWO-MLP and CLUST_(GMM)-SMOTE-GWO-MLP

Fault	Cluster	CLUST _(GMM) -GWO-MLP (Unbalanced)				CLUST _(GMM) -SMOTE-GWO-MLP (Balanced)		
		LRID	Neuron	Train MSE	Test accuracy	Neuron	Train MSE	Test accuracy
Accumulator	1	0.2465	7	1.12E-01	86.64	5	1.26E-01	85.71
	2	1.5756	26	4.24E-02	97.20	8	3.05E-02	98.40
	Aggregate			7.73E-02	91.92		7.81E-02	92.06
Cooler	1	1.8308	41	6.93E-18	100.00	46	5.41E-09	100.00
	2	1.9002	33	8.48E-20	100.00	45	7.43E-20	100.00
	3	1.838	32	3.78E-03	99.21	12	2.19E-05	99.82
	Aggregate			1.26E-03	99.74		7.31E-06	99.94
Pump	1	0.4088	8	3.62E-03	97.16	2	5.07E-03	99.09
	2	0.0627	4	7.15E-04	98.04	3	2.60E-03	99.30
	3	0.0067	15	1.12E-06	80.00	3	9.41E-05	87.14
	Aggregate			1.44E-03	91.73		2.59E-03	95.18
Valve	1	0.3052	49	1.95E-54	100.00	46	8.77E-56	100.00
	2	0.5411	15	1.17E-11	99.91	43	6.77E-20	100.00
	Aggregate			5.84E-12	99.96		3.39E-20	100.00

Table 9 Fault classification based on CLUST_(HAC)-GWO-MLP and CLUST_(HAC)-SMOTE-GWO-MLP

Fault	Cluster	CLUST _(HAC) -GWO-MLP (Unbalanced)				CLUST _(HAC) -SMOTE-GWO-MLP (Balanced)		
		LRID	Neuron	Train MSE	Test accuracy	Neuron	Train MSE	Test accuracy
Accumulator	1	0.1198	9	1.52E-01	84.19	6	1.27E-01	85.02
	2	1.6233	14	4.29E-02	92.05	6	3.85E-10	92.42
	Aggregate			9.77E-02	88.12		6.37E-02	88.72
Cooler	1	1.8380	25	7.57E-04	99.74	43	1.17E-08	100.00
	2	1.7958	43	1.10E-22	100.00	49	1.03E-21	100.00
	3	1.8131	35	8.61E-21	100.00	45	1.13E-19	100.00
	Aggregate			2.52E-04	99.91		3.89E-09	100.00
Pump	1	1.2585	47	2.43E-54	100.00	41	1.50E-50	100.00
	2	0.0023	39	1.24E-12	99.74	34	1.17E-10	100.00
	3	0.4531	21	5.34E-02	78.92	3	1.86E-02	94.32
	Aggregate			1.78E-02	92.88		6.19E-03	98.11
Valve	1	0.2466	49	1.76E-61	100.00	48	8.86E-61	100.00
	2	0.0006	47	3.84E-101	100.00	49	2.05E-104	100.00
	Aggregate			8.78E-62	100.00		4.43E-61	100.00

training accuracy), a 100% accuracy (with no misclassification) was achieved during testing. Several studies have indicated that the usual case where higher accuracy is expected during training data may not always be true for all cases since the optimisation technique used for fine-tuning the hyperparameters of classifiers tries to enhance the generalisability of the model with less regard to the training performance [110–112]. Other studies have linked this unusual outcome to the impact of feature selection technique in enhancing the generalisability of classifiers [113–115] while similar studies have attributed the phenomena to inadequate training

and test data samples [110]. However, relating this assertion to the results obtained from classifying cooler and valve conditions (Tables 6, 7, 8, 9), it can be seen that the unbalanced dataset is least regarded by the optimisers than the balanced dataset during training. This was manifested by the lower MSEs values obtained using the balanced dataset for the classification of cooler and valve conditions.

A visual representation of the relevance of balancing the clustering results with SMOTE before using it as input into the GWO-MLP classifier is presented in Fig. 4.

4.3.2 Effects of hybridising clustering and GWO-MLP

This section demonstrates and compares the relevance of systematically combining unsupervised clustering and supervised learning in fault classification. The comparison was done by applying the GWO-MLP classifier on both the original dataset (without clustering) and the balanced clustered dataset. The performance of GWO-MLP on these distinct datasets was assessed using the eight performance indicators of classifiers (Eqs. (40–47)).

