Postprint: Application of SADE-based Hybrid Kernel LSSVM in Compressor Fault Prediction

Authors: Du Jie, Wu Jing

Date: 2018-06-19T00:00:00+00:00

Abstract

To address the challenge of obtaining effective prediction models due to the diversity of screw refrigeration compressor types and the complexity of faults, a prediction model based on hybrid kernel Least Squares Support Vector Machine (LSSVM) optimized by Self-Adaptive Differential Evolution algorithm (SADE) is proposed. The SADE in this model features a simpler theoretical structure, fewer parameter settings, and stronger search capability compared to other intelligent optimization algorithms. During the optimization process, SADE achieves self-adaptation in differential strategy, scaling factor, and crossover probability, which ensures global search capability and population diversity in the early optimization stage while improving local search capability and convergence speed. By utilizing SADE to optimize kernel parameters, LSSVM parameters, and hybrid kernel adjustment parameters, the accuracy of the hybrid kernel LSSVM prediction model is enhanced. When applied to compressor fault prediction, experimental results demonstrate that the model can effectively predict compressor faults, thereby validating the feasibility of the proposed model.

Full Text

Preamble

Title: Application of Hybrid Kernel LSSVM Based on SADE in Compressor Fault Prediction

Authors: Du Jie[1], Wu Jing[2]
[1]School of Information Engineering, Southwest University of Science & Technology, Mianyang Sichuan 621010, China;
[2]Robot Technology Used for Special Environment Key Laboratory of Sichuan Province, Mianyang Sichuan 621010, China

Abstract: Due to the diversity of screw refrigeration compressor types and the complexity of faults, obtaining effective prediction models remains challenging.

This paper proposes a prediction model based on hybrid kernel Least Squares Support Vector Machine (LSSVM) optimized by Self-Adaptive Differential Evolution (SADE). Compared with other intelligent optimization algorithms, SADE offers a simpler theoretical structure, fewer parameters, and stronger search capability. During optimization, SADE achieves adaptivity in differential strategy, scaling factor, and crossover probability, ensuring global search ability and population diversity in the early stage while improving local search ability and convergence speed. By using SADE to optimize kernel parameters, LSSVM parameters, and hybrid kernel adjustment parameters, the prediction accuracy of the hybrid kernel LSSVM model is significantly enhanced. Applied to compressor fault prediction, experimental results demonstrate that the model effectively predicts compressor faults, validating its feasibility.

Keywords: LSSVM; hybrid kernel; SADE; optimization; fault prediction

0 Introduction

Screw refrigeration compressors represent a primary type of refrigeration compressor and constitute the core of refrigeration systems. Their safe and stable operation carries significant socioeconomic importance, making accurate and timely fault prediction essential for ensuring normal compressor operation. Current fault prediction methods primarily include time series analysis [1], grey prediction [2], and artificial neural networks [3]. However, time series and grey prediction methods rely on assumptions that often diverge from actual conditions. While neural networks are widely used for prediction, they are based on empirical risk minimization, which theoretically requires infinite samples for optimal accuracy—an impractical condition for real-world fault data.

To address these limitations, Suykens et al. [4] proposed Least Squares Support Vector Machine (LSSVM) based on Support Vector Machine (SVM) [5]. LSSVM effectively overcomes neural network deficiencies while handling high dimensionality and nonlinearity, making it particularly suitable for equipment fault prediction with limited samples. Compared with traditional SVM, LSSVM simplifies computational complexity, offers faster convergence, and delivers higher prediction accuracy. Researchers such as Dai Linchao and Lian Guangyao [6,7] have successfully applied LSSVM to equipment fault prediction modeling.

In LSSVM models, the kernel function critically affects prediction accuracy. Single-kernel models cannot simultaneously achieve high learning capability and generalization performance. Consequently, Smits et al. [8] proposed a hybrid kernel SVM that combines local and global kernel functions, yielding superior learning and generalization compared to single kernels. Ma Yan et al. [9,10] subsequently applied hybrid kernels to network traffic prediction and reducer fault prediction with promising results.

