ADJUSTING PARAMETERS OF ONE-SIDED CONTROL CHART-BASED SUPPORT VECTOR MACHINES USING GENETIC ALGORITHM AND PARTICLE SWARM OPTIMIZATION

Thi-Thao-Nguyen Nguyen, Thi Huynh Anh Le*

The University of Danang - University of Science and Technology, Danang, Vietnam

*Corresponding author: lthanh@dut.udn.vn

(Received: April 01, 2025; Revised: April 22, 2025; Accepted: April 25, 2025)

DOI: 10.31130/ud-jst.2025.23(6A).178E

Abstract - This study examines parameter optimization for one-sided control chart-based Support Vector Machines (SVM) using Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Statistical learning classification effectively addresses statistical process control issues, especially in managing big data complexities. While SVM has computational benefits, its predictive accuracy relies on proper parameter selection. This research presents two methods—D-SVM-GA and D-SVM-PSO—to optimize key parameters like moving windows *(N_w)*, the Gamma (σ^2), and the penalty cost *(C)*. Results show upper control limits of 0.518 for D-SVM-GA and 0.515 for D-SVM-PSO, indicating effective optimization. The ARL_1 values highlight improved detection performance, where higher N_w results in lower ARL_1, enhancing responsiveness to shifts. These findings underscore the effectiveness of evolutionary algorithms in optimizing SVM parameters, offering insights for future applications in process control and statistical learning, with potential for improved accuracy and efficiency in data-driven decision-making.

Key words - Genetic Algorithm; Parameter Optimization; Particle Swarm Optimization; Statistical Process Control; Support Vector Machine

1. Introduction

Statistical learning classification has been considered as a commonly technique to solve statistical process control problem. Statistical learning is utilized to understand and define a prediction function based on data. This technique has been used for machine learning regarding to statistics and function analysis [1]. This technique also has been used in process control to deal with data complexity and use the control limit. Big data has been known as a major trend in statistical learning and process control. Due to the advantage of computational time, support vector machine (SVM) has been a feasible approach to handle big data problem. However, the parameter selection would affect the accuracy of SVM so that it requires an evolutionary algorithm to enhance its accuracy [2]. Therefore, it is essential to determine optimal parameters for SVM to increase its accuracy.

To determine SVM's parameter, several algorithms have been used by previous studies. For instance, Wang and Du [3] used differential evolution (DE) while Hong et al. [4] applied Particle Swarm Optimization (PSO) to determine SVM's parameter. He et al. [5] proposed distance-based control chart that was called D-SVM to optimize SVM's parameter. Wang et al. [2] extended D-SVM approach to optimize the SVM's parameter by using DE which was called D-SVM-DE. The results showed that S-SVM-DE achieved the better value of average run length (ARL) than D-SVM chart.

This study further develops an on-site control chart based on SVM. Among the optimization techniques, PSO stands out for its simplicity and the minimal number of parameters that need adjustment compared to other evolutionary algorithms. Conversely, the Genetic Algorithm (GA) is a stochastic method that employs randomness in both selection and reproduction processes. The fundamental principle of GA involves managing a population of potential solutions simultaneously. Within the realm of GA, various procedures exist for parameter optimization. Specifically, parameter tuning allows for the identification of optimal values for GA parameters through an iterative search process, starting from an initial parameter vector.

Consequently, the application of GA and PSO for determining optimal SVM parameters leads to the designations D-SVM-GA and D-SVM-PSO, respectively. Additionally, this study evaluates how the parameters of the D-SVM chart influence the ARL_1 value. The primary parameters considered for the D-SVM in this research include the moving window size *(N_w)*, the Gamma (σ^2), and the penalty cost *(C)*.