Table 10 shows the performance of the proposed CLUST-SMOTE-GWO-MLP technique in classifying the accumulator condition. As observed, the CLUST-SMOTE-GWO-MLP showed better accuracy and generalisability when compared to the classification results from the original dataset (both balanced and unbalanced versions). That is, the techniques $\text{CLUST}_{(\text{KMEANS})}$ -SMOTE-GWO-MLP

(C_{KSGM}), $\text{CLUST}_{(\text{SOM})}$ -SMOTE-GWO-MLP (C_{SSGM}), $\text{CLUST}_{(\text{GMM})}$ -SMOTE-GWO-MLP (C_{GSGM}) and $\text{CLUST}_{(\text{HAC})}$ -SMOTE-GWO-MLP (C_{HSGM}) classified all the test instances with the accuracy of 88.27%, 91.65%, 92.06% and 88.72% respectively. These results demonstrate the superiority of the proposed CLUST-SMOTE-GWO-MLP in classifying accumulator conditions since lesser accuracies (87.13% and 86.44%) was obtained when both balanced and unbalanced versions of the original dataset were used as inputs for training GWO-MLP. The remaining seven classification metrics confirmed this assertion. However, an assessment of the variants of clusters of the proposed technique revealed an accuracy of 92.06% for C_{GSGM} which was 4.93% higher than the best from the original dataset. Hence, C_{GSGM} is more adequate for classifying the accumulator condition in the hydraulic system.

Fig. 4 Overall test accuracy of CLUST-GWO-MLP and CLUST-SMOTE-GWO-MLP in fault classification

Table 10 Classification performance of GWO-MLP, SMOTE-GWO-MLP and CLUST-SMOTE-GWO-MLP on accumulator condition

Metrics	Original dataset		CLUST-SMOTE-GWO-MLP (Balanced clustered dataset)			
	GWO-MLP (Unbalanced)	SMOTE-GWO-MLP (Balanced)	C_{KSGM}	C_{SSGM}	C_{GSGM}	C_{HSGM}
Accuracy	0.8644	0.8713	0.8827	0.9165	0.9206	0.8872
Error	0.1356	0.1287	0.1173	0.0835	0.0794	0.1128
Sensitivity	0.8459	0.8586	0.7299	0.8989	0.8939	0.8026
Specificity	0.9538	0.9571	0.9254	0.9719	0.9722	0.9361
Precision	0.8557	0.8590	0.7178	0.9006	0.8840	0.8236
F Score	0.8500	0.8582	0.7310	0.8994	0.8854	0.7952
MCC	0.8051	0.8157	0.6561	0.8718	0.8566	0.7508
GM	0.8162	0.8221	0.6642	0.8753	0.8595	0.7709

Better accuracy and generalisation performance were achieved when the proposed CLUST-SMOTE-GWO-MLP for cooler classification result was compared to standalone GWO-MLP classification based on the original dataset (Table 11). While the techniques C_K SGM, C_S SGM and C_H SGM classified all the test instances correctly (100%), C_G SGM produced a competitive result of 99.94%. In general, the proposed hybrid CLUST-SMOTE-GWO-MLP technique, irrespective of the clustering algorithm utilised, proved superior in classifying cooler condition when compared with the standalone (99.82%), even when SMOTE is applied on the original dataset (99.88%). This implies that the systematic combination of the clustering algorithms and MLP optimised with GWO (C_K SGM, C_S SGM, C_H SGM and C_G SGM) is efficient for classifying cooler conditions. This is also supported by the satisfactory results deduced from the remaining performance indicators (Table 11).

Similar to the case of accumulator condition (Table 10), the performance of the CLUST-SMOTE-GWO-MLP in classifying the condition of internal pump leakage was highly improved when comparing the various performance metrics shown in Table 12. This implies that the chances of classifying pump condition increased significantly when

clustering algorithms are combined with GWO-MLP (C_K SGM, C_S SGM, C_G SGM and C_H SGM). However, among the variants of clusters of the proposed technique, C_S SGM is more suitable for classifying the internal pump leakage in the hydraulic system. This is because even the best classification accuracy of the original data set was improved by 3.54%.

In the case of classifying the valve condition, a significant increase of more than 7% accuracy was achieved when the CLUST-SMOTE-GWO-MLP was implemented (Table 13). Interestingly Fig. 5 shows that, generally, any of the clustering algorithms applied in this study will suffice in producing quality clusters for the proposed hybrid technique when classifying cooler and valve conditions. However, SOM and GMM clustering algorithms produced better clusters for the proposed technique in the case of internal pump leakage and accumulator conditions respectively.

