Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.029210
Article

Tech Science Press

Fault Diagnosis in Robot Manipulators Using SVM and KNN

D. Maincer[1,*], Y. Benmahamed[2], M. Mansour[1], Mosleh Alharthi[3] and Sherif S. M. Ghonein[3]

[1]Laboratoire de Robotiques Parallélisme et Systèmes Embarquer, Département Automatique et Control, Université des Sciences et Technologie Houari Boumediene, Algiers, Algeria
[2]Laboratoire de Recherche en Electrotechnique, Ecole Nationale Polytechnique, B.P 182, El-Harrach, 16200, Algiers, Algeria
[3]Electrical Engineering Department, College of Engineering, Taif University, P. O. Box 11099, Taif, 21944, Saudi Arabia
*Corresponding Author: D. Maincer. Email: dmaincer@usthb.dz
Received: 27 February 2022; Accepted: 14 April 2022

Abstract: In this paper, Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) based methods are to be applied on fault diagnosis in a robot manipulator. A comparative study between the two classifiers in terms of successfully detecting and isolating the seven classes of sensor faults is considered in this work. For both classifiers, the torque, the position and the speed of the manipulator have been employed as the input vector. However, it is to mention that a large database is needed and used for the training and testing phases. The SVM method used in this paper is based on the Gaussian kernel with the parameters γ and the penalty margin parameter "*C*", which were adjusted *via* the PSO algorithm to achieve a maximum accuracy diagnosis. Simulations were carried out on the model of a Selective Compliance Assembly Robot Arm (SCARA) robot manipulator, and the results showed that the Particle Swarm Optimization (PSO) increased the performance of the SVM algorithm with the 96.95% accuracy while the KNN algorithm achieved a correlation up to 94.62%. These results showed that the SVM algorithm with PSO was more precise than the KNN algorithm when was used in fault diagnosis on a robot manipulator.

Keywords: Support Vector Machine (SVM); Particle Swarm Optimization (PSO); K-Nearest Neighbor (KNN); fault diagnosis; manipulator robot (SCARA)

1 Introduction

Modern robots are highly susceptible to faults during their execution due to the highly complex nature of new-generation robotic systems and the uncertain environments they occupy [1]. However, even well-designed robotic equipment will be subject to defects during its service life. For this purpose, numerous diagnostic techniques in the literature have been investigated to solve these problems [2–6]. The detection of emerging sensor/actuator failures has been given limited attention in the literature, which is important for the operation of the robot [7]. Indeed, detecting and isolating the defect sensor/ actuator in a manipulator is not easy during the operation of the robot. Sensor measurements describe the features of the monitored system and the sensor itself. For example, any abnormal deviation in sensor readings could be caused by a change in the monitoring system.

The system is also becoming more complex as the number of interconnected sensors and subsystems increases. This increase may cause defects to appear independently or concurrently. Furthermore, the measurements can be noisy because of the imperfect nature of the sensor [7].

Therefore, isolation and defect detection estimation (FDIE) methods are important for the diagnosis of manipulation defects [2,3]. Many studies have been developed based on artificial intelligence (AI) methods such as K-Nearest Neighbor (KNN) algorithm, artificial neural networks (ANNs), linear and nonlinear regression analyses, fuzzy logic, discriminant analysis and the combination of the Sequential Floating advance quest and Fisher Projection technique [8–13]. Artificial intelligence techniques were employed to derive affective states. Then they are adapted to such diagnostic issues. In addition to the aforementioned methods, Time Delay Control (TDC), Sliding Mode Control (SMC), tolerant control of faults which are applied to contract the fault effect based on fault detection, and neural networks with TDC [14–20].

In other words, the machine learning method was used for the automatic diagnosis of failures in many industries [21,22]. In particular, SVM is a significant machine learning method widely supported in many applications [23–25]. The SVM performance is highly dependent on the selection of certain parameters (such as kernel function and the parameters of regulation), the selection of kernel function and regulation parameters are important for SVM technique [26]. Even though, *v*-fold cross validation is the usually used procedure to establish the SVM model [2–4]. It requires t-times training, which in its turn needs a lot of computation (which is CPU-*intensive*). The SVM algorithm can be hybridized with other algorithms such as the Convolutional Neural Network (CNN) [27]. The aim of the combination is to achieve an accurate diagnosis.

The aim contribution of this paper is to approve the effectiveness of the machine learning method in diagnosing faults in a manipulator arm, it would be useful to do a comparative study between PSO-SVM and KNN algorithms. Recognizing that, the manipulating arm used here has three degrees of freedom and has been affected by defects on the three successive articulations. The proposed algorithms will be capable of detecting, isolating and identifying affected articulations, depending on the x-axis, y-axis, z-axis, xy-axis, xz-axis, yz-axis and xyz-axis. SVM parameters have been selected automatically by the PSO algorithm to improve diagnosis accuracy. In decree to have a dependable model, the Cross Validation (CV) method has been used. The application of PSO-SVM and the contrastive tests reveal the efficiency and the supremacy of the suggested technique compared to the KNN algorithm.

The remainder of the paper is organized as follows. Section 2 offers an overview of the dynamic model of the robot. Section 3 presents some reminders on KNN and SVM-PSO algorithms. Section 4 specifies the simulation results and discussions. Finally, Section 5 concludes the document and provides guidance for future work.

