



# Application of soft sensors and ant colony optimization for monitoring and managing defects in the automation industry

Wongchai A<sup>1</sup> · Mohammed A. S. Abourehab<sup>2,3</sup> · Mohammed Altaf Ahmed<sup>4</sup> · Saibal Dutta<sup>5</sup> · Koduganti Venkatrao<sup>6</sup> · Kashif Irshad<sup>7,8</sup>

Received: 17 August 2022 / Accepted: 18 December 2022  
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2023

## Abstract

In modern industrial processes, various types of soft sensors are used in process monitoring, control, and optimization, and the soft sensors designed to maintain or update these models are highly desirable in the industry. This paper proposes a novel technique for monitoring and control optimization of soft sensors in automation industry for fault detection. The fault detection has been carried out using probabilistic multi-layer Fourier transform perceptron (PMLFTP), and the input data has been pre-processed for removal of samples containing null values for fault detection and diagnosis process through Fourier transform-based detection and multi-layer perceptron-based diagnosis in the manufacturing process. The controlling of data in soft sensors has been optimized using auto-regression-based ant colony optimization (AR\_ACO), and the experimental results have been reported in terms of computational rate of 40%, QoS of 78%, RMSE of 45%, fault detection rate of 90%, and control optimization of 93%.

**Keywords** Soft sensors · Monitoring automation industry · Fault detection · PMLFTP · AR\_ACO

## 1 Introduction

It is tough and time-consuming to design defect detection methods for complex real-world industrial operations [1]. Numerous manufacturing cells execute a variety of assembly activities as well as functional tests in modern computer-based manufacturing systems. Computer software supervises a specific production process, many of which are custom created, and the cells are controlled by it. One of the most significant duties for computers assigned to manufacturing plant supervision is to detect as well as diagnose product problems. Obtaining data required for process analysis is the initial stage in this task. Only a few data-generating mechanisms and sensing devices were used in first inspection systems. As a result, engineers could only analyze a limited amount of data for fault diagnosis method, and a more method based on structured data analysis was needed [2]. Limit checking is still the only method of fault detection utilized in numerous manufacturing plants today [3]. In this example, for a given characteristic in production process for a product, maximal and minimum values are known as thresholds. When value of a feature is within these defined boundaries, it is said to be in a normal functioning state. Although simple, resilient, and trustworthy, this

---

✉ Wongchai A  
Wong\_anupong@outlook.com

- <sup>1</sup> Independent Researcher, Thailand, Thailand
- <sup>2</sup> Department of Pharmaceutics, College of Pharmacy, Umm Al-Qura University, Makkah 21955, Saudi Arabia
- <sup>3</sup> Department of Pharmaceutics and Industrial Pharmacy, Faculty of Pharmacy, Minia University, Minia 61519, Egypt
- <sup>4</sup> Department of Computer Engineering, College of Computer Engineering & Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
- <sup>5</sup> SAS Research and Development, Pune, India
- <sup>6</sup> Department of CSE, Guru Nanak Institution Technical Campus, Ibrahimpatnam, Hyderabad, India
- <sup>7</sup> Interdisciplinary Research Center for Renewable Energy and Power Systems (IRC-REPS), Research Institute, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia
- <sup>8</sup> K.A.CARE Energy Research & Innovation Center, Dhahran, Saudi Arabia

technique is sluggish to respond to deviations in a particular data characteristic as well as fails to detect complicated failures, which are detected by examining feature correlations. Another issue with this method is difficulty in defining threshold values for each attribute.

In terms of data analysis, industrial applications with heavy machinery, in particular, might advantage from a deeper understanding of underlying methods as well as equipment state to modify their maintenance plans [4]. These components are used in the machine’s rotating mechanics. These bearing parts deteriorate over time as a result of friction as the machine rotates. Traditionally, condition of a rolling bearing element was estimated using breakdown data from the past. Nowadays, state of a bearing element is determined by installing vibration sensors on certain sections of machines or detecting the motor currents of electrical engine that drives these elements. Vibrations, in particular, have proven to be beneficial in revealing underlying status of bearings. Raw signals must be denoised as well as pre-processed before analysis, utilizing complicated and time-consuming signal processing methods to obtain useful data, which is a vital requirement for an effective analysis. As a

result, the emphasis has switched to deep learning algorithms that can analyze raw data and create features automatically by recognizing patterns in input data. This automated approach saves time, is less prone to human error, and may need a specialized domain expert with less subject experience [5].

FDD (fault detection and diagnosis) is a vital control technique for achieving this task among numerous process supervision techniques because many industries seek to enhance their process performance by increasing their FDD capability. FDD’s primary functions are categorized into 2 sections: (1) monitoring process behavior and (2) disclosing presence, characteristics, and root causes of errors. To preserve high process output as well as throughput in industrial processes, rapid, significant detection tools for process or equipment failures that may affect whole system’s performance are needed [4–6]. The FDD for many different processes has gained a lot of attention from many industrial sectors and academia over the years due to the many significant benefits that may be obtained from lowering process or product-related costs, improving quality and productivity. FDD has played a key role in a variety of industrial engineering methods, including semiconductor