A careful study of Tables 10, 11, 12, 13 portray the superiority of the proposed CLUST-SMOTE-GWO-MLP over the other investigated methods in multiclass and imbalanced fault classification. The reason for the high accuracy could be attributed to the systematic clustering of the features

Table 11 Classification performance of GWO-MLP, SMOTE-GWO-MLP and CLUST-SMOTE-GWO-MLP on cooler condition

Metrics	Original dataset		CLUST-SMOTE-GWO-MLP (Balanced clustered dataset)			
	GWO-MLP (Unbalanced)	SMOTE-GWO-MLP (Balanced)	C_K SGM	C_S SGM	C_G SGM	C_H SGM
Accuracy	0.9982	0.9988	1.0000	1.0000	0.9994	1.0000
Error	0.0018	0.0012	0.0000	0.0000	0.0006	0.0000
Sensitivity	0.9556	0.9988	1.0000	1.0000	0.9994	1.0000
Specificity	0.9778	0.9994	1.0000	1.0000	0.9997	1.0000
Precision	0.9994	0.9988	1.0000	1.0000	0.9994	1.0000
F Score	0.9664	0.9988	1.0000	1.0000	0.9994	1.0000
MCC	0.9592	0.9982	1.0000	1.0000	0.9991	1.0000
GM	0.9772	0.9982	1.0000	1.0000	0.9991	1.0000

Table 12 Classification performance of GWO-MLP, SMOTE-GWO-MLP and CLUST-SMOTE-GWO-MLP on pump condition

Metrics	Original Dataset		CLUST-SMOTE-GWO-MLP (Balanced clustered dataset)			
	GWO-MLP (Unbalanced)	SMOTE-GWO-MLP (Balanced)	C_K SGM	C_S SGM	C_G SGM	C_H SGM
Accuracy	0.9323	0.9480	0.9821	0.9834	0.9518	0.9811
Error	0.0677	0.0520	0.0179	0.0166	0.0482	0.0189
Sensitivity	0.8936	0.9244	0.9181	0.9962	0.9540	0.9854
Specificity	0.9692	0.9746	0.9881	0.9743	0.9742	0.9880
Precision	0.8729	0.9402	0.9175	0.9745	0.9620	0.9887
F Score	0.9553	0.9276	0.9252	0.9845	0.9518	0.9857
MCC	0.8629	0.9070	0.9013	0.9713	0.9333	0.9762
GM	0.8460	0.9164	0.9066	0.9494	0.9371	0.9769

Table 13 Classification performance of GWO-MLP, SMOTE-GWO-MLP and CLUST-SMOTE-GWO-MLP on valve condition

Metrics	Original Dataset		CLUST-SMOTE-GWO-MLP (Balanced clustered dataset)			
	GWO-MLP (Unbalanced)	SMOTE-GWO-MLP (Balanced)	C _k SGM	C _s SGM	C _G SGM	C _H SGM
Accuracy	0.8964	0.9269	1.0000	1.0000	1.0000	1.0000
Error	0.1036	0.0731	0.0000	0.0000	0.0000	0.0000
Sensitivity	0.8513	0.8996	1.0000	1.0000	1.0000	1.0000
Specificity	0.9690	0.9717	1.0000	1.0000	1.0000	1.0000
Precision	0.9231	0.9579	1.0000	1.0000	1.0000	1.0000
F Score	0.8007	0.8739	1.0000	1.0000	1.0000	1.0000
MCC	0.8060	0.8712	1.0000	1.0000	1.0000	1.0000
GM	0.8945	0.9308	1.0000	1.0000	1.0000	1.0000

Fig. 5 Contribution of clustering algorithms to CLUST-SMOTE-GWO-MLP fault classification

into a homogeneous space, followed by the resampling strategy and finally the GWO-MLP classifier with adaptive parameters providing an ambient mechanism for exploring and exploiting the search space. That is, prior to the optimisation phase, the implementation of clustering reduced the complexity in exploration due to the transformation from a heterogeneous search space into a homogeneous one, thus, leading to a reduction in runtime. Moreover, the proposed technique's ability to self-balance the class outputs before training makes it robust as it reduces the level of biases introduced by the unbalanced class outputs during training. As seen in Tables 6, 7, 8, 9, the balancing strategy increased the performance accuracy of the proposed technique.

Although the proposed CLUST-SMOTE-GWO-MLP technique produces satisfactory results, the single-objective nature and the low capability of GWO to handle the complexities in a multi-modal search landscape may limit the search capabilities of the proposed technique. This might

result in a relatively long runtime during the optimisation phase. For these reasons, future works should explore the multi-objective and multi-modal alternatives for improving the exploration of GWO.