Hybrid kernel LSSVM requires optimization of numerous parameters. Common optimization algorithms include the Fruit Fly Algorithm [11], Particle Swarm Optimization (PSO) [12], and Genetic Algorithm (GA) [13]. While these reduce dependency on initial value selection, they suffer from complex principles, tendency to fall into local optima, and other issues. Genetic algorithms often exhibit premature convergence, limited scale, poor nonlinear problem handling, and instability, requiring substantial computation for complex problems. PSO utilizes three position information sources to guide particle movement but demands numerous initialization parameters and easily becomes trapped in local optima for complex equipment fault prediction problems. The Fruit Fly Algorithm, though faster and more robust than PSO and GA, requires careful selection of initial coordinates and movement parameters based on experience, compromising stability due to its reliance on random variables.

Given the complexity of screw refrigeration compressor faults and the multi-parameter optimization challenge, these algorithms prove inadequate. Storn et al.' s Differential Evolution (DE) algorithm offers simpler structure and fewer control parameters, yet its population diversity diminishes across generations, risking premature convergence to local minima. To address this, we propose SADE (Self-Adaptive Differential Evolution), which enhances global search capability, robustness, and optimization speed through adaptive mechanisms. SADE obtains optimal kernel parameters, LSSVM parameters, and hybrid kernel adjustment parameters, which are then applied to compressor fault prediction, thereby validating the reliability and effectiveness of the SADE-optimized hybrid kernel LSSVM model.

---

1 Compressor Fault Prediction Model Establishment

1.1 Least Squares Support Vector Machine

Support Vector Machine (SVM), proposed in the mid-1990s, differs from neural networks by employing structural risk minimization instead of empirical risk minimization, achieving favorable statistical performance even with limited samples. LSSVM transforms SVM' s quadratic programming problem into a linear equation system, dramatically simplifying computation and accelerating convergence.

In feature space F, the LSSVM regression model can be expressed as:

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

where w represents the feature space weight vector and b denotes the bias term. Applying structural risk minimization principles and KKT conditions through Lagrangian multipliers, and utilizing a kernel function $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ satisfying Mercer' s condition, we obtain the regression function:

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x) + b$$

1.2 Hybrid Kernel Function Principle

Kernel functions enable spatial transformation, converting low-dimensional nonlinear problems into high-dimensional linear ones. Commonly used kernel functions include:

- Linear kernel: $K(x_i, x) = x_i^T x$
- Polynomial kernel: $K(x_i, x) = (x_i^T x + 1)^d$
- Radial Basis Function (RBF) kernel: $K(x_i, x) = \exp\left(-\frac{\|x_i - x\|^2}{2\delta^2}\right)$
- Sigmoid kernel: $K(x_i, x) = \tanh(\eta x_i^T x + \theta)$

These kernels fall into two categories: local and global kernels. RBF and polynomial kernels typify local and global kernels, respectively. By combining their advantages and ensuring Mercer' s condition, we construct the hybrid kernel function:

$$K_{hybrid}(x_i, x) = \rho K_{poly} + (1 - \rho) K_{RBF}$$

where $K_{poly} = (x_i^T x + 1)^d$, $K_{RBF} = \exp\left(-\frac{\|x_i - x\|^2}{2\delta^2}\right)$, d represents the polynomial kernel degree, and $\rho \in [0, 1]$ serves as the hybrid weight coefficient.

1.3 Self-Adaptive Differential Evolution Algorithm

Differential Evolution (DE) is a swarm intelligence algorithm simulating natural evolution, featuring simple principles, minimal parameter control, and easy implementation through mutation, crossover, and selection operations. The scaling factor F and crossover probability CR significantly influence DE performance. A larger F expands the search range for potential solutions, while a smaller F accelerates convergence but risks local optima. For CR, larger values maintain population diversity and facilitate optimal solution discovery, whereas smaller values enable stable search within the solution space.