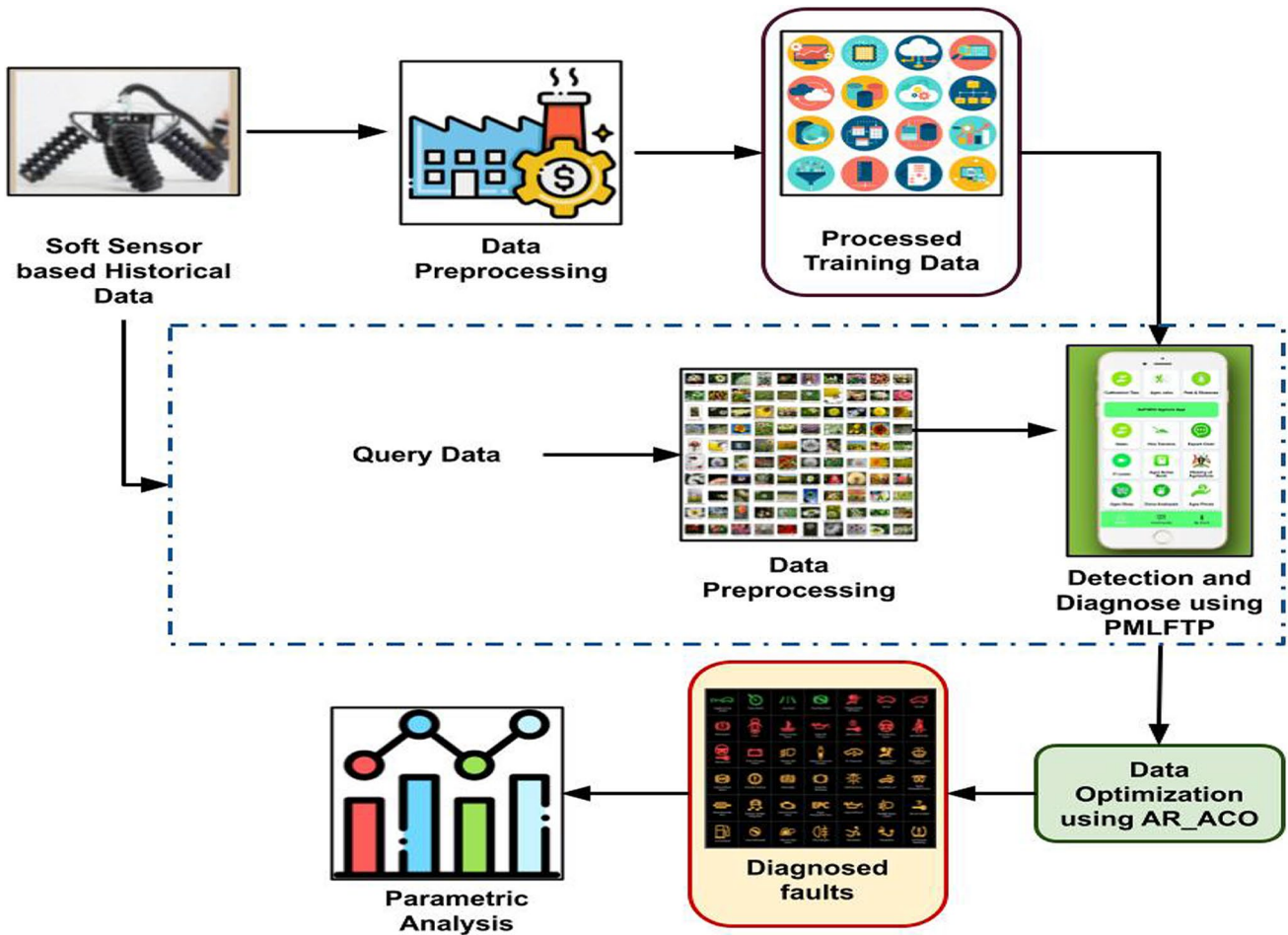


Fig. Overall proposed architecture

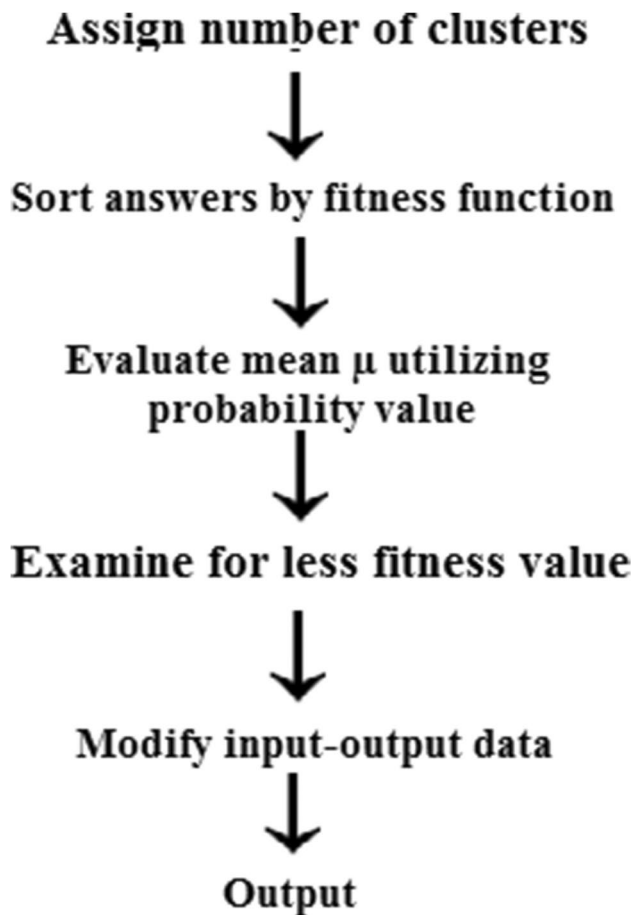


Fig. 2 Optimization algorithm

production and chemical and software engineering to name a few. As a result, there is an enhancing requirement for effective detection as well as diagnosis of suspicious defects to avert process deterioration, which could ultimately result in a decrease in product yield or process throughput. In general, the FDD work is conducted based on multiple processes as well as equipment data measured by instruments as a key way for process supervision [6].

Research contribution is as follows:

- To collect historical data from soft sensors in designing fault detection systems for monitoring and controlling with optimization
- To pre-process the collected data in removing null values and missing data
- To detect and diagnose the faults of processed data using probabilistic multi-layer Fourier transform perceptron (PMLFTP)
- To optimize and control the data of soft sensors using auto-regression-based ant colony optimization (AR\_ACO)

The experimental analysis has been carried out in terms of computational rate, quality of service (QoS), root mean square error (RMSE), fault detection rate, control optimization for various fault scenarios.

## 2 Related works

The value of foreign direct investment (FDI) was originally recognized in high-risk fields such as flight control, railways, medicine, nuclear power plants, and many others. Due to the growing use of computational intelligence for data analysis done by real-time methods, the necessity for problem detection has become even more pressing. This is particularly true in real-time energy-efficient management of distributed resources [7], real-time control as well as mobile crowd sensing [8], and protection of sensitive data collected by wearable sensors [9]. Regular inspections of sensor validation, measurement device calibration, software configuration, and preventative maintenance are required to ensure error-free operation [10]. According to [11], maintenance costs might be anything from 15 to 60% of the entire cost of manufacturing items. Within these margins, almost 33% of maintenance costs are directly related to redundant as well as inappropriate equipment maintenance. As a result, enhancing equipment efficiency while lowering the costs of costly maintenance could result in a significant reduction in overall production costs [12]. According to [13], equipment maintenance can be divided into three categories: (1) modification maintenance entails upgrading components to boost machine productivity as well as performance, (2) preventive maintenance entails replacing a component just before it fails, and (3) breakdown corrective maintenance is when a part fails and needs to be replaced, resulting in machine downtime. Focus on preventive maintenance in this paper, which is divided into 2 types: UBM (usage-based maintenance) and condition-based maintenance (CBM) are two types of maintenance. Time-domain analysis and frequency-domain analysis are two traditional methods for identifying representative characteristics to categorize signals [14, 15]. The large number of features derived from various domains results in an HD dataset, as one could expect. As a result, features are chosen [16] and methods like PCA (principal component analysis) [17] or LDA (linear discriminant analysis) [18] are typically employed to reduce dimensionality of these features. In addition, [19], for example, used data entropy to preprocess raw time series data. RPCA (recursive PCA) [20], DPCA (dynamic PCA) [21], and KPCA (kernel PCA) [22] are used to monitor a variety of industrial processes, including adaptive, dynamic, and nonlinear processes [23]. Two RPCA algorithms were published in [24] to adapt for regular process

