Soft Computing (2020) 24:15853–15869
https://doi.org/10.1007/s00500-020-04916-6

METHODOLOGIES AND APPLICATION

Industrial time series forecasting based on improved Gaussian process regression

Tianhong Liu[1,2] · Haikun Wei[3] · Sixing Liu[4] · Kanjian Zhang[3]

Published online: 30 April 2020

Abstract

Industrial processes often include shifting operating phases and dynamics, and system uncertainty. Industrial time series data may obey different distributions because of the time-varying characteristic. Therefore, a single global model cannot describe the local characteristics of multiple distributions. In this work, a hybrid GMM-IGPR model is proposed to solve this kind of time series prediction problem by using an improved Gaussian process regression (GPR) based on Gaussian mixture model (GMM) and a variant of the basic particles swarm optimization (PSO). In a first treatment to the time series, different distributions of the original dataset are characterized by adopting the GMM as a cluster method. Then, multiple localized GPR models are built to characterize the different properties between inputs and output within various clusters. In order to optimize the proposed algorithms, this paper utilizes the DEPSO which introduces differential evolution (DE) operator into the basic PSO algorithm to estimate hyperparameters of the GPR model, instead of using the traditional conjugate gradient (CG) method. Lastly, the Bayesian inference strategy is used to estimate the posterior probabilities of the test data with respect to different clusters. The various localized GPR models are integrated through these posterior probabilities as the weightings so that a global predictive model is developed for the final prediction. The effectiveness of the proposed algorithm is verified by means of a numerical example and a real industrial winding process. Statistical tests of experimental results compared with other popular prediction models demonstrate the good performance of the proposed model.

Keywords Predictive modeling · Industrial time series · Gaussian process regression · Gaussian mixture model · Particle swarm optimization

1 Introduction

Industrial data generally refer to a large amount of diversified time series generated at a high speed by automated industrial equipments and processes (General Electric Intelligent Platforms 2012). With the availability of large amounts of data, industries are increasingly looking toward new ways for extracting useful information. An accurate prediction of key variables of industrial processes plays a significant role in decision making in various kinds of applications such as industrial monitoring, prediction and diagnosing and becomes one of the most important aspects of industrial data applications. Due to the inherent time-varying characteristics and system uncertainty, industrial processes often show nonlinearity and non-Gaussianity features with shifting dynamics. A single global model based on all data cannot describe the local characteristics of multiple distributions. The shifting operating phases and dynamics may lead to biased and inefficient inferences if handled inappropriately. To solve this kind of prediction problem, the construction of an effective forecasting model should be considered.

Generally, time series forecasting techniques are either first principle or data-driven models. For complex industrial processes, the former methods usually require rich chemical

Communicated by V. Loia.

✉ Tianhong Liu
lthliu@163.com

[1] School of Information Engineering, Yangzhou University, Yangzhou 225127, People's Republic of China

[2] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, People's Republic of China

[3] Key Laboratory of Measurement and Control of CSE, Ministry of Education, School of Automation, Southeast University, Nanjing 210096, People's Republic of China

[4] School of Mechanical Engineering, Yangzhou University, Yangzhou 225127, People's Republic of China

or physical background knowledge and demand strict mathematical deduction. The lack of the background knowledge may restrict the rigorous theoretical modeling. The latter data-driven approaches depend upon historical measurement data with the minimal process knowledge requirement. These methods have gained extensive application in industrial time series forecasting (Zhao et al. 2012; Ferlito et al. 2017; He et al. 2016). Zhao et al. (2012) developed a two-phase data-driven-based forecasting and optimized adjusting method for gas real-time flow and gasholder-level prediction. In Ferlito et al. (2017), the attention was focused on eleven data-driven models to obtain 12 h ahead forecast of grid-connected photovoltaic plant production. He et al. (2016) achieved the multimode acid concentration prediction of cold-rolled strip steel pickling process by utilizing orthogonal signal correction-iteratively reweighted least squares (OSC-IRLS). Moreover, various machine learning methods such as artificial neural networks (ANN), support vector machine (SVM) and Bayesian inference are widely studied due to the nonlinear approximation capability of these models. Xu et al. (2014) developed a backpropagation neural network (BPNN) for the corner-wear prediction of a high-speed steel drill. An ANN-coupled multiobjective response surface methodology (MORSM) model had been developed by Bhowmik et al. (2018) for the performance-exhaust emission prediction of diesosenol-fueled diesel engine. Schenker and Agarwal (1995) combined a feedback neural network and the available partial process model to make the prediction of infrequently measurable quantities in poorly modeled processes. Liu and Chen (2013) constructed a just-in-time least squares support vector regression (JLSSVR) for online quality prediction of multigrade processes. A hybrid autoregressive fractionally integrated moving average and least square support vector machine (ARFIMA-LSSVM) model was proposed to forecast short-term wind power in Yuan et al. (2017). Herp et al. (2018) developed a statistical approach to abstract and predict turbine states in a Bayesian framework.

Apart from the above commonly used modeling methods, Gaussian process regression (GPR) (Rasmussen and Williams 2006) is also a popular data-driven approach for forecasting. In this work, GPR is applied to tackle the complex time series with shifting dynamics for nonlinear prediction. It provides a flexible nonparametric Bayesian model that permits a prior probability distribution to be defined over functions directly (Rasmussen and Williams 2006; Bishop 2006; Nabney 2002). One significant advantage of the GPR model over many other machine learning methods consists in its seamless integration of many machine learning tasks, including model training, hyperparameters and uncertainty estimation; thereby, the regression process is streamlined significantly and the results are less affected by subjectivity and more interpretable (Sun et al. 2014). It has been widely focused on by many researchers. Aye and Heyns (2017) proposed an integrated GPR model for the prediction of remaining useful life of slow-speed bearings and achieved lower prediction error. In Yu et al. (2013), a Bayesian model averaging-based multikernel Gaussian process regression (BMA-MKGPR) approach was developed for state estimation and quality prediction of nonlinear batch processes with multiple operating phases and between-phase transient dynamics. Ranjan et al. (2016) focused on estimation of parameters for robust GPR model which used heavy-tailed distributions instead of using normal distribution for modeling noise. Gregorčič and Lightbody (2009) introduced the GPR prior approach for the modeling of nonlinear hydraulic positioning system. Jin et al. (2015) put forward an online ensemble GPR soft sensor for nonlinear time-varying batch processes. Yu (2012) used GPR to predict the Tennessee Eastman Chemical process. Yang et al. (2016) proposed an ensemble just-in-time (JIT) Gaussian process regression (EJITGPR) framework for online quality prediction of industrial batch rubber mixing process.

The accuracy of the GPR model lies on whether the historical data can reflect the dynamic characteristics of the practical application processes and the predictive ability of the model. In the actual industrial processes, on the one hand, industrial processes may have many different possible states and conditions, and new data are sampled as new process state emerged. It is invalid to assume that the data are subject to a single distribution due to the multimode behavior of the processes. A single model-based prediction tends to be unreliable. Even though the deterministic strategies can classify the process data into different modes, the multiple local models which are built within these modes may not be the best option due to the uncertainty in the processes. An effective way to settle this problem is to identify the operating modes by estimating the probability density function of the input space data or by clustering the input data (Grbić et al. 2013). GMM is performed as a clustering method to describe the shifting operation conditions in this paper. It assumes that the data in each component follow Gaussian distribution (Reynolds 2015). Gaussian distribution is a good default choice in the context of lacking prior knowledge about what form a distribution over the real numbers should take (Goodfellow et al. 2016). The GMM can well describe the distribution and characteristics of training data in the parameter space which has the advantages of being simple and efficient. It has been successfully applied for clustering in many different applications (Nowakowska et al. 2015; Scrucca 2016; Elguebaly and Bouguila 2015).

On the other hand, for the purpose of improving the predictive ability of the model, an optimization algorithm is introduced to optimize the hyperparameters. As a Gaussian process is specified by its mean and covariance functions, optimization of hyperparameters is an important part of modeling, and it affects the reliability of the regression model. One of the most commonly used methods is the conjugate gradient (CG) algorithm (Rasmussen and Williams 2006). However, this algorithm relies heavily on the selection of initial value and it is hard to determine the number of iterations. Besides, for most of the GPR estimation, optimization of hyperparameters is not a convex optimization problem. The CG method can easily lead to local optimum. Therefore, a variant of the particle swarm optimization (PSO) algorithm is applied for the optimization of hyperparameters of the GPR model in this paper. PSO (Kennedy and Eberhart 1994) is a population-based search optimization technique. It is an evolutionary algorithm based on the flock foraging swarm intelligence relationship. Compared with the CG algorithm, this technique has the advantage of simple implementation, fewer control parameters and better convergence performance due to its information-sharing mechanism. It does not require a good initial solution to start the iteration and is less sensitive to the nature of objective function. PSO has been combined with different models in many different domains (Pradeepkumar and Ravi 2017; López et al. 2017; Sun et al. 2017; Singh and Borah 2014). Xu et al. (2011) use the PSO to optimize the hyperparameters of the GPR and form the PSO-GPR model. This model can improve the performance. Despite its simplicity and efficiency, conventional PSO suffers from two main limitations: premature convergence and being trapped in local minima. A major concern in the control of PSO performance is exploration and exploitation balance because excessive exploration wastes computational resources, while excessive exploitation leads to premature convergence. Many modified PSO variants have been proposed to improve the performance of original PSO. One of the hybrid versions, which introduces differential evolution (DE) (Xie et al. 2002) operator into PSO algorithm, is applied in this paper. This modified algorithm can maintain the diversity and avoid the problem of prematurity and easily trapping in local minima. By improving the optimization efficiency of the GPR model, the performance of the original GPR can be improved.

In this paper, a hybrid model based on GMM and an improved GPR (GMM-IGPR) is developed for complex industrial time series forecasting. As for industrial process with different operating phases, a single global GPR model may be unable to characterize the working conditions. The global modeling method is simpler but is more vulnerable to different distribution values. To solve this problem, the input data are categorized into different clusters by using the GMM firstly. The GMM not only gives the cluster of the data points, but also gives the probability of the data belonging to this cluster. Then localized GPR models are built to characterize the different properties between inputs and output. The hybrid particle swarm with differential evolution operator, termed DEPSO, is served as the GPR parametric optimizer to replace the traditional conjugate gradient method. Finally, the various localized GPR models are integrated and the proposed method is compared in terms of performance with several commonly used approaches. The main contributions of this paper can be summarized as follows: (i) The combination of the GMM and GPR is proposed with the distributions of the data identified by using the GMM; (ii) an improved PSO algorithm is employed to optimize the hyperparameters of the GPR model which are built to characterize the different properties of the data; (iii) the posterior probabilities of new samples which correspond to the likelihoods of different operation modes emerging in the processes are considered as the weighting coefficients in the final prediction model to cope with the uncertainty in the processes.