Liu Hao et al. [16] utilized information from superior individuals to overcome evolutionary blindness and enhance search capability. Zhang Jinhua et al. [14] proposed a weighted mutation strategy dynamic DE algorithm, though without addressing parameter control. Li Longshu et al. [15] balanced global and local search through opposition-based learning and Gaussian distribution randomness, but this approach proves unsuitable for our fault prediction problem. Therefore, we improve DE through adaptive mutation strategy selection and parameter control.

1.3.1 Mutation Strategy Selection In population-based optimization, the early search phase requires strong global exploration to maintain diversity, while the later phase demands enhanced local exploitation for precision and convergence speed. The mutation operator directly impacts search range. Common mutation operators include:

- $DE/rand/1$: $V_i^{G+1} = X_{r1}^G + F(X_{r2}^G - X_{r3}^G)$
- $DE/rand/2$: $V_i^{G+1} = X_{r1}^G + F(X_{r2}^G - X_{r3}^G) + F(X_{r4}^G - X_{r5}^G)$
- $DE/best/1$: $V_i^{G+1} = X_{best}^G + F(X_{r1}^G - X_{r2}^G)$
- $DE/best/2$: $V_i^{G+1} = X_{best}^G + F(X_{r1}^G - X_{r2}^G) + F(X_{r3}^G - X_{r4}^G)$
- $DE/rand-to-best/1$: $V_i^{G+1} = X_i^G + F(X_{best}^G - X_i^G) + F(X_{r1}^G - X_{r2}^G)$

where $r1, r2, r3, r4, r5 \in \{1, 2, ..., NP\}$ are mutually distinct random indices, X_{best}^G denotes the best individual in generation G, and F is the scaling factor.

The $DE/rand$ family employs randomly selected base vectors without fixed search directions, providing strong global search capability. The $DE/best$ family leverages historical population information, delivering better convergence precision and local search ability. Combining these advantages, we formulate the adaptive mutation operator:

$$V_i^{G+1} = \lambda[DE/rand/1] + (1-\lambda)[DE/best/1]$$

To achieve adaptive mutation, λ gradually decreases from 1 to 0 during search, expressed as:

$$\lambda = 1 - \frac{G}{T}$$

where G is the current generation and T is the maximum generation count.

1.3.2 Scaling Factor Adaptation Based on the scaling factor characteristics, F should be large during early search and small in later stages. We adjust F using:

$$F = F_0 \times 2^{(1-\frac{G}{T})}$$

where F_0 is the initial value. This formulation yields larger F values in early stages to expand search space, and smaller values in later stages to enhance convergence speed and accuracy.

1.3.3 Crossover Probability Adaptation Similarly, crossover probability CR should be large initially and small later. We employ:

$$CR = (CR_{max} - CR_{min})\left(1 - \frac{G}{T}\right)^2 + CR_{min}$$

where CR_{max} and CR_{min} represent the maximum and minimum crossover probability bounds.

1.3.4 SADE Optimization Procedure The complete SADE optimization steps are:

a) **Initialization**: Set population size $NP \in [5, 10] \times D$ (where D is individual dimension), maximum iterations T, initial CR and F, and initialize population covering the entire search space:

$$x_{i,j}^{(0)} = x_{j,min} + rand[0,1] \cdot (x_{j,max} - x_{j,min})$$

for $i = 1, 2, ..., NP$ and $j = 1, 2, ..., D$, where $x_{i,j}^{(0)}$ is the j-th dimension of individual i, and $rand[0,1]$ denotes uniform random distribution in $[0,1]$.

b) **Mutation**: Generate mutant individuals V_i^{G+1} using the adaptive mutation operator (Equation 4).

c) **Crossover**: Perform crossover between mutant individuals V_i^{G+1} and parent individuals X_i^G:

$$U_{i,j}^{G+1} = \begin{cases} V_{i,j}^{G+1} & \text{if } rand(0,1) \leq CR \\ X_{i,j}^{G} & \text{otherwise} \end{cases}$$

d) **Selection**: Employ greedy selection to retain superior individuals:

$$X_i^{G+1} = \begin{cases} U_i^{G+1} & \text{if } f(U_i^{G+1}) \leq f(X_i^G) \\ X_i^G & \text{otherwise} \end{cases}$$

e) **Next Generation**: Set $G = G + 1$.

f) **Termination**: If termination criteria or maximum iterations T are met, output the optimal solution; otherwise, return to step b).