changes in semiconductor production operations. They did this by iteratively updating the correlation matrix [25].

### 3 System model

This section discusses the fault detection in automation industry based on soft sensors in monitoring and controlling with optimization. The data has been collected from soft sensors in which the fault has to be detected. This collected data has been pre-processed for removal of samples containing null values. For this processed data, the detection and diagnosis have been carried out using probabilistic multi-layer Fourier transform perceptron (PMLFTP). Then, controlling of data in soft sensors has been optimized using auto-regression-based ant colony optimization (AR\_ACO) which has effect in increasing the production of industry automatically. The overall proposed architecture is shown in Fig. 1.

#### 3.1 Fault detection and diagnosis using probabilistic multi-layer Fourier transform perceptron (PMLFTP)

Input variables of model under study are given by N-dimensional vector  $x = \times 1, \times 2, \dots, \times N$ , and response variable is represented by  $g(x)$ . As stated in Eq. (1), the answer  $g(x)$  is a hierarchical correlated function expansion of input variables:

$$g(x) = g_0 + \sum_{i=1}^N g_i(x_i) + \sum_{1 \leq i_1 < i_2 \leq N} g_{i_1 i_2}(x_{i_1}, x_{i_2}) + \dots + \sum_{1 \leq i_1 < \dots < i_n \leq N} g_{i_1 i_2 \dots i_n}(x_{i_1}, x_{i_2}, \dots, x_{i_n}) + \dots + g_{12 \dots N}(x_1, x_2, \dots, x_N) \tag{1}$$

where  $g_0$  denotes the 0th order component function or mean response of  $g(x)$ . Function  $g_{i_1 i_2}(x_{i_1}, x_{i_2})$  is a 2nd-order that defines how variables  $x_{i_1}$  and  $x_{i_2}$  work together to produce output  $g(x)$ . Last one  $g_{12 \dots N}(x_1, x_2, \dots, x_N)$  comprises any residual dependency of all input variables linked together cooperatively to impact output  $g(x)$ . For component functions in Eq. (2), this approach reduces to the following relationship.

$$\begin{aligned} g_0 &= g(\mathbf{c}), \\ g_i(x_i) &= g(x_i, \mathbf{c}^i) - g_0 \\ g_{i_1 i_2}(x_{i_1}, x_{i_2}) &= g(x_{i_1}, x_{i_2}, \mathbf{c}^{i_1 i_2}) - g_{i_1}(x_{i_1}) - g_{i_2}(x_{i_2}) - g_0 \end{aligned} \tag{2}$$

$$\begin{aligned} g(x) &= g_0 + \sum_{i=1}^N g_i(x_i) + \mathcal{R}_2 \\ g(x) &= \sum_{i=1}^N g(c_1, \dots, c_{i-1}, x_i, c_{i+1}, \dots, c_N) - (N-1)g(\mathbf{c}) + \mathcal{R}_2 \end{aligned} \tag{3}$$

Now consider the 1st order of  $g(x)$  given by Eqs. (3) and (4):

$$\begin{aligned} \dot{g}(x) &= g(x_1, x_2, \dots, x_N) \\ &= \sum_{i=1}^N g(c_1, \dots, c_{i-1}, x_i, c_{i+1}, \dots, c_N) - (N-1)g(\mathbf{c}) \end{aligned} \tag{4}$$

Fourier transform pairs formulation is given by Eqs. (5) and (6):

$$M_Y(\theta) = \int_{-\infty}^{\infty} p_Y(y) e^{2\pi\theta iy} dy \tag{5}$$

$$p_Y(y) = \int_{-\infty}^{\infty} M_Y(\theta) e^{2\pi\theta iy} d\theta \tag{6}$$

The marginal density and features function of  $Y$  are  $p_Y(y)$  and  $M_Y(\theta)$  indicates imaginary number given as  $i = \sqrt{-1}$  given by Eq. (7):

$$\begin{aligned} f_p^{(m)}(t) &\equiv f_F^{(m)}(t) \\ &= C_0^{(m)} + \sum_{n=1}^{\infty} C_n^{(m)} \cos k(t - t^{(m)}) \\ &+ S_n^{(m)} \sin k(t - t^{(m)}) \quad k = \frac{2\pi n}{h^{(m)}}; \\ n &= 0, \pm 1, \pm 2, \dots; t \in (t^{(m)}; t^{(m)} + h^{(m)}); \end{aligned} \tag{7}$$

$$S_k^{(m)} = \frac{2}{h^{(m)}} \int_{t^{(m)}}^{t^{(m)}+h^{(m)}} f(t) \sin k(t - t^{(m)}) dt$$

Since the function  $f_p(t)$  in points  $t = t^{(m)}$  and  $t = t^{(m)} + h^{(m)}$  has discontinuities, for Fourier series following relations are valid by Eq. (8):

$$\begin{aligned} f_p(t^{(m)}) &= f_p(t^{(m)} + h^{(m)}) = f(t^{(m)}) \\ &+ \frac{1}{2} \{f(t^{(m)} + h^{(m)}) - f(t^{(m)})\} \\ &= f_i^{(m)} + \frac{1}{2} \Delta f_i^{(m)} = C_0^{(m)} + \sum_{k=1}^{\infty} C_k^{(m)} \end{aligned} \tag{8}$$