2. Literature review

2.1. Support Vector Machine

Support vector machine (SVM) is first introduced by Vapnik (1998). Considering a training dataset of N pairs $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N), i = 1,2, \dots, N,$ where $x_i \in R^P$ is a p-dimensional input vector and $y_i \in \{-1,1\}$ is the class label. A linear decision function $f(x)$ which classifies x, a p-dimensional observation that may not necessarily be from the training set, can be defined as:

$$f(x) = w^T x + b, \tag{1}$$

where T is a transpose operator, w is normal to the hyperplane, and b is a threshold value. For each training sample x_i, this function provides $f(x_i) \geq 0$ for $y_i = 1$ and $f(x_i) < 0$ for $y_i = -1$. Due to the existence of many hyperplanes that separate two classes, the SVM classifier is based on a hyperplane that maximizes the margin between two classes and expressed as optimization problem as follows:

$$\text{Min } \frac{1}{2} w^T w = \frac{1}{2} \parallel w \parallel^2, \tag{2}$$

subject to

$$y_i(w^T w + b) \geq 1, i = 1,2, \dots, N, \quad (3)$$

where $\|.\|$ is the Euclidian norm. In current practice, the hyperplane may not separate the training data completely. Hence, we can relax the constraint as Eq. (4).

$$y_i(w^T w + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1,2, \dots, N, \quad (4)$$

where ξ_i is a slack variable related to training errors. Therefore, we can modify the function in Eq. (2) as follows:

$$\text{Min} \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} \xi_i, \quad (5)$$

where $C > 0$ is a regularizing constant that controls the trade-off between classification errors and margin.

However, in most cases, the linear separation is still not practically used. Therefore, to get the nonlinear separating surface, the input space is mapped into a feature space by using kernel functions and by searching for an optimal hyperplane in the resulting feature space. The nonlinear decision function is given by Eq. (6).

$$f(x) = w^T \phi(x) + b, \quad (6)$$

where $\phi(.)$ is a nonlinear mapping function. Under the given parameters $C > 0$ and $\varepsilon > 0$, the standard form of SVM can be defined as Eq. (7).

$$\text{Min}_{w,b,\xi,\xi^*} \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} (\xi_i + \xi^*), \quad (7)$$

subject to

$$\begin{aligned} & w^T \phi(x_i) + b - y_i \leq \varepsilon + \xi_i, \\ & y_i - w^T \phi(x_i) - b_i \leq \varepsilon + \xi^*, \\ & \xi_i, \xi^* \geq 0, i = 1,2, \dots, N, \end{aligned} \quad (8)$$

where ξ^* denotes training errors above ε and ξ denotes training errors below $-\varepsilon$. Equation 7 can be solved by considering its dual problem that is a quadratic programming problem. After the quadratic optimization with inequality constraint is solved, the SVM decision function is expressed by Eq. (9).

$$f(x) = \text{sgn}(\sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b), \quad (9)$$

where x is a new observation to be classified, $0 \leq \alpha_i < C$ are coefficients satisfying $\sum_{i=1}^{N} \alpha_i y_i = 0$, x_i represents those inputs with nonzero corresponding coefficients called support vectors (SVs), and $K(x, x_i)$ is a kernel function. One of the kernel functions that is commonly used is Gaussian radial basis function (RBF) and expressed as Eq. (10).

$$\begin{aligned} & K(x_j, x_i) = \exp(-\gamma \| x_j - x_i \|^2), \\ & \qquad j = 1,2, \dots, N_w \text{ and } x_i \in SV, \end{aligned} \quad (10)$$

where $\gamma = 1/2\sigma^2$ and σ^2 is the spread parameter.

2.2. D-SVM Chart

SVMs and relevant modifications have been widely implemented in solving control chart pattern recognition problems. He at al. (2018) [5] developed the distance-based control chart that called D-SVM chart using SVMs. The D-SVM chart has some different characteristics compared to traditional distance-based multivariate control. D-SVM chart considers the average distance between real-time observations in a moving window and a classification boundary. It is obtained by dynamically training an SVM model with a data composed of fixed reference dataset and real-time moving window samples. It differs from T^2 chart that considers the Mahalanobis distance between the centers of a reference dataset and moving window samples as a monitoring statistic. A moving window sample of size N_w consists of $N_w - 1$ preceding observations and one real-time observations. The fixed in-control reference dataset and the moving window samples are denoted as S_0 and S_t respectively, where t is the time period at which a real-time observation is collected (i.e. $t = 1,2, \dots, N$).

