

Article

Metaheuristics and Support Vector Data Description for Fault Detection in Industrial Processes

Jesús Alejandro Navarro-Acosta ¹, Irma D. García-Calvillo ¹  and Vanesa Avalos-Gaytán ¹
and Edgar O. Reséndiz-Flores ^{2,*} 

¹ Research Center on Applied Mathematics, Autonomous University of Coahuila, Prolongación David Berlanga, Edificio S, Primer Piso, Camporredondo, Saltillo 25115, Mexico; alejandro.navarro@uadec.edu.mx (J.A.N.-A.); irma.garcia@uadec.edu.mx (I.D.G.-C.); vanesaavalos@uadec.edu.mx (V.A.-G.)

² Division of Postgraduate Studies and Research, Tecnológico Nacional de México/IT de Saltillo. Blvd. Venustiano Carranza 2400, Colonia Tecnológico, Saltillo 25280, Mexico

* Correspondence: eresendiz@itsaltillo.edu.mx; Tel.: +52-844-4389-500

Received: 28 November 2020; Accepted: 16 December 2020; Published: 21 December 2020



Abstract: In this study, a system for faults detection using a combination of Support Vector Data Description (SVDD) with metaheuristic algorithms is presented. The presented approach is applied to a real industrial process where the set of measured faults is scarce. The original contribution in this work is the industrial context of application and the comparison of swarm intelligence algorithms to optimize the SVDD hyper-parameters. Four recent metaheuristics are compared hereby to solve the corresponding optimization problem in an efficient manner. These optimization techniques are then implemented for fault detection in a multivariate industrial process with non-balanced data. The obtained numerical results seem to be promising when the considered optimization techniques are combined with SVDD. In particular, the Spotted Hyena algorithm outperforms other metaheuristics reaching values of F1 score near 100% in fault detection.

Keywords: support vector data description; metaheuristics; fault detection; one class classification

1. Introduction

Currently, machine learning and nature-inspired algorithms are being applied in several research fields to obtain optimal results. Some real applications include medical diagnosis based on the patient's symptoms [1], fraud detection in economic transactions [2], identification of patterns of investment in order to buy/sell in a more efficient manner [3], image detection to predict city traffic, machine failure and the design of autonomous vehicles, among others [4]. There are some previous studies regarding fault detection and fault diagnosis based on Dynamic Weight Principal Component Analysis (PCA) [5], Principal polynomial analysis [6], PCA and a Bayesian network [7], deep convolutional neural network [8], Hidden Markov Model and Bayesian Network [9], among others. However, statistical assumptions about the distribution of the process data should be made in some of these approaches. Support Vector Data Description (SVDD) [10] and Artificial Neural Networks (ANN) algorithms share the same concept using the linear learning model for pattern recognition. ANN tries to converge to a local minimum using the gradient descent learning algorithm and suffers from overfitting problems. On the other hand, SVDD tends to find a global solution during training since the complexity of the model has been taken into account as a structural risk in SVDD formation. ANN minimizes only empirical risk learned from training samples and SVDD considers not only the empirical risk but the structure risk. Thus, SVDD training results show better generalization capability than those obtained with ANN.

In industry, machine learning classifiers are implemented to focus on faults detection. In [11,12], reviews of machine learning applications in the manufacturing industry are presented. The use of artificial neural networks in the modeling and optimization of processes is emphasized as well as the application of SVM for quality assessment in industries. Another application of SVM for the early failure prediction in the oil and gas industry is shown in [13]. In [14], a monitoring platform using Artificial Neural Network and the Support Vector Machine is proposed and applied to the prediction of the performance of aeronautical engines and health diagnosis. Nevertheless, conventional algorithms are also created to perform two-class or multi-class classification tasks. Because of this, data containing all class of information on the processes are required. Recently, machine learning algorithms have been used in fault detection, but there is sometimes the downside of having more information available from one of the kind. For example, if a relatively new machine is in operation, it is very likely that only the data corresponding to normal operation are available. In this study, a method for fault detection is proposed which is able to deal with processes or machines where observations or data about the faults are scarce for the training phase by using the one class classifier known as Support Vector Data Description (SVDD) [10]. The choice of hyper-parameters is one of the most important features for SVDD. These parameters are associated with the hyper-sphere as well as with the kernel function chosen for the classification. In this study, these hyper-parameters are simultaneously optimized using metaheuristic techniques. Furthermore, a comparison among four optimization techniques is reported in order to evaluate the quality of the obtained results. The results are satisfactory when applied to a process of fault detection at low computational cost.

There are some studies covering different SVDD algorithm applications [15–17]—most of them optimizing the hyper-parameters using approaches like grid search, which is computationally expensive. To obtain these parameters in a more efficient manner, some authors have considered metaheuristic algorithms. Particle swarm optimization (PSO) and genetic algorithms can be often found in literature. Ref. [18] presents a brief general description about hyper-parameter optimization in Support Vector Machines (SVM). In this context, the ant colony algorithm is chosen in [19] for feature selection and parameter optimization in SVM for fault diagnosis. Ref. [20] used genetic algorithms to optimize parameters corresponding to the kernel function. Optimization of the hyper-sphere parameters together with those associated with the Gaussian radial base kernel function are studied by [21] using grid search. Ref. [22] obtained optimal results by the combination of grid search and PSO.

In the last few decades, researchers have developed several nature-inspired optimization algorithms that mimic some biological behaviors or physical phenomena. Techniques based on swarm intelligence mimic the socially intelligent behavior of groups of species. Search algorithms start with a group of randomly generated solutions generally naming a population evolving throughout successive generations and promote the population improvement throughout the iterations. In this study, a performance comparison is presented corresponding to four of these metaheuristics for calculation of the hyper-parameters. Besides PSO, which has been widely used for these purposes, the considered algorithms in this study have demonstrated their efficiency in solving optimization problems applied to engineering. Furthermore, the performance is tested for the Spotted Hyena Optimization (SHO) algorithm, Krill herd (KH) algorithm, and Squirrel Search Algorithm (SSA).

The SHO is a recent metaheuristic based on swarms which mimics the social behavior of these animals in nature. Among the vast amount of metaheuristics, SHO has shown great advantages in the exploration of diverse search spaces compared to other state-of-the-art approaches [23]. SHO has been implemented to solve a wide range of engineering problems like prediction of materials resistance during cutting processes [24]. SHO is used in combination with neural networks to improve the prediction ability of these algorithms. In [25], two complex engineering problems with restrictions like the design of bar armor and the design of multiple disk clutch brakes are solved. The results show the efficiency of SHO in solving these problems compared to other optimization algorithms like PSO and ACO. It showed its applicability in environments of a high dimension with a low computational cost. Ref. [26] implements SHO for design optimization in aerodynamic surface and optical buffer

problems where the obtained numerical results are then compared with algorithms like Grey Wolf Optimizer, Genetic Algorithm, and PSO, among others. Likewise, the binary version of SHO has been developed and used to build wrapping approaches for feature selection in different sets of UCI data [27]. Moreover, SHO has been hybridized with other algorithms like PSO to improve its abilities to solve various engineering problems [28]. Recently, Ref. [29] implements an SHO version able to deal with multi-objective problems for the prediction of characteristics in gene selection through its combination with machine learning algorithms like SVM.