5 Conclusions

In this study, a hybrid CLUST-SMOTE-GWO-MLP technique has been proposed as a novelty for performing multiclass and imbalance classification of different fault conditions. The superiority of the proposed CLUST-SMOTE-GWO-MLP was demonstrated by comparing to its distinct modifications (variants). It was observed that the proposed hybrid technique outperformed all the compared modification in terms of test accuracy and seven other evaluation metrics considered. Interestingly, the choice of clustering algorithm depended on the specific classification task at hand. In that, any of the clustering algorithms will suffice

in producing quality clusters for the proposed hybrid technique when classifying cooler and valve conditions whilst neural network-based SOM and distribution-based GMM clustering algorithms produced optimal clusters in the case of internal pump leakage and accumulator conditions respectively. Also, it was observed that irrespective of the clustering technique, balancing the class outputs of the training dataset with SMOTE before training the classifier enhanced the classification results, especially for the pump and accumulator fault conditions. The cooler and valve conditions, on the other hand, did improve for some specific cases after balancing, however, an aggregate test performance of 100% accuracy was generally obtained from both balanced and unbalanced training of classifiers. This phenomenon was attributed to the correlation-based feature selection technique used since all the selected features were generally correlated with the class outputs of cooler and valve conditions. Although the proposed technique improves the classification of fault conditions, it may be limited by the single-objective nature of the applied GWO algorithm as well as its low capability to handle the complexities in a multi-modal search landscape. Hence, future works should explore GWO with multi-objective and multi-modal capabilities. This will ultimately improve the performance of the MLP classifier of fault conditions as well as making it versatile to handle both single or multi-modal engineering problems.

Declarations

Conflict of interest The authors declare that they have no competing interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Liu Y, Ge Z (2018) Weighted random forests for fault classification in industrial processes with hierarchical clustering model selection. *J Process Control* 64:62–70. <https://doi.org/10.1016/j.procont.2018.02.005>
- Di Z, Kang Q, Peng D, Zhou M (2019) Density Peak-based Pre-clustering Support Vector Machine for Multi-class Imbalanced Classification. In: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC). pp 27–32
- Chen G, Liu Y, Ge Z (2019) K-means Bayes algorithm for imbalanced fault classification and big data application. *J Process Control* 81:54–64
- Chen G, Ge Z (2019) SVM-tree and SVM-forest algorithms for imbalanced fault classification in industrial processes. *IFAC J Syst Control* 8:100052
- Chakraborty T (2017) EC3: Combining Clustering and Classification for Ensemble Learning. In: 2017 IEEE International Conference on Data Mining (ICDM). pp 781–786
- Maheshwari A, Agarwal V, Sharma SK (2018) Transmission line fault classification using artificial neural network based fault classifier. *Int J Electr Eng Technol (IJEET)* 9:170–181
- Resmi R, Vanitha V, Aravind E, Haritha S (2019) Detection, Classification and Zone Location of Fault in Transmission Line using Artificial Neural Network. In: 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICEECT). IEEE, pp 1–5
- Kaur H, Kaur M (2020) Fault Classification in a Transmission Line Using Levenberg–Marquardt Algorithm Based Artificial Neural Network. In: *Data Communication and Networks*. Springer, pp 119–135
- Mahmud MN, Ibrahim MN, Osman MK, Hussain Z (2017) Support vector machine (SVM) for fault classification in radial distribution network. *Adv Sci Lett* 23:4124–4128
- Gao X, Wei H, Li T, Yang G (2020) A rolling bearing fault diagnosis method based on LSSVM. *Adv Mech Eng* 12:1687814019899561
- Helwig N, Pignatelli E, Schütze A (2015) Condition monitoring of a complex hydraulic system using multivariate statistics. In: *Instrumentation and Measurement Technology Conference (I2MTC), 2015 IEEE International*. IEEE, pp 210–215
- Schneider T, Helwig N, Schütze A (2018) Automatic feature extraction and selection for condition monitoring and related datasets. In: 2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC). IEEE, pp 1–6
- Xiao L, Sun H, Zhang L, Niu F, Yu L, Ren X (2019) Applications of a strong track filter and LDA for on-line identification of a switched reluctance machine stator inter-turn shorted-circuit fault. *Energies* 12:134
- Aker E, Othman ML, Veerasamy V, Aris I, bin, Wahab NIA, Hizam H, (2020) Fault detection and classification of shunt compensated transmission line using discrete wavelet transform and naive bayes classifier. *Energies* 13:243
- Wu Y, Fu Z, Fei J (2020) Fault diagnosis for industrial robots based on a combined approach of manifold learning, treelet transform and Naive Bayes. *Rev Sci Instrum* 91:015116
- Zhang X-Y, Yang P, Zhang Y-M, Huang K, Liu C-L (2014) Combination of classification and clustering results with label propagation. *IEEE Signal Process Lett* 21:610–614
- Amruthnath N, Gupta T (2018) A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. In: 2018 5th International Conference on Industrial Engineering and Applications (ICIEA). IEEE, pp 355–361
- Park S, Park S, Kim M, Hwang E (2020) Clustering-based self-imputation of unlabeled fault data in a fleet of photovoltaic generation systems. *Energies* 13:737
- Zhu J, Ge Z, Song Z (2018) Distributed Gaussian mixture model for monitoring plant-wide processes with multiple operating modes. *IFAC J Syst Control* 6:1–15
- Hong Y, Kim M, Lee H, Park JJ, Lee D (2019) Early fault diagnosis and classification of ball bearing using enhanced Kurtogram