The following integral is local Fourier transform given by Eqs. (9) and (10):

$$F\{f(t)\} = F(m, k) = \frac{2}{h^{(m)}} \int_{t^{(m)}}^{t^{(m)}+h^{(m)}} f(t) e^{-\beta(t)} (t - t^{(m)}) dt, \tag{9}$$

where  $k = \frac{2\pi n}{h^{(m)}} = \omega^{(m)} n; n = 0, \pm 1, \pm 2, \dots; m = 1, 2, \dots$

$$\begin{aligned} f(t^{(m)}) &= \frac{1}{2} \sum_{n=-\infty}^{\infty} F(m, k) - \frac{1}{2} \Delta f(t^{(m)}) \\ &= \frac{1}{2} F(m, 0) + \sum_{n=1}^{\infty} C_n^{(m)} - \frac{1}{2} \Delta f(t^{(m)}) \end{aligned} \tag{10}$$

Steps in proposed technique for fault probability evaluation are as follows:

1. If  $u = \{u_1, u_2, \dots, u_N\}^T \in \mathfrak{R}^N$  is standard Gaussian variable, let  $u^* = \{u_1^*, u_2^*, \dots, u_N^*\}^T$  be MPP or design point. Create an orthogonal matrix  $R \in \mathfrak{R}^{N \times N}$  whose  $N$ -th column is  $\alpha^* = u^* / \beta_{HL}$ , i.e.,  $R = [R_1 | \alpha^*]$  where  $R_1 \in \mathfrak{R}^{N \times (N-1)}$  satisfies  $\alpha^{*T} R_1 = 0 \in \mathfrak{R}^{1 \times (N-1)}$ . MPP has a distance  $\beta_{HL}$ , which is commonly mentioned as Hasofer-Lind reliability index [1-3]. For an orthogonal transformation  $u = Rv$ . Let  $v = \{v_1, v_2, \dots, v_N\}^T \in \mathfrak{R}^N$  indicates rotated Gaussian space with MPP  $v^* = \{v_1^*, v_2^*, \dots, v_N^*\}^T$ . Gaussian space  $v$  with  $v^* = \{v_1^*, v_2^*, \dots, v_N^*\}^T$  as reference point as Eq. (11):

$$\begin{aligned} \dot{g}(\mathbf{v}) &\equiv g(v_1, v_2, \dots, v_N) \\ &= \sum_{i=1}^N g(v_1^*, \dots, v_{i-1}^*, v_i, v_{i+1}^*, \dots, v_N^*) - (N-1)g(\mathbf{v}^*) \\ \sum_{n=-\infty}^{\infty} F(m, k) &= -j\pi \sum_{i=1}^l A_i(m) \sum_{s=1}^S b_{si} \text{ctg}(j\pi a_{si}) \end{aligned} \tag{11}$$

Furthermore, MPP is selected as reference point. Terms  $g(v_1^*, \dots, v_{i-1}^*, v_i, v_{i+1}^*, \dots, v_N^*)$  are individual component functions which are independent of each other as shown by Eq. (12).

$$\begin{aligned} \dot{g}(\mathbf{v}) &= a + \sum_{i=1}^N g(v_i, \mathbf{v}^{*i}) \\ f(t^{(m)}) &= \frac{-j\pi}{2} \sum_{i=1}^l A_i(m) \sum_{s=1}^S b_{si} \text{ctg}(j\pi a_{si}) - \frac{1}{2} \Delta f(t^{(m)}) \end{aligned} \tag{12}$$

The Park-Goreva equations for synchronous machine are stated utilizing the relative unit method given by Eq. (13).

$$\begin{aligned} -u_d &= r_i i_d + \frac{d}{d\theta} (x_d i_d + x_{ad} i_f + x_{ad} i_{1d}) + x_q i_q + x_{aq} i_{1q} \\ -u_q &= r_i i_q + \frac{d}{d\theta} (x_q i_q + x_{aq} i_{1q}) - (x_d i_d + x_{ad} i_f + x_{ad} i_{1d}) \\ u_f &= r_f i_f + \frac{d}{d\theta} (x_{ad} i_d + x_f i_f + x_{ad} i_{1d}) \\ 0 &= r_{1d} i_{1d} + \frac{d}{d\theta} (x_{ad} i_d + x_{ad} i_f + x_{1d} i_{1d}) \end{aligned} \tag{13}$$

New intermediate variables are given as Eq. (14)

$$z_i = g(v_i, \mathbf{v}^{*i}) \tag{14}$$

These new variables are used to convert approximation function into the following form using Eq. (15).

$$\dot{g}(\mathbf{v}) = a + z_1 + z_2 + \dots + z_N \tag{15}$$

In system of walking coordinates given by Eq. (16), the following equation is obtained in  $m^{\text{th}}$  local interval of recurrence of converter within the duration of commutation of stages  $\gamma$  related to valve switching:

$$u_f^{(m)} = \sqrt{3}E \cos(\theta - \pi/3) - 2r_c i_f^{(m)} - 2x_c \frac{di_f^{(m)}}{d\theta} + r_c i_\gamma^{(m)} + x_c \frac{di_\gamma^{(m)}}{d\theta} \tag{16}$$

where  $\theta = \omega t, \theta \in [\alpha; \alpha + \gamma]$ ;  $i_\gamma^{(m)}$  – exciter current stage, ending switching and  $i_\gamma^{(m)}(\alpha) = i_\gamma^{(m)}(\alpha), i_\gamma^{(m)}(\alpha + \gamma) = 0$ .

The index  $l$  corresponds to value of variable at switching point, i.e., when controlling signal is submitted to next thyristor of converter, as demonstrated by following formula, Eq. (17):

$$\Delta I_{fl}^{(m)} = i_f^{(m)}(\alpha + \pi/3) - i_f^{(m)}(\alpha) = I_{fl}^{(m+1)} - I_{fl}^{(m)} \tag{17}$$

Consider above-mentioned synchronous generator has outputs coupled to active and inductive loads  $r_w, x_w$ , as given in Eq. (18):

$$\left. \begin{aligned} u_d &= r_w i_d + x_w \frac{di_d}{d\theta} + x_w i_q \\ u_q &= r_w i_q + x_w \frac{di_q}{d\theta} - x_w i_d \end{aligned} \right\} \tag{18}$$

After applying LFT to Eqs. (19), (20), and (21) in the field, obtain the synchronous generator's equations as well as its activator.