The KH algorithm is a recent nature-based algorithm inspired by the individual krill herding behavior. This algorithm was introduced in [30]. The KH algorithm works to achieve the minimum distance between individual krill and the nearest food. This algorithm has been successfully used in solving many problems of numerical optimization, electric and power system problems, text grouping, breast cancer detection, and the training of neural networks [31–33]. More recently, the possibility of applying this algorithm for the grouping of text documents is studied in [34], while the same problem is addressed in [35] using a hybrid algorithm based on KH. The grouping based on the KH algorithm is proposed in [36] for the network of wireless sensors. A method for diagnosing bearing failure based on KH and a kernel extreme learning machine is proposed in [37]. An improvement of KH applied to fault diagnosis with a Support Vector Machine to solve the power transformers' fault diagnosis problem based on the analysis of dispersed gases is presented in [38].

The SSA is a metaheuristic approach based on the behavior of flying squirrels which are a diversified nocturnal tree rodent group that are highly adjusted to gliding locomotion. The SSA mimics the dynamic food search behavior of flying squirrels and their effective form of locomotion, known as gliding, which is a very effective mechanism for traveling long distances. The algorithm has been recently proposed [39,40]. Moreover, it has proven its good performance in some applications, for example in [41], is applied for the optimization of a backpropagation artificial neural network (BPNN) using a multi-objective method based on SSA to optimize the main parameters of a continuous galvanization process for advanced DP Steels. In [42], SSA is used to optimize a complex problem where combined heat and energy distribution for various regions is modeled integrating renewable energy sources. In [43], a hybrid algorithm based on the combination of SSA and the optimization of invasive weed is proposed. This algorithm is combined with the Support Vector Machine and the deterministic maximum likelihood algorithm to perform the classification of air quality levels. Recently, a Chaotic SSA variant for optimum programming of multiple tasks in an infrastructure cloud environment as service is reported in [44].

The main contribution of this paper consists of a fault detection system using a combination of SVDD with some optimization methods like SHO, KH, SSA, and the well-known PSO. The effectiveness of the used different metaheuristics, for the parameters optimization corresponding to the hyper-sphere as well as those associated with the Gaussian radial basis kernel used in SVDD, is compared. As it will be shown, promising results are obtained when the mentioned swarm intelligence algorithms are combined with SVDD and applied to fault detection in a real industrial problem.

The rest of the paper is organized as follows: theoretical background is presented in Section 2, the proposed methodology for fault detection is described in Section 3, and this is followed by the industrial application section and finally some conclusions and future work are discussed in the last section.

2. Theoretical Background

2.1. Support Vector Data Description

Tax and Duin [10] proposed the SVDD classification method which determines a close boundary around the data set for a given class: a hyper-sphere characterized by a center and a radius $R \geq 0$ that defines a separation between the inner region with high data density and the outer region with low density. The data that lie right at the limit of the hyper-sphere are called the support vectors while

those outside are the outliers. Let $\{\mathbf{x}_i : i = 1, 2, 3, \dots, N\}$ be a column vector set and $\mathbf{x}^2 = \mathbf{x} \cdot \mathbf{x}$ the training set for which a description must be specified and assume that x_i 's show variances in all given directions. The data set delimitation of the inner region of the hyper-sphere will be minimized with an error function which minimizes the possibility of accepting outliers and such a function is defined as:

$$\begin{aligned} \min F(R, \mathbf{a}, \xi_i) &= R^2 + C \sum_i \xi_i \\ \text{s.t. } \|\mathbf{x}_i - \mathbf{a}\|^2 &\leq R^2 + \xi_i, \xi_i \geq 0 \forall i \end{aligned} \tag{1}$$

where ξ_i are the slack variables that will penalize the largest distances, and C is a control parameter of the trade-off between volume of the hyper-sphere and the errors [45]. Applying the Lagrange multipliers method, the following equations arise:

$$\begin{aligned} L(R, \mathbf{a}, \alpha_i, \gamma_i, \xi_i) &= R^2 + C \sum_i \xi_i - \sum_i \alpha_i \\ &\left\{ R^2 + \xi_i - \left(\|\mathbf{x}_i\|^2 - 2\mathbf{a} \cdot \mathbf{x}_i + \|\mathbf{a}\|^2 \right) \right\} - \sum_i \gamma_i \xi_i \end{aligned} \tag{2}$$

with $\alpha_i \geq 0$ as Lagrangian multipliers and $\gamma_i > 0$.

The dual formulation of the equations in (2) can be obtained by solving the KKT conditions, which reads

$$\begin{aligned} \min \quad & \sum_i \alpha_i (\mathbf{x}_i, \mathbf{x}_i) - \sum_i \sum_j \alpha_i \alpha_j (\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & 0 \leq \alpha \leq C \end{aligned} \tag{3}$$

In case a given \mathbf{x}_i satisfies $\|\mathbf{x}_i - \mathbf{a}\|^2 < R^2 + \xi_i$, then $\alpha_i = 0$; otherwise, when \mathbf{x}_i satisfies $\|\mathbf{x}_i - \mathbf{a}\|^2 = R^2 + \xi_i$, the corresponding Lagrange multiplier α_i is strictly greater than zero.

The vectors \mathbf{x}_i corresponding to $\alpha_i > 0$ represent the set of vectors necessary to characterize a set of data, and this set of vectors can be called support vectors of the description [46]. When there is a new vector \mathbf{z} , the distance to the center of the sphere can be computed. If this distance is smaller than R, \mathbf{z} is accepted as a new vector in the description of the data, that is,

$$\|\mathbf{z} - \mathbf{a}\|^2 = (\mathbf{z} \cdot \mathbf{z}) - 2 \sum_{i=1}^l \alpha_i (\mathbf{z} \cdot \mathbf{x}_i) + \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \leq R^2 \tag{4}$$

Note that, if the inner product in Equation (2) is replaced with a kernel function $K(\mathbf{x}_i, \mathbf{x}_i)$ a description for nonlinear data sets can then be obtained. In this manner, data are mapped to a higher dimension feature space by means of the kernel function, and this makes the nonlinear data separable [47]. Thus, the problem can be reformulated as follows:

$$\begin{aligned} \min \quad & \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_i \sum_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & 0 \leq \alpha \leq C \end{aligned} \tag{5}$$

where α_i remains as the Lagrange multipliers and $K(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function used as a functional mapping.

2.2. Spotted Hyena Optimizer (SHO)

Proposed by Dhiman and Kumar [48], SHO is a recent optimization technique that mimics the behavior of spotted hyenas when hunting. Spotted hyenas are social animals, and they hunt for prey by means of trusted friends groups and through their great ability to recognize their prey. These groups can be of up to 100 hyenas, and this is why the hunting method tends to be very effective and gets results in short periods of time. The main stages of this algorithm include searching, surrounding, and attacking the prey, in addition to other spotted hyena-seeking behaviors. In SHO, there is a search agent leader, and it is assumed that it knows the location of the prey. In this way, the other agents update their position to form friend groups around the leader. Next, the mathematical models corresponding to the mentioned several stages of this algorithm are described.