The distance from a real-time observation x_t to the support vectors (SVs), produced by the SVM classifier $C_t = (\alpha, b)$ can be calculated by using Eq. (11).

$$M(x_t|C_t) = \sum_{x_i \in SV} \alpha_i y_i K(x_j, x_i) + b, j = 1,2, \dots, N_w, \quad (11)$$

where $i = 1,2,\dots$, length of SVs at time t. The SVM scores $(M(x))$, can be positive or negative. The transformation $g(x_t|C_t) = \frac{1}{1+\exp(-M(x|C_t))}$ makes the monitoring statistic to go upward while there is a process shift. Then, the monitoring statistic of the D-SVM chart can be expressed by Eq. (12).

$$p(S_t|C_t) = \frac{1}{N_w} \sum_{x_j \in S_t} g(x_j|C_t), \quad (12)$$

where N_w is the length of a moving window and S_t is the moving window samples for $t = 1,2, \dots, N$.

2.3. Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a population-based optimization algorithm that originally introduced by Eberhart and Kennedy [6] and basically inspired by the social behavior of bird flocks. It is one of the popular algorithms that is very easy to implement and has fewer parameters to adjust when compared with other evolutionary algorithms [7]. In PSO, the population is called a swarm, while the individuals are called particles. The position of a particle (search point) in n-dimensional space is expressed by Eq. (13).

$$x_i = (x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{in})^T, \quad (13)$$

where x_{ij} is the vector component of the i^{th} particle in the j^{th} dimension. The movement of each particle is specified by the velocity or change in position vector.

$$v_i = (v_{i1}, v_{i2}, \dots, v_{ij}, \dots, v_{in})^T, \quad (14)$$

and each particle established its best position found during the search that is expressed by Eq. (15). The value of the function at that position denotes as $f(pbest_i)$.

$$pbest_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{ij}, \dots, pbest_{in})^T, \quad (15)$$

On the other side, the swarm remembers the overall best position found by any of the swarm's particles during the search that is expressed by Eq. (16) and its corresponding function value that denotes as $f(gbest)$.

$$gbest_i = (gbest_{i1}, gbest_{i2}, \dots, gbest_{ij}, \dots, gbest_{in})^T, \quad (16)$$

Updated velocity vector that denotes as v_i^{k+1}, specifying the movement from the current position of the particle (x_i^k) is defined by a weighted linear combination of three vectors: the vector from the current particle position to its own best position $(pbest_i^k - x_i^k)$; the vector from the current particle position to the global best position $\left(gbest^k - x_i^k\right)$; and the velocity vector found in the previous iteration (v_i^k). In other words, the velocity vector (v_{ij}^{k+1}), for the i^{th} particle at the $(k+1)^{th}$ step is given by Eq. (17).

$$v_{ij}^{k+1} = w.v_{ij}^k + c_1.\mathrm{rand}_{1ij}.\left(pbest_{ij}^k - x_{ij}^k\right) + c_2.\mathrm{rand}_{2ij}.\left(gbest_{ij}^k - x_{ij}^k\right), \quad (17)$$

where w, c_1, and c_2 are positive constants, and rand_{1ij} and rand_{2ij} are randomly generated uniform distributed number in the range (0,1). Then, the next position of the particle (x_{ij}^{k+1}) is expressed by Eq. (18).

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1}. \quad (18)$$

2.4. Genetic Algorithm (GA)

GA is initially introduced by John Holland (1975) as a heuristic method to solve optimization problems based on what is called "survival of the fittest" and proved as an effective method for optimization and currently it is widely used in artificial life system. GA is a stochastic algorithm that using random procedure in both selection and reproduction. GA keeps the memory of its solutions and always considers a population of individuals to generate new ones.

The general concept of GA is simple where it handles a set of possible evolution simultaneously. Each of possible solution is represented by what we called chromosome where the reproduction process will be started. The reproduction process uses a set of operators directly on the chromosome that include crossover and mutation. Replacement is then applied to decide whatever the new generated offspring to be inserted in the population or not. To evolve the population of individuals, this process is iterated until its population converges to an area in the search space that contains the best solution to a problem. Two types of parameters that must be distinguished include: Quantitative Parameter and Qualitative Parameter. Quantitative Parameter is a parameter with numerical value that take the value from (0,1) and includes for instance crossover rate, mutation rate, etc. Qualitative Parameter is a parameter that has finite domain and often symbolic e.g. mutation operator, crossover operator. It does not have any specific distance metric e.g. one point, two points, three points, etc.