$$\begin{aligned} (r_s + jkx_{ds}) \dot{I}_d(m, k) + jkx_{ad} \dot{I}_f(m, k) + jkx_{ad} \dot{I}_{1d}(m, k) + x_{qs} \dot{I}_q(m, k) + \\ + x_{aq} \dot{I}_{1q}(m, k) = -\frac{6}{\pi} x_{ds} \Delta I_d^{(m)} - \frac{6}{\pi} x_{ad} \Delta I_f^{(m)} - \frac{6}{\pi} x_{ad} \Delta I_{1d}^{(m)}, \\ -x_{ds} \dot{I}_d(m, k) - x_{ad} \dot{I}_f(m, k) - x_{ad} \dot{I}_{1d}(m, k) + (r_s + jkx_{qs}) \dot{I}_q(m, k) + \\ + jkx_{aq} \dot{I}_{1q}(m, k) = -\frac{6}{\pi} x_{qs} \Delta I_q^{(m)} - \frac{6}{\pi} x_{aq} \Delta I_{1q}^{(m)}, \\ jkx_{ad} \dot{I}_d(m, k) + (r_{fs} + jkx_{fs}) \dot{I}_f(m, k) + jkx_{ad} \dot{I}_{1d}(m, k) = -\frac{6}{\pi} x_c I_f^{(m)} + \end{aligned} \tag{19}$$

$$\begin{aligned} + \dot{B}(m, k) (r_c + jkx_c) I_f^{(m)} - \frac{6\sqrt{3}}{\pi} E \frac{\cos(\alpha - \frac{\pi}{6}) + jk \sin(\alpha - \frac{\pi}{6})}{k^2 - 1} \\ - \frac{6}{\pi} x_{ad} \Delta I_d^{(m)} - \frac{6}{\pi} x_{fs} \Delta I_f^{(m)} - \frac{6}{\pi} x_{ad} \Delta I_{1d}^{(m)}, \end{aligned} \tag{20}$$

$$\begin{aligned} jkx_{ad} \dot{I}_d(m, k) + jkx_{ad} \dot{I}_f(m, k) + (r_{1d} + jkx_{1d}) \dot{I}_{1d}(m, k) \\ = -\frac{6}{\pi} x_{ad} \Delta I_d^{(m)} - \frac{6}{\pi} x_{ad} \Delta I_f^{(m)} - \frac{6}{\pi} x_{1d} \Delta I_{1d}^{(m)} \\ jkx_{aq} \dot{I}_q(m, k) + (r_{1q} + jkx_{1q}) \dot{I}_{1q}(m, k) \\ = -\frac{6}{\pi} x_{aq} \Delta I_q^{(m)} - \frac{6}{\pi} x_{1q} \Delta I_{1q}^{(m)} \end{aligned} \tag{21}$$

In this context, it is generally known that the equation gives MLPNN state space formulation with one hidden layer (22).

$$\begin{aligned} \mathbf{z}(n) &= \mathbf{A}^T(n) \begin{bmatrix} \mathbf{x}(n) \\ 1 \end{bmatrix} \mathbf{u}(n) \\ &= \mathbf{f}(\mathbf{z}(n)) = [f(z_0(n)) \dots f(z_{l-1}(n))]^T \mathbf{y}(z) = \mathbf{b}^T(n) \begin{bmatrix} \mathbf{u}(n) \\ 1 \end{bmatrix} \\ \mathbf{x}(n) &= [x(n) \dots x_i(n)] = \tanh(z_i(n)), i = 1, \dots, l \end{aligned} \tag{22}$$

where  $\mathbf{x}(n) = [x(n) \dots x(n - K + 1) \ 1]^T$  is  $(K+1) \times 1$  input vector, which is constituted by elements of perceptual feature vector and bias of MLPNN;  $\mathbf{z}(n) = [z_0(n) \dots z_{l-1}(n)]^T$  is neuron output vector in hidden layer.

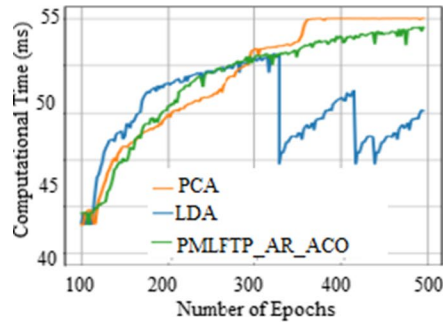
$$\begin{aligned} \mathbf{w}(n) &= [\mathbf{a}^T(n) \mathbf{b}^T(n)]^T \\ E_T(\mathbf{w}(n)) &= \sum e(n) = \sum \frac{1}{2} (y(n) - y_d(n))^2 \\ \nabla \mathbf{E}_T(n) &= \nabla \mathbf{E}_T(\mathbf{w}(n)) = [\nabla \mathbf{E}_a^T(n) \nabla \mathbf{E}_b^T(n)]^T \end{aligned} \tag{23}$$

The desired output is  $y_d(n)$ , output error is  $e(n)$ , and the error measure's gradients concerning  $\mathbf{a}(n)$  and  $\mathbf{b}(n)$  are  $\nabla \mathbf{E}_a(n)$  and  $\nabla \mathbf{E}_b(n)$ . From Eq. (24) concept of error measures, in the least-squares sense, it is observed that MLPNN seeks to make its output as near to subjective measure  $y_d(n)$  as possible.

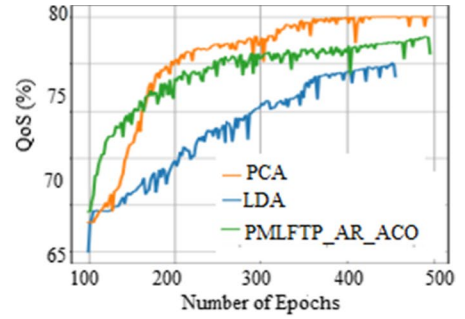
$$\begin{aligned} \nabla \mathbf{E}_A(n) &= \frac{\partial e(n)}{\partial \mathbf{A}(n)} = \begin{bmatrix} \overline{\partial e(n)} & \dots & \partial e(n) \\ \overline{\partial a_{1,1}(n)} & & \partial a_{1,l}(n) \\ \vdots & \ddots & \vdots \\ \overline{\partial e_{(K),1}(n)} & \dots & \overline{\partial a_{(K+1),l}(n)} \end{bmatrix} \\ \nabla \mathbf{E}_A(n) &= \begin{bmatrix} \mathbf{x}(n) \\ 1 \end{bmatrix} \frac{\partial e(n)}{\partial \mathbf{z}(n)} \frac{\partial e(n)}{\partial \mathbf{z}(n)} = \begin{bmatrix} \frac{\partial e(n)}{\partial z_1(n)} & \dots & \frac{\partial e(n)}{\partial z_l(n)} \end{bmatrix}^T \\ \frac{\partial \mathbf{f}(n)}{\partial \mathbf{s}(n)} &= \dot{\mathbf{f}}(n) = \begin{bmatrix} \frac{\partial f_1(n)}{\partial s_1(n)} & \dots & \frac{\partial f_l(n)}{\partial s_l(n)} \end{bmatrix}^T \frac{\partial e(n)}{\partial \mathbf{z}(n)} = (\mathbf{b}(n) \cdot \dot{\mathbf{f}}(n)) e(n) \end{aligned} \tag{24}$$