2 Problem formulations

Manipulative robots are represented by dynamic equations that are found thanks to the Lagrangian formulation, so the equation of a manipulator is written as follows [28]:

$$M(q)\,\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{1}$$

where $G(q)$ is the vector of gravitational force, $C(q,\dot{q})$ is the matrix of centrifuge and Coriolis, $M(q)$ the inertia matrix, $\tau\varepsilon R^n$ is the vector torque, $\overset{\varepsilon}{q}\,R^n, \dot{q}\varepsilon R^n, q\varepsilon R^n$ are angular acceleration, the angular velocity, and the angle of the joints, respectively [28]. In this paper, the dynamic equations of the *SCARA* robot have been utilized for the simulation.

The additive fault sensors signified by:

$$q_t = q + \Delta q \tag{2}$$

where:

- q_t is the sensor's joint measurement;
- q is the sensor's joint fault-free measurement;
- Δq is the supposed sensor fault;

Then the dynamic Eq. (3) can be obtained by replacing Eq. (2) in Eq. (1).

$$M(q_t)\,\ddot{q}_t + C(q_t, \dot{q}_t)\,\dot{q}_t + G(q_t) = \tau_t \tag{3}$$

Where:

- $M(q_t)$ is the inertia matrix;
- $C(q_t, \dot{q}_t)$ is the matrix of Coriolis and centrifuge;
- $G(q_t)$ is the vector of gravitational force;
- τ_t is the control input torque;

3 K-Nearest Neighbor (KNN)

KNN, which is a supervised learning algorithm, usually applied to signal processing, pattern recognition, image processing, medical, etc. [29]. The algorithm assumes that the closest position for acquisitions falls into the same category. The technique is developed to compute the distance between the test point and the drive samples [30]. Thus, the nearest neighbors are found from the training dataset to determine which class label is the most common. The *k* closest data points are studied with the aim of assigning to the data points being analyzed [31].

The value of *k* is chosen so that it is not too small, thus that the noise effect is minimized in the training dataset. On the other hand, when the value of *k* is large, this influences the increase in the computation time. This consists in choosing for an example *x* the class $y = \omega_j$ of its closest neighbor j [31]:

$$D_{KNN}(x) = \omega_j | \, j = argmin_{1 \le i \le m} d(x, x_i) \tag{4}$$

With $d(x,\ x_i)$ is a metric allowing to measure the distance between two examples. To avoid being too sensitive to noisy data, we often search for the KNN. Let us denote by $N_N(x, k, \omega_i, Z_m)$ the function returning the number of examples of the class ω_i among the *k* nearest neighbors of *x*. The selected class is the one which is the majority among these *k* neighbors, which corresponds to the following decision function:

$$D_{kNN}(x) = argmax_{\omega_i \in y} N_N(x, k, \omega_i, Z_m) \tag{5}$$

The pseudo code of KNN algorithm can be established as follows:

```
Begin
For each $example(x, c) \varepsilon A$
Calculate the distance $d(x', x)$
End
For each $x \varepsilon\ KNN(x')$ do
Count the number of occurrences of each class
End
Assign to $x'$ the most frequent class
End
```

4 Support Vectors Machine (SVM)

The SVMs are a pioneer machine learning tool suited for classification, prediction and regression. The purpose of using SVM is to find the optimal solution between two classes. When the number of samples is infinitely large one obtains the optimal solution. In particular, it has a good generalization even when the samples are few [32].

The SVM decision function can be written as follows:

$$f(x) = w_x + b = \sum_{k=1}^{m} w_k x_k = 0 \tag{6}$$

where $f(x)$ results Classifications of test data, w *weight*, b *biase*, x Kernel test data calculation [33]. In order to satisfy the constraints, the hyperplane must be definitely separated. In other words:

$f(x_k) \geq 1$ if $y_k = 1 => x_k$ be owned by to class 1 and

$f(x_k) \leq -1$ if $y_k = -1 => x_k$ be owned to class -1

The merging of these constraints gives the following result:

$$y_k(wx_k + b) \geq 1 \;\; for\; k = 1, 2, \ldots, m \tag{7}$$

In the case of non-linear separation of two classes, the variable ξ_k is introduced in Eq. (7), and it then becomes:

$$y_k(wx_k + b) \geq 1 - \xi_k \;\; for\; k = 1, 2, \ldots, m\;,\;\; and\; 0 < \xi_k \tag{8}$$

The conditions below must be satisfied by the separating hyperplane:

$$\begin{cases} y_k(wx_k + b) \geq 1 \\ min\dfrac{1}{2}\|w\|^2 \end{cases} \tag{9}$$

Obviously, $\|w\|^2 = w^T w$, minimizing the norm makes the margin maximum.

