


RESEARCH

Open Access



Supervised machine learning-based salp swarm algorithm for fault diagnosis of photovoltaic systems

Amal Hichri¹, Mansour Hajji^{1*}, Majdi Mansouri² , Hazem Nounou² and Kais Bouzrara³

*Correspondence:
majdi.mansouri@qatar.tamu.edu

¹ Research Unit Advanced Materials and Nanotechnologies, Higher Institute of Applied Sciences and Technology of Kasserine, Kairouan University, Kairouan, Tunisia

² Electrical and Computer Engineering Program, Texas A&M University at Qatar, Doha, Qatar

³ Laboratory of Automatic Signal and Image Processing, National School of Engineers of Monastir, University of Monastir, 5019 Monastir, Tunisia

Abstract

The diagnosis of faults in grid-connected photovoltaic (GCPV) systems is a challenging task due to their complex nature and the high similarity between faults. To address this issue, we propose a wrapper approach called the salp swarm algorithm (SSA) for feature selection. The main objective of SSA is to extract only the most important features from the raw data and eliminate unnecessary ones to improve the classification accuracy of supervised machine learning (SML) classifiers. Subsequently, the selected features are used to train supervised machine learning (SML) techniques in distinguishing between various operating modes. To evaluate the efficiency of the technique, we used healthy and faulty data from GCPV systems that have been injected with frequent faults, 20 different types of faults were introduced, including line-to-line, line-to-ground, connectivity faults, and those affecting the operation of bay-pass diodes. These faults present diverse conditions, such as simple and multiple faults in the PV arrays and mixed faults in both arrays. The performances of the developed SSA-SML are compared with those using principal component analysis (PCA) and kernel PCA (KPCA) based SML techniques through different criteria (i.e., accuracy, recall, precision, F1 score, and computation time). The experimental findings demonstrated that the proposed diagnosis paradigm outperformed the other techniques and achieved a high diagnostic accuracy (an average accuracy greater than 99%) while significantly reducing computation time.

Keywords: Fault diagnosis (FD), Feature selection (FS), Photovoltaic (PV) systems, Salp swarm algorithm (SSA), Supervised machine learning (SML)

Introduction

In huge datasets, the process of assessing data becomes more difficult since not all of the data is appropriate. Feature selection is the process of selecting the most important features and removing the repetitious ones in order to solve classification issues. The selected subset of features will improve classification accuracy while decreasing classification time, providing the same or even better classification accuracy than using all of the features [1]. The goal is to identify a set of significant s features from a set of S features ($s < S$) in a given dataset [2]. S is composed of all the features of a particular data collection; it may include noisy, repetitive, and misleading features. As a result,

a complete search cannot be used in practice since it scans the whole solution space, which takes a long time [3]. We intended to save only a subset of the relevant features. Unnecessary features are not only useless for classification, but they may significantly decrease classification accuracy. By removing unnecessary features, computational efficiency, and classification accuracy may be improved. The search criteria contain two types of FS methods: filter-based and wrapper-based. The filter-based techniques choose the feature subset independently of the predictors. Filtering-based FS methods include the gain ratio [4] and information gain (IG) [5]. Wrapper-based techniques, as opposed to filter-based approaches, apply predictors to evaluate the quality of the chosen features [6, 7]. These techniques like sequential backward selection (SBS) [8], sequential forward selection (SFS) [9], and neural network-based methods [10]. Several search approaches, in particular the random search and the greedy search, have been employed to find the most suitable subset of features [11]. Greedy search approaches create and assess all possible combinations of characters, making this strategy time-demanding. Meanwhile, random search approaches scan the search space at random for the best subset of features. However, these approaches have several disadvantages, such as being easily stuck at local optimal points and having a high search space and time complexity. Metaheuristic approaches were employed to address the limitations of the previously discussed FS methods. Metaheuristic techniques are approaches to global optimization that mimic the biological, physical, and animal social behaviors in nature [12]. When applied to FS issues, they can explore the search space both globally and locally. Particle swarm optimization (PSO) [13], genetic algorithms (GAs) [14], differential evolution (DE) [14], Ant lion optimization (ALO) [15], grey wolf optimizer (GWO) [16], and artificial bee colony optimization [17] are all well-known instances of metaheuristics. In the preceding two decades, metaheuristics have proved their efficiency and productivity in solving difficult and large-scale challenges in engineering design and machine learning data mining applications [18]. Several studies have been conducted to evaluate the effectiveness of various metaheuristic algorithms for feature selection. In [19], the authors introduced a binary version of the ant lion optimizer (ALO) to find the optimal set of features and demonstrated that their proposed algorithm outperformed other algorithms in terms of accuracy. In [20], the authors modified the parameter used to balance exploration and exploitation in ALO and introduced a chaotic ALO (CALO), which was shown to outperform standard ALO, particle swarm optimization (PSO), and genetic algorithm (GA). Meanwhile, in [21], the authors proposed a feature selection technique based on a modified Cuckoo Search algorithm with rough sets and showed that their proposed method was superior to other optimizers. In [22], the authors improved the binary iteration of the whale optimization algorithm (WOA) for feature selection, resulting in an improved algorithm (IWOA) that outperformed other algorithms in terms of classification accuracy and feature reduction. In [23], the authors introduced a chaotic version of the moth-flame optimization (MVO) algorithm, called CMVO, which was found to be superior to other optimizers. Finally, in [24], the authors proposed a binary version of the hybrid grey wolf optimization and particle swarm optimization algorithm (BGWOPSO), which outperformed other binary optimization algorithms for accuracy, feature selection, and computational time. Another approach to feature selection is using machine learning

algorithms such as artificial neural networks (ANN). In [25], the authors proposed a feature selection approach based on an extension of particle swarm optimization (PSO) for wind energy conversion (WEC) systems, which demonstrated improved classification performance with reduced computation time. Similarly, in [26], the authors proposed using genetic algorithm (GA) for feature selection in combination with ANN for fault diagnosis in grid-connected photovoltaic (GCPV) systems, which proved to be feasible and effective with low computation time.