- and Gaussian mixture model. *IEEE Trans Instrum Meas* 68:4746–4755
21. Lu Y, Tian Z, Peng P, Niu J, Li W, Zhang H (2019) GMM clustering for heating load patterns in-depth identification and prediction model accuracy improvement of district heating system. *Energy Buildings* 190:49–60. <https://doi.org/10.1016/j.enbuild.2019.02.014>
 22. Blue J, Roussy A, Thieullen A, Pinaton J (2012) Efficient FDC based on hierarchical tool condition monitoring scheme. In: 2012 SEMI Advanced Semiconductor Manufacturing Conference. IEEE, pp 359–364
 23. Barmada S, Romano F, Tucci M (2014) Hierarchical clustering applied to measured data relative to pantograph-catenary systems as a predictive maintenance tool
 24. Raptodimos Y, Lazakis I (2018) Using artificial neural network-self-organising map for data clustering of marine engine condition monitoring applications. *Ships Offshore Struct* 13:649–656. <https://doi.org/10.1080/17445302.2018.1443694>
 25. Skowron M, Wolkiewicz M, Orlowska-Kowalska T, Kowalski CT (2019) Application of self-organizing neural networks to electrical fault classification in induction motors. *Appl Sci* 9:616
 26. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. *Jair* 16:321–357. <https://doi.org/10.1613/jair.953>
 27. Adel A, Fawzi G (2018) Gear multi-fault feature extraction and classification based on fuzzy entropy of local mean decomposition, singular value decomposition and MLP neural network. In: International Conference on Advanced Mechanics and Renewable Energies ICAMRE2018
 28. Aljohani A, Aljurbua A, Shafullah M, Abido MA (2018) Smart Fault Detection and Classification for Distribution Grid Hybridizing ST and MLP-NN. In: 2018 15th International Multi-Conference on Systems, Signals & Devices (SSD). IEEE, pp 94–98
 29. Wilson K, Wang J (2019) Optimized artificial neural network method for underground cables fault classification. *Annal Electric Electron Eng* 2:18–24
 30. Mirjalili S (2015) How effective is the Grey Wolf optimizer in training multi-layer perceptrons. *Appl Intell* 43:150–161
 31. Ali MA, Bingamil AA, Jarndal A, Alsyoud I (2019) The Influence of Handling Imbalance Classes on the Classification of Mechanical Faults Using Neural Networks. In: 2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO). pp 1–5
 32. Lemnaru C, Potolea R (2011) Imbalanced classification problems: systematic study, issues and best practices. In: International Conference on Enterprise Information Systems. Springer, pp 35–50
 33. Hertz JA, Krogh AS, Palmer RG (1991) Introduction To The Theory Of Neural Computation, 1st edn. Westview Press, Redwood City, Calif
 34. Whitley D (1994) A genetic algorithm tutorial. *Stat Comput* 4:65–85. <https://doi.org/10.1007/BF00175354>
 35. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks. pp 1942–1948 vol.4
 36. Ilonen J, Kamarainen J-K, Lampinen J (2003) Differential evolution training algorithm for feed-forward neural networks. *Neural Process Lett* 17:93–105. <https://doi.org/10.1023/A:1022995128597>
 37. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. *IEEE Comput Intell Mag* 1:28–39. <https://doi.org/10.1109/MCI.2006.329691>
 38. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. *Inf Sci* 179:2232–2248
 39. Karaboga D, Gorkemli B, Ozturk C, Karaboga N (2014) A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artif Intell Rev* 42:21–57. <https://doi.org/10.1007/s10462-012-9328-0>
 40. Mosavi MR, Khishe M, Ghamgosar A (2016) Classification of sonar data set using neural network trained by Gray Wolf Optimization. *Neural Netw World* 26:393
 41. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
 42. Mirjalili S, Hashim SZM (2010) A new hybrid PSO-GSA algorithm for function optimization. In: 2010 International Conference on Computer and Information Application. pp 374–377
 43. Mosavi MR, Khishe M (2017) Training a feed-forward neural network using particle swarm optimizer with autonomous groups for sonar target classification. *J Circuit Syst Comp* 26:1750185. <https://doi.org/10.1142/S0218126617501857>
 44. Branch M (2012) A multi-layer perceptron neural network trained by invasive weed optimization for potato color image segmentation. *Trends Appl Sci Res* 7:445–455
 45. Yu JJQ, Lam AYS, Li VOK (2011) Evolutionary artificial neural network based on Chemical Reaction Optimization. In: 2011 IEEE Congress of Evolutionary Computation (CEC). pp 2083–2090
 46. Salimi H (2015) Stochastic fractal search: a powerful metaheuristic algorithm. *Knowl-Based Syst* 75:1–18
 47. Mosavi MR, Khishe M, Akbarisani M (2017) Neural network trained by biogeography-based optimizer with chaos for sonar data set classification. *Wireless Pers Commun* 95:4623–4642. <https://doi.org/10.1007/s11277-017-4110-x>
 48. Mosavi MR, Khishe M, Naseri MJ, Parvizi GR, Mehdi A (2019) Multi-layer perceptron neural network utilizing adaptive best-mass gravitational search algorithm to classify sonar dataset. *Arch Acoust* 44:137–151
 49. Khishe M, Mosavi MR (2020) Classification of underwater acoustical dataset using neural network trained by chimp optimization algorithm. *Appl Acoust* 157:107005. <https://doi.org/10.1016/j.apacoust.2019.107005>
 50. Khishe M, Safari A (2019) Classification of sonar targets using an MLP neural network trained by dragonfly algorithm. *Wireless Pers Commun* 108:2241–2260. <https://doi.org/10.1007/s11277-019-06520-w>
 51. Khishe M, Mohammadi H (2019) Passive sonar target classification using multi-layer perceptron trained by salp swarm algorithm. *Ocean Eng* 181:98–108. <https://doi.org/10.1016/j.oceaneng.2019.04.013>
 52. Gupta S, Deep K, Mirjalili S (2020) An efficient equilibrium optimizer with mutation strategy for numerical optimization. *Appl Soft Comput* 96:106542
 53. Gupta S, Deep K (2020) A novel hybrid sine cosine algorithm for global optimization and its application to train multilayer perceptrons. *Appl Intell* 50:993–1026. <https://doi.org/10.1007/s10489-019-01570-w>
 54. Gupta S, Deep K, Mirjalili S, Kim JH (2020) A modified sine cosine algorithm with novel transition parameter and mutation operator for global optimization. *Expert Syst Appl* 154:113395
 55. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
 56. Khishe M, Mosavi MR (2019) Improved whale trainer for sonar datasets classification using neural network. *Appl Acoust* 154:176–192. <https://doi.org/10.1016/j.apacoust.2019.05.006>
 57. Qiao W, Khishe M, Ravakhah S (2021) Underwater targets classification using local wavelet acoustic pattern and Multi-Layer Perceptron neural network optimized by modified Whale Optimization Algorithm. *Ocean Eng* 219:108415. <https://doi.org/10.1016/j.oceaneng.2020.108415>