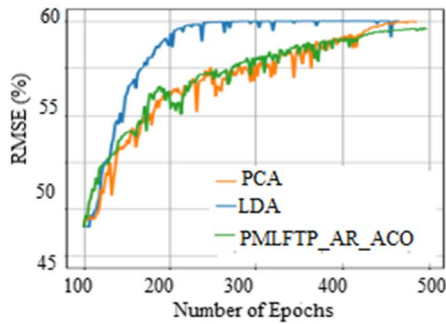
**Fig. 3** Analysis of fault situation 1 in terms of **a** computational rate, **b** QoS, **c** RMSE, **d** fault detection rate, **e** control optimization



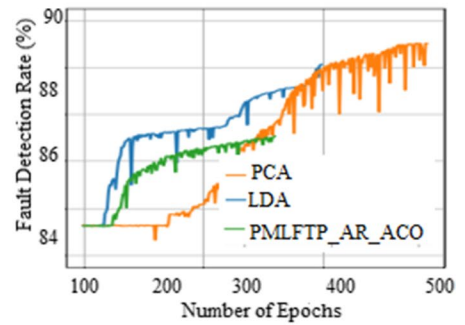
(a) computational rate



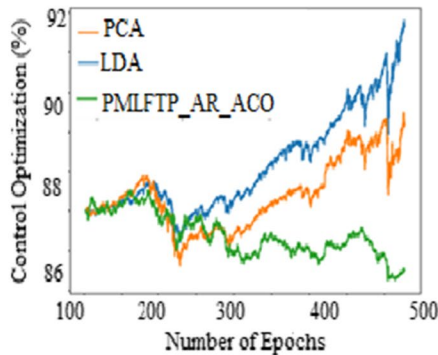
(b) QoS



(c) RMSE



(d) fault detection rate



(e) control optimization

Let the differential operator be expressed by Eq. (25),

$$R_d\{g(w(n))\} \equiv \left. \frac{\partial}{\partial \alpha} g(w(n) + \alpha d(n)) \right|_{\alpha=0} \quad (25)$$

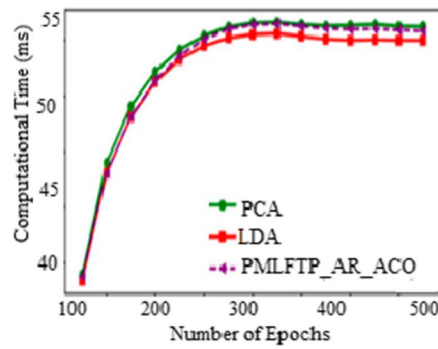
Then,  $H(w(n))d(n)$  is given by Eq. (26)

$$H(w(n))d(n) = \Re\{\nabla E_T(w(n))\} \quad (26)$$

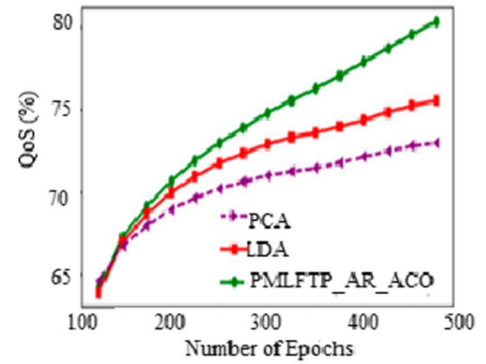
During training and tests were calculated based on correlation,  $\rho$ , and variance of error  $\sigma_e^2$ , given by Eqs. (27) and (28):

$$\rho = \frac{\sum_{i=0}^{N-1} (x_i(n) - \hat{x}(n))(y_i(n) - \hat{y}(n))}{\sqrt{\sum_{i=0}^{N-1} (x_i(n) - \hat{x}(n))^2 \sum_{i=0}^{N-1} (y_i(n) - \hat{y}(n))^2}} \quad (27)$$

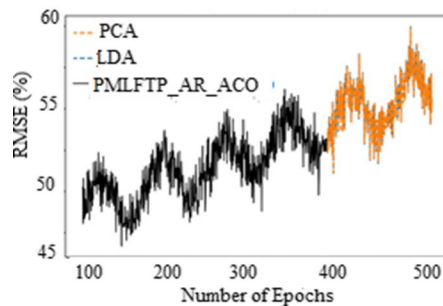
**Fig. 4** Analysis of fault situation 2 in terms of **a** computational rate, **b** QoS, **c** RMSE, **d** fault detection rate, **e** control optimization



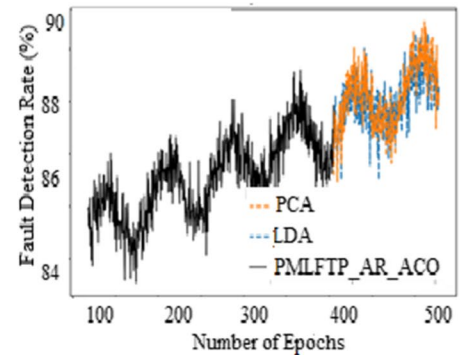
(a) computational rate



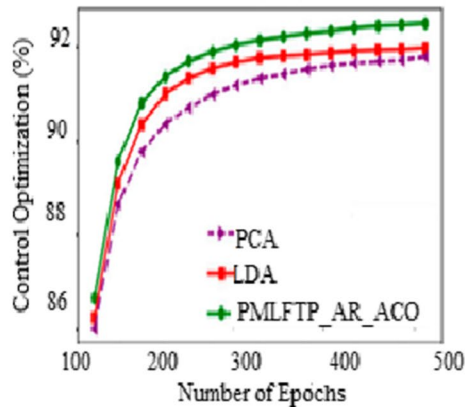
(b) QoS



(c) RMSE



(d) fault detection rate



(e) control optimization

$$\sigma_e^2 = \frac{1}{N} \sum_{i=1}^N \left[ (x_i(n) - y_i(n)) - (\hat{x}(n) - \hat{y}(n)) \right]^2 \quad (28)$$