In the current study, we present a novel fault diagnosis paradigm for photovoltaic (PV) systems utilizing a feature selection method called SSA-SML. The proposed approach aims to address the complex nature of GCPV systems and the high similarity between different faults, which makes it challenging to diagnose faults accurately and ensure high-performance functioning. The main contributions of our work include:

- The first step in our approach is to select the most important and sensitive features from the data, which can be challenging in nonlinear systems. While PCA is a commonly used method, it is not always effective for fault classification. Therefore, an alternative method called KPCA was developed. However, KPCA can be computationally challenging for large datasets.
- To overcome these challenges, we propose an SSA-based SML technique for detecting faults and distinguishing between operating modes in PV systems. SSA offers several advantages, such as being a new algorithm, easier to implement, having fewer parameters, and having a low computational cost [27].
- The salp swarm algorithm (SSA) is utilized for feature selection by eliminating unnecessary features, while supervised machine learning is used for fault diagnosis. This approach tackles the issues of statistical, multivariate, and nonlinear feature selection and fault diagnosis in GCPV systems while improving classification accuracy, limiting the number of chosen features, and significantly reducing computation time.

The rest of the paper is organized as follows: [Sect. 2](#) gives a brief theoretical overview of PCA, KPCA, and SSA, which are employed in feature extraction and selection. [Section 3](#) is devoted to the discussion of supervised machine-learning techniques. [Section 4](#) presents the proposed methodology for fault diagnosis and classification utilizing an SSA-based SML algorithm. [Section 5](#) presents the simulation results that evaluate the performance of the proposed SSA-based SML. [Section 6](#) concludes the paper.

Methods

Feature extraction and selection

Principal component analysis

Principal component analysis (PCA) is a descriptive method for analyzing existing relationships between system variables without taking the system's model into account [28]. Originally developed by Karl Pearson to describe and summarize the information contained in a dataset, Harold Hotelling later improved it as a technique for analyzing existing relationships between variables [29].

Consider the data matrix $X(N, m)$ of a system, where N represents the number of measurements or observations and m represents the number of sensors or variables. Before running the analysis, it is necessary to perform preprocessing, which includes centering and reducing the data. The goal of this preprocessing is to keep certain variables from dominating the analysis simply due to their high amplitude in comparison to other variables. The following relation then centers each column X_1 of the matrix $X(N \cdot m)$

$$X_i = \frac{X_i - M_i}{\sigma_i^2} \quad (1)$$

Where X_i is the i th column of the matrix X , M_i is the mean of the i th column and $X\sigma_{i_1}^2$ is the variance of the i th column, respectively. The new centered and reduced data matrix is as follows:

$$X = [X_1 X_2 \dots X_m] \quad (2)$$

After obtaining the new data matrix, the covariance matrix Φ is computed as follows:

$$\Phi = \frac{1}{N-1} X^T \cdot X \quad (3)$$

The principal component analysis thus consists of breaking down the matrix as follows:

$$T = P^T X \quad (4)$$

where the principal components of X are represented by the columns of the matrix T . The eigenvectors of the covariance matrix Φ are represented by the columns of the matrix P . In terms of linear systems, PCA is quite effective. Due to the nonlinear nature of the majority of current systems, PCA is ineffective in these systems. In order to get around PCA's difficulties, a number of nonlinear-based PCA techniques have been developed, including kernel principal component analysis (KPCA).

Kernel principal component analysis

Kernel PCA (KPCA) depends on translating data into a higher-dimensional space where the data becomes linear. Consider a data matrix with m variables and N observations that have been normalized.

$$X = [x_1 x_2 \dots x_N]^T \in \mathcal{R}^{N \times m} \quad (5)$$

The data are projected onto the characteristic space H using the function $\varnothing : x_i \in \mathcal{R}^m \rightarrow \varnothing_i = \varnothing(x_i) \in \mathcal{R}^h$ of dimension $h \gg m$. The dot product of two vectors $\varnothing(x_i)$ and $\varnothing(x_j)$ is an important characteristic in the feature space, and it is as follows:

$$\varnothing(x_j)^T \varnothing(x_j) = k(x_i, x_j) \tag{6}$$

where k denotes the kernel function and $i, j = 1, \dots, N$

In this study, we utilized the radial basis function defined as follows:

$$k(x, y) = \exp\left(-\frac{(x - y)^T(x - y)}{c}\right) \tag{7}$$

where c is the width of the Radial basis kernel function.

The KPCA model, like the linear PCA model, is derived by looking at the eigenvalues and eigenvectors of the covariance matrix in the new space. In the case of a collection of centered and reduced data, $\varphi = [\varnothing(x_1)L\varnothing(x_1) \cdots \varnothing(x_{N1})]^T$ The covariance matrix C is defined in the space of the characteristics by:

$$\begin{aligned} (N - 1)C &= \varphi^T \varphi \\ &= \sum_{i=1}^N \varnothing_i \varnothing_i^T \end{aligned} \tag{8}$$

The following equation will be solved to determine the eigenvalues λ and eigenvectors v of the covariance matrix C .

$$\varphi^T \varphi v = \sum_{i=1}^N \varnothing_i \varnothing_i^T v = \lambda v \tag{9}$$

Equation (2) may be expressed from the Gram’s matrix $K = \varphi\varphi^T$ as follows:

$$K\alpha = \lambda\alpha \tag{10}$$

where λ and α are the eigenvalues and eigenvectors of K . It being important to identify the first ℓ kernel principal components. The cumulative percent variance (CPV) criteria [30] are utilized to calculate the number of significant ℓ KPCs. As determined by the first ℓ KPCs, the CPV is a measure of the percent variance:

$$CPV(\ell) = \frac{\sum_{j=1}^{\ell} \lambda_j}{\sum_{j=1}^N \lambda_j} \times 100 \tag{11}$$

Subsequently, the kernel principal components are calculated using

$$t = \Lambda^{-1/2} P^T k(x) \tag{12}$$

where the ℓ principal eigenvectors $P = [\alpha_1, K, \dots, \alpha_1]$ of K are those that correspond to its largest eigenvalues $\Lambda = \text{diag}\{\lambda_1, K, \dots, \lambda_\ell\}$.

To select the effective features, Hotelling’s T^2 and SPE are also used in addition to the ℓ first KPCs. These are the statistical characteristics defined:

$$T^2(x) = k(x)PA^{-1}P^T k(x) \tag{13}$$

$$SPE(x) = k(x, x) - k(x)^T Ck(x) \tag{14}$$

Where $\Lambda = (\Lambda_1, K, \dots, \Lambda_l)$ and $C = PA^{-1}P^T$. Where $k(x)$ is the kernel vector of the measured variable x and is denoted by

$$k(x) = [k(x_1, x)K, \dots, k(x_{N'}x)]^T \tag{15}$$

Figure 1 illustrates the main stages of the KPCA technique for feature extraction and selection.