In the area of GA, there exists a different procedure in determining the value of the parameters. Specific on the parameter tuning, this method allows the search method to find the best values for a set of parameters of a GA. Starting from a vector of parameters, parameter tuning is searching for the best vector of values by iteratively generating. This section presents the impact of differentiating D-SVM chart's parameters toward the ARL_1. The parameters involved in this study are: Number of moving windows (N_w), σ^2, and penalty cost (C). An upper control limit needs to be set up before calculating the ARL_1. for each different parameter value. This work utilizes $ARL_0 \approx 200$ to determine the appropriate control limit. The procedure to determine the upper control limit is shown in Algorithm 2. Number of simulations to obtain a specific value of either ARL_0 or ARL_1 set to 100 replications. Moreover, this study uses three different values for each considered parameter that are defined in Error! **Not a valid bookmark self-reference.** shows several types of information which are written in the x, y, z format. x represents different parameter values that will be used for numerical studies. Secondly, the associated upper control limit is denoted by y. Lastly, z is the obtained ARL_0 after implementing Algorithm 2. It can be seen that, there is an increasing trend in the UCL of the D-SVM chart as the value of N_w increases. Conversely, an increase in σ^2 and C leads to a decrease in the UCL. This relationship suggests that larger values of moving windows enhance the control limits, while higher values of variance and C diminish them.

Table 1 to perform the numerical studies on the D-SVM chart. This section presents the impact of differentiating D-SVM chart's parameters toward the ARL_1. The parameters involved in this study are: Number of N_w, σ^2, and C. An upper control limit needs to be set up before calculating the ARL_1 for each different parameter value. This work utilizes $ARL_0 \approx 200$ to determine the appropriate control limit. The procedure to determine the upper control limit is shown in Algorithm 2. Number of simulations to obtain a specific value of either ARL_0 or ARL_1 set to 100 replications. Moreover, this study uses three different values for each considered parameter that are defined in Error! **Not a valid bookmark self-reference.** shows several types of information which are written in the x, y, z format. x represents different parameter values that will be used for numerical studies. Secondly, the associated upper control limit is denoted by y. Lastly, z is the obtained ARL_0 after implementing Algorithm 2. It can be seen that, there is an increasing trend in the UCL of the D-SVM chart as the value of N_w increases. Conversely, an increase in σ^2 and C leads to a decrease in the UCL. This relationship suggests that larger values of moving windows enhance the control limits, while higher values of variance and C diminish them.

Table 1 to perform the numerical studies on the D-SVM chart with different parameter vectors. The quality of the given parameter vector is defined by the performance of its relative GA using its parameter values.

3. Methodology

This section presents the method of implementing GA and PSO to adjust parameter σ^2 of the D-SVM chart to obtain the upper control limit of the chart. The procedure is basically divided into two stages. Firstly, a simulation approach is conducted to obtain a set of run-length values that result in an average run length (ARL). Secondly, the ARL obtained from the simulation approach will be compared with the predetermined in-control ARL (ARL_0). These two stages will be continuously repeated until the resulting ARL is greater or equal to the predetermined ARL_0. Algorithms 1 and Algorithm 2 are presented to show the procedures of implementation.

Algorithm 1. Function of obtaining the ARL

1. Set the necessary parameters (dimensionality of the observed data, value of moving windows, mean values of data, covariance matrix of data, number of simulation)
2. Generate a vector of data, $S_0 = \{1, 2, \dots, N_0\}$ as a reference set where N_0 is the necessary number of generated reference data.
3. **for** i in 1 $\boldsymbol{to}$ $number\ of\ simulation$:
4.   Generate a vector of data, $S_1 = \{1, 2, \dots, N_1\}$ as an artificial set where N_1 is the necessary number of generated artificial data.
5.   **for** j in 1 $\boldsymbol{to}$ $(N - N_w + 1)$:
6.   Take the part of S_1, starting from index j to $j + N_w - 1$ and place it in S_w
7.   Combine both S_0 and S_w into S_s
8.   Set the labels for S_s, -1 for the part from S_0 and 1 for the part from S_w
9.   Set a function of SVM with Gaussian radial basis function that takes one parameter to run, $SVM(\sigma^2)$ and MAE as the performance measurement of the SVM.
10.   Set the lower value, l, and the upper value, u, of σ^2
11.   Set an optimization function that is based on GA and PSO. This function takes the l and u for the range of σ^2 and the objective is to minimize the MAE resulted from implementing the $SVM(\sigma^2)$. The result of the function is the value of σ^2 resulting in the smallest MAE, namely σ_*^2
12.   Run the $SVM(\sigma^2)$ with $\sigma^2 = \sigma_*^2$
13.   Extract the coefficients and the intercept of the $SVM(\sigma_*^2)$
14.   Calculate the monitoring statistics, p, defined in equation (12)
15.   **if** $p > h$
16.   set j as the run length of the iteration i
17.   **break**
18.   **end for**
19.   Set $ARL = mean(run\ length)$
20. **end for**