2.2.1. Encircling Prey

In this stage, the best current potential solution is considered like the prey. In that way, the other hyenas update their position around it. The mathematical model corresponding to this behavior is as follows:

$$\vec{D}_h = \left| \vec{B} \cdot \vec{P}_p(x) - \vec{P}(x) \right| \tag{6}$$

$$\vec{P}(x+1) = \vec{P}_p(x) - \vec{E} \cdot \vec{D}_h \tag{7}$$

where \vec{D}_h determines the distance between the prey and the hyena, x shows the current iteration, \vec{B} , and \vec{E} are the coefficient vectors. \vec{P}_p is the position vector of the prey while \vec{P} is the position vector of the spotted hyena. The vectors \vec{B} and \vec{E} are calculated as follows:

$$\vec{B} = 2 \cdot r \cdot \vec{d}_1 \tag{8}$$

$$\vec{E} = 2 \vec{h} \cdot r \cdot \vec{d}_2 - \vec{h} \tag{9}$$

$$\vec{h} = 5 - (\text{iteration} * (5/\text{max}_{iteration})) \tag{10}$$

where \vec{h} decreases linearly from 5 to 0 in the course of the highest number of iterations. $r \vec{d}_1$ and $r \vec{d}_2$ are random vectors in $[0, 1]$.

2.2.2. Hunting

Given that spotted hyenas hunt in “trusted friend” groups, the searching agents must form conglomerates around the best agent. The following equations model such behavior,

$$\vec{D}_h = \left| \vec{B} \cdot \vec{P}_h - \vec{P}_k \right| \tag{11}$$

$$\vec{P}_k = \vec{P}_h - \vec{E} \cdot \vec{D}_h \tag{12}$$

$$\vec{C}_h = \vec{P}_k + \vec{P}_{k+1} + \dots + \vec{P}_{k+N} \tag{13}$$

where \vec{P}_h defines the position of the first best hyena, \vec{P}_k shows the position of the rest of the hyenas, and N refers to the number of hyenas calculated as follows:

$$N = \text{Count}_{nos} \left(\vec{P}_h, \vec{P}_{h+1}, \dots, (\vec{P}_h + \vec{M}) \right) \tag{14}$$

similarly, \vec{M} is a random vector in $[0, 1]$.

2.2.3. Attacking the Prey

The mathematical formulation for attacking the prey reads

$$\vec{P}(x + 1) = \frac{\vec{C}_h}{N} \tag{15}$$

where $\vec{P}(x + 1)$ saves the best solution and updates the positions of other searching agents according to the position of the best searching agent.

2.2.4. Searching for Prey (Exploration)

The vector \vec{B} previously defined provides random values for the exploration during all iterations. Therefore, this mechanism effectively allows for avoiding local optima even in the final iterations.

2.3. Krill Herd Algorithm (KH)

The KH algorithm is inspired by the simulation of small crustaceans (Krill) behavior which live underwater. These crustaceans have the ability to form large swarms to avoid predators. The fitness function in the KH algorithm used to solve global optimization problems is based on the density of the swarm and the location of the food. Each krill migrates toward the area of highest density and at the same time continues to search for the places that contain the most food. Increasing density and foraging are used as a means to bring krill to global optimum levels at the end.

During the moving process, each krill moves towards the best option based on three essential movements:

- (i) movement generated by other krill;
- (ii) food search activity;
- (iii) physical diffusion.

The equation describing the krill moving process is as follows:

$$\frac{dX_i}{dt} = N_i + F_i + D_i,$$

where N_i is the motion produced by other krill, F_i is the food search motion, and D_i is the random diffusion of the i th krill individual.

The direction of induced motion α_i is decided by the following parts: target effect, local effect, and a repulsive effect. For a krill individual, this movement can be defined as

$$N_i^{new} = N_i^{max} \alpha_i + \omega_n N_i^{old}$$

where N_i^{new} , ω_n , and N_i^{old} denote the maximum induced speed, the inertia weight, and the last motion, respectively.

The food searching motion is influenced by two components: the food location and the previous experience about food location. For the i th krill, this motion can be expressed as follows:

$$F_i = V_f \beta_i + \omega_f F_i^{old}$$

where V_f is the feeding speed, ω_f is the inertia weight, F_i^{old} is the last feeding motion, and

$$\beta_i = \beta_i^{food} + \beta_i^{best}$$

The physical diffusion is essentially a random process. This motion can be computed based on a maximum diffusion speed D^{\max} and a random directional vector δ as follows:

$$D_i = D^{\max} \delta$$

The position in KH from t to $t + \Delta t$ is given by the following equation:

$$X_i(t + \Delta t) = X_i(t) + \Delta t \frac{dX_i}{dt}$$

The interested reader is referred to [30] for more detailed information about the KH algorithm.

2.4. Squirrel Search Algorithm SSA

Recently, Ref. [39] proposed an innovative nature-inspired algorithm for optimization, the Squirrel Search Algorithm, which has been very efficient in solving unconstrained numerical optimization problems. The algorithm mimics the strategies of flying squirrels in searching for food sources and escaping predators. A summer and winter phases are considered since the motion dynamics are different depending on the season. This strategy allows it to escape from local minima, thus raising the likelihood of reaching the global optimum. The algorithm considers a certain number of flying squirrels in a forest. It is assumed that each squirrel is located on a tree. Each squirrel searches for food by gliding among trees looking for the best food source and there are three types of trees: normal tree (no food), oak tree (acorn nuts food source), and hickory tree (hickory nuts food source). It is supposed that there is a population of N flying squirrels in the forest, one at hickory tree, N_{fs} at acorn trees, and the rest ($1 \leq N_{fs} \leq N$) at normal trees. Each squirrel is represented by a vector with D components corresponding to the dimension of the problem. Initially, the flying squirrels are in a random position to start the algorithm and the location of the squirrels can be represented by the following expression:

$$FS_i = FS_L + rand(1, D) \times (FS_U - FS_L)$$

Since each row represents one squirrel, this matrix can be initialized in a random manner with a uniform distribution between $(0, 1)$ with the lower and upper dimensions of each squirrel as FS_U and FS_L , respectively. Considering the full matrix, then the fitness evaluation corresponding to the location for each squirrel gives a fitness vector with the value of the objective function. This vector is arranged in ascending order in order to identify the best value associated with the best food source (hickory tree F_h), another food source (acorn tree F_a), and the squirrels in normal trees F_n , in such a manner that each squirrel can be identified.

Considering the case when foraging squirrels do not run into a predator, flying squirrels then look for better food sources in the forest, which implies that F_h remains unchanged. The destination of F_a is F_h and the destination of F_n is random between F_a or F_h . In the case when they find a predator, they are forced to seek and find shelter in a random location. Their behavior can be mathematically described as follows:

Case 1. The flying squirrels in acorn trees move to hickory trees, according to the following equation:

$$FS_i^{t+1} = \begin{cases} FS_i^t + d_g \times G_c \times (F_h^t - FS_i^t) & \text{if } r \geq P_{dp} \\ \text{random location,} & \text{otherwise} \end{cases}$$

Case 2. Some of the flying squirrels on normal trees move to the acorn trees looking for better food and some that have already been fed move to hickory trees in order to store food. The new locations read:

$$FS_i^{t+1} = \begin{cases} FS_i^t + d_g \times G_c \times (F_a^t - FS_i^t) & \text{if } r \geq P_{dp} \\ \text{random location} & \text{otherwise} \end{cases}$$

where $r \in (0, 1)$ is a random number, P_{dp} represents the predator appearance probability, t is the current iteration, G_c a constant, and d_g is the gliding distance. The detailed calculation of these parameters are introduced in [39].