Salp swarm algorithm (SSA)

SSA is one of the algorithms with a random population that Mirjalili et al. [27]. proposed in 2017. SSA mimics the swarming behavior of salps during ocean foraging. Salps typically form a swarm known as a salp chain in heavy oceans. The salp at the front of the chain is the leader in the SSA algorithm, while the remaining salps are referred to as followers. The position of salps is defined in a d -dimensional search space, where d is the number of variables in a particular problem, similar to previous swarm-based methods. Therefore, a two-dimensional matrix called x is utilized to store the positions of all salps. Additionally, it is believed that the swarm will use S as its aim to find a food source in the search space. The following is the provided mathematical model for SSA. Using the next equation, the leader salp can change its position:

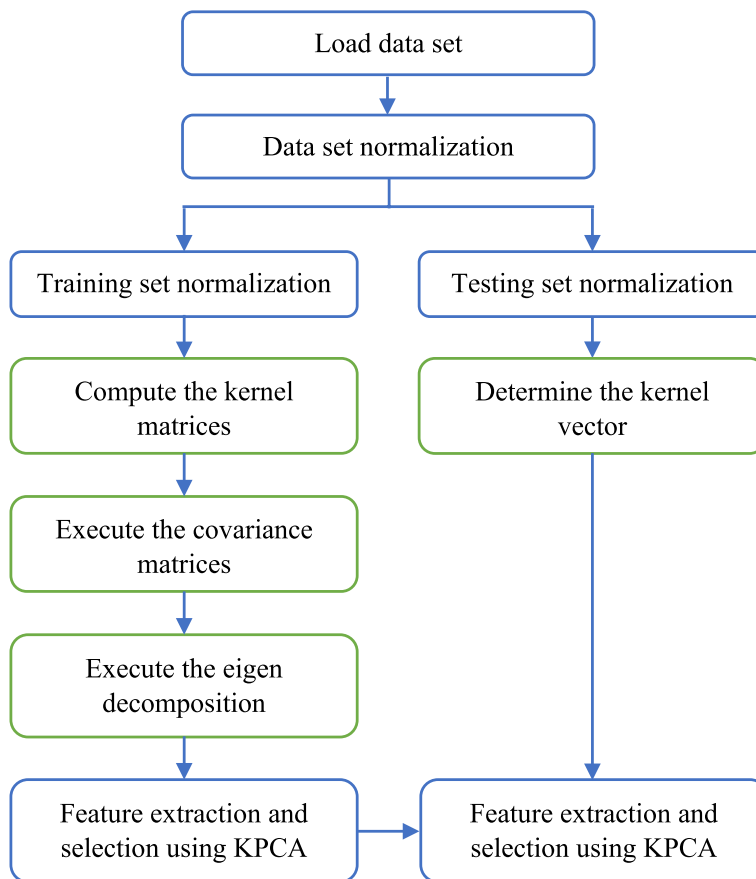


Fig. 1 KPCA-based feature extraction and selection flowchart

$$x_j^1 = \begin{cases} S_j + c_1((ub_j - lb_j)c_2 + lb_j)c_3 \geq 0 \\ S_j + c_1((ub_j - lb_j)c_2 + lb_j)c_3 < 0 \end{cases} \tag{16}$$

where x_j^1 represents the position of the first salp (leader) in the j th dimension, S_j represents the position of the food source in the j th dimension, ub_j and lb_j represent the upper and lower bounds of the j th dimension, respectively, and c_1 , c_2 , and c_3 are random numbers. Equation 16 demonstrates that the leader only changes its position in relation to the food source. Because it balances exploration and exploitation, the coefficient c_1 is the most crucial parameter in the SSA.

$$c_1 = 2e^{-\left(\frac{l}{L}\right)^2} \tag{17}$$

where L is the maximum number of iterations and l represents the current iteration. Random variables in the interval $[0,1]$ are generated uniformly for the parameters c_2 and c_3 . The following equations (Newton’s law of motion) are used to update the position of the followers:

$$x_j^i = \frac{1}{2}\lambda t^2 + \delta_0 t \tag{18}$$

where $i \geq 2$, x_j^1 depicts the position of the i th follower salp in the j th dimension, t denotes time, δ_0 denotes the beginning speed, and $\lambda = \frac{\delta_{final}}{\delta_0}$ where $\delta = \frac{x-x_0}{t}$

The discrepancy between iterations is equal to 1 because the time in optimization is iterated, and since $\delta_0 = 0$, this equation can be written as follows:

$$x_j^i = \frac{1}{2}(x_j^i + x_j^{i-1}) \tag{19}$$

where $i \geq 2$, x_j^i represent the position of the i th following salp in the j th dimension, respectively. It is possible to mimic the salp chains using Eqs. 16 and 19.

SSA-based feature selection

The following is a list of the requirements to develop the SSA-based feature selection paradigm:

Encoding scheme We encoded the individuals using a vector of real numbers. The vector is applied for features that are randomly mapped in the interval $[0,1]$. As a result, if the component value is equal to or greater than 0.5, it is replaced with 1 and the feature is selected. However, the value is estimated to be 0 and the feature is not picked.

Objective function The classification accuracy rate is calculated from Eq. 20, which is our objective function based on computing accuracy for each selection.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{20}$$

where TP (true positive) refers to correctly classified positive observations, TN (true negative) refers to correctly classified negative samples, FP (false positive) refers to

incorrectly classified positive observations, and FN (false negative) refers to incorrectly classified negative observations.

Architecture system In this part, we discussed our suggested system, the SSA-based feature selection architecture. Previous research employed the term 'System Architecture' [31, 32]. The following are the primary components of SSA-based feature selection: Data normalization is a typical preprocess in feature selection. We normalized the features to exist in the interval [0,1] in order to eliminate the negative effects of existing bias values in particular features; this normalization was accomplished by identifying the selected feature by N in Eq. 21:

$$N = \frac{F - \min_F}{\max_F - \min_F} \quad (21)$$

Salps individuals decoding: our vector has been occupied by the selected features in this stage.

Identifying training and testing sets: we partitioned the dataset into training sets ($X_{\text{train}}, Y_{\text{train}}$) and testing sets ($X_{\text{test}}, Y_{\text{test}}$). The main features are represented by $X = [X_1, X_2, \dots, X_n]$ and the main class is Y . To build the model, SML classifiers are used to manage X_{train} and Y_{train} . Finally, we evaluate the model's accuracy by using X_{test} as an input to the model.

Select a feature subset: we picked features with a value of 1 from the training set.

Fitness evaluation: we used training set vectors to train our classifier and then used Eq. 20 to estimate classification accuracy.

Termination condition: we stopped the entire operation by limiting the number of iterations. Figure 2 depicts the entire system workflow for feature selection-based SSA.