### 3.2 Data controlling and optimization using auto-regression-based ant colony optimization (AR\_ACO)

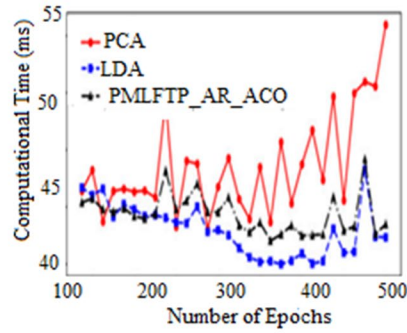
The autoregressive model AR(p) uses a linear combination of the  $p$  last values by Eq. (29) to determine the value of a process at an arbitrary time step  $t$ .

$$y_t = \phi_1 \cdot y_{t-1} + \phi_2 \cdot y_{t-2} + \dots + \phi_p \cdot y_{t-p} + \varepsilon_t \quad (29)$$

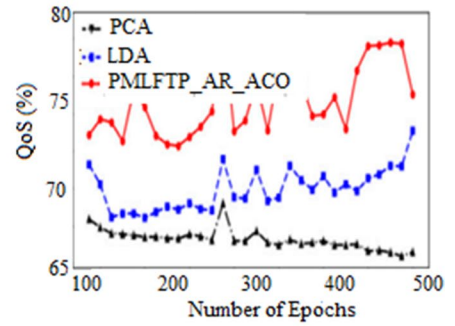
The order of the AR model is denoted by the number  $p$ . The model parameters are the weights  $\phi_i$  of the linear combination. They are thought to be constant. Furthermore, an AR model requires that this process is superimposed by white noise.  $\varepsilon_t$  are regarded uncorrelated in time and identically distributed, with a zero expected value and finite variance. AR(p) is the abbreviation for this model.

The AR(p) model is used to characterize a given time series, as shown in Eq. (30):

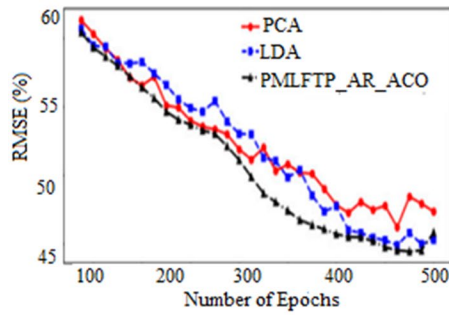
**Fig. 5** Analysis of fault situation 3 in terms of **a** computational rate, **b** QoS, **c** RMSE, **d** fault detection rate, **e** control optimization



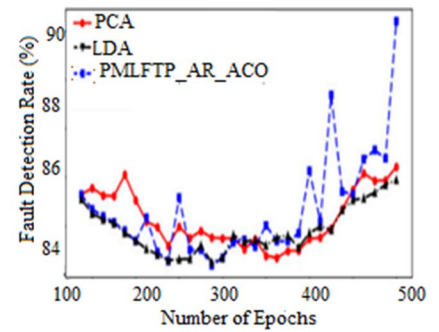
(a) computational rate



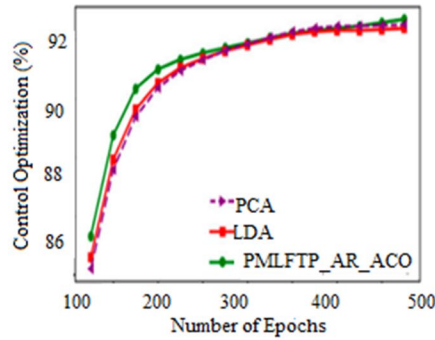
(b) QoS



(c) RMSE



(d) fault detection rate



(e) control optimization

$$y_{t-3}, y_{t-2}, y_{t-1}, y_t, y_{t+1}, y_{t+2}, y_{t+3}, \dots \quad (30)$$

This calculation assumes a one-unit change over time. In general, the time step can be of any unit, and it can be substituted by  $\Delta t$  by altering the unit of time, and the equation can be rewritten as Eq. (31):

$$\frac{y_t - y_{t-\Delta t}}{\Delta t} = \phi_0 + (\phi_1 - 1)y_{t-\Delta t} \quad (31)$$

The AR(p) model can be thought of as a linear operator that is applied to an initial vector of time series data. In this view, the definition's equation is expressed as a matrix equation by Eqs. (32) and (33):

$$\underbrace{\begin{pmatrix} y_t \\ y_{t-1} \\ \vdots \\ y_{t-p+1} \end{pmatrix}} = A_p \cdot \begin{pmatrix} y_{t-1} \\ y_{t-2} \\ \vdots \\ y_{t-p} \end{pmatrix} \quad (32)$$

with a matrix

$$A_p = \begin{pmatrix} \phi_1 & \phi_2 & \phi_3 & \dots & \phi_{p-2} & \phi_{p-1} & \phi_p \\ 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 1 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (33)$$



**Table 1** Comparative analysis for various fault situations between proposed and existing technique

Fault situations	Techniques	Computational Time	QoS	RMSE	Fault detection rate	Control optimization
Fault situation-1	PCA	55	76	60	86	86
	LDA	53	78	58	88	90
	PMLFTP_AR_ACO	50	80	57	89	92
Fault situation-2	PCA	55	73.2	58	84	91
	LDA	53	75	55	88.7	91.8
	PMLFTP_AR_ACO	52	79.9	50	89.9	92
Fault situation-3	PCA	55	65	52	85	91
	LDA	43	74	48	86	92
	PMLFTP_AR_ACO	40	78	45	90	93

The attribute graph’s nodes are arranged in the same order as the composite vector’s boundary values. There are N edges  $e_{ik}, k1::N$  between two consecutive nodes  $v_i$  and  $v_{i+1}$  when N binary BCs are combined. Each edge,  $e_{ik}$  in E, shows a couple of conditional probabilities (i.e.,  $P([v_i, v_{i+1}] | c_1), P([v_i, v_{i+1}] | \bar{c}_2)$ ) integrated with attribute interval  $[v_i, v_{i+1}]$ . Conditional probability distribution of original instance of attribute from kth BC is used to calculate these probabilities. Conditional probabilities labelling edge  $e_{i1} = [v_1, v_2]$  are evaluated in the following way by Eq. (34):

$$P([v_1, v_2] | c_1) = \frac{P([v_1^1, v_2^1] | c_1) * (v_2 - v_1)}{(v_2^1 - v_1^1)} \tag{34}$$