The rest of the paper is organized as follows. Section 2 presents the basic methodology used in this study. The proposed method for industrial time series prediction is given in Sect. 3. In Sect. 4, the model is evaluated on a simulated example and a real industrial winding process. The experimental procedures and performances are outlined. Finally, conclusions are given in Sect. 5.

2 Theoretical aspects

2.1 Gaussian mixture model

GMM is a parametric probability distribution for continuous random variables where the density function is defined as the weighted sum of multiple Gaussian densities (Reynolds 2015).

Consider a superposition of K Gaussian densities of the form

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(x|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \tag{1}$$

where $\mathbf{x}$ is the random variable. Each Gaussian density $\mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$ is called a *component* of the mixture and has its own mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$. The parameters π_k in Eq. (1) are called *mixing coefficients* which satisfies $\sum_{k=1}^{K} \pi_k = 1$ and $0 \leq \pi_k \leq 1$.

The form of the Gaussian mixture distribution is governed by the parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, which is denoted by $\boldsymbol{\pi} \equiv \{\pi_1, \ldots, \pi_K\}$, $\boldsymbol{\mu} \equiv \{\mu_1, \ldots, \mu_K\}$, $\boldsymbol{\Sigma} \equiv \{\Sigma_1, \ldots, \Sigma_K\}$. One way to set the values of these parameters is to use maximum likelihood:

$$\begin{aligned} \ln\ p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \sum_{n=1}^{N} \ln\ p(\mathbf{x}_n|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \end{aligned} \tag{2}$$

where $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. Maximizing the log-likelihood function in Eq. (2) for a GMM turns out to be a more complex problem than for the case of a single Gaussian due to the presence of the summation over k that appears inside the logarithm. As a result, the maximum likelihood solution for the parameters no longer has a closed-form analytical solution. Alternatively we can employ a powerful framework called EM algorithm to estimate the parameters.

For N observations $X = [x_1, x_2, \ldots, x_N]$, we use $Z = [z_1, z_2, \ldots, z_N]$ for the corresponding unobserved data. Before maximizing the likelihood function, a quantity that will play an important role is given out. This quantity called posterior probabilities or responsibilities is the conditional probability of Z given X.

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j))} \tag{3}$$

Set the derivatives of $\ln\, p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to the means $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ and π_k of the Gaussian components to zero, respectively, in Eq. (2). First, initialize the means $\boldsymbol{\mu_k}$, covariances $\boldsymbol{\Sigma_k}$, mixing coefficients π_k, and evaluate the initial value of the likelihood. Then the estimation procedures contain the following two steps called: *expectation* step (E step) and *maximization* step (M step). In the E step the current values of the parameters are used to evaluate the posterior probabilities or responsibilities $\gamma(z_{nk})$ as given in Eq. (3). Then these probabilities are used in the M step, to re-estimate the means, covariances and mixing coefficients. Finally, if the convergence criterion is satisfied, stop the iteration; if not, return to the E step. The optimization process consists of initializing the parameter values and iteratively updating until convergence. At each iteration n, the parameters of GMM are updated as follows:

$$\boldsymbol{\mu}_k^{(n+1)} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma^{(n)}(z_{nk})x_n \tag{4}$$

$$\boldsymbol{\Sigma}_k^{(n+1)} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma^{(n)}(z_{nk})\left(\mathbf{x}_n - \boldsymbol{\mu}_k^{n+1}\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_k^{n+1}\right)^{\mathrm{T}} \tag{5}$$

$$\pi_k^{(n+1)} = \frac{N_k^{(n)}}{N} \tag{6}$$

where $N_k = \frac{1}{\sum_{n=1}^{N}\gamma(z_{nk})}$. By using an adequate number of Gaussian, and by adjusting their parameters in the linear combination, almost any continuous density can be approximated to arbitrary precision (Bishop 2006).

Traditionally the parameter update method used by GMM is common batch EM algorithm which requires the whole data to be available at each iteration. In this paper, we apply an online variant of the EM which makes it possible to update the parameters without storing the data. Online EM algorithms are incremental, in which at each EM cycle only one data point is processed at a time. There are two main approaches in the literature. The first one is incremental EM (Neal and Hinton 1998). The second one is known as stepwise EM (Cappé 2011; Sato and Ishii 2000; Cappé and Moulines 2009). The incremental EM optimizes the lower bound of the observations log-likelihood sequentially and requires storing the expected sufficient statistics for each data. The stepwise EM which is based on stochastic approximation theory requires only constant memory use. In this study, the online EM proposed by Cappé and Moulines (2009) is considered. The main idea is to replace a stochastic approximation step with the expectation step of the common EM algorithm Titterington (1984), while keeping the maximization step unchanged. Consider the case of Gaussian mixture, a data point m is performed an update. In the E step the corresponding old and new values of the responsibilities are denoted $\gamma^{\mathrm{old}}(z_{mk})$ and $\gamma^{\mathrm{new}}(z_{mk})$. Then the means $\boldsymbol{\mu}_k^{\mathrm{new}}$, covariances $\boldsymbol{\Sigma}_k^{\mathrm{new}}$, mixing coefficients π_k^{new} in the M step are given as:

$$\boldsymbol{\mu}_k^{\mathrm{new}} = \boldsymbol{\mu}_k^{\mathrm{old}} + \left(\frac{\gamma^{\mathrm{new}}(z_{mk}) - \gamma^{\mathrm{old}}(z_{mk})}{N_k^{\mathrm{new}}}\right)\left(\mathbf{x}_m - \boldsymbol{\mu}_k^{\mathrm{old}}\right) \tag{7}$$

$$\begin{aligned}\boldsymbol{\Sigma}_k^{\mathrm{new}} &= \boldsymbol{\Sigma}_k^{\mathrm{old}} \\ &\quad + \left(\frac{\gamma^{\mathrm{new}}(z_{mk}) - \gamma^{\mathrm{old}}(z_{mk})}{N_k^{\mathrm{new}}}\right)\left(\mathbf{x}_m - \boldsymbol{\mu}^{\mathrm{new}}\right) \\ &\quad \left(\mathbf{x}_m - \boldsymbol{\mu}^{\mathrm{new}}\right)^{\mathrm{T}}\end{aligned} \tag{8}$$

$$\pi_k^{\mathrm{new}} = \frac{N_k^{\mathrm{new}}}{N} \tag{9}$$

Together with

$$N_k^{\mathrm{new}} = N_k^{\mathrm{old}} + \gamma^{\mathrm{new}}(z_{mk}) - \gamma^{\mathrm{old}}(z_{mk}) \tag{10}$$

Thus, both the E step and the M step take fixed time that is independent of the total number of data points. Owing to space constraints, detailed derivation process can be referred to the literature (Cappé and Moulines 2009).

Besides the internal parameters of the model, one parameter that has to be validated is the number of components K. Validation step is important in order to prevent the model from overfitting, and many model selection criteria have been proposed to combat this problem. Two most popular criteria for model selection are Akaike's information criterion (AIC) and Bayesian information criterion (BIC).

2.2 Gaussian process regression

A GPR is a nonparametric probabilistic approach for modeling and forecasting. Over the past decade, Gaussian process regression has attracted much attention in machine learning because it provides a principled, practical and probabilistic approach for kernel learning machines (Rasmussen and Williams 2006). For a multiple-input single-output (MISO) nonlinear system, let $D = \{(x_i, y_i), i = 1, 2, \ldots, n\}$ denotes a set of input–output data, where n is the number of data points. The input and output samples are $\mathbf{x} = [x_1\ x_2 \ldots\ x_n]^{\mathrm{T}}$ and $\mathbf{y} = [y_1\ y_2 \ldots\ y_n]^{\mathrm{T}}$. The regression model can be formulated as

$$\mathbf{y} = f(\mathbf{x}) + \epsilon \tag{11}$$

where f is the true underlying function and $\mathbf{y}$ is the observed target value. ϵ is assumed to be an independent, identically distributed Gaussian distribution with zero mean and variance σ_n^2, i.e.,$\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. GPR model is used to infer the latent function $f(\cdot)$ from the dataset D. It can be completely specified by its mean and covariance functions, written as $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}^{'}))$. The mean function $m(\mathbf{x})$ encodes the central tendency and is often assumed to be zero. The covariance function $k(\mathbf{x}, \mathbf{x}^{'})$ encodes information about the shape and structure the function expected to have. The corresponding mean and covariance functions are defined as:

$$m(\mathbf{x}) = E[f(\mathbf{x})] \tag{12}$$

$$k(\mathbf{x}, \mathbf{x}^{'}) = E[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \tag{13}$$

A common covariance function is the squared exponential function (SE) (Tobias et al. 2006), which is written as $k(\mathbf{x}, \mathbf{x}^{'}) = \sigma_f^2 exp\left(-\frac{(x-x^{'})^2}{2l^2}\right) + \sigma_n^2 \delta_{x,x'}$ or $\mathrm{cov}(y) = K(X, X) + \sigma_n^2 \mathbf{I}$, where σ_f is the standard deviation of the signal which controls the prior variance and l is an isotropic length scale parameter that controls the rate of decay of the covariance. $\delta_{x,x'}$ is a Kronecker delta which is one iff $x = x^{'}$ and zero otherwise. Then the likelihood is given by

$$p(\mathbf{y}|\mathbf{f}) \sim \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma_n^2 \mathbf{I}) \tag{14}$$

where $\mathbf{I}$ is the identity matrix. The goal of the regression model is to predict the target value $y_* \in R$ at a new point x_*. The joint distribution of the training outputs $\mathbf{y}$ and the test output y_* according to the prior is

$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} = \left(\begin{bmatrix} \mathbf{f} \\ f_* \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon} \\ \epsilon_* \end{bmatrix}\right) \tag{15}$$

$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X, X) + \sigma_n^2 \mathbf{I} & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right) \tag{16}$$