Faults classification using supervised machine learning techniques Supervised machine learning classifiers are then applied to these features for the goal of fault classification once the most informative features of the data have been extracted and chosen using PCA, KPCA, and SSA approaches. These classifiers include K-nearest neighbors (KNN), discriminant analysis (DA), decision trees (DT), and support vector machines (SVM).

K-nearest neighbors

The K-nearest neighbors (KNN) technique is a widely used machine learning algorithm for classification and regression tasks. It is a simple yet effective non-parametric method for classifying new observations based on their similarity to previously observed data [33].

Discriminant analysis

Discriminant analysis (DA) is a well-known machine-learning technique for classification tasks. It is a statistical method for determining a linear combination of features that best divides into two or more classes of objects. The purpose of the DA is to find a function that can accurately forecast the grouping or classification of new observations based on their predictor variable values [34, 35].

Decision trees

The decision tree (DT) is a common machine-learning technique that represents a decision-making process using a tree-like structure. Each node in the tree represents a

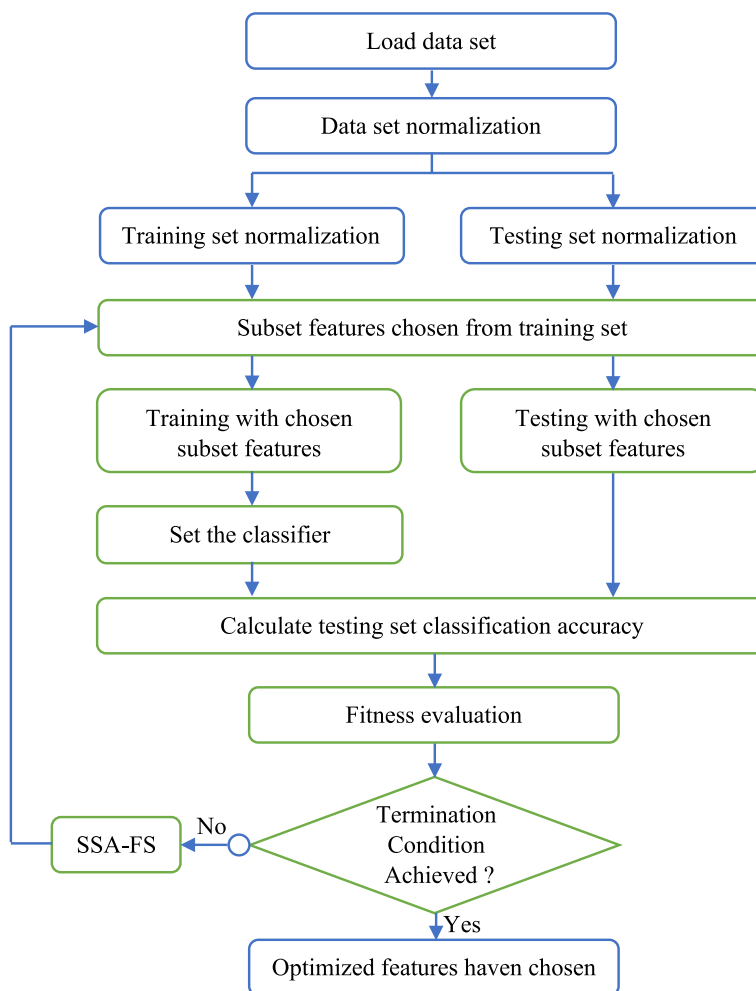


Fig. 2 SSA-based feature selection flowchart

decision based on a certain feature or attribute, and the branches indicate various outcomes or decisions based on that feature [36].

Support vector machines

Support vector machine (SVM) is a supervised machine learning model. It is based on the concept of a hyperplane classifier, also known as linear separability. The purpose of SVM is to identify a linear optimal hyperplane that maximizes the margin of separation between the two classes [37, 38].

Fault diagnosis and classification using SSA-based SML technique The proposed methodology for fault diagnosis in GCPV systems consists of two primary steps: feature selection and fault classification. The approach utilizes filter and wrapper methods for feature selection, and the supervised machine learning (SML) classifier for fault diagnosis. The aim is to simplify the classification process due to the complex nature of GCPV systems and the high similarity between different faults. The first step involves collecting GCPV data, which is then subjected to PCA, KPCA, and SSA to extract and select the

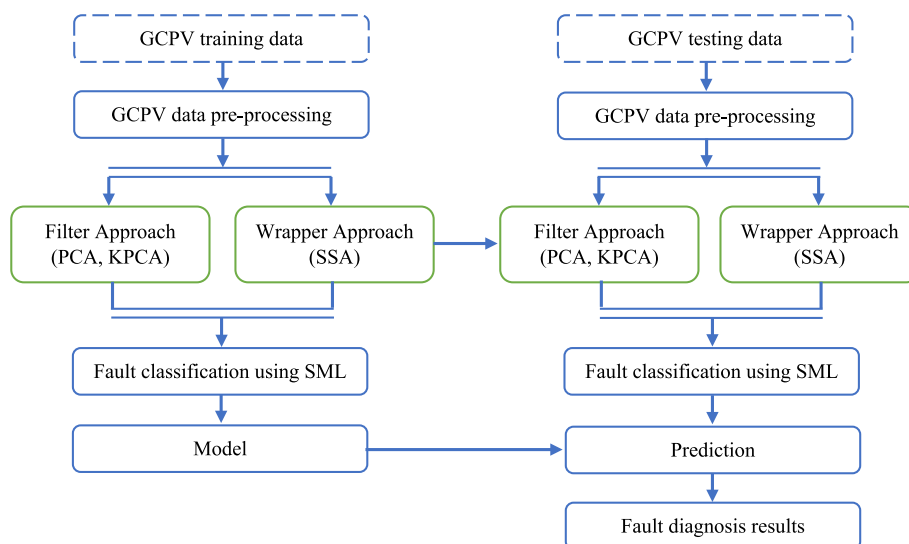


Fig. 3 Illustration of SML-based features selection procedures for PV fault diagnosis

most efficient and pertinent features. The selected feature subset is then used as input to the SML classifier to differentiate between operating modes and classify faults. The proposed technique is summarized in the block diagram shown in Fig. 3. This study presents an effective fault diagnosis technique based on the SSA model and SML classifiers. Although PCA is highly efficient for linear systems, it is inappropriate for most nonlinear systems, which are prevalent in GCPV systems. Moreover, KPCA may be inadequate for real-world applications with large datasets. To address these challenges, an optimized SSA-based SML classifier technique is proposed, which utilizes SSA for feature selection and SML for fault classification.