Algorithm 2. Function of obtaining appropriate upper-control limit

1. Set the initial value of h_0, the upper-control limit of D-SVM chart
2. **repeat**
3.   Implement **Algorithm 1.** with $h = h_0$ and obtain the ARL
4.   **if** $ARL > ARL_0$

        break

6.   **else**

        $h_0 = h_0 + 0.002$

8.   **end if**

4. Computational results

The experiment coded by R language run on a computer with 3.60 GHz and 16.0 GB RAM. Sensitivity analyses on the impact of change of D-SVM parameters toward the ARL_1 are first performed. The obtained upper control limits by implementing the GA and PSO to optimize parameter σ^2 of the D-SVM chart are then proposed with a brief discussion.

4.1. Testing on D-SVM Parameters

This section presents the impact of differentiating D-SVM chart's parameters toward the ARL_1. The parameters involved in this study are: Number of N_w, σ^2, and C. An upper control limit needs to be set up before calculating the ARL_1. for each different parameter value. This work utilizes $ARL_0 \approx 200$ to determine the appropriate control limit. The procedure to determine the upper control limit is shown in Algorithm 2. Number of simulations to obtain a specific value of either ARL_0 or ARL_1 set to 100 replications. Moreover, this study uses three different values for each considered parameter that are defined in Error! **Not a valid bookmark self-reference.** shows several types of information which are written in the x, y, z format. x represents different parameter values that will be used for numerical studies. Secondly, the associated upper control limit is denoted by y. Lastly, z is the obtained ARL_0 after implementing Algorithm 2. It can be seen that, there is an increasing trend in the UCL of the D-SVM chart as the value of N_w increases. Conversely, an increase in σ^2 and C leads to a decrease in the UCL. This relationship suggests that larger values of moving windows enhance the control limits, while higher values of variance and C diminish them.

Table 1 to perform the numerical studies on the D-SVM chart.

Error! Not a valid bookmark self-reference. shows several types of information which are written in the x, y, z format. x represents different parameter values that will be used for numerical studies. Secondly, the associated upper control limit is denoted by y. Lastly, z is the obtained ARL_0 after implementing Algorithm 2. It can be seen that, there is an increasing trend in the UCL of the D-SVM chart as the value of N_w increases. Conversely, an increase in σ^2 and C leads to a decrease in the UCL. This relationship suggests that larger values of moving windows enhance the control limits, while higher values of variance and C diminish them.

Table 1. *Tested parameters of D-SVM chart with the obtained upper-control limit (UCL) and* ARL_0

$\boldsymbol{N_w}$	10, **0.428**, 207.14
	14, **0.524**, 210.88
	18, **0.576**, 242.3
$\boldsymbol{\sigma^2}$	2.5, **0.472**, 208.27
	3.5, **0.428**, 207.14
	4.5, **0.372**, 226.86
$\boldsymbol{C}$	0.6, **0.556**, 244.73
	0.8, **0.492**, 234.26
	1, **0.428**, 207.14

***Table 2.** ARL_1 comparison under different values of N_w, σ^2, and C for multivariate normal data with different mean vector shifts*

Shift	N_w		
	10	**14**	**18**
(1,1,1, 0,0,0,0,0,0,0)	84.12	36.47	22.89
(1,1,1,1,1,0,0,0,0,0)	35.38	21.7	14.6
(1,1,1,1,1,1,1,0,0,0)	16.43	8.94	5.71
Shift	σ^2		
	2.5	**3.5**	**4.5**
(1,1,1, 0,0,0,0,0,0,0)	36.95	84.12	94.83
(1,1,1,1,1,0,0,0,0,0)	11.24	35.38	52.03
(1,1,1,1,1,1,1,0,0,0)	5.26	16.43	29.04
Shift	C		
	0.6	**0.8**	**1**
(1,1,1, 0,0,0,0,0,0,0)	74.52	70.96	84.12
(1,1,1,1,1,0,0,0,0,0)	33.46	34.08	35.38
(1,1,1,1,1,1,1,0,0,0)	16.92	18.33	16.43