In order to get the optimal solution of the hyperplane, the Lagrangian principle has been utilized:

$$L(w, b, \propto) = \frac{1}{2}\|w\|^2 - \sum_{k=1}^{m} \alpha_k \left[y_k\left(w^T x_k + b\right) - 1\right] \tag{10}$$

$\propto$ is a set of Lagrnage coefficients $(\propto_k) > 0$

By introducing the following optimal conditions

$$\begin{cases} \dfrac{\partial L(w, b, a)}{\partial w} = 0 \\ \dfrac{\partial L(w, b, a)}{\partial b} = 0 \end{cases} \tag{11}$$

We obtain:

$$\begin{cases} w = \sum_{k=1}^{m} \propto_k x_k y_k \\ \sum_{k=1}^{m} \propto_k x_k = 0 \end{cases} \tag{12}$$

We solve the problem using the following form:

$$\begin{cases} \max L(w,b,a) \\ w = \sum_{k=1}^{m} \propto_k x_k y_k \\ \sum_{k=1}^{m} \propto_k x_k = 0 \quad \forall k, \propto_k \geq 0 \end{cases} \tag{13}$$

This is equivalent to the following equation:

$$\begin{cases} \max \sum_{k=1}^{m} \propto_k - \frac{1}{2}\sum_{k,j} \propto_k \propto_j y_k y_j \left(x_k, x_j\right) \\ \sum_{k=1}^{m} \propto_k x_k = 0 \; \forall k, \; \propto_k \geq 0 \end{cases} \tag{14}$$

We integrate the deviation variable *k* as follows:

$$y_k(wx_k + b) \geq 1 - \xi_k \quad such\ as \;\; k = 1, 2, \ldots, m \;\; \forall k, \; 0 < \xi < 1 \tag{15}$$

We obtain the following optimization problem:

$$\begin{cases} min \frac{1}{2}\|w\|^2 + C\sum_{k=1}^{m} \xi_k \\ y_k(wx_k + b) \geq 1 - \xi_k \; such\ as \;\; \forall k, \; 0 < \xi_k < 1 \;\; k = 1, 2, \ldots, m \end{cases} \tag{16}$$

where C is the margin parameter.

The lagrangian is defined as follows:

$$L(w, b, \xi, x) = \frac{1}{2}\|w\|^2 + c\sum_{k=1}^{m} \xi_k - \sum_{k=1}^{m} \propto_k \left[y_k\left(w^T x_k + b\right) - 1 + \xi_k\right. \tag{17}$$

If we apply the Karush-Kuhn-Kucker conditions which are the second order optimization conditions:

$$\begin{cases} \frac{\partial L(w, b, \xi, \propto)}{\partial w} = 0 \\ \frac{\partial L(w, b, \xi, \propto)}{\partial b} = 0 \\ \frac{\partial L(w, b, \xi, \propto)}{\partial \epsilon} = 0 \end{cases} \tag{18}$$

We obtain:

$$\begin{cases} \max \sum_{k=1}^{m} \alpha_k - \frac{1}{2}\sum_{k,j} \alpha_k \alpha_j y_k y_j \varnothing(x_k)\varnothing\left(x_j\right) \; such\ as \; \forall k, \; \alpha_k \geq 0 \\ \sum_{k=1}^{m} \propto_k x_k = 0 \end{cases} \tag{19}$$

In the SVM networks, the product $\varnothing(x_k)\varnothing\left(x_j\right)$ may be supplanted by the positive kernel function. In our case, the following Gaussian kernel function has been selected:

$$k\left(x_k, x_j\right) = \exp\left(-\gamma||x - x'||^2\right) \tag{20}$$

w and b define the position of the separating hyperplane as explained in the Fig. 1:

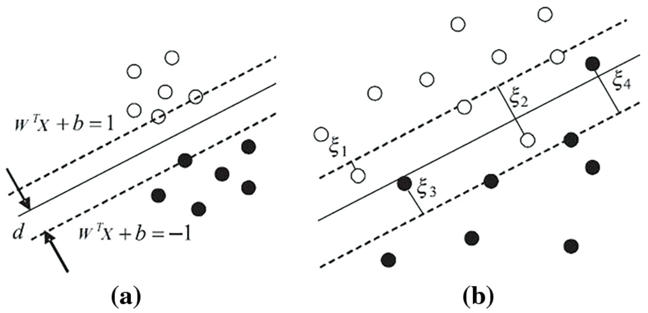

Figure 1: Separation of two classes: (a) Linear separation (b) Non-linear separation

4.1 SVM Parameters Optimization using PSO algorithm

The PSO algorithm has been developed by Kennedy and Eberhart [34] since 1995. The parameters (C,γ) of the SVM model have been optimized for the progression of the PSO algorithm, in order to increase the diagnostic precision. This algorithm was built after having had a long observation about a flock of birds and also a school of fish, while they are in search of their food. Every fish has its way between food and itself. This action is a social behavior, intelligence in swarms, leads all natural processes to find a shorter path to their walls [35].

The PSO algorithm ensures auto-selection of the SVM parameters which achieve the best diagnostic accuracy.

The optimization procedure of the SVM parameters through PSO technique is interpreted in different step as follows:

1. Initialization: N particles of populations are randomly generated by PSO, each particle with a position and a velocity in the d dimensional search space.

Moreover, the particles are named thus, $P_i = (P_{i1}, P_{i2}, \ldots, P_{id})$ position and $V_i = (V_{i1}, V_{i2}, \ldots, V_{id})$ speed, knowing that, P_i and V_i are matrices of $N * d$. (C, γ) conditioning parameter for SVM variables

2. Training: The SVM model is trained using 90% of the database, for each particle

3. Testing: 10% of the database has been reserved to test the SVM model. The evaluation criteria representing the accuracy of each particle are calculated through Eq. (21).

$$f_i = \frac{y_t - y_{fi}}{y_t} \times 100 \tag{21}$$

Where y_{fi} is the total number of false classified data, y_t is the total number of testing data.