The proposed SSA-based SML technique is a promising solution for detecting and identifying faults in GCPV systems. It leverages the strengths of SSA for feature selection and SML for fault classification to address the challenges posed by nonlinear systems and large datasets.

Experimental results and discussions

System description

Figure 4 shows a photovoltaic system setup with a DC bus voltage of 500 V. The PV side is made up of 3 PV networks with a maximum power of 4 kW each. A single set of PV arrays is composed of 2 parallel chains where each chain has 24 modules connected in series. Every module has 20 cells [26].

In this study, the two parallel PV fields, PV₁ and PV₂, underwent different scenarios representing five types of faults, as outlined in Table 1. The simple fault in PV₁ involved four fault scenarios:

- Bypass diode fault: The bypass is emulated by changing the resistance.

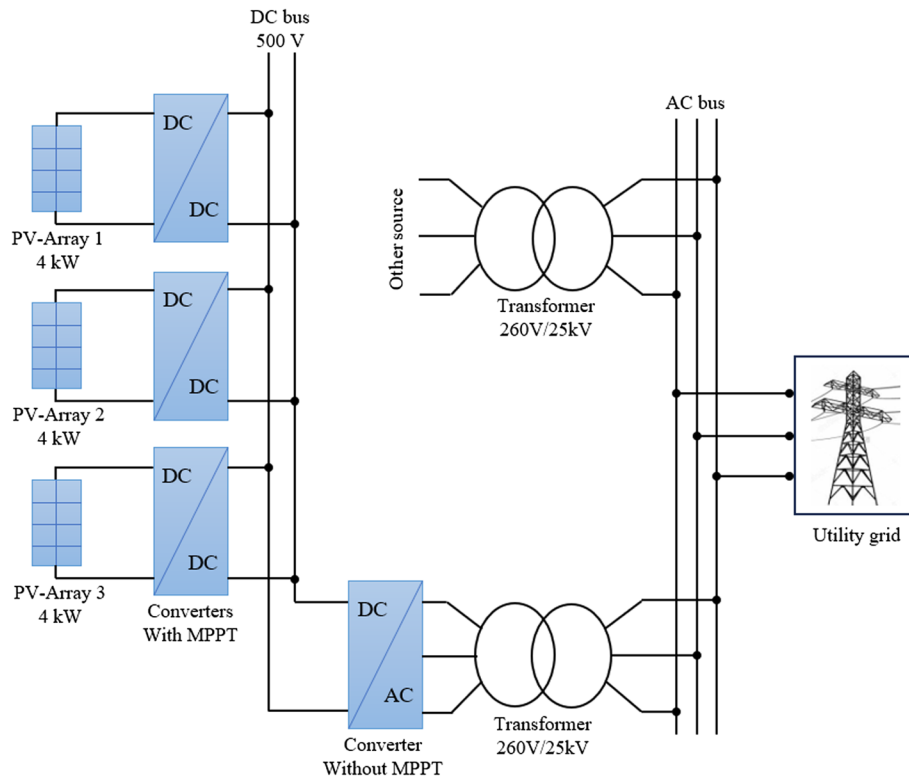


Fig. 4 Scheme of a parallel structure on a direct bus

Table 1 Description and characteristics of the different labeled injected faults

Types of faults	Fault label	Description
Simple fault in PV ₁	F ₁	Bypass diode fault (BD ₁)
	F ₂	Connectivity fault (Cn ₁)
	F ₃	Line to line fault (LL ₁)
	F ₄	Line to ground fault (LG ₁)
Simple fault in PV ₂	F ₅	Bypass diode fault (BD ₂)
	F ₆	Connectivity fault (Cn ₂)
	F ₇	Line to line fault (LL ₂)
	F ₈	Line to ground fault (LG ₂)
Multiple fault	F ₉	LL ₁ + LG ₁
	F ₁₀	LL ₁ + BD ₁
	F ₁₁	LL ₂ + LG ₂
	F ₁₂	LG ₂ + Cn ₂
Mixed fault	F ₁₃	LL ₁ + LL ₂
	F ₁₄	LG ₁ + LG ₂
	F ₁₅	LL ₁ + BD ₂
	F ₁₆	LG ₁ + Cn ₁
	F ₁₇	LL ₁ + LG ₁ + BD ₂
	F ₁₈	BD ₁ + BD ₂ + LG ₂
	F ₁₉	BD ₁ + BD ₂ + LL ₂
	F ₂₀	LL ₁ + BD ₁ + Cn ₂ + LG ₂

- Connectivity fault: the connectivity fault is considered in the string of the PV system, between two modules. This fault was modeled by a serial variable resistance.
- Line-to-line fault: LL is described by the variation in resistance that is situated between any two points in the PV array.
- Line to ground fault: LG is described by the variation in resistance that is situated between one point and the ground.

This study deals with various fault scenarios, and each scenario includes several cases, as shown in Table 2.

- The first scenario refers to simple faults that only affect the PV1 array.
- The second scenario represents simple faults that solely affect the PV2 array.
- The third scenario deals with multiple faults on the same array. In this case, we address multiple faults on both PV1 and PV2 separately.
- The fourth scenario examines mixed faults that might occur on both arrays at the same time.
- The fifth scenario integrates all of the preceding scenarios to monitor the system in all of its states.

Table 3 shows the various simulated 8 variable measurements that were collected in order to carry out the various experiments for fault diagnosis purposes. These variables represent one healthy (attributed to class C_0) and 20 different faulty operating modes of GCPV (assigned to $C_i, i = 1, \dots, 20$), respectively, as shown in Table 2. The

Table 2 Construction of database for GCPV fault diagnosis system

Types of faults	Class	State	Training set	Testing set
Normal	C_0	Healthy	6000	6000
	C_1	F_1	6000	6000
Simple fault in PV ₁	C_2	F_2	6000	6000
	C_3	F_3	6000	6000
	C_4	F_4	6000	6000
Simple fault in PV ₂	C_5	F_5	6000	6000
	C_6	F_6	6000	6000
	C_7	F_7	6000	6000
Multiple fault	C_8	F_8	6000	6000
	C_9	F_9	6000	6000
	C_{10}	F_{10}	6000	6000
Mixed fault	C_{11}	F_{11}	6000	6000
	C_{12}	F_{12}	6000	6000
	C_{13}	F_{13}	6000	6000
	C_{14}	F_{14}	6000	6000
	C_{15}	F_{15}	6000	6000
	C_{16}	F_{16}	6000	6000
	C_{17}	F_{17}	6000	6000
	C_{18}	F_{18}	6000	6000
	C_{19}	F_{19}	6000	6000
	C_{20}	F_{20}	6000	6000

Table 3 Variables description

Variables	Descriptions
x_1	I_{pv1} : Output current of the PV ₁ panel (A)
x_2	V_{pv1} : Output voltage of the PV ₁ panel (V)
x_3	I_{pv2} : Output current of the PV ₂ panel (A)
x_4	V_{pv2} : Output voltage of the PV ₂ panel (V)
x_5	V_{dc} : Grid voltage phase dc (V)
x_6	i_a : Grid current phase a (A)
x_7	i_b : Grid current phase b (A)
x_8	i_c : Grid current phase c (A)

collected dataset was divided into two categories, namely, training and testing datasets, and the same observations were used for both. To validate the testing dataset, we added noise of significant magnitude.