The season changes that help the algorithm to escape from local optima are considered by calculating the season constant as follows:

$$S_c^t = \sqrt{\sum_{k=1}^D (F_{ai,k}^t - F_{h,k}^t)^2} \quad i = 1, 2, \dots, N_{fs}$$

$$S_{\min} = \frac{10e^{-6}}{(365)^{t/(T/2.5)}}$$

where T is the maximum number of iterations and c is the current iteration. Moreover, the condition is verified if $S_c^t < S_{\min}$. If this happens, the flying squirrels are relocated according to the following equation:

$$FS_i^{t+1} = FS_L + \text{Lévy}(n) \times (FS_U - FS_L)$$

All parameters suggested in [39] have been used.

2.5. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population search algorithm based on the simulation of the social behavior of birds, bees, or a school of fishes [49]. In this stochastic search technique, a multidimensional vector represents a particle in a multidimensional search space. These particles move toward the next position depending on the velocity vector associated with them. The velocity is updated based on the current velocity and the best position it has explored so far. This algorithm used the global best solution concept to obtain the optimal solution. At each iteration, the global best solution is recorded and updated [50].

Let $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{id})^t$ be the i -th particle of the swarm in a d -dimensional search space with a corresponding velocity $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{id})^t$. Thus, the equations that conduct the particle's movements read

$$\mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + c_1\phi_1(\mathbf{p}_{ibest} - \mathbf{y}_i(t)) + c_2\phi_2(\mathbf{p}_{gbest} - \mathbf{y}_i(t))$$

$$\mathbf{y}_i(t+1) = \mathbf{y}_i(t) + \mathbf{v}_i(t+1)$$

where c_1 and c_2 are acceleration coefficients, and ϕ_1 and ϕ_2 are random variables with uniform distribution in $[0, 1]$. \mathbf{p}_{ibest} and \mathbf{p}_{gbest} are the best local and global particle positions so far, respectively. w denotes the inertia weight which shows the effect of previous velocity vector on the new vector.

3. Methodology for Fault Detection

The methodology presented in this study consists of pre-processing of data, that is, cleaning and structuring data to obtain a matrix of m observations with n process variables. Once this is done, the different metaheuristics are implemented to optimize the training of a one-class classifier (SVDD), that is, the hyper-parameters are optimally found that improve the abilities of SVDD for faults detection in industrial processes where there is not much faulty information available. In particular, advantages of the exploration and exploitation of metaheuristic algorithms are taken in order to tune the hyper-parameters C in SVDD and s for the RBF kernel (16). Once the SVDD training is optimized,

it will be ready to monitor new data and detect possible faults in the multivariate industrial process. Figure 1 shows the process of the described methodology:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2s^2\right) \tag{16}$$

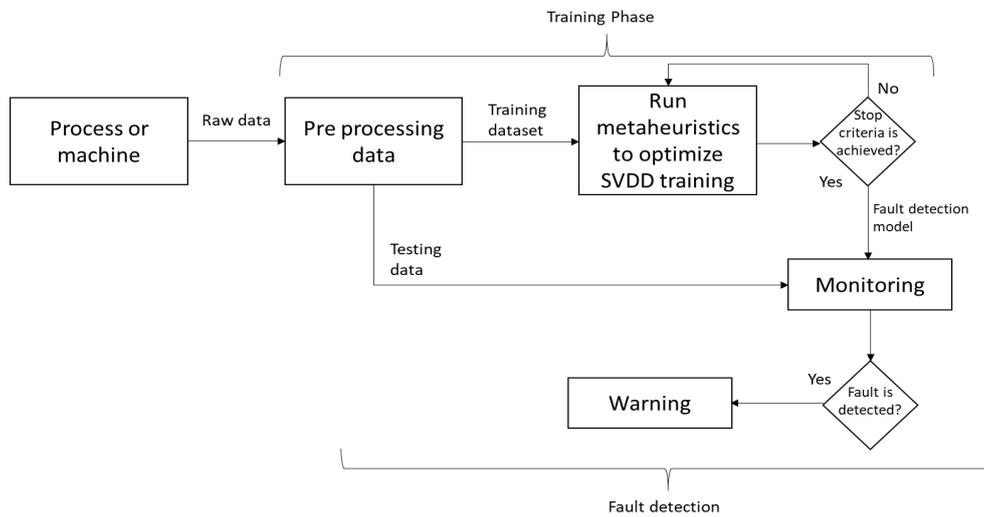


Figure 1. Fault detection system.

With the purpose of performance comparison during the optimization and training processes of SVDD, some metaheuristic algorithms such as SHO, KH, SSA, and PSO are implemented which show great efficiency in solving engineering problems. Thus, four approaches arise and need to be compared, that is: (1) SHO-SVDD, (2) KH-SVDD, (3) SSA-SVDD, and (4) PSO-SVDD. The metric under consideration is the well known F1 score (17) due to the fact that this metric takes into account precision and sensitivity. Such metric is computed based on the true and false positives (TP and FP) and negatives (TN and FN) (18) and (19). Moreover, this is highly relevant in the detection of faults in industrial processes. The F1 score value is found between 0 and 1, that is, a value of 1 means that the algorithm can detect the faults without errors. In this sense, the possible solutions generated by the different metaheuristics (particles, group of hyenas, etc.) for both hyper-parameters (C and s) are used in the training and testing phases of SVDD yielding several values of F1 score. Therefore, the different metaheuristics will adjust their solutions through the iterations in order to maximize the F1 score value:

$$F1 \text{ score} = \frac{2 \cdot Re \cdot Pr}{Re + Pr} \tag{17}$$

$$Pr = \frac{TP}{TP+FP} \tag{18}$$

$$Re = \frac{TP}{TP+FN} \tag{19}$$

4. Industrial Application

The four approaches described above are applied to the injection moulding process of car pedals in a local automotive industry. The implementation is carry out in a plastic injection machine with the capability to produce four pieces per cycle and save the relevant information in each run i.e., the parameters used to produce such pieces are automatically stored in a database. Subsequently, the pieces are classified as good or bad through several quality tests carry out by the technical staff. The injection moulding process involves 36 variables listed in Table 1. This data set was selected by the expert personnel of the process and consists of variables of diverse nature like temperatures, pressure,

time, etc., corresponding to several machine operational features such as heating power, mold position, and injection time. Since this machine just recently started operating, the primary data provided by the company include 154 observations of normal operation mode. Once the data are cleaned up and structured, the training of the one-class classifier takes place. As soon as the algorithm is trained in order to determine the system's capability, 64 new observations provided by the company are tested. This new data set includes observations corresponding to machine faults, that is, measurements where the injection process does not meet the quality standards. Then, the F1 score metric is calculated and the metaheuristics tune the SVDD hyper-parameters in order to achieve a better F1 score.

Table 1. Variables description.