4. Updating: each particle updates the position and speed using the following equations:

$$V_i^d(t+1) = WV_i^d(t) + C_1r_1(t)\left(pbest^d(t) - p_i^d(t)\right) - C_2r_2(t)(gbest^d(t) - p_i^d(t) \tag{22}$$

$$p_i^d(t+1) = p_i^d(t) + V_i^d(t+1) \tag{23}$$

- C_1 *and* C_2 are acceleration coefficients;
- P_i represents the position of the "i" particle;

- $V_i$represents the velocity of the "i" particle;
- r_1 and r_2 are random number in the range [0,1];
- V is the inertia weight.

5. Stop criterion: Steps 2 to 4 are duplicated until reaching the maximum number of iterations.
6. Exploitation: The SVM optimal parameters acquired are utilized for classification.

The calculation process of SVM model can be summarized by the flowing flowchart in Fig. 2:

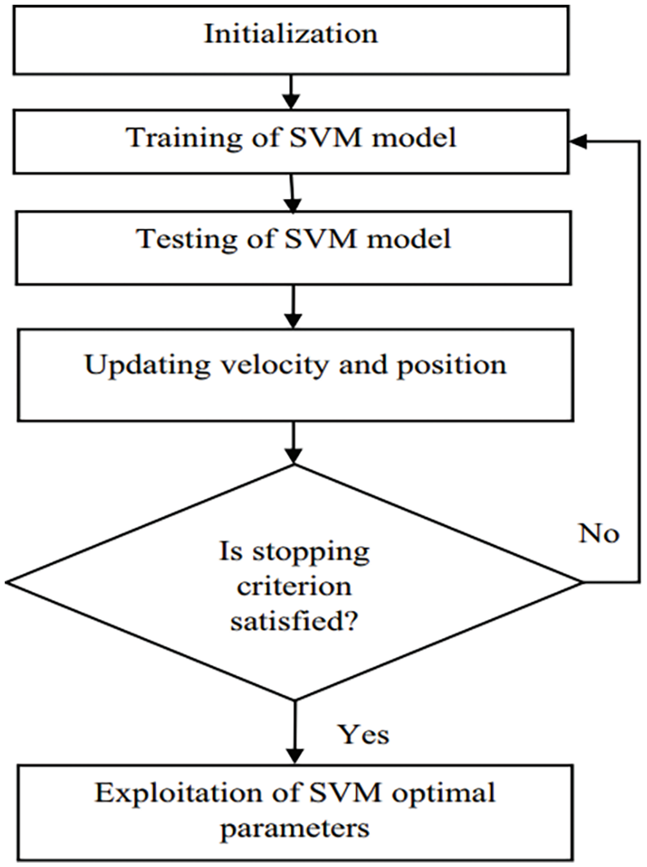
Initialization
Training of SVM model
Testing of SVM model
Updating velocity and position
Is stopping criterion satisfied?
No
Yes
Exploitation of SVM optimal parameters

Figure 2: The SVM model with the calculation process [36]

5 Cross Validation

V-fold cross validation is a procedure for ranking or comparing the robustness of a classification algorithm on a set of data. The classification algorithm is randomly distributed by a bend dataset of equal size, and each fold is used to test the induced pattern of the other folds. Evaluating the mean of the precisions of v leads the cross-validation of v-fold to better performance of the classification algorithm, implying the level of averaging is supposedly to be in the fold. Knowing that each fold contains the same number of instances [37].

6 Simulations and results

A robot manipulator arm has been considered in this work with 3 degrees of freedom, called *SCARA* as described in Section 2. The robot manipulator parameters used in the simulations are introduced as follows, in order to diagnose the manipulator faults: the moments of inertia are I1 = 0.02, I2 = 0.03 and I3 = 0.05 (rad/s), m1 = 0.5, m2 = 0.3, m3 = 0.1 Kg, the length of the links is L = 1m and the sampling time is equal to $ts = 0.01\ s$. In order to test the diagnosis of faulty joints, 7 classes have been supported on the x-axis, y-axis, z-axis, xy-axis, xz-axis, yz-axis and xyz-axis. One on the x-axis (D1), 10 additive samples faults have been applied with a 0.001 sampling step over an interval [0,0.02]. The second class (D2) on the y-axis, also 10 additive samples faults with a 0.001 have been introduced over the interval [0,0.02]. On the interval [0,0.03], the 10 additive samples faults have been injected on the z-axis and considered as the third class (D3). In the fourth class (D4), the 10 additive faults have been implanted on the xy-axis, on the x-axis over the interval [0,0.02] and along the y-axis on the interval [0,0.02]. On the interval [0,0.02] and [0,0.03] according to the x-axis and z-axis, the 10 additives faults have been inserted to create the D5 class. However, in the sixth class (D6), and in the intervals [0,0.02] and [0,0.03] interposed to the y-axis and z-axis, we apply 10 additive samples faults. The last class D7 (xyz-axis) which includes three variation intervals [0,0.02], [0,0.02] and [0,0.03], we introduce 10 additive samples faults. Thus, the output vector consists in: [D1 D2 D3 D4 D5 D6 D7]. The whole of the injected faults according to the 7 classes have been established with a database of 63000 samples.