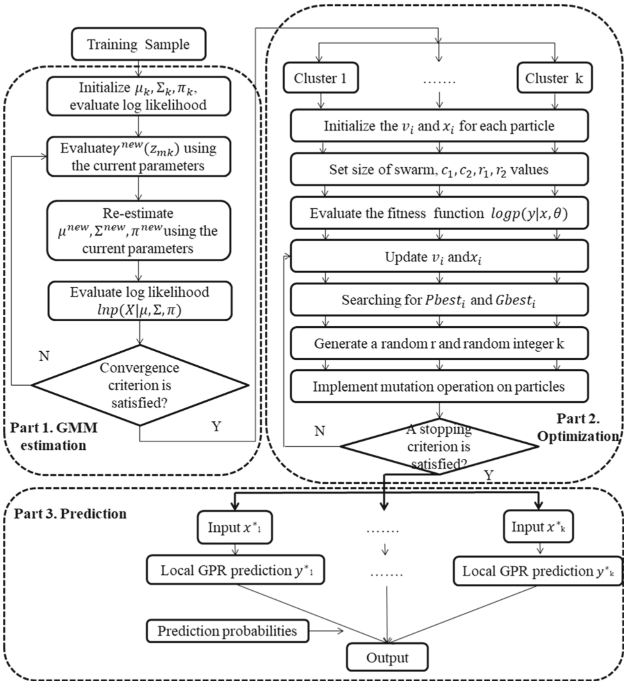
Training Sample
Initialize μ_k, Σ_k, π_k, evaluate log likelihood
Evaluate$\gamma^{new}(z_{mk})$ using the current parameters
Re-estimate $\mu^{new}, \Sigma^{new}, \pi^{new}$using the current parameters
Evaluate log likelihood $lnp(X|\mu, \Sigma, \pi)$
N
Convergence criterion is satisfied?
Y
Part 1. GMM estimation
Cluster 1
....
Cluster k
Initialize the v_i and x_i for each particle
Set size of swarm, c_1, c_2, r_1, r_2 values
Evaluate the fitness function $logp(y|x, \theta)$
Update v_i andx_i
Searching for $Pbest_i$ and $Gbest_i$
Generate a random r and random integer k
Implement mutation operation on particles
N
A stopping criterion is satisfied?
Y
Part 2. Optimization
Part 3. Prediction
Input x^*_1
....
Input x^*_k
Local GPR prediction y^*_1
....
Local GPR prediction y^*_k
Prediction probabilities
Output

Fig. 1 Flow diagram of the GMM-IGPR model

where

$$K(X, X) = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k(\mathbf{x}_1, \mathbf{x}_n) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k(\mathbf{x}_2, \mathbf{x}_n) \\ \cdots & \cdots & \ddots & \cdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & k(\mathbf{x}_n, \mathbf{x}_2) & \cdots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix} \tag{17}$$

The prediction output y_* and variance Σ_* are given by

$$y_* = K(X_*, X)(K(X, X) + \sigma_n^2 \mathbf{I})^{-1} y \tag{18}$$

$$\Sigma_* = K(X_*, X_*) - K(X_*, X)(K(X, X) + \sigma_n^2 \mathbf{I})^{-1} K(X, X_*) \tag{19}$$

The inverse of the covariance matrix $(K(X, X) + \sigma^2 \mathbf{I})^{-1}$ can be calculated using the Cholesky decomposition (Rasmussen and Williams 2006).

It can be seen from the above discussion, covariance function $\mathrm{cov}(y)$ exerts an important influence on GPR, because it controls how much the data are smoothed in estimating the unknown function. Thus, the parameter selection of the SE covariance function directly affects the performance of the Gaussian process. We rewrite Eq. (14) specifically as

$$\log\ p(\mathbf{y}|X, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^{\mathrm{T}} K_y^{-1} \mathbf{y} - \frac{1}{2}\log|K_y| - \frac{n}{2}\log 2\pi \tag{20}$$

where $K_y = K(X, X) + \sigma_n^2 \mathbf{I}$. The parameters can be optimized by maximizing this marginal likelihood. In Eq. (20), only the first term involves the observed targets; the second

Table 1 Criterion of MAPE

MAPE (%)	Forecasting power
10 <	Excellent
10 − 20	Good
20 − 50	Reasonable
> 50	Incorrect

term is the complexity penalty depending only on the covariance function and the inputs; $n\log(2\pi)/2$ is a normalization constant. In this paper, an improved PSO is introduced into GPR model to replace the gradient descent algorithm.

2.3 Particle swarm optimization

2.3.1 Standard PSO

PSO is an optimization technology proposed by Kennedy and Eberhart (1994). It originates from the concepts of swarm intelligence and evolutionary computation. This technique is considered as a useful tool for finding the parameters required for maximizing a particular objective.

In PSO, each swarm has a population of N particles, and each particle in the swarm represents the candidate solution to an optimization problem. The particle is expressed by the position vector x_i and velocity vector v_i, which is represented by $x_i = \{x_{i1}, x_{i2}, \ldots, x_{iD}\}$ and $v_i = \{v_{i1}, v_{i2}, \ldots, v_{iD}\}$, respectively, when searching a D-dimensional problem. PSO initializes the particles randomly over the search space with random velocity values. Then, each particle's velocity is updated using its previous historical best position $pbest_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$. These particles search along the entire space to find the whole swarm's best experience $gbest_i = (g_{i1}, g_{i2}, \ldots, g_{iD})$. The specific update processes can be expressed as follows (Shi and Eberhart 1998):

$$v_{iD}(t+1) = wv_{iD}(t) + c_1r_1(pbest_{iD}(t) - x_{iD}(t)) + c_2r_2(gbest_{iD}(t) - x_{iD}(t)) \quad (21)$$

$$x_{iD}(t+1) = x_{iD}(t) + v_{iD}(t+1) \quad (22)$$

where $v_{iD}(t)$ is the velocity of ith particle at tth iteration, $x_{iD}(t)$ is the position of ith particle at tth iteration, w is an inertia weight; c_1 and c_2 are acceleration coefficients; r_1 and r_2 are two randomly generated numbers in the range of [0,1]. Generally speaking, the inertia weight reflects the succession of the current velocity and is designed to balance exploration and exploitation search capabilities. A small inertia weight is convenient for local search, and a large inertia weight is more suitable for global search. In Eq. (21), w decreases according to the linearly decreasing weight (LDW) strategy: $w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} * iter$, with the maximum inertial factor $w_{\max} = 0.9$ and the minimum inertial factor $w_{\min} = 0.4$. In the above update functions, the suitable values of parameters have an important effect on the convergence of a swarm.

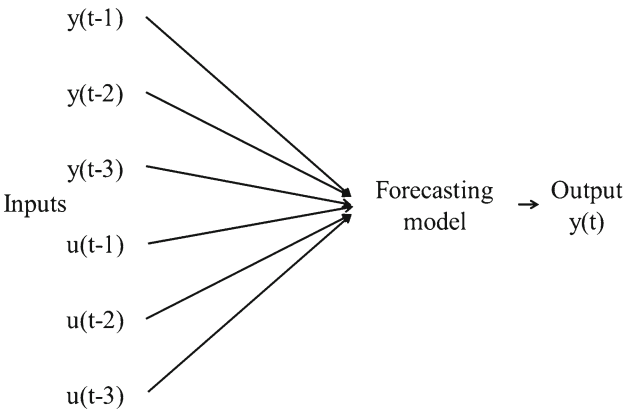

Fig. 2 Inputs and output of the numerical example

2.3.2 DEPSO

In the traditional PSO, the concept of a more-or-less permanent social topology is fundamental to PSO (Kennedy and Eberhart 1994; Kennedy 1997), which means the *pbest* and *gbest* should not be too closed to make some particles inactive in certain stage of evolution (Higashi and Iba 2003; Xie et al. 2002). In the latter stage of evolution process, the *pbest* and *gbest* either do not change or change very little. If v_{iD} value is small in equation (20), it cannot back to large value and loses exploration capability in some generations. When the v_{iD} is close to zero, it will be damped quickly with the ratio w.

In order to keep the diversity of the particle swarm, we need to make sure that the speed will not be smaller and smaller. Instead, a large update can be restored when a certain limit is reached. In the DEPSO algorithm, PSO maintains the particle swarm dynamics while keeps population diversity with the differential evolution. After the *pbest* and *gbest* of the swarm have been obtained, the mutations are provided by DE operator on the *pbest*, with a trail point $Trail_i$, whose dth dimension is

$$\begin{aligned} &if\ (rand() < CR \quad or \quad d == k), \\ &\ then \quad Trail_{id} = g_{id} + F \cdot (p_{1d} + p_{2d} - p_{3d} - p_{4d}) \end{aligned} \quad (23)$$

where rand() is a random number within [0,1]; CR is a crossover constant $\in$ [0, 1]; k is a random integer value within $[1, D]$; F is a real and constant factor $\in$ [0, 2] which controls the amplification of the differential variation; p_{1d}, p_{2d}, p_{3d}, p_{4d} are chosen randomly from *pbest*. The mutation is performed on the *pbest* in order to prevent the swarm from disorganizing by unexpected fluctuations.

Table 2 The performance criteria values of different prediction models

Models	GMM-IGPR	ANN	SVR	GPR
RMSE	1.8264	2.1805	2.4118	2.3842
MAE	1.4753	1.6193	1.9068	1.8744
MAPE(%)	0.6479	1.1032	1.1497	0.9638
R	0.9743	0.9626	0.9605	0.9741

$Trail_{id}$ will replace $pbest_i$ only if it is better than $pbest_i$. The original PSO operator and the DE operator will be performed alternately.

3 Forecasting procedure of GMM-IGPR

In the current paper, a hybrid model based on GMM and DEPSO-improved GPR (GMM-IGPR) is proposed for complex industrial process time series forecasting. The specific forecasting procedures of the proposed model mainly contains five steps and is summarized as follows:

Step 1. Samples are prepared and the training inputs are first used to estimate the GMM so that the different components are identified. The probability distribution parameters $\mathbf{9} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K\}$ are estimated by online EM algorithm (Eqs. 7–9).

Step 2. The training inputs are categorized by maximizing the posterior probabilities:

$$k\,(x_i) = \underset{j}{argmax}\, p(\Psi_j|x_i) \tag{24}$$

where $k\,(x_i)$ denotes the particular cluster that input x_i is classified into. The clustered training samples can be denoted by $\left\{(X^{(1)}, Y^{(1)}), \ldots, (X^{(K)}, Y^{(K)})\right\}$ because the corresponding output y_i is attributed into the same cluster. For a new data point x_t, the posterior probability which denotes that the probability of x_t belonging to the j-th cluster is given by

$$p(\Psi_j|x_t) = \frac{\pi_j \mathcal{N}(y_t|\mu_j, \Sigma_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(y_t|\mu_k, \Sigma_k))} \tag{25}$$

Step 3. Optimize the hyperparameters of the GPR for each cluster.

Step 3.1. Set the size of swarm, the values of c_1, c_2, r_1, r_2 and initialize the velocity v_i and position x_i for each particle. Set the velocity range $[v_{\min}, v_{\max}]$ and the position range $[x_{\min}, x_{\max}]$ to refrain the dimensional value from moving too far out of the search space;

Step 3.2. Evaluate fitness function Eq. (19) of each particle $x_i(t)$;

Step 3.3. Implement mutation operation on particles. Generate a random integer value k within $[1, D]$ and a random number within [0,1]. The dth dimension of the trail point is given in Eq. (22).