	Description
xf_1	[F°], Nozzle 1
xf_2	[F°], Nozzle 2
xf_3	[Percent], Heating power zone 1
xf_4	[Percent], Heating power zone 2
xf_5	[Percent], Heating power zone 3
xf_6	[Percent], Heating power zone 4
xf_7	[Percent], Heating power zone 5
xf_8	[Percent], Heating power zone 6
xf_9	[in], Mold position value
xf_{10}	[in], Opening run
xf_{11}	[US ton], Closing force peak value
xf_{12}	[US ton], Closing force real value
xf_{13}	[s], Mold protection time
xf_{14}	[F°], Oil temperature
xf_{15}	[F°], Traverse
xf_{16}	[s], Cooling time
xf_{17}	[psi], Backpressure
xf_{18}	[in ³], Volume end screw holding pressure
xf_{19}	[psi], Holding pressure
xf_{20}	[in ³ /s], Dosage power
xf_{21}	[psi], Pressure at Switchover
xf_{22}	[s], Cycle time
xf_{23}	[lbf-ft], Mean spin
xf_{24}	[lbf-ft], Peak value at spin
xf_{25}	[psi], Specific injection pressure
xf_{26}	[in ³], Dosage volume
xf_{27}	[in ³], Injection volume
xf_{28}	[s], Dosing time
xf_{29}	[s], Injection time
xf_{30}	[F°], Cylinder zone 1
xf_{31}	[F°], Cylinder zone 2
xf_{32}	[F°], Cylinder zone 3
xf_{33}	[F°], Cylinder zone 4
xf_{34}	[ft/s], Revolutions
xf_{35}	[Wh], Injection work
xf_{36}	[in ³], Switching volume

The parameters of the different metaheuristics were selected according to their authors and the corresponding values are specified in Table 2. The number of iterations was selected in experimental testing where it was determined as the maximum amount of iterations that produced changes in the results. The search range is from 0.001 to 600 for the hyper-parameters. Table 3 shows the different values and descriptive statistics of the F1 score metric and the computational times corresponding to 30 runs using each of the four approaches. As it can be seen, the SHO algorithm reaches the highest mean value of F1 score (0.9702) with a small variability i.e., standard deviation (std) equal to 0.0074.

Furthermore, this approach presents the highest F1 score values in 80% of the experiments while SSA, KH, and PSO achieve similar performance only in 16, 13, and 0%, respectively. On the other hand, SHO presents the least computational time in the 100% of the runs achieving execution times two and even three times less than SSA and PSO. In addition, SHO presents the minimum variability with respect to the remaining approaches. In order to confirm the statistical significance of the results, a Mardia test is first performed to determine if the data present multivariate normal distribution. Since the data set is not normally distributed, the non-parametric ANOVA test of the Kruskal–Wallis is used. Figures 2 and 3 show the corresponding box plots for the implemented approaches. As it can be seen in Figure 2, SHO reaches the highest median F1 score value with low variability. On the other hand, the remaining approaches present the equal median F1 score values. However, KH shows the highest variability on its results. Although SSA sometimes yields similar results than those of SHO, they are so sporadic throughout the runs which is the reason for the boxplot to show these values as outliers. Figure 3 clearly depicts that SHO achieves the best computational times since the highest outliers of SHO are below of the lowest valued reported by KH. Tables 4 and 5 present the numerical results of the ANOVA test where for both cases (F1 score and computational time) the p -value is much lower than the significance level of 0.05. Thus, there is a statistical significance of the difference between the means of the results obtained by the four approaches. On the other hand, the high error in the sum of the squares (SS) in the F1 score analysis shows that there is a great variability which can not be explained by the predictors. Once considerable differences have been detected when using the Kruskal–Wallis ANOVA, a post hoc Tukey test is performed to determine the mean with the most significant difference. This analysis compares the means of all treatments with the mean of every other treatment and the best available method is considered in cases when confidence intervals are desired (for details, refer to [51]). Figures 4 and 5 show the graph of the estimates and comparison interval. Each group mean is represented by a little circle and the interval is represented by a line extending out from the circle. Two group means are significantly different if their intervals are disjoint and they are not significantly different if their intervals overlap. As it can be seen, the SHO approach does not overlap with any other in both cases (F1 score and computational times), i.e., the Tukey test concludes that there is a significant difference between the SHO algorithm and the remaining approaches. This shows that SHO-SVDD offers a higher performance for fault detection in the plastic injection machine (F1 score is equal to 0.9702 in average) with a shorter computational training time. Finally, the overfitting effect is analyzed to test the SHO-SVDD generalization performance. The complete data set (normal and mixed operation samples) is used to implement a K-fold cross validation procedure. In addition, 5-fold (test 1) and 10-fold (test 2) cross validation are computed 30 times, and F1 score mean value is calculated i.e., a total of 150 and 300 training-test experiments are computed, respectively. The obtained F1 score mean values are 0.9701 for test 1 and 0.9714 for test 2, both are very close to F1 score mean value of 0.9702 obtained in the previous analysis. Considering this, it is concluded that the SHO-SVDD approach for fault detection presents a good generalization for this particular data set.

Table 2. Parameters configuration for each algorithm.

	SHO	KH	SSA	PSO
Number of iterations	200	200	200	200
Population size	50	50	50	50
Other parameters		$v_f = 0.02,$ $D^{max} = 0.005,$ $N^{max} = 0.01$ $w_n = 0.1 + 0.8(1 - i/200)$	$N_{fs} = 3$	$w = 0.5$ $c_1 = c_2 = 2,$

Table 3. F1 score—Times.

	F1 Score				Time			
	SHO	KH	SSA	PSO	SHO	KH	SSA	PSO
1	0.9620	0.9189	0.9610	0.9189	76.6396	112.7972	139.2550	207.6410
2	0.9744	0.9189	0.9189	0.9189	78.2157	114.3666	149.3458	209.0972
3	0.9512	0.9189	0.9750	0.9189	77.4362	113.8831	129.8195	208.6143
4	0.9744	0.9189	0.9189	0.9189	77.4033	113.4381	149.1254	208.5458
5	0.9744	0.9189	0.9189	0.9189	77.6753	114.3179	147.2649	207.9583
6	0.9750	0.9189	0.9189	0.9189	76.9510	113.6539	155.0708	209.6714
7	0.9744	0.9189	0.9744	0.9189	77.3957	114.0773	142.7828	211.6702
8	0.9750	0.9189	0.9750	0.9333	79.0279	113.6960	128.8649	212.5049
9	0.9750	0.9189	0.9189	0.9189	78.7634	113.6895	143.1924	208.9668
10	0.9744	0.9189	0.9211	0.9189	83.3741	113.5960	142.3234	209.6382
11	0.9630	0.9189	0.9750	0.9189	80.4753	113.8772	116.8845	210.7641
12	0.9750	0.9189	0.9189	0.9189	82.3348	114.1014	151.0041	209.1223
13	0.9750	0.9189	0.9189	0.9189	79.5617	114.0026	149.3981	208.3499
14	0.9750	0.9189	0.9189	0.9189	78.9463	114.1080	146.5968	208.5490
15	0.9750	0.9189	0.9189	0.9189	79.7457	113.7620	151.1163	210.4518
16	0.9744	0.9189	0.9351	0.9189	88.0263	113.7218	135.8897	209.4198
17	0.9512	0.9189	0.9189	0.9189	78.0516	113.8042	147.1367	209.8243
18	0.9620	0.9189	0.9189	0.9189	77.6847	113.7368	147.1347	213.8261
19	0.9744	0.9744	0.9189	0.9189	77.9648	98.1384	145.6228	208.7176
20	0.9620	0.9744	0.9189	0.9189	79.1865	114.2702	152.3700	208.1235
21	0.9744	0.9744	0.9189	0.9189	79.1609	114.2038	146.7939	209.6390
22	0.9750	0.9744	0.9189	0.9189	76.4243	114.1394	148.8082	210.0689
23	0.9630	0.9744	0.9189	0.9189	76.2201	113.5049	147.7095	207.0849
24	0.9750	0.9744	0.9189	0.9189	78.9390	113.6086	145.8267	208.5851
25	0.9750	0.9744	0.9189	0.9189	75.2608	113.7726	149.5031	225.5017
26	0.9744	0.9744	0.9744	0.9189	74.3404	113.8676	129.5610	210.0427
27	0.9744	0.9744	0.9189	0.9189	74.6434	114.3396	148.6370	207.7600
28	0.9620	0.9744	0.9750	0.9189	74.9797	113.4536	131.1206	209.6813
29	0.9620	0.9750	0.9189	0.9189	74.6099	93.0891	154.4285	207.8903
30	0.9750	0.9750	0.9189	0.9189	76.3885	114.4280	157.7352	213.1721
Mean	0.9702	0.9411	0.9321	0.9194	78.1942	112.6482	144.3441	210.0294
Std	0.0074	0.0277	0.0232	0.0026	2.8170	4.6904	9.1777	3.3325

Table 4. F1 score Kruskal–Wallis test.