The following criteria have been approved for use in evaluating and comparing performance: accuracy, precision, recall, F1 score, and computation time (CT) [39].

Simulation results In this section, the proposed methods PCA, KPCA, and SSA-based SML are applied for monitoring the GCPV system, a tenfold cross-validation approach was used. In order to perform the proposed FD paradigm, four conditions are considered including the first condition (attributed to Cd_1), which represents a healthy mode, Simple fault in PV1 (F_3), and simple fault in PV2 (F_7) modes. The second condition (Cd_2), which represents a healthy mode, Simple fault in PV1 (F_2) and simple fault in PV2 (F_6) modes. The third condition (Cd_3), which represents a healthy mode and Mixed fault mode (F_{15}). Finally, the last condition (Cd_4), which represents a healthy mode and all faults modes (F_1 to F_{20}).

The PCA and KPCA algorithms are used as a feature selection technique in a filter mode. In this study and in regard to the PCA model, 3 groups of features are used, containing group 1: (T_ρ), group 2: (T_ρ , SPE), and group 3: (T_ρ , T^2 , SPE). Group 2 (the first $\ell=6$ PCs and SPE statistics) provides the best results in terms of classification accuracy. Where 6 Principal components have been retained to be used in a supervised machine learning classifier in all faults. Thus, due to its underlying linearity assumption, PCA performs quite poorly for fault classification in some nonlinear systems. KPCA was developed to deal with nonlinear relationships between process variables. Where, the 95% cumulative variance criteria are used to identify the retained KPCs, with 53 KPCs remaining.

On the other hand, the SSA algorithm is used as a feature selection technique in a wrapper mode by applying the KNN, DA, DT, and SVM classifiers as a fitness function (where $K=5$, $nSplit=50$, $Disc='l'$ and $Kernel=r$). In this work, these classifiers are used as a classification algorithm to evaluate the quality of the chosen subset of features. The SSA parameters are set as follows, the population size (number of salps) is 10 and the maximum number of iterations is 50. The results presented in Table 4 show that the SSA-SML selects a minimal number of features in all faults.

Table 4 SSA-based feature selection in all faults

Algorithm	All features	No. of selected	Features
SSA-KNN	8	5	1, 2, 3, 4, 7
SSA-DA	8	3	2, 4, 5
SSA-DT	8	4	1, 2, 4, 8
SSA-SVM	8	6	1, 2, 3, 4, 5, 7

Discussions

Various classifiers are used in this study, and the best classifier is chosen based on classification performance. Table 5 depicts the overall performance accuracy.

Firstly, PCA-SML achieved low accuracies in some cases. In Cd_1 , in this case, all the developed approaches had high diagnosis performance, with accuracy rates of 87.96%, 84.89%, 85.47%, and 97.18% for KNN, DA, DT, and SVM classifiers, respectively. However, the results decreased in Cd_2 compared to the previous condition. Additionally, The fault diagnosis techniques showed poor performances in Cd_3 , with an accuracy rate of 59.86% for the SVM classifier. When dealing with all fault conditions (Cd_4), PCA-based DA and DT had low accuracy rates of 48.85% and 47.64%, respectively, and were inefficient in distinguishing between different operating modes.

Secondly, KPCA-SML achieved accuracies between 5.50 and 99.89%. In Cd_1 , the KNN, DT, and SVM classifiers showed good results in terms of performance classification, except for the DA classifier with an accuracy rate of 33.35%. However, KNN and SVM had very high computation times during the testing stage. Moreover, the outcomes decreased in Cd_2 compared to the initial condition. Consequently, in Cd_3 , the Fault diagnosis techniques showed good results, except for DA, with an accuracy rate of 46.11%. When dealing with all fault conditions (Cd_4), KNN and SVM classifiers achieved high accuracy rates of 98.83% and 90.91%, respectively. However, DA and DT showed very poor classification with accuracy rates of 5.50% and 15.75%, respectively.

Finally, SSA achieved the highest accuracies (57.62 – 99.98%) using the all conditions. SSA-ML had the best overall performance with accuracies of 99.98% and 99.91% for SVM and DT classifiers, respectively, in Cd_1 . In Cd_2 , all the developed approaches had high diagnosis performance. Then in Cd_3 , SSA improved the performance classification of KPCA-based DA classifier with an accuracy rate increasing from 46, 11% to 83.77%. In the last condition, SSA enhanced the results of KPCA-DA from 5.50 to 57.62%, from 15.75 to 69.72%, and from 90.91 to 99.46% for the DA, DT, and SVM classifiers, respectively. Besides, the proposed method led to a significant reduction in computation time compared to the other methods. Furthermore, SSA outperformed other techniques in terms of classification accuracy, recall, precision, F1 score, and computation time, due to its ability to explore the feature space intelligently. These results confirmed the effectiveness of the SSA in analyzing the feature space and selecting the best subset that resulted in higher classification performance.