58. Ho Y-C, Pepyne DL (2002) Simple explanation of the no-free-lunch theorem and its implications. *J Optim Theory Appl* 115:549–570
59. Prajindra SK, Kiong TS, Siaw Paw JK (2017) Dynamic social behavior algorithm for real-parameter optimization problems and optimization of hyper beamforming of linear antenna arrays. *Eng Appl Artif Intell* 64:401–414. <https://doi.org/10.1016/j.engappai.2017.06.027>
60. Faris H, Aljarah I, Al-Betar MA, Mirjalili S (2018) Grey wolf optimizer: a review of recent variants and applications. *Neural Comput & Applic* 30:413–435. <https://doi.org/10.1007/s00521-017-3272-5>
61. Huang H, Fan Q, Wei J, Huang D (2019) An intelligent fault identification method of rolling bearings based on SVM optimized by improved GWO. *Syst Sci Control Eng* 7:289–303
62. Liwei XIE, Yong LI, Longfu LUO, Yijia CAO, Wei HU, ZHANG Y, Xiaohui S (2019) An Identification Method of Fault Type Based on GWO-SVM for Distribution Network. In: 2019 IEEE Sustainable Power and Energy Conference (iSPEC). IEEE, pp 1970–1974
63. Helwig N, Pignaneli E, Schütze A (2015) Detecting and Compensating Sensor Faults in a Hydraulic Condition Monitoring System. In: In proceeding Sensor. SENSOR, pp 641–646
64. Schneider T, Helwig N, Schütze A (2017) Automatic feature extraction and selection for classification of cyclical time series data. *tm-Technisches Messen* 84:198–206
65. Binsaeid S, Asfour S, Cho S, Onar A (2009) Machine ensemble approach for simultaneous detection of transient and gradual abnormalities in end milling using multisensor fusion. *J Mater Process Technol* 209:4728–4738. <https://doi.org/10.1016/j.jmatprotec.2008.11.038>
66. Lee J, Lapira E, Bagheri B, Kao H (2013) Recent advances and trends in predictive manufacturing systems in big data environment. *Manufact Lett* 1:38–41
67. Schmidt B, Wang L (2018) Cloud-enhanced predictive maintenance. *Int J Adv Manuf Technol* 99:5–13. <https://doi.org/10.1007/s00170-016-8983-8>
68. Rodriguez MZ, Comin CH, Casanova D, Bruno OM, Amancio DR, da Costa L, F, Rodrigues FA, (2019) Clustering algorithms: a comparative approach. *PLoS ONE* 14:e0210236. <https://doi.org/10.1371/journal.pone.0210236>
69. Fränti P, Virtamäki O (2006) Iterative shrinking method for clustering problems. *Pattern Recogn* 39:761–775
70. Kinnunen T, Sidoroff I, Tuononen M, Fränti P (2011) Comparison of clustering methods: a case study of text-independent speaker modeling. *Pattern Recogn Lett* 32:1604–1617
71. MacQueen J (1967) Some methods for classification and analysis of multivariate observations. *The Regents of the University of California*
72. Xu D, Tian Y (2015) A Comprehensive survey of clustering algorithms. *Ann Data Sci* 2:165–193. <https://doi.org/10.1007/s40745-015-0040-1>
73. Kohonen T (1990) The self-organizing map. *Proc IEEE* 78:1464–1480
74. Nourani V, Mehrvand M, Baghanam AH (2014) Implication of SOM-ANN based clustering for multistation rainfall-runoff modeling. *J Urban Environ Eng* 8:198–210
75. Melka J, Mariage J-J (2017) Efficient Implementation of Self-Organizing Map for Sparse Input Data. In: *IJCCI*. pp 54–63
76. Rasmussen CE (2000) The infinite Gaussian mixture model. In: *Advances in neural information processing systems*. pp 554–560
77. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. *J Roy Stat Soc: Ser B (Methodol)* 39:1–22
78. Johnson SC (1967) Hierarchical clustering schemes. *Psychometrika* 32:241–254
79. Liao TW, Hua G, Qu J, Blau PJ (2006) Grinding wheel condition monitoring with hidden Markov model-based clustering methods. *Mach Sci Technol* 10:511–538
80. Ward JH (1963) Hierarchical grouping to optimize an objective function. *J Am Stat Assoc* 58:236–244
81. Babnik T, Aggarwal RK, Moore PJ (2008) Principal component and hierarchical cluster analyses as applied to transformer partial discharge data with particular reference to transformer condition monitoring. *IEEE Trans Power Delivery* 23:2008–2016. <https://doi.org/10.1109/TPWRD.2008.919030>
82. Arroyo Á, Herrero Á, Tricio V, Corchado E (2017) Analysis of meteorological conditions in Spain by means of clustering techniques. *J Appl Log* 24:76–89
83. Caliński T, Harabasz J (1974) A dendrite method for cluster analysis. *Commun Statist-theor Method* 3:1–27
84. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 20:53–65
85. Davies DL, Bouldin DW (1979) A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence* 224–227
86. Tibshirani R, Walther G, Hastie T (2001) Estimating the number of clusters in a data set via the gap statistic. *J Royal Stat Soc: Series B (Stat Methodol)* 63:411–423
87. Krzanowski WJ, Lai YT (1988) A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics* 23–34
88. Batista GE, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorat News* 6:20–29
89. Prati RC, Batista GE, Silva DF (2015) Class imbalance revisited: a new experimental setup to assess the performance of treatment methods. *Knowl Inf Syst* 45:247–270
90. Wang S, Yao X (2012) Multiclass imbalance problems: analysis and potential solutions. *IEEE Trans Syst, Man, Cybernet Part B (Cybernetics)* 42:1119–1130
91. Ortigosa-Hernández J, Inza I, Lozano JA (2017) Measuring the class-imbalance extent of multi-class problems. *Pattern Recogn Lett* 98:32–38
92. Zhu R, Wang Z, Ma Z, Wang G, Xue J-H (2018) LRID: A new metric of multi-class imbalance degree based on likelihood-ratio test. *Pattern Recogn Lett* 116:36–42
93. Hellinger E (1909) Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *J für die reine und angewandte Mathematik (Crelles Journal)* 1909:210–271
94. Rice JA (2006) *Mathematical statistics and data analysis*. Cengage Learning
95. Avadh Kishor, Singh PK (2016) Empirical Study of Grey Wolf Optimizer. In: Pant M, Deep K, Bansal JC, Nagar A, Das KN (eds) *Proceedings of Fifth International Conference on Soft Computing for Problem Solving*. Springer, Singapore, pp 1037–1049
96. Long W, Liang X, Cai S, Jiao J, Zhang W (2017) A modified augmented Lagrangian with improved grey wolf optimization to constrained optimization problems. *Neural Comput Appl* 28:421–438
97. Yang B, Zhang X, Yu T, Shu H, Fang Z (2017) Grouped grey wolf optimizer for maximum power point tracking of doubly-fed induction generator based wind turbine. *Energy Convers Manage* 133:427–443
98. Malik MRS, Mohideen ER, Ali L (2015) Weighted distance grey wolf optimizer for global optimization problems. In: 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICIC). IEEE, pp 1–6