1.4 SADE-Based Hybrid Kernel LSSVM Prediction Model

Following the principles of hybrid kernels, LSSVM, and SADE improvements, establishing the SADE-based hybrid kernel LSSVM prediction model essentially involves integrating SADE into the hybrid kernel LSSVM training process to locate optimal parameters. The parameter combination optimized by SADE is (δ, C, d, ρ), where δ is the RBF kernel parameter, C is the penalty factor, d is the polynomial coefficient, and ρ is the hybrid kernel weight coefficient.

During training, real-world data suffers from missing values, noise, and inconsistencies, necessitating data preprocessing. After preprocessing, we initialize parameters including termination criteria, population size NP, scaling factor F, crossover probability CR, and the parameter combination (δ, C, d, ρ). The initial values train the hybrid kernel LSSVM via Equations (2) and (3) to obtain predictions, from which individual fitness is calculated using Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t^* - y_t)^2}$$

where L_n is the prediction horizon, y_t^* is the predicted value at time t, and y_t is the actual value. If termination criteria are unsatisfied, the algorithm proceeds to the next generation ($G = G + 1$), generating new parameter combinations through Equations 4-10 and the SADE procedure described in Section 1.3. This process repeats until termination, yielding the optimal parameter combination. The optimal parameters are then substituted into Equations (2) and (3) to establish the SADE-based hybrid kernel LSSVM prediction model. This model predicts test sample data, and its effectiveness is evaluated by comparing predictions with actual values using Equation (11). The complete modeling procedure is illustrated in [Figure 1: see original paper].

---

2 Experimental Validation

To verify algorithm effectiveness, we collected monitoring data from a cold chain company including temperature, humidity, high pressure, low pressure, and various voltage and current values—totaling 10 attributes. Data were sampled every 5 minutes over one day, comprising 200 training samples and 88 test samples. DE parameters were set as: maximum iterations $T = 200$, population size $NP = 50$, initial scaling factor $F = 0.5$, crossover probability bounds $CR_{min} = 0.2$ and $CR_{max} = 0.8$, RBF parameter δ and penalty factor C in $[2^{-15}, 2^{15}]$, weight coefficient $\rho = 0.9$, and polynomial coefficient $d = 1$. All data were normalized before SADE optimization, with the SADE optimization process shown in [Figure 2: see original paper].

The SADE-obtained optimal parameter combination was $\delta = 1.93$, $C = 43.26$, $d = 1.42$, and $\rho = 0.82$. These parameters trained the hybrid kernel LSSVM model for test sample prediction, with results compared against GA-based and PSO-based hybrid kernel LSSVM models. [FIGURE:3-5] compare predictions from PSO-LSSVM, GA-LSSVM, and our SADE-LSSVM against actual values.

Since screw refrigeration compressors stop when temperature reaches setpoints, values around 0.1 indicate shutdown periods, values exceeding 0.3 signify faults, and values between 0.2-0.3 represent normal operation. The figures clearly demonstrate that SADE-optimized predictions outperform both GA and PSO algorithms.

To quantitatively evaluate model effectiveness, we analyzed maximum relative error (E_{max}), RMSE, and runtime for all three models, as summarized in .

The SADE-LSSVM model achieved a maximum relative error of 3.1423%, representing reductions of 7.6931% and 4.6701% compared to GA-LSSVM and PSO-LSSVM, respectively. The RMSE also decreased by 0.006087 and 0.004699 compared to GA-LSSVM and PSO-LSSVM, confirming higher reliability. Runtime analysis shows SADE-LSSVM as the fastest, indicating SADE' s superior optimization speed over GA and PSO. Comprehensive analysis confirms that SADE-LSSVM surpasses GA-LSSVM and PSO-LSSVM in prediction accuracy, reliability, and computational efficiency, establishing it as a robust model for screw refrigeration compressor fault prediction.