Concerning the input vector, the composition consists of: the torque vector $\tau(\tau_x, \tau_y, \tau_z)$, position vector $q(q_x, q_y, q_z)$ and manipulator speed vector $\dot{q}\left(\dot{q}_x, \dot{q}_y, \dot{q}_z\right)$, and it will be represented by: $\left[\tau_x\ \tau_y\ \tau_y\ q_x\ q_y\ q_z\ \dot{q}_x\ \dot{q}_y\ \dot{q}_z\right]$. For classifiers, the cross validation method has been used.

The simulations have been carried out using the MATLAB software. The results are presented below.

6.1 KNN results

For both classifiers, the training and test procedures were initiated using the cross-validation method. 10-fold cross validations have been chosen and the database D is composed of : $D = [V_1\ V_2\ V_3\ V_4\ V_5\ V_6\ V_7\ V_8\ V_9\ V_{10}]$. The accuracy diagnosis is assessed 10 times and the grader is assessed on the basis of the average value. Every time, for the ith test V_i is used, and the remaining of the database is reserved for the training process. Then, all accuracies rates have been evaluated on the tenth of samples population (63 000 samples) for the testing phase and on ninth of the samples population (56 700) for the training phase. Tab. 1 explains the decomposition of the data of the 10-fold cross validation method.

Table 1: Data decomposition

Fold	Training	Testing
1	$[V_1\ V_2\ V_3\ V_4\ V_5\ V_6\ V_7\ V_8\ V_9]$	$[V_{10}]$
2	$[V_1\ V_2\ V_3\ V_4\ V_5\ V_6\ V_7\ V_8\ V_{10}]$	$[V_9]$
3	$[V_1\ V_2\ V_3\ V_4\ V_5\ V_6\ V_7\ V_9\ V_{10}]$	$[V_8]$
4	$[V_1\ V_2\ V_3\ V_4\ V_5\ V_6\ V_8\ V_9\ V_{10}]$	$[V_7]$
5	$[V_1\ V_2\ V_3\ V_4\ V_5\ V_7\ V_8\ V_9\ V_{10}]$	$[V_6]$
6	$[V_1\ V_2\ V_3\ V_4\ V_6\ V_7\ V_8\ V_9\ V_{10}]$	$[V_5]$
7	$[V_1\ V_2\ V_3\ V_5\ V_6\ V_7\ V_8\ V_9\ V_{10}]$	$[V_4]$
8	$[V_1\ V_2\ V_4\ V_5\ V_6\ V_7\ V_8\ V_9\ V_{10}]$	$[V_3]$
9	$[V_1\ V_3\ V_4\ V_5\ V_6\ V_7\ V_8\ V_9\ V_{10}]$	$[V_2]$
10	$[V_2\ V_3\ V_4\ V_5\ V_6\ V_7\ V_8\ V_9\ V_{10}]$	$[V_1]$

To obtain the highest possible diagnostic accuracy, several types of distance were used for the KNN algorithm, namely Cityblock, Cosine, Correlation and Euclidean. For the training phase, the number of neighbors k has been varied from 1 to 100 where the value corresponding to the better accuracy rate is maintained. The effect of the number of neighbors was also investigated, Fig. 3 shows clearly this influence (the first cross validation taken as example).

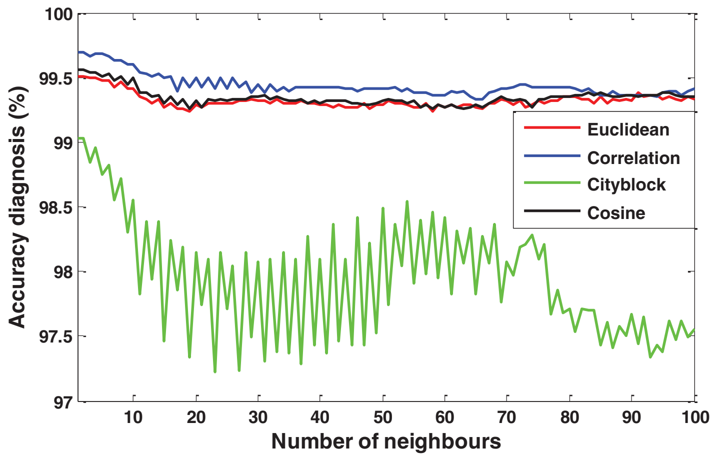

Figure 3: The impact of the distance type and number of neighbors on accuracy diagnosis

From the obtained results in Fig. 3 of the first fold validation, it becomes clear the extent of the effect of the number of neighbors on the diagnosis accuracy. An accuracy diagnosis of 99.69% has been obtained for correlation distance with k = 1, the cosine comes in second place with an accuracy of 99.55 and k = 1. Accuracy of 99.50% and 99.03% have been achieved for the Euclidian and city block respectively with k = 1 for both distances. The results of the 10 folds validation are summarized in Tab. 2. For each validation, it is clear the extent of the impact of distance type and neighbors value.