99. Rodríguez L, Castillo O, Soria J (2016) Grey wolf optimizer with dynamic adaptation of parameters using fuzzy logic. In: 2016 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp 3116–3123
100. Gupta S, Deep K (2019) Hybrid grey wolf optimizer with mutation operator. In: Soft computing for problem solving. Springer, pp 961–968
101. Gupta S, Deep K (2019) An opposition-based chaotic grey wolf optimizer for global optimisation tasks. *J Exp Theor Artif Intell* 31:751–779
102. Gupta S, Deep K, Assad A (2020) Reliability–redundancy allocation using random walk gray wolf optimizer. In: Soft computing for problem solving. Springer, pp 941–959
103. Gupta S, Deep K (2020) A memory-based grey wolf optimizer for global optimization tasks. *Appl Soft Comput* 93:106367. <https://doi.org/10.1016/j.asoc.2020.106367>
104. Gupta S, Deep K, Moayedi H, Foong LK, Assad A (2020) Sine cosine grey wolf optimizer to solve engineering design problems. *Eng Comput*. <https://doi.org/10.1007/s00366-020-00996-y>
105. Yan F, Xu J, Yun K (2019) Dynamically Dimensioned Search Grey Wolf Optimizer Based on Positional Interaction Information. In: Complexity. <https://www.hindawi.com/journals/complexity/2019/7189653/>. Accessed 14 Feb 2021
106. Alsalem MA, Zaidan AA, Zaidan BB, Hashim M, Albahri OS, Albahri AS, Hadi A, Mohammed KI (2018) Systematic review of an automated multiclass detection and classification system for acute Leukaemia in terms of evaluation and benchmarking, open challenges, issues and methodological aspects. *J Med Syst* 42:204
107. Agaian S, Madhukar M, Chronopoulos AT (2014) Automated screening system for acute myelogenous leukemia detection in blood microscopic images. *IEEE Syst J* 8:995–1004
108. Boughorbel S, Jarray F, El-Anbari M (2017) Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PloS one* 12:
109. Hossin M, Sulaiman MN (2015) A review on evaluation metrics for data classification evaluations. *Int J Data Min Knowledge Manag Process* 5:1
110. Schutten M, Wiering MA (2016) An analysis on better testing than training performances on the Iris dataset. In: Proceedings of Belgian Dutch Artificial Intelligence Conference, Amsterdam, The Netherlands. pp 10–11
111. Silva R, Gomes V, Mendes-Faia A, Melo-Pinto P (2018) Using support vector regression and hyperspectral imaging for the prediction of oenological parameters on different vintages and varieties of wine grape berries. *Remote Sensing* 10:312
112. Balantic CM, Donovan TM (2020) Statistical learning mitigation of false positives from template-detected data in automated acoustic wildlife monitoring. *Bioacoustics* 29:296–321. <https://doi.org/10.1080/09524622.2019.1605309>
113. Chen Q, Xue B, Niu B, Zhang M (2016) Improving generalisation of genetic programming for high-dimensional symbolic regression with feature selection. In: 2016 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp 3793–3800
114. Chen Q, Zhang M, Xue B (2017) Feature selection to improve generalization of genetic programming for high-dimensional symbolic regression. *IEEE Trans Evol Comput* 21:792–806
115. Al-Fugara A, Ahmadlou M, Shatnawi R, AlAyyash S, Al-Adamata R, Al-Shabeeb AA-R, Soni S (2020) Novel Hybrid Models Combining Meta-heuristic Algorithms with Support Vector Regression (SVR) for Groundwater Potential Mapping. *Geocarto International* 1–20

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.