---

3 Conclusion

Addressing the diversity of screw refrigeration compressor types and fault complexity, we adopted the hybrid kernel LSSVM model as a suitable solution. For its multi-parameter optimization challenge, we proposed SADE, which offers simpler theoretical structure, fewer parameters, and stronger search capability compared to alternative algorithms. Building upon traditional DE, SADE introduces optimizations in mutation strategy and parameter control. Experimental comparisons verify that the SADE-based hybrid kernel LSSVM model delivers superior reliability and accuracy over PSO-based and GA-based counterparts while requiring less computational time. Consequently, the proposed SADE-LSSVM model provides an effective solution for screw refrigeration compressor fault prediction.

References

[1] Dong R J, Pedrycz W. A granular time series approach to long-term forecasting and trend forecasting [J]. Physica A: Statistical Mechanics and its Applications, 2008, 387(13): 3253-3270.

[2] Zhang L, Ji P R, Du A H, et al. Comparison and application of several grey forecasting models for medium and long term power load forecasting [J]. Journal of China Three Gorges University: Natural Science Edition, 2009, 31(3): 41-46.

[3] Jin J, Rui Y N, Guo X H, et al. Study on fault prediction of large-scale compressor based on grey theory and radial basis function neural network [J]. Journal of Suzhou University: Engineering Science Edition, 2004, (5): 124-127.

[4] Suykens J A K, Vandewalle J. Least squares support vector machine classifiers [J]. Neural Processing Letters, 1999, 9(3): 293-300.

[5] Vapnik V N. The nature of statistical learning theory [M]. New York: Springer-Verlag, 1995.

[6] Dai L C, Wu L L, Zhao H N, et al. Fault prediction method based on

least squares support vector machine [J]. Journal of Central South University (Natural Science Edition), 2009, (S1): 253-257.

[7] Lian G Y, Lv X M, Huang K L, et al. Research on fault prediction model of complex equipment based on least squares support vector machine [J]. Computer Measurement and Control, 2011, 19(5): 1030-1032.

[8] Smits C F, Jordaan E M. Improved SVM regression using mixtures of kernels [C]//Proc of International Joint Conference on Neural Networks. 2002: 2785-2790.

[9] Ma Y, Yan B. Network traffic prediction model based on mixed kernel least squares support vector machine [J]. Microcomputer Application, 2015, 31(02): 50-53.

[10] Zhang H W, Zuo X Y, Pan H. Fault diagnosis of main reducer based on hybrid kernel learning support vector machine [J]. Computer Application and Software, 2017, 34(05): 91-98.

[11] Geng L Y, Chen L H. Prediction of railway freight volume based on FOA optimized mixed kernel LSSVM [J]. Computer Application Research, 2017, 34(02): 409-412.

[12] Li W Y, Yan H H, Yao H J. Application of least squares support vector machine based on particle swarm optimization in communication equipment fault prediction [J]. Microelectronics and Computer, 2013, 30(02): 99-102.

[13] Wang Y Y. Research on internal fault prediction model of large power transformer based on genetic algorithm [D]. Chongqing: Chongqing University, 2003.

[14] Zhang J H, Song L S, Zhang Y H, et al. Dynamic differential evolution algorithm with weighted variation strategy [J]. Computer Engineering and Application, 2017, 53(04): 156-162.

[15] Li L S, Weng Q Q. Adaptive differential evolution algorithm based on reverse learning [J]. Computer Application, 2018, 38(02): 399-404.

[16] Liu H. Differential evolution algorithm based on optimal learning strategy [C]//Proc of the 26th China Process Control Conference. 2015.

Note: Figure translations are in progress. See original paper for figures.