Conclusions

In this study, we focused on diagnosing various incipient faults of grid-connected photovoltaic (GCPV) systems during different operation modes. We identified 20 different types of faults, including line-to-line and line-to-ground faults, connectivity faults,

Table 5 Summary performances of different classifiers

Classifiers	Features selection	Phase	Global performance				
			Accuracy	Recall	Precision	F ₁ score	CT(s)
KNN	PCA method	<i>Cd</i> ₁	87.96%	87.96%	88.04%	87.99%	2.63
		<i>Cd</i> ₂	73.39%	73.40%	73.72%	73.56%	2.14
		<i>Cd</i> ₃	88.97%	88.97%	89.03%	88.99%	1.02
		<i>Cd</i> ₄	70.53%	70.53%	70.58%	70.55%	15.7
DA		<i>Cd</i> ₁	84.89%	84.89%	86.35%	85.61%	0.03
		<i>Cd</i> ₂	70.50%	70.50%	72.03%	71.26%	0.02
		<i>Cd</i> ₃	61.44%	61.45%	61.44%	61.44%	0.09
		<i>Cd</i> ₄	48.85%	48.85%	51.27%	50.03%	0.9
DT		<i>Cd</i> ₁	85.47%	85.47%	86.84%	86.15%	0.02
		<i>Cd</i> ₂	67.02%	67.02%	68.95%	67.97%	0.02
		<i>Cd</i> ₃	53.86%	53.86%	62.37%	57.80%	0.03
		<i>Cd</i> ₄	47.64%	47.64%	50.81%	49.17%	0.2
SVM	<i>Cd</i> ₁	79.18%	79.18%	81.17%	80.16%	2.66	
	<i>Cd</i> ₂	59.86%	59.86%	68.15%	63.74%	5.17	
	<i>Cd</i> ₃	65.38%	65.39%	75.85%	70.23%	0.5	
	<i>Cd</i> ₄	63.85%	62.30%	63.85%	62.54%	6.31	
KNN	<i>Cd</i> ₁	99.89%	99.90%	99.90%	99.90%	26.34	
	<i>Cd</i> ₂	99.21%	99.21%	99.22%	99.21%	30.12	
	<i>Cd</i> ₃	99.78%	99.78%	99.78%	99.78%	16.08	
	<i>Cd</i> ₄	98.83%	98.83%	98.85%	98.84%	145.8	
DA	<i>Cd</i> ₁	33.35%	33.35%	33.48%	33.41%	0.16	
	<i>Cd</i> ₂	32.33%	32.34%	31.56%	31.95%	0.20	
	<i>Cd</i> ₃	46.11%	46.11%	41.43%	43.64%	0.19	
	<i>Cd</i> ₄	5.50%	5.50%	6.78%	6.07%	5.22	
DT	<i>Cd</i> ₁	92.27%	92.27%	93.36%	90.68%	0.01	
	<i>Cd</i> ₂	86.86%	86.68%	87.01%	86.84%	0.03	
	<i>Cd</i> ₃	99.37%	99.37%	99.38%	99.37%	0.11	
	<i>Cd</i> ₄	15.75%	15.75%	42.33%	22.96%	0.28	
SVM	<i>Cd</i> ₁	99.08%	99.08%	99.09%	99.08%	5.13	
	<i>Cd</i> ₂	98.84%	98.84%	98.88%	98.86%	6.48	
	<i>Cd</i> ₃	99.52%	99.53%	99.53%	99.53%	1.55	
	<i>Cd</i> ₄	90.91%	90.91%	92.27%	91.58%	21.76	
KNN	<i>Cd</i> ₁	99.79%	99.79%	99.79%	99.79%	0.25	
	<i>Cd</i> ₂	99.78%	99.78%	99.78%	99.78%	0.25	
	<i>Cd</i> ₃	99.85%	99.85%	99.85%	99.85%	0.14	
	<i>Cd</i> ₄	99.38%	99.39%	99.38%	99.38%	1.9	
DA	<i>Cd</i> ₁	99.71%	99.71%	99.71%	99.71%	0.1	
	<i>Cd</i> ₂	87.67%	87.67%	87.96%	87.81%	0.12	
	<i>Cd</i> ₃	83.77%	83.77%	87.64%	85.66%	0.05	
	<i>Cd</i> ₄	57.62%	57.62%	63.83%	60.57%	0.67	
DT	<i>Cd</i> ₁	99.91%	90.90%	99.90%	90.90%	0.03	
	<i>Cd</i> ₂	99.06%	99.06%	99.07%	99.06%	0.015	
	<i>Cd</i> ₃	99.94%	99.94%	99.94%	99.94%	0.03	
	<i>Cd</i> ₄	69.72%	69.72%	70.61%	70.16%	0.25	
SVM	<i>Cd</i> ₁	99.98%	99.98%	99.98%	99.98%	0.55	
	<i>Cd</i> ₂	99.98%	99.98%	99.98%	99.98%	1.10	
	<i>Cd</i> ₃	99.94%	99.95%	99.95%	99.95%	0.55	
	<i>Cd</i> ₄	99.46%	99.47%	99.46%	99.46%	3.37	

and faults affecting the operation of bypass diodes. These faults presented diverse conditions, such as simple and multiple faults in the PV arrays and mixed faults in both arrays. To address the complexity and similarity between faults, we developed a feature selection tool to enhance the accuracy of the supervised machine learning (SML) models. Firstly, we applied the salp swarm algorithm (SSA) for feature selection to select the most effective features from the raw data. Then, we fed these significant and sensitive features into the SML model for classification purposes. The results confirmed that the developed paradigm significantly improved the diagnosis performance when applied to GCPV systems. The diagnosis accuracies of the proposed SSA-SML were compared to those using PCA and kernel PCA-based SML methods through different metrics (i.e., accuracy, recall, precision, F_1 score, and computation time). The obtained results confirmed that the development paradigm outperformed the other methods and achieved a high diagnostic accuracy (an average accuracy greater than 99%) and low computation time using GCPV data.

Abbreviations

GCPV	Grid-connected photovoltaic system
FD	Fault diagnosis
FS	Feature selection
SML	Supervised machine learning
PCA	Principal component analysis
KPCA	Kernel principal component analysis
SSA	Salp swarm algorithm
KNN	K-nearest neighbors
SVM	Support vector machine
DT	Decision tree
DA	Discriminant analysis
CT	Computation time

Acknowledgements

The publication is the result of the Qatar National Research Fund (QNRF) research grant.

Author's contributions

Amal Hichri: writing—original draft, software. Mansour Hajji: writing—original draft, software. Majdi Mansouri: supervision, methodology, reviewing, and editing. Hazem Nounou: visualization. Kais Bouzrara: visualization. All authors read and approved the final manuscript.

Funding

Funding is provided by the Qatar National Library.