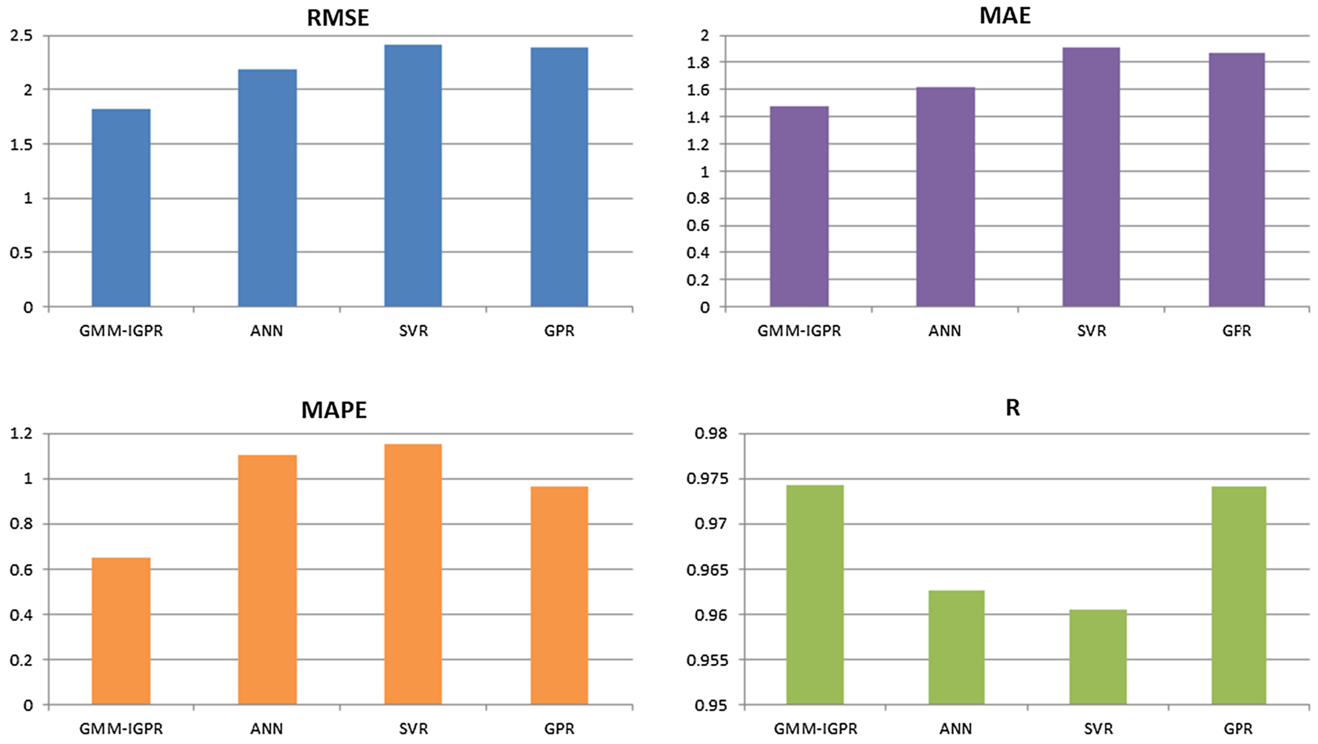

Fig. 3 The visualization of prediction performances

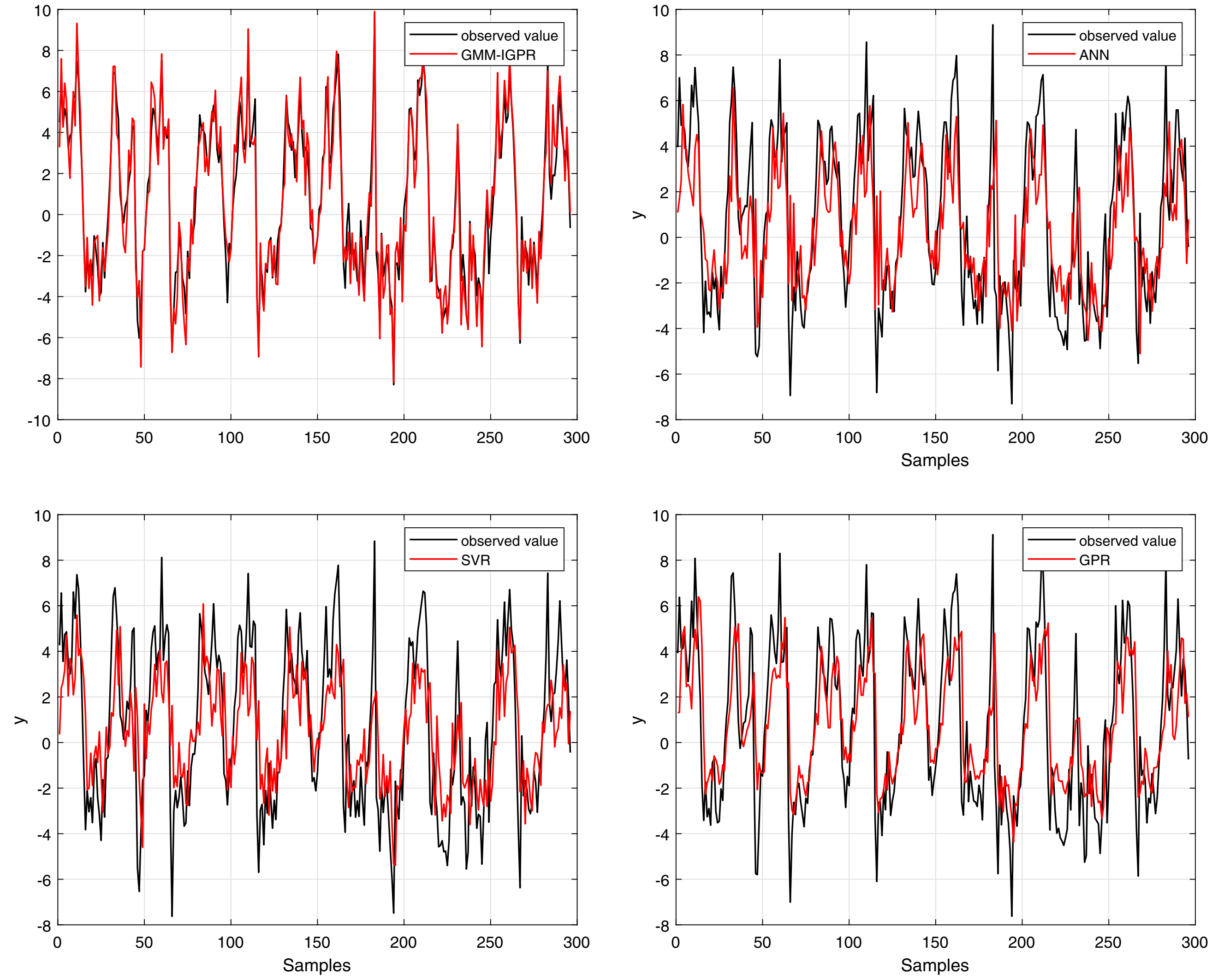

Fig. 4 The prediction performances of different models

Step 3.4. Update the velocity and position of the particles iteratively according to Eqs. (21) and (22). The inertia weight is updated by using LDW strategy.
Step 3.5. If the predefined number of iterations I_{it} is reached or an optimum solution is found, stop the procedure; otherwise, repeat until a stopping criterion is satisfied.

Step 4. Construct the localized GPR models. The output of a localized GPR model for x_t within the j-th cluster is given by

$$y_t^{(j)} = k(x_t, X^{(j)})^{\mathrm{T}}(k(X^{(j)}, X^{(j)}) + \sigma^2\mathbf{I})^{-1}Y^{(j)} \tag{26}$$

Step 5. Ensemble all the localized models by integrating them through Bayesian inference-based posterior probabilities

$$\begin{aligned} y_t &= \sum_{j=1}^{K}\left\{\hat{y}_t^{(j)} p\left(\Psi_j | x_t\right)\right\} \\ &= \sum_{j=1}^{K}\left\{\frac{\pi_j \mathcal{N}(y_t|\mu_j, \Sigma_j)k(x_t, X^{(j)})^{\mathrm{T}}(K_y^j)^{-1}Y^{(j)}}{\sum_{k=1}^{K}\pi_k \mathcal{N}(y_t|\mu_k, \Sigma_k))}\right\} \end{aligned} \tag{27}$$

The above forecasting procedure of the GMM-IGPR approach is illustrated in Fig. 1

4 Application examples

In order to verify the reliability of the proposed method and quantify its performance, we test it on two examples. The

Table 3 The impact of parameter optimization on prediction performance

Models	GMM-IGPR	GMM-GPR
RMSE	1.8264	2.1537
MAE	1.4753	1.8629
MAPE(%)	0.6479	0.9230
R	0.9643	0.9712

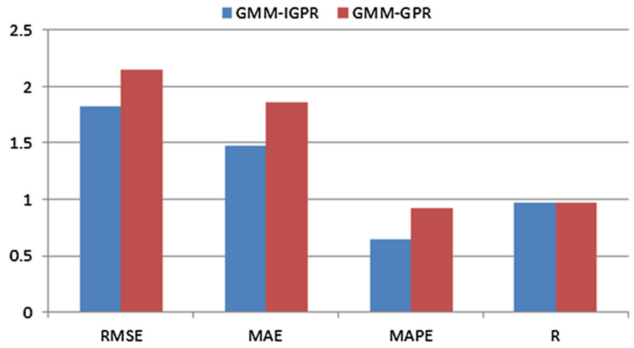

Fig. 5 The comparison of forecasting results

first one is a numerical simulation example which is used for the improved forecasting model demonstration. The second one is a real industrial winding process application case. The results of experiments are compared with several forecasting methods including artificial neural network (ANN), support vector regression (SVR), Gaussian process regression (GPR) and GMM-GPR models. The structure of ANN (BP neural network) model is determined by the trials. For the SVR model, the radial basis function (RBF) is selected as the kernel function. Two parameters, the regularization parameter C and the kernel function parameter σ, need to be identified. The parameters for GPR are determined based on the stochastic gradient descent (SGD). The GMM-GPR model applies GMM to identify the clusters of the original samples and then construct local models. In this study, four commonly used performance criteria are adopted to evaluate the accuracy of the forecasting models, including root-mean-squared error (RMSE) (Eq.28), mean absolute error (MAE) (Eq.29), mean absolute percentage error (MAPE) (Eq.30) and correlation coefficient (R) (Eq.31).

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[f(i) - y_i]^2} \tag{28}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|f(i) - y_i| \tag{29}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{f(i)}{y_i} - 1\right| \times 100\% \tag{30}$$

Table 4 Friedman ranks (numerical example)

Models	GMM-IGPR	ANN	SVR	GPR	GMM-GPR
Avg. rank	1.00	3.00	5.00	3.67	2.33

Table 5 Test statistics

N	3
Chi-square	10.667
df	4
Asymp.sig	0.031

Friedman test

Table 6 p values of the comparisons between our proposal and the others

Model 1–Model 2	Sig.
GMM-IGPR - GMM-GPR	0.0390
GMM-IGPR - ANN	0.0302
GMM-IGPR - GPR	0.0213
GMM-IGPR - SVR	0.0121

$$R = \frac{\sum_{i=1}^{N}(f(i) - \bar{f})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(f(i) - \bar{f})^2} \cdot \sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{31}$$

In Eqs. (27)–(30), N is the number of observed samples, and $f(i)$ and y_i $(i = 1, 2, \ldots, N)$ denote the predicted value and actual value at time t, respectively. And $\bar{f}$ and $\bar{y}$ denote the mean of the predicted values and the actual values, respectively. All of them are ways of comparing forecasts with their eventual outcomes. RMSE represents the sample standard deviation of the differences between predicted values and observed values. MAE reflects the average error that reveals the similar magnitude between the predicted values and observed values. The criterion of MAPE corresponding to the forecasting power is listed in Table 1 (Liu et al. 2016). R is a numerical measure of correlation, meaning a statistical relationship between two variables. All the models are built and tested by using MATLAB software. For the proposed GMM-IGPR model, the GPML (Gaussian processes for machine learning) toolbox (Rasmussen and Nickisch 2010) is used to offer some helps.