Table 2 compares ARL_1 under various shifts for N_w, σ^2, and C. The results reveal a notable trend: ARL1 decreases as the magnitude of the shift increases. Additionally, changes in N_w, and σ^2 significantly influence the performance of the D-SVM chart, whereas the parameter C appears to have a minimal impact. Specifically, a higher value of N_w, correlates with a lower ARL_1, indicating improved detection performance. In contrast, the relationship between σ^2 and ARL_1 is inversely proportional; as σ^2 increases, ARL_1 also increases, suggesting that higher variance leads to a longer average run length, potentially indicating a slower response to shifts.

4.2. Upper Control Limits of D-SVM-GA and D-SVM-PSO Charts

This section shows the initial implementation of GA and PSO to optimize parameter σ^2 of D-SVM chart. To conduct this experiment, the other parameters are determined beforehand, i.e. $N_w = 10$ and $C = 1$. The main difficulty encountered during this experiment is extremely high computational time due to adding an additional step. For example, performing GA or PSO before applying the D-SVM chart over the provided dataset. Consequently, the conducted simulation experiment only takes 10 replications because of the time limitation.

***Table 3.** Summary of GA and PSO optimization results for D-SVM chart*

	Upper control limit	Accuracy	Computational time (Seconds)
D-SVM-GA	0.518	92%	150
D-SVM-PSO	0.515	89%	90

Based on the simulation results of Table 3, the upper-control limit of D-SVM-GA chart and D-SVM-PSO chart are 0.518 and 0.515, respectively. The results also showed that the computational time of D-SVM-GA is significantly higher than the computational time of D-SVM-PSO, highlighting the efficiency of PSO in this context.

This analysis delineates the effects of varying parameters on the performance of the D-SVM chart, particularly focusing on ARL_1. The findings underscore the importance of parameter selection in optimizing chart performance and highlight the trade-offs between computational efficiency and the accuracy of parameter tuning in D-SVM applications.

5. Conclusion and future work

This work has provided fundamental analyses to show the impact of varying parameters of D-SVM chart toward the ARL_1. In addition, GA and PSO are both utilized to optimize parameter σ^2 of the D-SVM chart, namely D-SVM-GA and D-SVM-PSO. However, due to time limitation, this work only shows the UCL resulted from D-SVM-GA and D-SVM-PSO. In the future, an extensive computational experiment could be conducted to fairly compare the GA and PSO algorithm.

REFERENCES

[1] T. Hastie, R. Tibshirani, and Friedman, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction,* 2nd ed. Berlin: Springer, 2009.

[2] F. K. Wang, B. Bizuneh, and C. Xiaobin, "One-sided control chart based on support vector machines with differential evolution algorithm", *Quality and Reliability Engineering International*, vol. 35, 2019, doi: 10.1002/qre.2465.

[3] F. K. Wang and T. Du, "Implementing support vector regression with differential evolution to forecast motherboard shipments", *Expert Systems with Applications*, vol. 41, pp. 3850–3855, 2014, doi: 10.1016/j.eswa.2013.12.022.

[4] W. C. Hong, "Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model", *Energy Conversion and Management,* vol. 50, pp. 105-117, 2009, doi: 10.1016/j.enconman.2008.08.031.

[5] S. He, W. Jiang, and H. Deng, "A distance-based control chart for monitoring multivariate processes using support vector machines", *Annals of Operations Research*, vol. 263, no. 1, pp. 191-207, 2018, doi: 10.1007/s10479-016-2186-4.

[6] R. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory". *Proceedings of the sixth interna- tional symposium on micro machine and human science. IEEE*, 1995: p. 39-43.

[7] S. Du, J. Lv, and L. Xi, "On-line classifying process mean shifts in multivariate control charts based on multiclass support vector machines", *International Journal of Production Research*, vol. 50, pp. 1-23, 2012, doi: 10.1080/00207543.2011.631596.