Table 2: KNN accuracy analysis

Cross Validation	Best neighbors (k)				Accuracy diagnosis (%)			
	Euclidean	Correlation	City block	Cosine	Euclidian	Correlation	City block	Cosine
1st validation	1	1	1	1	99.50	**99.69**	99.03	99.55
2nd validation	3	1	1	1	**99.82**	**99.82**	99.80	**99.82**
3rd validation	1	1	1	1	**99.90**	**99.90**	**99.90**	**99.90**
4th validation	1	1	1	1	**99.90**	**99.90**	**99.90**	**99.90**
5th validation	1	1	1	1	**99.90**	**99.90**	**99.90**	**99.90**
6th validation	1	1	3	1	**99.90**	**99.90**	**99.90**	**99.90**
7th validation	57	1	43	57	99.80	**99.88**	99.50	99.80
8th validation	75	1	94	75	97.61	**97.95**	96.84	97.71
9th validation	55	1	3	41	86.68	**87**	85.79	86.60
10th validation	1	1	1	1	62.04	**62.28**	60.44	62.25
Overall accuracy					94.50	**94.62**	94.1	94.53

The overall accuracy of each distance type is represented in Tab. 3. The results are given for the four distance types, such as, Euclidean, Correlation, City block and cosine. It was found that the best accuracy of the KNN algorithm was 94.62% with a correlation distance type against 94.53 for cosine distance type. The Euclidean and the City block distances have achieved 94.50% and 94.10% respectively.

Table 3: Overall accuracy

Distance type	Euclidean	Correlation	City block	Cosine
Accuracy diagnosis (%)	94.50	**94.62**	94.1	94.53

6.2 SVM results

The SVM parameters C and □ are adjusted using PSO. The results of the simulations show that the approach makes it possible not only to select important characteristics, but also to obtain high precision for the classification of faults. PSO affects the precision of the SVM after hybridization of SVM-PSO. The SVM-PSO method performs better than the SVM on the benchmark dataset of the reviewer's dataset. Fig. 4 shows the evolution performance of the SVM classifier during optimization process using PSO algorithm for the first validation. The PSO algorithm has been employed to the auto selection of parameters (C, γ) for the gaussian kernel. Fig. 4 proves the influence of parameters values on the accuracy results.

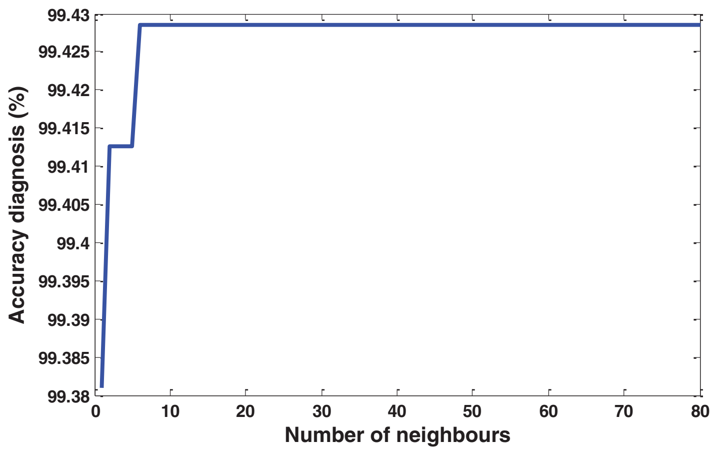

Figure 4: Performance of PSO-SVM classifiers

The results presented in Tab. 4 for cross-validation 10; in each cross-validation, the parameters (C, γ) that provided the highest possible accuracy are different. Subsequently, the PSO algorithm was helpful in adjusting the C and γ parameters.

Tab. 5 presents the dataset used in the simulations. In fact, has been finding 7 classes of faults in which the value of the fault for each input is given.

The simulation results illustrate that the PSO-SVM and KNN algorithms detect immobilization in various faults. The test result presented in Tab. 6 shows that the PSO-SVM method could diagnose errors correctly and efficiently with an accuracy of 96.95%. On the other hand, the KNN algorithm has a fault diagnosis with a precision of 94.53%. In this regard, we conclude that the PSO-SVM algorithm provides a better accuracy for the error classification, compared to the KNN algorithm.

Table 4: PSO-SVM analysis of accuracy

Validation	(C, γ)	Accuracy diagnosis (%)
1st validation	(187.62, 0.2560)	99.42
2nd validation	(268.23, 0.0555)	99.90
3rd validation	(223.73, 0.3137)	99.90
4th validation	(310.89 ,0.0213)	99.90
5th validation	(464.98, 0.1076)	99.90
6th validation	(21.64, 0.0482)	99.87
7th validation	(556.05, 0.0802)	99.90
8th validation	(23.69, 0.0581)	99.84
9th validation	(1.08, 1.5039)	96.22
10th validation	(5.37, 0.0867)	74.65
Overall accuracy		96.95

Table 5: Diagnosis result of 7 samples

N	τ_x	τ_y	τ_y	q_x	q_y	q_z	$\dot{q}_x$	$\dot{q}_y$	$\dot{q}_z$	Sensor fault	KNN	PSO-SVM
1	0,3	3e − 9	−10,5	0,0014	6e − 7	−1e − 7	2,72e − 5	0,056	0,4905	D1	D1	D1
2	0,3	1.28e − 7	−10,5	0,0126	2,43e − 5	−6.59e − 6	0,0011	0,4479	0,4906	D2	D1	D2
3	0,3	1,28e − 7	−10,5	0,0126	6e − 7	−6.59e − 6	0,0011	0,4479	0,4906	D3	D3*	D3
4	0,3	3e − 9	−10,5	0,0014	6e − 7	−1e − 7	2,72e − 5	0,056	0,4905	D4	D4	D4
5	0,3	3e − 9	−10,5	0,0014	6e − 7	−1e − 7	2,72e − 5	0,056	0,4905	D5	D5	D5
6	0,3	3e − 9	−10,5	0,0014	6e − 7	−1e − 7	2,72e − 5	0,0560	0,4905	D6	D6	D6
7	0,3	3e − 9	−10,5	0,0014	6e − 7	−1e − 7	2,72e − 5	0,0560	0,4905	D7	D7	D7