4.1 A simulation study

In the simulation study, a nonlinear function is used as a numerical simulation experiment in order to verify the validity of the model. Time series data are generated according to the following nonlinear equations (Narenda & Li,1996):

$$y(t) = \frac{x_1(t)}{1 + 0.5\sin(x_2(t))} + \frac{x_2(t)}{1 + 0.5\sin(x_1(t))} + \varepsilon(t) \tag{32}$$

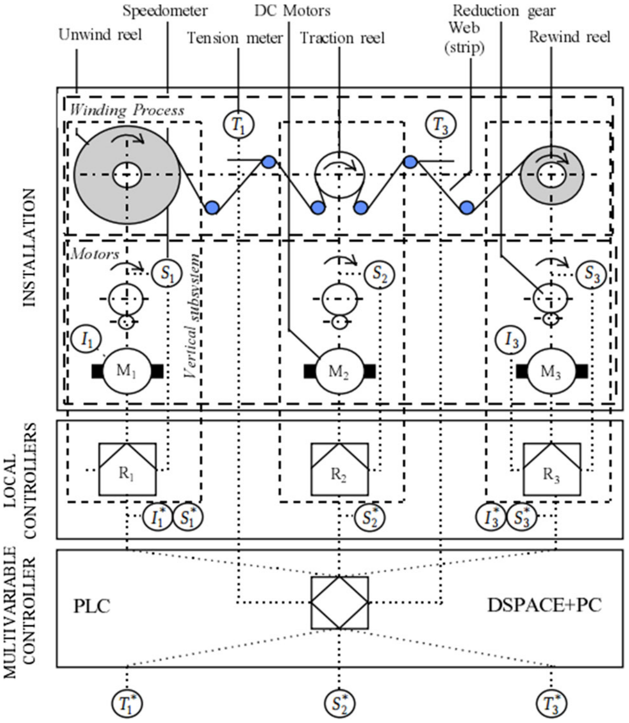

Fig. 6 Winding pilot system

Table 7 Input and output variables in the winding process

Input variables	Output variables
Angular speed of reel 1 $(S1)(x_1)$	
Angular speed of reel 2 $(S2)(x_2)$	Tension in the web between reel 1 and 2 $(T1)(y_1)$
Angular speed of reel 3 $(S3)(x_3)$	Tension in the web between reel 2 and 3 $(T3)(y_2)$
Setpoint current at motor 1 $(I1*)(x_4)$	
Setpoint current at motor 2 $(I3*)(x_5)$	

Table 8 The statistical information of winding process data

Var.	Max	Median	Min	Mean	Std
x_1	5.4800	− 0.0400	− 6.6000	6.0000e− 05	1.0002
x_2	1.6300	0.4150	− 5.1000	− 8.0000e− 05	1.0002
x_3	3.1900	− 0.0300	− 7.2000	− 1.6000e− 04	1.0003
x_4	1.8900	0.5300	− 2.1300	− 3.6000e− 05	1.0001
x_5	1.1600	0.7900	− 1.3700	− 9.2000e− 05	1.0002
y_1	2.4200	− 0.2500	− 1.5400	− 9.2000e− 05	1.0002
y_2	4.8900	0.1000	− 3.3000	4.0000e− 06	1.0003

$$x_1(t+1) = \left(\frac{x_1(t)}{1+x_1^2(t)} + 1\right)\sin(x_2(t)) \tag{33}$$

$$\begin{aligned} x_2(t+1) =& x_2(t)\cos(x_2(t)) + \exp(-\frac{x_1^2(t)+x_2^2(t)}{8})x_1(t) \\ &+ \frac{u^3(t)}{1+u^2(t)+0.5\cos(x_1(t))+x_2(t)} \end{aligned} \tag{34}$$

where $x_1(t)$ and $x_2(t)$ are state variables. $y(t)$, $u(t)$ and $\varepsilon(t)$ denote the output, input and white noise of the nonlinear functions, respectively. Suppose the state of the system is unpredictable, the output can be predicted by using the given information of inputs and output. The model inputs are expressed as $\phi(t-1) = (y(t-1), y(t-2), y(t-3), u(t-1), u(t-2), u(t-3))^{\mathrm{T}}$, which can be explained in Fig. 2. $u(t)$ is a random signal belonging to $[-2.5, 2.5]$ and $\varepsilon(t) \in \mathcal{N}(0, 0.1)$. The training samples are 3000. The signal $u(t) = \sin(0.2\pi t) + \sin(0.08\pi t)$ is applied to the system to generate 300 test samples. The forecasting models are trained by the train set and validated on the test set. For the proposed GMM-IGPR model, the parameters are set as $I_{it} = 1000$, $c_1 = c_2 = 2$, $CR = 0.4$, $F = 1.2$. The ANN model is composed of one input layer, one hidden layer and one output layer. The number of hidden neurons is 15. The regularization parameter C and the kernel function parameter σ of the SVR are optimized as $C = 11.3137$, $\sigma = 0.1768$ by fivefold cross-validation method. They are optimized with the values within the range of the default values of 2^{-5} to 2^5, step 0.01. The forecasting results of different prediction models are listed in Table 2, and the visualizations of the results are given in Fig. 3. The prediction performances are plotted in Fig. 4. For the purpose of verifying the effectiveness of the parameter optimization, the comparisons between the GMM-IGPR and GMM-GPR are made. As mentioned above, the GMM-IGPR model introduces differential evolution operator into the basic PSO algorithm to estimate hyperparameters of the GPR model, instead of using the traditional conjugate gradient method. Table 3 presents the impact of parameter optimization on prediction performance. Figure 5 visualizes the comparison results. From Table 2 and Fig. 3, it can be seen that the proposed model performs better than the compared methods. Figure 4 shows apparently a better curve fitting of the observed output when contrasted with other forecasting results. The comparisons between the GMM-IGPR and GMM-GPR in Table 3 reveal that the parameter optimization is helpful to improve the model performance.

In order to test whether the forecast accuracy of the proposed method is significantly better than the accuracy of the other prediction models, statistical tests including Friedman test and the post hoc test are used to rank different methods with much more statistical reliability. In the literature (Demšar 2006), Demšar considered that nonparametric tests had been used because the conditions that guarantee the reliability of the parametric tests might not be met (Martínez et al. 2018). According to the methodology described in

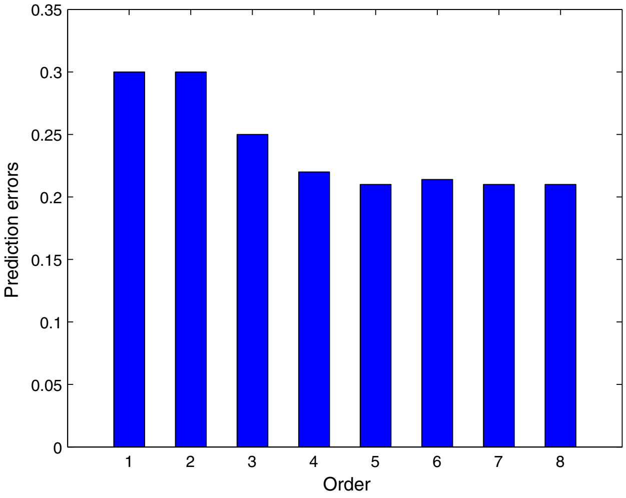

Fig. 7 Prediction error versus order

Table 9 The specific input variables

Orig.	Selected inputs
$S1$	$S1(t-5), S1(t-4), S1(t-3), S1(t-2), S1(t-1)$
$S2$	$S2(t-5), S2(t-4), S2(t-3), S2(t-2), S2(t-1)$
$S3$	$S3(t-5), S3(t-4), S3(t-3), S3(t-2), S3(t-1)$
$I1*$	$I1*(t-5), I1*(t-4), I1*(t-3), I1*(t-2), I1*(t-1)$
$I3*$	$I3**(t-5), I3*(t-4), I3*(t-3), I3*(t-2), I3*(t-1)$
$T1$	$T1(t-3), T1(t-2), T1(t-1)$
$T3$	$T3(t-3), T3(t-2), T3(t-1)$

Demšar (2006), the Friedman test is used to test whether these five methods have the same prediction accuracy. If the null hypothesis is rejected, then the corresponding post hoc tests are applied to test whether the proposed model is significantly superior to other comparison models.

The Friedman test (Friedman 1937, 1940) is a nonparametric test which is used to detect differences in treatments across multiple test attempts. In a finite sample of size n, the observed data $x = (x_i), i = 1, n; x_i = (x_{ij}); j = 1, k$ consist of n independent realizations of the random variable. Usually, the data are written as a matrix with the n rows and k columns. Columns 1 to k indicate different models, while the rows correspond to criteria 1 to n used to evaluate the performances (Joachim 1997). In this case, five methods ($k = 5$) are compared in three evaluation criteria datasets ($n = 3$). The test ranks the algorithms for each dataset according to their forecast accuracy. The best performing algorithm gets the rank of 1, the second best is ranked as 2 and so on. If there are tied values, assign to each tied value the average of the ranks that would have been assigned without ties. Let us denote R_j^i as the rank of the j-th algorithms ($j \in k$) on the i-th dataset ($i \in n$). The average rank of every algorithm j is computed as $R_j = \frac{1}{n}\sum_i R_j^i$. The null hypothesis of the Friedman test states that all the methods are equivalent and their ranks R_j should be equal. The Friedman statistic is expressed as (Demšar 2006):

$$\chi_F^2 = \frac{12n}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{35}$$

which is distributed according to the Chi-squared distribution with $k - 1$ degrees of freedom. Table 4 depicts the average ranks computed through the Friedman test. As can be seen in the table, GMM-IGPR is the best performing algorithm of this example, whereas SVR is the worst. The Friedman test

$$\chi_F^2 = 10.667 \tag{36}$$

With the significant level $\alpha = 0.05$ and the $k - 1 = 4$ degrees of freedom, the critical value of $F_{0.05}[3]$ is 7.815. Since $\chi_F^2 > F_{0.05}[3]$, we reject the null hypothesis. Therefore, there are significant differences among different models. The test statistics results shown in Table 5 also verify this conclusion. Next, a post hoc test is done to test whether the proposed algorithm is significantly better than the other competitors. Table 6 shows the p values of the comparisons between our proposal and the other approaches by using the Bonferroni method of multiple comparisons in SPSS. If $p > 0.05$, there are no significant differences, while $0.01 < p < 0.05$ means there are significant differences. As can be seen from Table 6, the p values mean that there are significant differences between our approach and the others.