Source	SS	df	MS	Chi-sq	p-Value
Columns	56,089.6	3	18,696.5	57.3	2.2188×10^{-12}
Error	60,398.9	116	520.7		
Total	116,488.5	119			

Table 5. Computational times Kruskal–Wallis test.

Source	SS	df	MS	Chi-sq	p-Value
Columns	135,000	3	45,000	111.57	5.0394×10^{-24}
Error	8990	116	77.5		
Total	143,990	119			

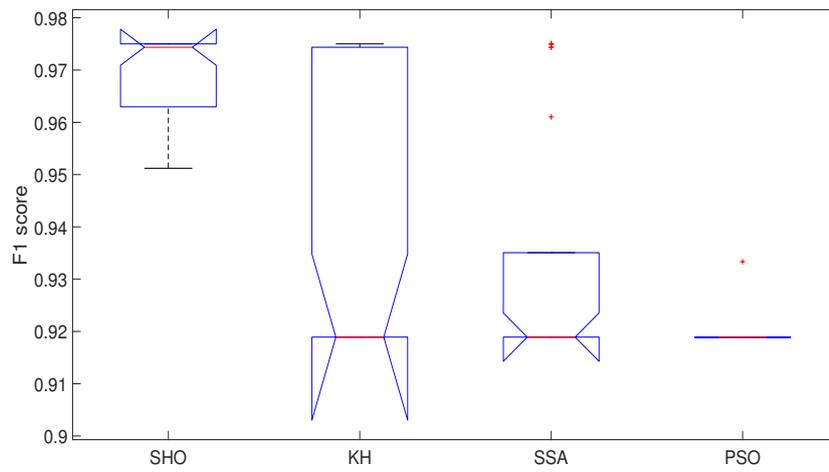


Figure 2. F1 score box plot.

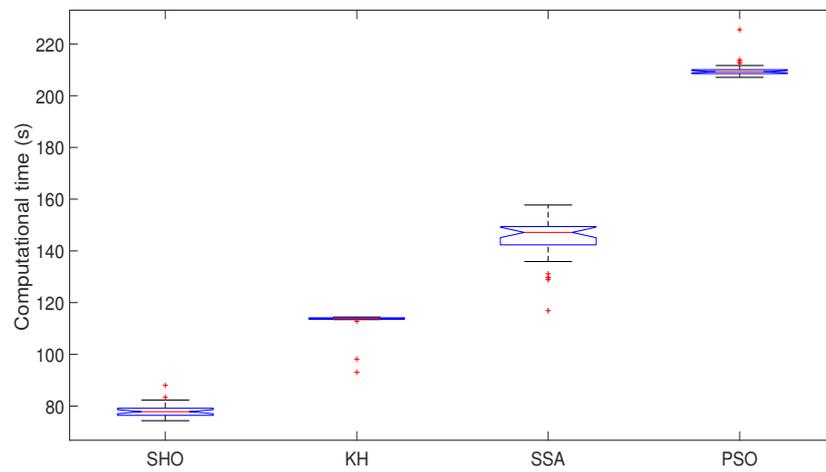


Figure 3. Computational times box plot.

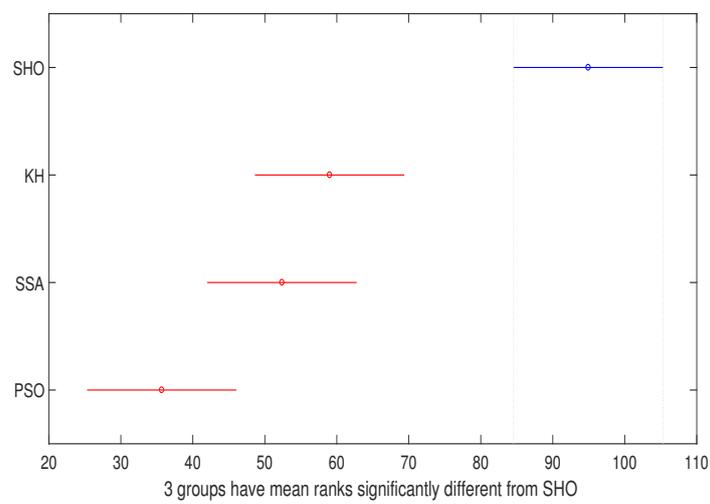


Figure 4. F1 score Tukey test.

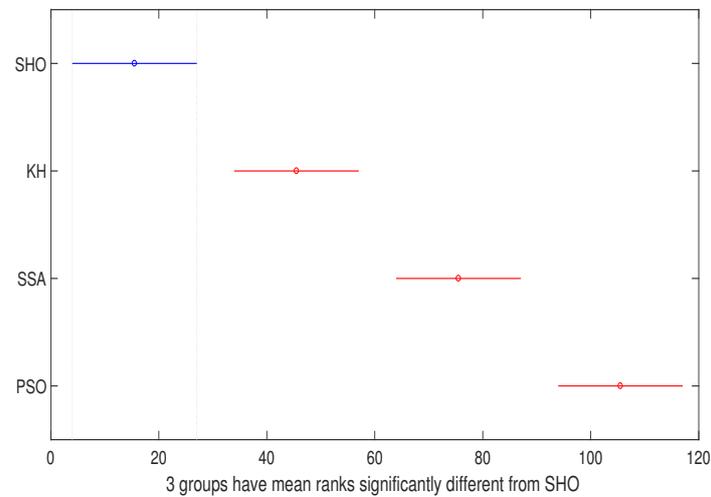


Figure 5. Computational times Tukey test.

5. Conclusions and Future Work

In this paper, a methodology for fault detection in multivariate processes is presented. This proposal is able to work in industrial processes and machines which are recently operated. It can deal with little data about faults by the implementation of a one class classifier (SVDD). Such algorithm is optimized during its training stage, that is, its hyper-parameters are tuned using a recent metaheuristic based on the behavior of spotted hyenas (SHO). In order to evaluate its computational performance, SHO has been compared with three other metaheuristics (KH, SSA y PSO) that have been successfully implemented for solving several engineering problems. The approaches were tested using data taking from a plastic injecting machine in the automotive industry. The results after 30 runs show that SHO reaches higher values in the F1 score metric, that is, it shows higher performance for fault detection in considerably shorter computing time with respect to the rest of the considered approaches. Furthermore, a non-parametric statistical analysis was performed to prove the statistical significance of the superior performance of SHO. The Kruskal–Wallis test and the post hoc analysis show that there is a significant statistical difference between SHO and the other metaheuristics. Moreover, SHO-SVDD was tested for generalization performance using 5-fold and 10-fold cross validation. The obtained F1 score mean values were 0.9701 and 0.9714, respectively, which are very close to the F1 score mean value of 0.9702 obtained by the proposed analysis. Considering this, it is concluded that the SHO-SVDD method presents a good generalization to classify this data set. Based on this, the SHO-SVDD approach seems to be the best to carry out fault detection for this particular industrial application. As future work, some feature selection approaches will be implemented to obtain a subset of variables that allows for performing a fault diagnosis in order to trace back operational issues or other problems in the process.