Availability of data and materials

Data will be made available on request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 12 April 2023 Accepted: 15 December 2023

Published online: 05 January 2024

References

1. Harvey DY, Todd MD (2014) Automated feature design for numeric sequence classification by genetic programming. *IEEE Trans Evol Comput* 19(4):474–489
2. Oh I-S, Lee J-S, Moon B-R (2004) Hybrid genetic algorithms for feature selection. *IEEE Trans Pattern Anal Mach Intell* 26(11):1424–1437
3. Cover TM, Van Campenhout JM (1977) On the possible orderings in the measurement selection problem. *IEEE Trans Syst Man Cybern* 7(9):657–661
4. Witten IH, Frank E (2002) Data mining: practical machine learning tools and techniques with java implementations. *ACM SIGMOD Rec* 31(1):76–77

5. Quinlan JR (1986) Induction of decision trees. *Mach Learn* 1(1):81–106
6. Ma L, Li M, Gao Y, Chen T, Ma X, Qu L (2017) A novel wrapper approach for feature selection in object-based image classification using polygon-based cross-validation. *IEEE Geosci Remote Sens Lett* 14(3):409–413
7. Zhu Z, Ong Y-S, Dash M (2007) Wrapper-filter feature selection algorithm using a memetic framework. *IEEE Transactions on Systems, Man, and Cybernetics. Part B (Cybernetics)* 37(1):70–76
8. Gheyas IA, Smith LS (2010) Feature subset selection in large dimensionality domains. *Pattern Recogn* 43(1):5–13
9. Bermejo P, J. A. G´omez, J. M. Puerta, Incremental wrapper-based subset selection with replacement: An advantageous alternative to sequential forward selection, in, (2009) *IEEE Symposium on Computational Intelligence and Data Mining*. IEEE 2009:367–374
10. Setiono R, Liu H (1997) Neural-network feature selector. *IEEE Trans Neural Networks* 8(3):654–662
11. Langley P, et al. (1994) Selection of relevant features in machine learning, in: *Proceedings of the AAAI Fall symposium on relevance*, Vol. 184, Citeseer, pp. 245–271.
12. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
13. Banks A, Vincent J, Anyakoha C (2008) A review of particle swarm optimization. part ii: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. *Natural Computing* 7(1):109–124.
14. Han K-H, Kim J-H (2002) Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Trans Evol Comput* 6(6):580–593
15. Mafarja MM, Mirjalili S (2019) Hybrid binary ant lion optimizer with rough set and approximate entropy reduces for feature selection. *Soft Comput* 23(15):6249–6265
16. Ibrahim RA, Elaziz M. Abd, Lu S (2018) Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization. *Expert Syst Appl* 108:1–27.
17. Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. *Appl Math Comput* 214(1):108–132
18. Javidi M, Emami N (2016) A hybrid search method of wrapper feature selection by chaos particle swarm optimization and local search. *Turk J Electr Eng Comput Sci* 24(5):3852–3861
19. Emary E, Zawbaa HM, Hassanien AE (2016) Binary ant lion approaches for feature selection. *Neurocomputing* 213:54–65
20. Zawbaa HM, Emary E, Grosan C (2016) Feature Selection via Chaotic Antlion Optimization. *PLoS ONE* 11(3):e0150652. <https://doi.org/10.1371/journal.pone.0150652>
21. Aziz MAE, Hassanien AE (2018) Modified cuckoo search algorithm with rough sets for feature selection. *Neural Comput Appl* 29(4):925–934
22. Hegazy AE, Makhoulouf M, El-Tawel GS (2018) Dimensionality reduction using an improved whale optimization algorithm for data classification. *Int J Modern Educ Comput Sci* 11(7):37
23. Ewees AA, El Aziz MA, Hassanien AE (2019) Chaotic multi-verse optimizer-based feature selection. *Neural Comput Appl* 31(4):991–1006.
24. Al-Tashi Q, Kadir SJA, Rais HM, Mirjalili S, Alhussain H (2019) Binary optimization using hybrid grey wolf optimization for feature selection, *IEEE Access* 7:39496–39508
25. Mansouri M, Dhibi K, Nounou H, Nounou M (2022) An effective fault diagnosis technique for wind energy conversion systems based on an improved particle swarm optimization. *Sustainability* 14(18):11195
26. Hichri A, Hajji M, Mansouri M, Abodayeh K, Bouzrara K, Nounou H, Nounou M (2022) Genetic-algorithm-based neural network for fault detection and diagnosis: Application to grid-connected photovoltaic systems. *Sustainability* 14(17):10518
27. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 114:163–191
28. Harkat MF (2003) D´etection et localisation de d´efauts par analyse en composantes principales, Ph.D. thesis, Institut National Polytechnique de Lorraine-INPL.
29. Chouaib C (2016) Diagnostic et surveillance des proc´ed´es industriels et de leur environnement sur la base de l’analyse de donn´ees, Ph.D. thesis, Badji Mokhtar-Annaba University.
30. Maulud A, Wang D, Romagnoli J (2006) A multi-scale orthogonal nonlinear strategy for multi-variate statistical process monitoring. *J Process Control* 16(7):671–683
31. Braga PL, Oliveira AL, Meira SR (2008) A ga-based feature selection and parameters optimization for support vector regression applied to software effort estimation, in: *Proceedings of the 2008 ACM symposium on Applied computing*, pp. 1788–1792. <https://doi.org/10.1145/1363686.1364116>
32. Faris H, Hessonah MA, Al-Zoubi A, Mirjalili S, Aljarah I (2018) A multiverse optimizer approach for feature selection and optimizing svm parameters based on a robust system architecture. *Neural Comput Appl* 30(8):2355–2369
33. Wang Y, Pan Z, Pan Y (2019) A training data set cleaning method by classification ability ranking for the k-nearest neighbor classifier. *IEEE transactions on neural networks and learning systems* 31(5):1544–1556
34. Klecka WR, Iversen GR, Klecka WR (1980) *Discriminant analysis*, Vol. 19, Sage. <https://doi.org/10.4135/9781412983938>
35. Balakrishnama S, Ganapathiraju A (1998) *Linear discriminant analysis-a brief tutorial*. Institute for Signal and information Processing 18(1998):1–8
36. Breiman L, Friedman JH, Olshen RA, Stone CJ (2017) *Classification and regression trees*, Routledge. *Nat Methods* 14:757–8. <https://doi.org/10.1038/nmeth.4370>
37. Dietrich R, Opper M, Sompolinsky H (1999) Statistical mechanics of support vector networks. *Phys Rev Lett* 82(14):2975
38. Kaper M, Meinicke P, Grossekhoefer U, Lingner T, Ritter H (2004) Bci competition 2003-data set iib: support vector machines for the p300 speller paradigm. *IEEE Trans Biomed Eng* 51(6):1073–1076
39. Mandal JK, Bhattacharya D (2020) Emerging technology in modelling and graphics, *Advances in Intelligent Systems and Computing (AISC, volume 937)*, Springer

Publisher’s Note

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