Table 10 Performance evaluations of different models in winding process

Model	$T1$(kg)				$T3$(kg)			
	RMSE	MAE	MAPE (%)	R	RMSE	MAE	MAPE (%)	R
GMM-IGPR	0.1068	0.0703	0.2010	0.9938	0.3347	0.2169	0.6951	0.9659
ANN	0.1452	0.0820	0.3361	0.9889	0.3521	0.2310	0.7828	0.9344
SVR	0.8008	0.7124	1.1942	0.5876	0.6672	0.5635	0.9795	0.7751
GPR	0.1445	0.0809	0.3236	0.9918	0.3681	0.2333	0.7664	0.9414

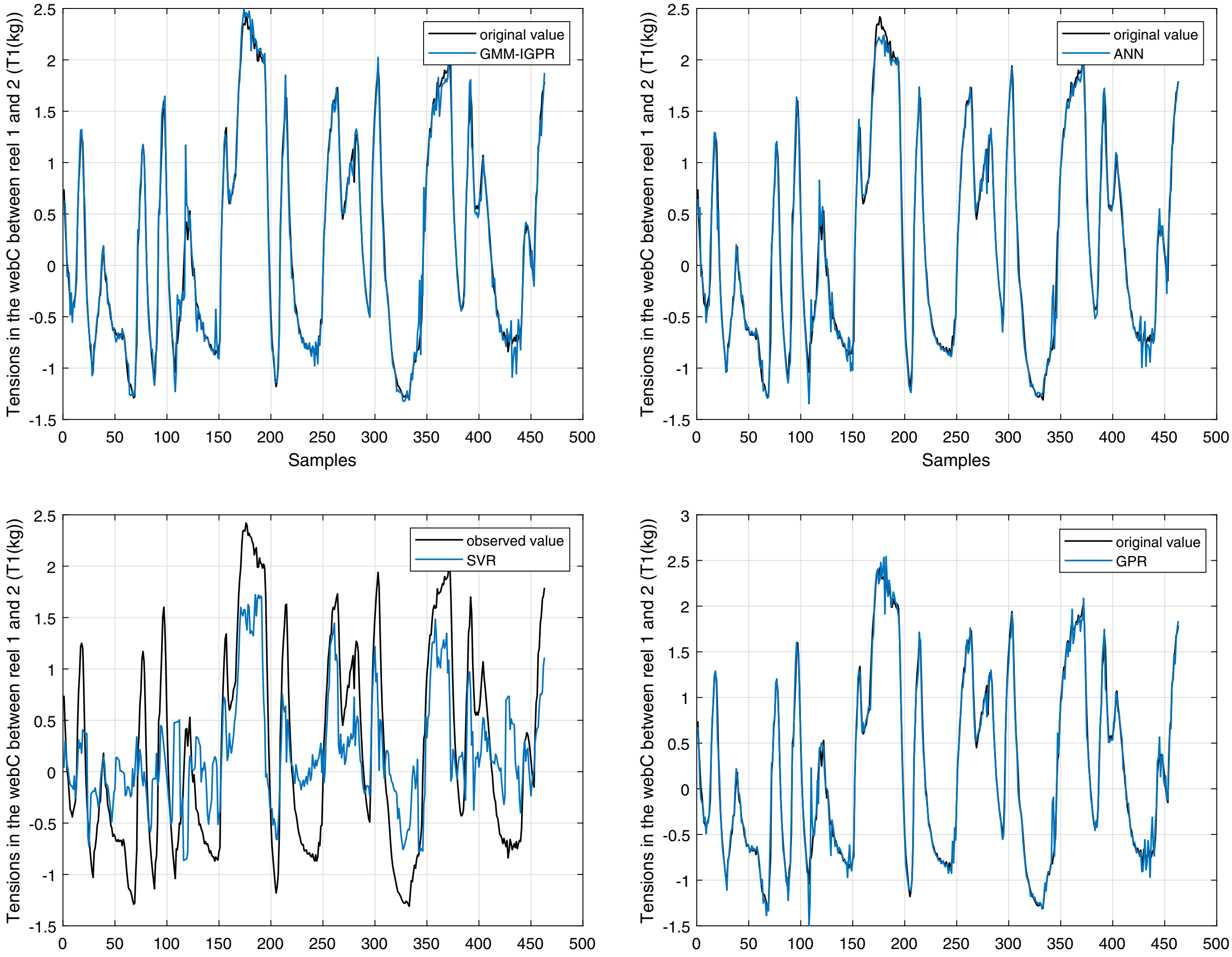

Fig. 8 The prediction performances of different models (T1)

4.2 Industrial application

4.2.1 Process description

The process is a test setup of an industrial winding process. A diagram of the plant is shown in Fig. 6 (Bastogne et al. 1997). This dataset is provided by Thierry Bastogne of the Henri Poincare University and can be found in Database for the Identification of Systems (DaISy) (DaISy 2006). The main part of the plant is composed of a plastic web that is unwinded from first reel (unwinding reel), goes over the traction reel and is finally rewinded on the rewinding reel. The angular speed of each reel (S1, S2 and S3) and the tensions in the web between reel 1 and 2 ($T1$) and between reel 2 and 3 (T3) are measured by dynamo tachometers and tension meters, respectively. Reel 1 and reel 3 are coupled with a DC motor that is controlled with input setpoint currents I1* and I3*. The DC motor is controlled by a local regulator composed of one or two PI controllers. The setpoints of these controllers are computed by a programmable logical controller (PLC) in order to control the two tensions and the linear velocity of the strip. The final objective of the winding process is to control the linear velocity (V_2) and the tensions ($T1$) and ($T3$) around operating points. This dataset has 2500 samples with the sampling time of 0.1 s. We use the two tensions as the model outputs in this paper. The variables are shown in Table 7. Suppose we denote the input and output as $\{x_1, x_2, x_3, x_4, x_5\}$ and $\{y_1, y_2\}$, respectively. The statistical information of this dataset is given in Table 8. The specific description of the process is given in the literature (Bastogne et al. 1997).

4.2.2 Order selection

Before using the samples to train the forecasting models, model order should be determined firstly. The original features are constructed based on the potential relationship between the original time series and their lags. In this paper,

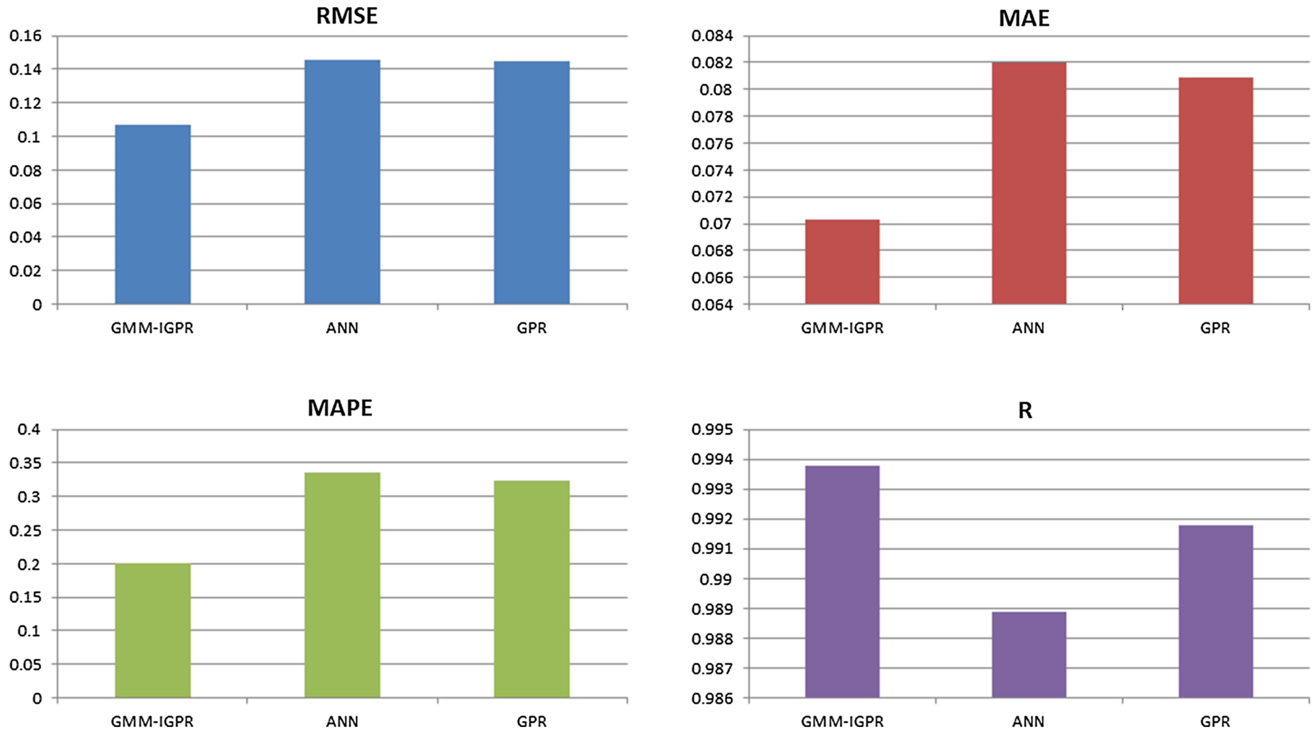

Fig. 9 The visualization of prediction performances of GMM-IGPR, ANN and GPR models ($T1$)

Table 11 The impact of parameter optimization on prediction performance

Model	$T1$(kg)				$T3$(kg)			
	RMSE	MAE	MAPE (%)	R	RMSE	MAE	MAPE (%)	R
GMM-IGPR	0.1068	0.0703	0.2010	0.9938	0.3347	0.2169	0.6951	0.9659
GMM-GPR	0.1327	0.0794	0.3023	0.9842	0.3496	0.2274	0.7283	0.9477

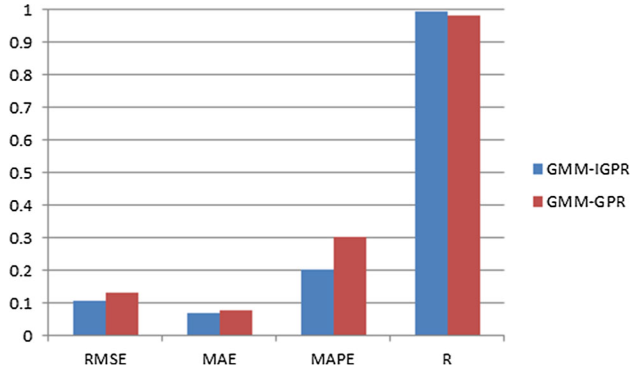

Fig. 10 The comparison of forecasting results ($T1$)

the actually most practical way consists in examining the decrease of prediction error with the order illustrated in Fig. 7. A validation set with 500 data is used for order determination. In this case, it seems that no significant improvement of the prediction error (RMSE) can be expected by increasing the order upper than 5. Same method is also used for the determination of output order. The minimum error is corresponded to the output order of 3. The previous observations of $T1$ and $T3$ are also used as model inputs to predict the present values. Thus we have totally 28 input variables in this experiment as shown in Table 9 to predict the outputs $T1(t)$ and $T3(t)$, respectively.