Author Contributions: Conceptualization, J.A.N.-A. and I.D.G.-C.; Data curation, E.O.R.-F.; Formal analysis, E.O.R.-F.; Investigation, J.A.N.-A., I.D.G.-C. and V.A.-G.; Methodology, J.A.N.-A. and I.D.G.-C.; Project administration, J.A.N.-A., I.D.G.-C., V.A.-G. and E.O.R.-F.; Resources, J.A.N.-A., I.D.G.-C., V.A.-G. and E.O.R.-F.; Software, J.A.N.-A. and I.D.G.-C.; Supervision, J.A.N.-A., I.D.G.-C. and E.O.R.-F.; Validation, J.A.N.-A. and I.D.G.-C.; Visualization, J.A.N.-A. and V.A.-G.; Writing—original draft, J.A.N.-A., I.D.G.-C., V.A.-G. and E.O.R.-F.; Writing—review & editing, V.A.-G. and E.O.R.-F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Autonomous University of Coahuila Grant No. C01-2019-49, and the APC was funded by the Autonomous University of Coahuila.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ghaheri, A.; Shoar, S.; Naderan, M.; Hoseini, S. The Applications of Genetic Algorithms in Medicine. *Oman. Med. J.* **2015**, *30*, 406–416. [[CrossRef](#)] [[PubMed](#)]
2. Vats, S.; Dubey, S.K.; Pandey, N.K. Genetic algorithms for credit card fraud detection. In Proceedings of the International Conference on Education and Educational Technologies, Barcelona, Spain, 1–3 July 2013.
3. Azzini, A.; De Felice, M.; Tettamanzi, A.G.B. A Comparison between Nature-Inspired and Machine Learning Approaches to Detecting Trend Reversals in Financial Time Series. In *Natural Computing in Computational Finance: Volume 4*; Brabazon, A., O'Neill, M., Maringer, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 39–59. [[CrossRef](#)]
4. Chiroma, H.; Gital, A.Y.; Rana, N.; Abdulhamid, S.M.; Muhammad, A.N.; Umar, A.Y.; Abubakar, A.I. Nature Inspired Meta-heuristic Algorithms for Deep Learning: Recent Progress and Novel Perspective. In *Advances in Computer Vision*; Arai, K., Kapoor, S., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 59–70.
5. Tao, Y.; Shi, H.; Song, B.; Tan, S. A Novel Dynamic Weight Principal Component Analysis Method and Hierarchical Monitoring Strategy for Process Fault Detection and Diagnosis. *IEEE Trans. Ind. Electron.* **2020**, *67*, 7994–8004. [[CrossRef](#)]
6. Zhang, X.; Kano, M.; Li, Y. Principal Polynomial Analysis for Fault Detection and Diagnosis of Industrial Processes. *IEEE Access* **2018**, *6*, 52298–52307. [[CrossRef](#)]
7. Amin, M.T.; Imtiaz, S.; Khan, F. Process system fault detection and diagnosis using a hybrid technique. *Chem. Eng. Sci.* **2018**, *189*, 191–211. [[CrossRef](#)]
8. Wu, H.; Zhao, J. Deep convolutional neural network model based chemical process fault diagnosis. *Comput. Chem. Eng.* **2018**, *115*, 185–197. [[CrossRef](#)]
9. Galagedarage Don, M.; Khan, F. Dynamic process fault detection and diagnosis based on a combined approach of hidden Markov and Bayesian network model. *Chem. Eng. Sci.* **2019**, *201*, 82–96. [[CrossRef](#)]
10. Tax, D.M.; Duin, R.P. Support Vector Data Description. *Mach. Learn.* **2004**, *54*, 45–66. [[CrossRef](#)]
11. Paturi, U.; Cheruku, S. Application and performance of machine learning techniques in manufacturing sector from the past two decades: A review. *Mater. Today Proc.* **2020**. [[CrossRef](#)]
12. Rostami, H.; Dantan, J.; Homri, L. Review of data mining applications for quality assessment in manufacturing industry: Support vector machines. *Int. J. Metrol. Qual. Eng.* **2015**, *6*, 401. [[CrossRef](#)]
13. Orru, P.; Zoccheddu, A.; Sassu, L.; Mattia, C.; Cozza, R.; Arena, S. Machine learning approach using MLP and SVM algorithms for the fault prediction of a centrifugal pump in the oil and gas industry. *Sustainability* **2020**, *12*, 4776. [[CrossRef](#)]
14. Giorgi, M.D.; Campilongo, S.; Ficarella, A. A diagnostics tool for aero-engines health monitoring using machine learning technique. *Energy Procedia* **2018**, *148*, 860–867. [[CrossRef](#)]
15. Jiang, Z.; Hu, M.; Feng, K.; Wang, H. A SVDD and-Means Based Early Warning Method for Dual-Rotor Equipment under Time-Varying Operating Conditions. *Shock Vib.* **2018**, *2018*. [[CrossRef](#)]
16. Jazi, A.Y.; Liu, J.J.; Lee, H. Automatic inspection of TFT-LCD glass substrates using optimized support vector machines. *IFAC Proc. Vol.* **2012**, *45*, 325–330. [[CrossRef](#)]
17. Zhuang, L.; Dai, H. Parameter optimization of kernel-based one-class classifier on imbalance text learning. In *Pacific Rim International Conference on Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 434–443.
18. Cao, Q.; Yu, L.; Cheng, M. A Brief Overview on Parameter Optimization of Support Vector Machine. *DEStech Trans. Mater. Sci. Eng.* **2016**. [[CrossRef](#)]
19. Zhang, X.; Chen, W.; Wang, B.; Chen, X. Intelligent fault diagnosis of rotating machinery using support vector machine with ant colony algorithm for synchronous feature selection and parameter optimization. *Neurocomputing* **2015**, *167*, 260–279. [[CrossRef](#)]
20. Lessmann, S.; Stahlbock, R.; Crone, S.F. Genetic algorithms for support vector machine model selection. In Proceedings of the 2006 IEEE International Joint Conference on Neural Network, Vancouver, BC, Canada, 16–21 July 2006; pp. 3063–3069.
21. Theissler, A.; Dear, I. Autonomously Determining the Parameters for SVDD with RBF Kernel from a One-Class Training Set. *World Academy of Science, Engineering and Technology, Open Science Index* **79**. *Int. J. Comput. Inf. Eng.* **2013**, *7*, 949–957.