Table 6: Comparison accuracy between KNN and PSO-SVM

	KNN	PSO-SVM
Accuracy (%)	94.53	**96.95**

7 Conclusions

In this paper a comparative study was made between the PSO-SVM and KNN methods for the classification of faults in a manipulator robot arm. In addition to the classification-based approach for fault detection, a fault detection method based on a dynamic model of the interaction of the SCARA robot was also introduced. Various parameters are considered as input vectors for classifying seven classes of faults within the manipulator. The diagnostic accuracy of the two classifiers was estimated using 63,000 samples from the associated database. The classification method by SVM has a strong capability of learning the characteristics of the system (manipulator), which avoids the problems of low exemplary and poor distinction of traditional manual uprooting of the characteristics, and improves the diagnostic accuracy of the characteristics faults. In order to improve the SVM method, a PSO algorithm has been proposed in this paper to perform automatic optimization of the SVM parameters, which facilitates the efficiency of optimization of these parameters. Extensive simulations were conducted to demonstrate the efficacy of the proposed method. The numerical optimization simulations were carried out based on 7 widely applied fault classes and simulation results indicated that the proposed PSO variant has better performance in terms of search accuracy and speed of convergence. The results also demonstrate that the PSO-SVM algorithm offers a more accurate diagnosis than the KNN algorithm. The SVM algorithm may be a promising technique to diagnose defects in robotic manipulators. In the future, it is recommended that the SVM algorithm be linked to other optimization techniques in order to achieve better results.

Acknowledgement: The authors acknowledge the financial support received from Taif University Researchers Supporting Project (Number TURSP-2020/122), Taif University, Taif, Saudi Arabia.

Funding Statement: This work was supported by Taif University Researchers Supporting Project (Number TURSP-2020/122), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

[1] F. K. Cao, Y. Y. Zhuang, F. Yan, Q. F. Yang and W. Wang, “Long-term autonomous environment adaptation of mobile robots: State-of-the-art methods and prospects,” *Acta Automatica Sinica*, vol. 46, no. 2, pp. 205–221, 2020.

[2] F. Caccavale, P. Cilibrizzi, F. Pierri and L. Villani, “Actuators fault diagnosis for robot manipulators with uncertain model,” *Control Engineering Practice*, vol. 17, no. 1, pp. 146–157, 2009.

[3] M. Van, P. Franciosa and D. Ceglarek, “Fault diagnosis and fault-tolerant control of uncertain robot manipulators using high-order sliding mode,” *Mathematical Problems in Engineering*, vol. 2016, no. 6, pp. 1–14, 2016.

[4] G. Betta and A. Pietrosanto, “Instrument fault detection and isolation: State of the art and new research trends,” *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 1, pp. 100–107, 2000.

[5] V. Verma, G. Gordon, R. Simmons and S. Thrun, “Real-time fault diagnosis [robot fault diagnosis],” *IEEE Robotics & Automation Magazine*, vol. 11, no. 2, pp. 56–66, 2004.

[6] S. J. Qin and W. Li, “Detection and identification of faulty sensors in dynamic processes,” *AIChE Journal*, vol. 47, no. 7, pp. 1581–1593, 2001.

[7] S. Alag, A. Agogino and M. Morjaria, “A methodology for intelligent sensor measurement, validation, fusion, and fault detection for equipment monitoring and diagnostics,” *Ai Edam-Artificial Intelligence for Engineering Design Analysis and Manufacturing*, vol. 15, no. 4, pp. 307–320, 2001.

[8] T. Batool, S. Abbas, Y. Alhwaiti, M. Saleem, M. Ahmad *et al.,* “Intelligent model of ecosystem for smart cities using artificial neural networks,” *Intelligent Automation and Soft Computing*, vol. 30, no. 2, pp. 513–525, 2021.

[9] J. Li, “An improved K-nearest neighbor algorithm using tree structure and pruning technology,” *Intelligent Automation and Soft Computing*, vol. 25, no. 1, pp. 35–48, 2019.

[10] S. A. Ageev, A. A. Privalov, V. V. Karetnikov and A. A. Butsanets, “An Adaptive method for assessing traffic characteristics in high-speed multiservice communication networks based on a fuzzy control procedure,” *Automation and Remote Control*, vol. 82, no. 7, pp. 1222–1232, 2021.

[11] V. A. Petrushin, “Emotion recognition agents in real world,” In: *AAAI Fall Symp. on Socially Intelligent Agents: Human in the Loop*, North Falmouth, Massachusetts, USA, pp. 136–138, 2000.

[12] D. S. Lavrova and A. A. Shtyrkina, “The analysis of artificial neural network structure recovery possibilities based on the theory of graphs,” *Automatic Control and Computer Sciences*, vol. 54, no. 8, pp. 977–982, 2020.