4.2.3 Simulation results and discussion

In this section, the industrial winding process datasets are used to construct the experiment. The proposed model is compared with the ANN, SVR, GPR and GMM-GPR models. The parameters of these models are determined based on the validation set. The ANN model is composed of one hidden layer and 5 hidden neurons. The parameters of the SVR are optimized as $C = 2.00, \sigma = 0.0313$ by fivefold cross-validation method. Other parameters are the same as the above section. The forecasting results of different prediction models are listed in Table 10, and the visualizations of the results are given in Figs. 8 and 11. It is observed that the performance of the SVR is significantly inferior to other models in this case. Therefore, Figs. 9 and 12 only show

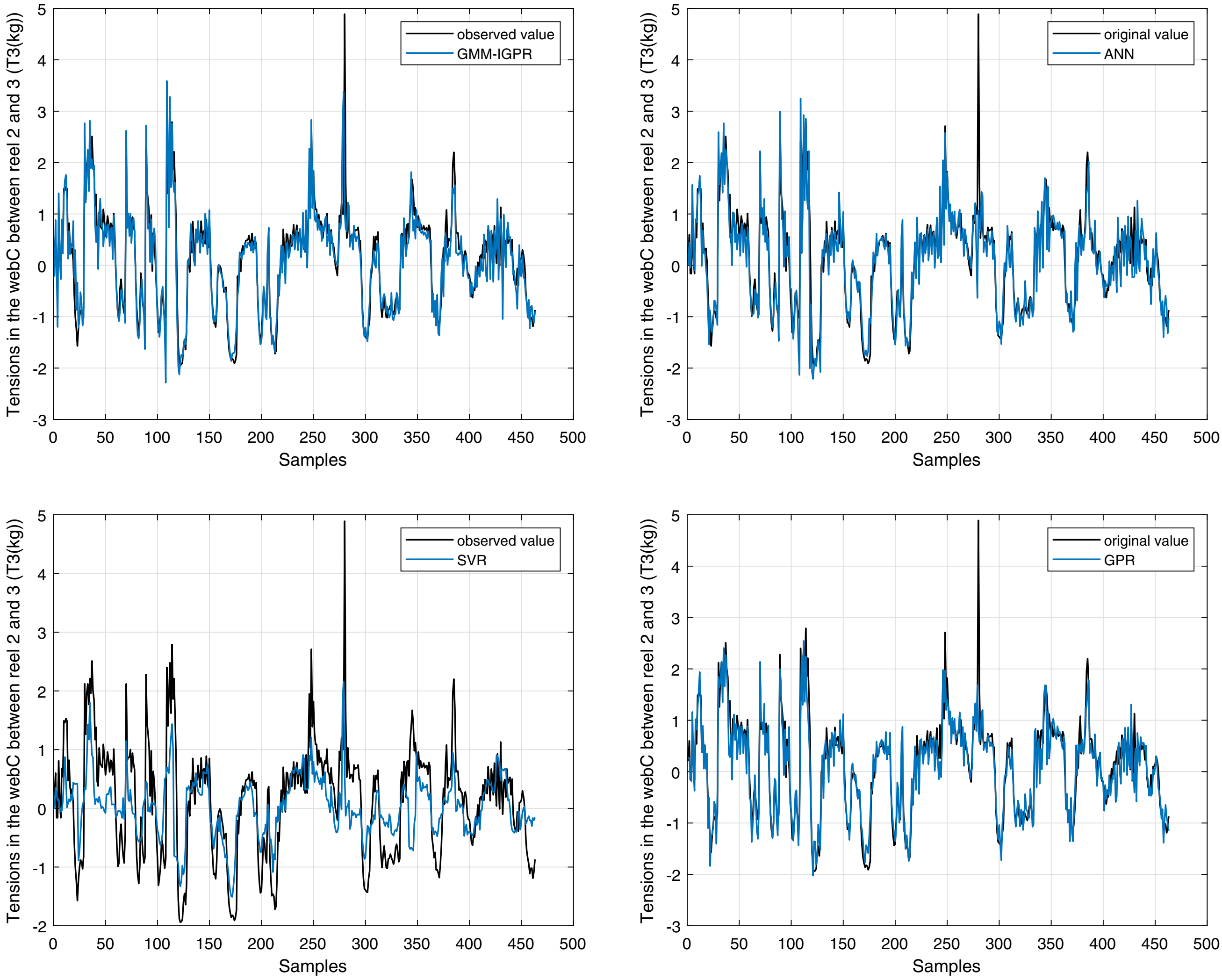

Fig. 11 The prediction performances of different models (T3)

the prediction performances of the other three models. The comparisons between the GMM-IGPR and GMM-GPR are performed, and the impact of parameter optimization on prediction performance is given in Table 11 and Figs. 10 and 13. For the tension in the web between reel 1 and 2 ($T1$), it can be seen from Table 10 and Fig. 9 that the proposed GMM-IGPR model performs better than other compared approaches with the minimum value of RMSE as 0.1068, MAE as 0.0703 and MAPE as 0.2010% and the maximum value of R as 0.9938. Table 11 reveals that the proposed model improves the prediction accuracy and outperforms the GMM-GPR models. The corresponding error measures are also illustrated in Fig. 10. It can be observed that the model can capture the tendency of real values. Although we did not plot all the predictions in the same graph, it is easy to find that the proposed model outperforms the other four models in terms of RMSE, MAE, MAPE and R and leads to more accurate predictions on the tensions $T1$. Likewise, for the prediction of the tension in the web between reel 2 and 3 (T3), the proposed GMM-IGPR model performs better than other compared approaches with the minimum value of RMSE as 0.3347, MAE as 0.2169 and MAPE as 0.6951% and the maximum value of R as 0.9659. It tracks the real tension values much more accurately than the other four models as shown in Fig. 11. The corresponding errors are tabulated in Table 10 and 11 and plotted in Figs. 12 and 13.

Similar to the simulation study, the statistical tests are also used to prove that the observed differences are not merely random. The same procedure is performed in this winding process. The average Friedman ranks are given in Table 12. The Friedman test

$$\chi^2_{F(T1)} = 12.000 \tag{37}$$

$$\chi^2_{F(T3)} = 11.465 \tag{38}$$

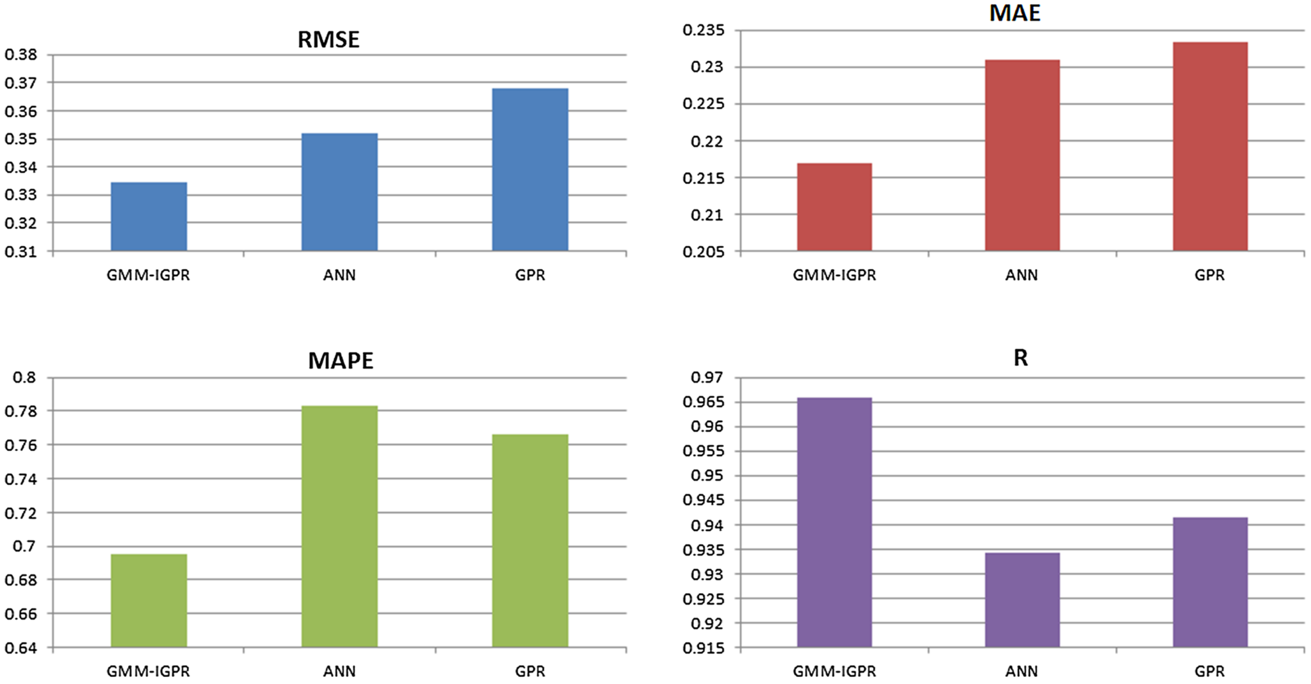

Fig. 12 The visualization of prediction performances of GMM-IGPR, ANN and GPR models (T3)

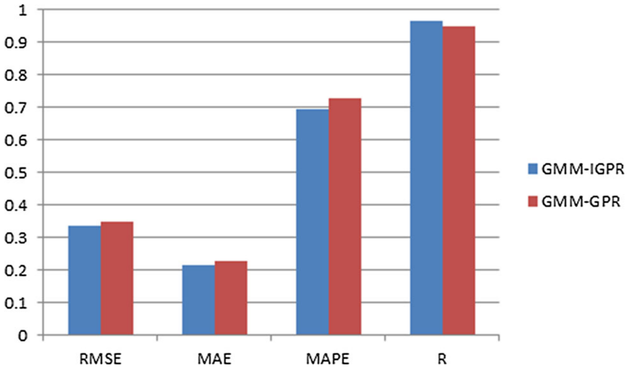

Fig. 13 The comparison of forecasting results ($T3$)

As the critical value of $F_{0.05}[3]$ is 7.815. Since $\chi^2_{F(T1)} = 12.000 > F_{0.05}[3]$, $\chi^2_{F(T3)} = 11.465 > F_{0.05}[3]$, we reject the null hypothesis. Therefore, there are significant differences among different models. The test statistics results shown in Table 13 also verify this conclusion.