22. Xiao, T.; Ren, D.; Lei, S.; Zhang, J.; Liu, X. Based on grid-search and PSO parameter optimization for Support Vector Machine. In Proceedings of the 11th World Congress on Intelligent Control and Automation, Shenyang, China, 29 June–4 July 2014; pp. 1529–1533.
23. Panda, N.E.A. Oppositional Spotted Hyena Optimizer with Mutation Operator for Global Optimization and Application in Training Wavelet Neural Network. *J. Intell. Fuzzy Syst.* **2020**. [[CrossRef](#)]
24. Moayedi, H.; Bui, D.T.; Dounis, A.; Kalantar, B. Spotted Hyena Optimizer and Ant Lion Optimization in Predicting the Shear Strength of Soil. *Appl. Sci.* **2019**, *9*, 4738. [[CrossRef](#)]
25. Dhiman, G.; Chahar, V. Spotted Hyena Optimizer for Solving Complex and Nonlinear Constrained Engineering Problems: Theory and Applications. In *Harmony Search and Nature Inspired Optimization Algorithms*; Springer: Singapore, 2019.
26. Dhiman, G.; Kaur, A. Optimizing the Design of Airfoil and Optical Buffer Problems Using Spotted Hyena Optimizer. *Designs* **2018**, *2*, 28. [[CrossRef](#)]
27. Chahar, V.; Kaur, A. Binary Spotted Hyena Optimizer and its Application to Feature Selection. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *11*, 2625–2645.
28. Dhiman, G.; Kaur, A. A Hybrid Algorithm Based on Particle Swarm and Spotted Hyena Optimizer for Global Optimization. In *Soft Computing for Problem Solving*; Bansal, J.C., Das, K.N., Nagar, A., Deep, K., Ojha, A.K., Eds.; Springer: Singapore, 2019; pp. 599–615.
29. Divya, S.; El, K.; Rao, M.; Vemulapati, P. Prediction of Gene Selection Features Using Improved Multi-objective Spotted Hyena Optimization Algorithm. In *Data Communication and Networks*; Springer: Singapore, 2020.
30. Gandomi, A.H.; Alavi, A.H. Krill herd: A new bio-inspired optimization algorithm. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 4831–4845. [[CrossRef](#)]
31. Bolaji, A.L.; Al-Betar, M.A.; Awadallah, M.A.; Khader, A.T.; Abualigah, L.M. A comprehensive review: Krill Herd algorithm (KH) and its applications. *Appl. Soft Comput.* **2016**, *49*, 437–446. [[CrossRef](#)]
32. Abualigah, L.M.; Khader, A.T.; Al-Betar, M.A.; Awadallah, M.A. A krill herd algorithm for efficient text documents clustering. In Proceedings of the 2016 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Batu Feringghi, Malaysia, 30–31 May 2016; pp. 67–72.
33. Abualigah, L.M.; Khader, A.T.; Hanandeh, E.S.; Gandomi, A.H. A novel hybridization strategy for krill herd algorithm applied to clustering techniques. *Appl. Soft Comput.* **2017**, *60*, 423–435. [[CrossRef](#)]
34. Abualigah, L.M.; Khader, A.T.; Hanandeh, E.S. A combination of objective functions and hybrid krill herd algorithm for text document clustering analysis. *Eng. Appl. Artif. Intell.* **2018**, *73*, 111–125. [[CrossRef](#)]
35. Abualigah, L.M.; Khader, A.T.; Hanandeh, E.S. Hybrid clustering analysis using improved krill herd algorithm. *Appl. Intell.* **2018**, *48*, 4047–4071. [[CrossRef](#)]
36. Karthick, P.; Palanisamy, C. Optimized cluster head selection using krill herd algorithm for wireless sensor network. *Automatika* **2019**, *60*, 340–348. [[CrossRef](#)]
37. Wang, Z.; Zheng, L.; Wang, J.; Du, W. Research on novel bearing fault diagnosis method based on improved krill herd algorithm and kernel extreme learning machine. *Complexity* **2019**, 2019. [[CrossRef](#)]
38. Zhang, Y.; Li, X.; Zheng, H.; Yao, H.; Liu, J.; Zhang, C.; Peng, H.; Jiao, J. A fault diagnosis model of power transformers based on dissolved gas analysis features selection and improved krill herd algorithm optimized support vector machine. *IEEE Access* **2019**, *7*, 102803–102811. [[CrossRef](#)]
39. Jain, M.; Singh, V.; Rani, A. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. *Swarm Evol. Comput.* **2019**, *44*, 148–175. [[CrossRef](#)]
40. Zheng, T.; Luo, W. An improved squirrel search algorithm for optimization. *Complexity* **2019**, 2019. [[CrossRef](#)]
41. Altamirano-Guerrero, G.; Garcia-Calvillo, I.D.; Resendiz-Flores, E.O.; Costa, P.; Salinas-Rodriguez, A.; Goodwin, F. Intelligent design in continuous galvanizing process for advanced ultra-high-strength dual-phase steels using back-propagation artificial neural networks and MOAMP-Squirrels search algorithm. *Int. J. Adv. Manuf. Technol.* **2020**, *110*, 2619–2630. [[CrossRef](#)]
42. Basu, M. Squirrel search algorithm for multi-region combined heat and power economic dispatch incorporating renewable energy sources. *Energy* **2019**, *182*, 296–305. [[CrossRef](#)]
43. Hu, H.; Zhang, L.; Bai, Y.; Wang, P.; Tan, X. A hybrid algorithm based on squirrel search algorithm and invasive weed optimization for optimization. *IEEE Access* **2019**, *7*, 105652–105668. [[CrossRef](#)]
44. Sanaj, M.; Prathap, P.J. Nature inspired chaotic squirrel search algorithm (CSSA) for multi objective task scheduling in an IAAS cloud computing atmosphere. *Eng. Sci. Technol. Int. J.* **2020**, *23*, 891–902. [[CrossRef](#)]

45. Shen, F.; Song, Z.; Zhou, L. Improved PCA-SVDD based monitoring method for nonlinear process. In Proceedings of the 2013 25th Chinese Control and Decision Conference (CCDC), Guiyang, China, 25–27 May 2013; pp. 4330–4336. [[CrossRef](#)]
46. Yin, G.; Zhang, Y.T.; Li, Z.N.; Ren, G.Q.; Fan, H.B. Online fault diagnosis method based on Incremental Support Vector Data Description and Extreme Learning Machine with incremental output structure. *Neurocomputing* **2014**, *128*, 224–231. [[CrossRef](#)]
47. Liu, C.; Gryllias, K. A semi-supervised Support Vector Data Description-based fault detection method for rolling element bearings based on cyclic spectral analysis. *Mech. Syst. Signal Process.* **2020**, *140*, 106682. [[CrossRef](#)]
48. Dhiman, G.; Chahar, V. Spotted Hyena Optimizer: A Novel Bio-inspired based Metaheuristic Technique for Engineering Applications. *Adv. Eng. Softw.* **2017**. [[CrossRef](#)]
49. Kennedy, J.; Eberhart, R.C. Particle Swarm Optimization. *IEEE Int. Conf. Neural Netw.* **1995**, *4*, 1942–1948.
50. Abualigah, L.M.; Khader, A.T.; Hanandeh, E.S. A new feature selection method to improve the document clustering using particle swarm optimization algorithm. *J. Comput. Sci.* **2018**, *25*, 456–466. [[CrossRef](#)]
51. Herve Abdi, L.J.W. *Encyclopedia of Research Design*; SAGE Publications: New York, NY, USA, 2010.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).