[13] P. A. Mukhachev, T. R. Sadretdinov, D. A. Pritykin, A. B. Ivanov and S. V. Solov’ev, “Modern machine learning methods for telemetry-based spacecraft health monitoring,” *Automation and Remote Control*, vol. 82, no. 8, pp. 1293–1320, 2021.

[14] A. Mellit and S. A. Kalogirou, “Artificial intelligence techniques for photovoltaic applications: A review,” *Progress in Energy and Combustion Science*, vol. 34, no. 5, pp. 574–632, 2008.

[15] C. E. Boudjedir, D. Boukhetala and M. Bouri, "Nonlinear PD plus sliding mode control with application to a parallel delta robot," *Journal of Electrical Engineering*, vol. 69, no. 5, pp. 329–336, 2018.

[16] J. Baek, M. Jin and S. Han, "A new adaptive sliding-mode control scheme for application to robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3628–3637, 2016.

[17] M. Galicki, "Finite-time control of robotic manipulators," *Automatica*, vol. 51, no. 5, pp. 49–54, 2015.

[18] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki and J. Schröder, Analysis based on components and architecture. In: *Diagnosis and Fault-tolerant Control*, Second. ed., Berlin Heidberg: Springer- Verlag, 2006.

[19] R. J. Patton, "Fault-tolerant control: the 1997 situation," *IFAC Proceedings*, vol. 30, no. 18, pp. 1029–1051, 1997.

[20] D. Maincer, M. Mansour, A. Hamache, C. Boudjedir and M. Bounabi, "Switched time delay control based on artificial neural network for fault detection and compensation in robot manipulators," *SN Applied Sciences*, vol. 3, no. 4, pp. 1–13, 2021.

[21] N. Stroppa and F. Yvon, "Analogical learning and formal proportions: Definitions and methodological issues, ENST Paris Report," 2005.

[22] M. A. Khan, M. Ali, M. Shah, T. Mahmood, M. Ahmad *et al.,* "Machine learning-based detection and classification of walnut fungi diseases," *Intelligent Automation and Soft Computing*, vol. 30, no. 3, pp. 771–785, 2021.

[23] M. R. Huq, A. Ali and A. Rahman, "Sentiment analysis on twitter data using KNN and SVM," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, pp. 19–25, 2017.

[24] O. Kherif, Y. Benmahamed, M. Teguar, A. Boubakeur and S. S. Ghoneim, "Accuracy improvement of power transformer faults diagnostic using KNN classifier with decision tree principle," *IEEE Access*, vol. 9, pp. 81693–81701, 2021.

[25] S. Shamshirband, A. Mosavi, T. Rabczuk, N. Nabipour and K. W. Chau, "Prediction of significant wave height; comparison between nested grid numerical model, and machine learning models of artificial neural networks, extreme learning and support vector machines," *Engineering Applications of Computational Fluid Mechanics*, vol. 14, no. 1, pp. 805–817, 2020.

[26] S. Sanner and E. Abbasnejad, "Symbolic variable elimination for discrete and continuous graphical models," in *Twenty-Sixth AAAI Conf. on Artificial Intelligence*, Toronto, Ontario, Canada, 2012.

[27] W. Wang, B. Zhang, K. Wu, S. A. Chepinskiy, A. A. Zhilenkov *et al.,* "A visual terrain classification method for mobile robots' navigation based on convolutional neural network and support vector machine," *Transactions of the Institute of Measurement and Control*, vol. 44, no. 4, pp. 744–753, 2021.

[28] T. C. Hsia and L. S. Gao, "Robot manipulator control using decentralized linear time-invariant time-delayed joint controllers," in *Proc of the IEEE Conf. on International Conference on Robotics and Automation*, Cincinnati, OH, USA, pp. 2070–2075, 1990.

[29] S. S. Istia and H. D. Purnomo, "Sentiment analysis of law enforcement performance using support vector machine and K-nearest neighbor," in *Proc. of the IEEE Conf. on 3rd Int. Conf. on Information Technology, Information System and Electrical Engineering (ICITISEE)*, Yogyakarta, Indonesia, pp. 84–89, 2018.

[30] M. R. Huq, A. Ali and A. Rahman, "Sentiment analysis on twitter data using KNN and SVM," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, pp. 19–25, 2017.

[31] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[32] S. Y. Liang and J. H. Lv, "Least squares support vector machine for fault diagnosis optimization," in *Applied Mechanics and Materials*. Trans Tech Publications Ltd, Switzerland, pp. 505–508, 2013.

[33] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[34] J. F. Schutte and A. A. Groenwold, "A study of global optimization using particle swarms," *Journal of Global Optimization*, vol. 31, no. 1, pp. 93–108, 2005.

[35] Q. Bai, "Analysis of particle swarm optimization algorithm," *Computer and Information Science*, vol. 3, no. 1, pp. 180, 2010.

[36] Y. Benmahamed, M. Teguar and A. Boubakeur, "Application of SVM and KNN to duval pentagon 1 for transformer oil diagnosis," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 24, no. 6, pp. 3443–3451, 2017.

[37] T. T. Wong, "Performance evaluation of classification algorithms by K-fold and leave-one-out cross validation," *Pattern Recognition*, vol. 48, no. 9, pp. 2839–2846, 2015.