The post hoc test is also done to test whether the proposed algorithm is significantly better than the other competitors in this winding process. Table 14 shows the p values of the comparisons between our proposal and the other approaches by using the multiple comparisons in SPSS. As can be seen from this table, with the significant level $\alpha = 0.05$, there are significant differences between our approach and the others.

5 Conclusion

Complex industrial process usually has the characteristics of strong nonlinearity, time-varying and disturbance. In this paper, a novel method based on the Gaussian process regression by integrating the GMM and DEPSO is introduced. The presented approach adopts the multiple model method according to the different characteristics of sample information for fitting. In order to strengthen the clustering effect, GMM is employed to identify clusters. An online EM algorithm is used instead of batch version. A procedure for parameter optimization based on PSO is adopted. Then localized GPR in the high-dimensional kernel space is performed to characterize the nonlinear dynamics within every single mode. Furthermore, the prediction is conducted in a stochastic fashion by adopting Bayesian inference strategy to combine the localized GPR models. The effectiveness of the proposed model is verified using a numerical example and an industrial process dataset. In the experiment, the results of the developed model demonstrate that the proposed GMM-IGPR method leads to a higher prediction accuracy and reliability than other approaches. The performance of the GMM-IGPR method is better than other models. Moreover, GMM-IGPR also realizes the best ranks in the statistical tests of experimental results. The superior capability of the GMM-IGPR approach to handle switching dynamics can be attributed to its local–global complex model structure, stochastic design based on Bayesian inference and nonlinearity characterization by GPR. The results obtained for the two examples

Table 12 Friedman ranks (Winding process)

Model	GMM-IGPR	ANN	SVR	GPR	GMM-GPR
Avg. rank ($T1$)	1.00	4.00	5.00	3.00	2.00
Avg. rank ($T3$)	1.00	3.33	5.00	3.67	2.00

Table 13 Test statistics

	$T1$	$T3$
N	3	3
Chi-square	12.000	11.467
df	4	4
Asymp.sig	0.017	0.022

Friedman test

Table 14 *p* values of the comparisons between our proposal and the others

model 1-model 2	Sig.($T1$)	Sig.($T3$)
GMM-IGPR - GMM-GPR	0.041	0.032
GMM-IGPR - ANN	0.039	0.024
GMM-IGPR - GPR	0.007	0.020
GMM-IGPR - SVR	0.002	0.002

suggest that the proposed method is an effective way to improve the winding process forecasting accuracy and can be a useful tool for complex industrial process forecasting modeling.

Acknowledgements The authors gratefully acknowledge the financial support of this research by the Natural Science Foundation of Jiangsu Province (Nos. BK20170500, BK20190876), Natural Science Foundation of China (Grant No.61773118), Natural Science Foundation of the Jiangsu Higher Education Institutions (Nos. 19KJB520063, 18KJB460032).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Aye SA, Heyns PS (2017) An integrated Gaussian process regression for prediction of remaining useful life of slow speed bearings based on acoustic emission. Mech Syst Signal Process 84:485–498

Bastogne T, Noura H, Richard A, Hittinger JM (1997) Application of subspace methods to the identification of a winding process. In: IEEE European control conference (ECC), pp 2168–2173

Bhowmik S, Paul A, Panua R, Ghosh SK, Debroy D (2018) Performance-exhaust emission prediction of diesosenol fueled diesel engine: an ANN coupled MORSM based optimization. Energy 153:212–222

Bishop CM (2006) Pattern recognition and machine learning. Springer, New York

Cappé O (2011) Online expectation-maximisation. In: Mengersen K, Robert C, Titterington M (eds) Mixtures: estimation and applications. Wiley, New York, pp 1–53

Cappé O, Moulines E (2009) On-line expectation-maximization algorithm for latent data models. J R Stat Soc B 71(3):593–613

DaISy: Database for the Identification of Systems (2006). In: De Moor BLR (ed) Department of Electrical Engineering, ESAT/STADIUS. KU Leuven, Belgium. http://homes.esat.kuleuven.be/~smc/daisy/daisydata.html

Demšar J (2006) Statistical comparisions of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Elguebaly T, Bouguila N (2015) Simultaneous high-dimensional clustering and feature selection using asymmetric Gaussian mixture models. Image Vis Comput 34:27–41

Ferlito S, Adinolfi G, Graditi G (2017) Comparative analysis of data-driven methods online and offline trained to the forecasting of grid-connected photovoltaic plant production. Appl Energy 205:116–129

Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32:675–701

Friedman M (1940) A comparison of alternative tests of significance for the problem of *m* rankings. Ann Math Stat 11(1):86–92

General Electric Intelligent Platforms, (2012). The Rise of Industrial Big Data, 2012, White Paper

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Boca Raton

Grbić R, Slišković D, Kadlec P (2013) Adaptive soft sensor for online prediction and process monitoring based on a mixture of Gaussian process models. Comput Chem Eng 58:84–97

Gregorčič G, Lightbody G (2009) Gaussian process approach for modelling of nonlinear systems. Eng Appl Artif Intell 22:522–533

He F, Li M, Wang BJ (2016) Mult-mode acid concentration prediction models of cold-rolled strip steel pickling process. J Process Control 24:916–923

Herp J, Ramezani MH, Bach-Andersen M, Pedersen NL, Nadimi ES (2018) Bayesian state prediction of wind turbine bearing failure. Renew Energy 116:164–172

Higashi N, Iba H (2003) Particle swarm optimization with Gaussian mutation. In: IEEE swarm intelligence symposium, pp 72–79

Jin H, Chen X, Wang L, Yang K, Wu L (2015) Adaptive soft sensor development based on online ensemble Gaussian process regression for nonlinear time-varying batch processes. Ind Eng Chem Res 54(30):7320–7345

Joachim R (1997) The permutation distribution of the Friedman test. Comput Stat Data Anal 26:83–99

Kennedy J (1997) The particle swarm: social adaptation of knowledge. In: IEEE international conference on evolutionary computation, pp 303–308

Kennedy J, Eberhart RC (1994) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, Perth, Australia 4:1942–1948

Liu Y, Chen JH (2013) Integrated soft sensor using just-in-time support vector regression and probabilistic analysis for quality prediction of multi-grade processes. J Process Control 23:793–804

Liu L, Wang Q, Wang J, Liu M (2016) A rolling grey model optimized by particle swarm optimization in economic prediction. Comput Intell 32(3):391–419

López C, Zhong W, Zheng ML (2017) Short-term electric load forecasting based on wavelet neural network, particle swarm optimization and ensemble empirical mode decomposition. Energy Proc 105:3677–3682

Martínez F, Frías MP, Pérez-Godoy MD, Rivera AJ (2018) Dealing with seasonality by narrowing the training set in time series forecasting with kNN. Expert Syst Appl 103:38–48

Nabney IT (2002) NETLAB algorithms for pattern recognition. Springer, Great Britain

Neal RM, Hinton GE (1998) A view of the EM algorithm that justifies incremental, sparse, and other variants Learning in graphical models. In: Jordan MI (ed) Learning in graphical models, NATO ASI series, vol 89. Springer, Netherlands, pp 355–368

Nowakowska E, Koronacki J, Lipovetsky S (2015) Clusterability assenssment for Gaussian mixure models. Appl Math Comput 256:591–601

Pradeepkumar D, Ravi D (2017) Forecasting financial time series volatility using particle swarm optimization trained quantile regression neural network. Appl Soft Comput 58:35–52

Ranjan R, Huang B, Fatehi A (2016) Robust Gaussian process modeling using EM algorihtm. J Process Control 42:125–136

Rasmussen CE, Nickisch H (2010) Gaussian processes for machine learning (GPML) toolbox. J Mach Learn Res 11:3011–3015

Rasmussen CE, Williams CKI (2006) Gaussian processes for machine learning. The MIT Press, Cambridge

Reynolds D (2015) Gaussian mixture models. Encycl Biometrics 741:827–832

Sato MA, Ishii S (2000) On-line EM algorithm for the normalized Gaussian network. Neural Comput 12(2):407–432

Schenker B, Agarwal M (1995) Prediction of infrequently measurable quantities in poorly modeled processes. J Process Control 5:329–339

Scrucca L (2016) Identifying connected components in Gaussian finite mixture models for clustering. Comput Stat Data Anal 93:5–17

Shi Y, Eberhart RC (1998) Parameter selection in particle swarm optimization. In: International conference on evolutionary programming, pp 591–601

Singh P, Borah B (2014) Forecasting stock index price based on M-factors fuzzy time series and particle swarm optimization. Int J Approx Reason 55:812–833

Sun AY, Wang D, Xu X (2014) Monthly streamflow forecasting using Gaussian process regression. J Hydrol 511:72–81

Sun W, Wang CF, Zhang CC (2017) Factor analysis and forecasting of CO_2 emissions in Hebei, using extreme learning machine based on particle swarm optimization. J Clean Prod 162:1095–1101

Titterington DM (1984) Recursive parameter estimation using incomplete data. J R Stat Soc B 46:257–267

Tobias P, Malte K, Carl ER (2006) Nonstationary gaussian process regression using a latent extension of the input space. In: ISBA eighth world meeting on Bayesian statistics

Xie XF, Zhang WJ, Yang ZL (2002) A dissipative particle swarm optimization. In: IEEE congress on evolutionary computation, pp 1456–1461

Xu C, Liu BG, Liu KY, Guo JQ (2011) Intelligent analysis model of landslide displacement time series based on coupling PSO-GPR. Rock Soil Mech 32(6):1669–1675

Xu J, Yamada K, Seikiya K, Tanaka R, Yamane Y (2014) Effect of different features to drill-wear prediction with back propagation neural network. Precis Eng 38(4):791–798

Yang K, Jin H, Chen X (2016) Soft sensor development for online quality prediction of industrial batch rubber mixing process using ensemble just-in-time Gaussian process regression models. Chemom Intell Lab 155:170–182

Yu J (2012) Online quality prediction of nonlinear and non-Gaussian chemical processes with shifting dynamics using finite mixture model based Gaussian process regression approach. Chem Eng Sci 82:22–30

Yu J, Chen K, Rashid MM (2013) A Bayesian model averaging based multi-kernel Gaussian process regression framework for nonlinear state estimation and quality prediction of multiphase batch processes with transient dynamics and uncertainty. Chem Eng Sci 93:96–109

Yuan X, Tan Q, Lei X, Yuan Y, Wu X (2017) Wind power prediction using hybrid autoregressive fractionally integrated moving average and least square support vector machine. Energy 129:122–137

Zhao J, Liu QL, Wang W, Pedrycz W, Cong L (2012) Hybrid neural prediction and optimized adjustment for coke oven gas system in steel industry. IEEE Trans Neural Netw Learn Syst 23:439–450