The characteristic polynomial of  $A_p$  is  $\chi_p(\lambda) = (-1)^p * (\lambda^p - \phi_1 \lambda^{p-1} - \dots - \phi_{p-1} \lambda - \phi_p)$   
 The case  $p = 1$  with Eq. (35)

$$\det(A_1 - \lambda I) = \det(\phi_1 - \lambda) = \phi_1 - \lambda = -1 * (\lambda - \phi_1) = \chi_1(\lambda). \tag{35}$$

Induction step starts with Eq. (36):

$$\det(A_p - \lambda I) = \det \begin{pmatrix} \phi_1 - \lambda & \phi_2 & \phi_3 & \dots & \phi_{p-2} & \phi_{p-1} & \phi_p \\ 1 & -\lambda & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & -\lambda & \dots & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & -\lambda & 0 & 0 \\ 0 & \dots & 0 & 0 & 1 & -\lambda & 0 \\ 0 & \dots & 0 & 0 & 0 & 1 & -\lambda \end{pmatrix} \tag{36}$$

As indicated in Eq. (37), the determinant of this matrix will be determined by utilizing the Laplace expansion along the last column.

$$\det(A_p - \lambda I) = -(-1)^p * \phi_p * \det \begin{pmatrix} 1 & -\lambda & 0 & \dots \\ 0 & 1 & -\lambda & \dots \\ \vdots & \ddots & \ddots & \ddots \\ 0 & \dots & 0 & 1 \end{pmatrix} + (-\lambda) * \det(A_{p-1} - \lambda I) \tag{37}$$

The Laplace expansion reduces the  $p \times p$  matrix  $(A_p - \lambda I)$  into two matrices. The first matrix’s determinate is one since its structure has a lower triangular matrix of zeros, and the diagonal’s product is one. The induction hypothesis is met by the second matrix, which is given by Eq. (38):

$$\det(A_p - \lambda I) = (-1)^p * (-\phi_p) - \lambda * \det(A_{p-1} - \lambda I) = (-1)^p * (-\phi_p) - \lambda * [(-1)^{p-1} * (\lambda^{p-1} - \phi_1 \lambda^{p-2} - \dots - \phi_{p-2} \lambda - \phi_{p-1})] \tag{38}$$

characteristic polynomial of  $A_p$  is  $\chi_p(\lambda)$   
 k-fold application of linear operator  $A_p$  is represented by Eq. (39):

$$A_p^k = (TDT^{-1})^k = TD \underbrace{T^{-1} * T}_{=I} DT^{-1} \dots TDT^{-1} = TD^k T^{-1} \tag{39}$$

Using corresponding eigenvectors, matrix  $A_2$  is decomposed into  $A_2 = T * D * T^{-1}$  shown in Eq. (40):

$$T = \begin{pmatrix} \frac{1}{2} - \frac{\sqrt{5}}{10} & \frac{1}{2} + \frac{\sqrt{5}}{10} \\ -\frac{\sqrt{5}}{5} & \frac{\sqrt{3}}{5} \end{pmatrix} \text{ and } D = \begin{pmatrix} \frac{1}{2} - \frac{1}{2}\sqrt{5} & 0 \\ 0 & \frac{1}{2} + \frac{1}{2}\sqrt{5} \end{pmatrix} \tag{40}$$

closed form of this AR(2)-model is represented by Eq. (41):

$$\begin{pmatrix} y_{k+1} \\ y_k \end{pmatrix} = A_2^k * \begin{pmatrix} y_1 \\ y_0 \end{pmatrix} = T * \begin{pmatrix} (\frac{1}{2} - \frac{1}{2}\sqrt{5})^k & 0 \\ 0 & (\frac{1}{2} + \frac{1}{2}\sqrt{5})^k \end{pmatrix} * T^{-1} * \begin{pmatrix} y_1 \\ y_0 \end{pmatrix} = (\frac{1}{2} - \frac{3}{10}\sqrt{5}) * (\frac{1}{2} - \frac{1}{2}\sqrt{5})^k + (\frac{1}{2} + \frac{3}{10}\sqrt{5}) * (\frac{1}{2} + \frac{1}{2}\sqrt{5})^k \tag{41}$$

The key principle behind understanding a differential equation as an AR model is that it is symmetric; not only can the difference equation be understood as an AR model, but it can also be reversed. Furthermore, higher-order AR models relate to differential equations of increasing degree. Aside from the following difference quotients, namely, forward difference by Eq. (42):

$$\frac{\Delta y}{\Delta t} = \frac{y_{t+\Delta t} - y_t}{\Delta t} = \frac{y_t - y_{t-\Delta t}}{\Delta t} = \frac{y_{t+\Delta t} - y_{t-\Delta t}}{2\Delta t} \quad (42)$$

There are also difference quotients for the numerical calculation of higher derivatives to approach the first-order derivative. Equation (43) provides a recursive definition of higher-order central difference quotients:

$$\frac{\Delta^n y}{\Delta t^n} = \frac{1}{\Delta t^n} \cdot \sum_{k=0}^n (-1)^k \binom{n}{k} y_{t+k-n/2} \quad (43)$$

for even degrees of  $n$ , and by Eq. (44):

$$\begin{aligned} \frac{\Delta^n y}{\Delta t^n} &= \frac{1}{2\Delta t^n} \cdot \sum_{k=0}^{n-1} (-1)^k \binom{n-1}{k} \cdot (y_{t+k+1-(n-1)/2} - y_{t+k-1-(n-1)/2}) a_n y^{(n)} a_{n-1} y^{(n-1)} \\ &+ \dots + a_2 y'' + a_1 y' + a_0 y = f(x) y^{(n)} + a_{n-1} y^{(n-1)} + \dots + a_2 y'' + a_1 y' + a_0 y = f(x). \end{aligned} \quad (44)$$

by substituting  $k$ -th derivative by  $\lambda^k$  in Eq. (45):

$$\chi(\lambda) = \lambda^n + a_{n-1} \lambda^{(n-1)} + \dots + a_2 \lambda^2 + a_1 \lambda + a_0 \quad (45)$$

The roots are real or occur in conjugate pairs if all coefficients  $a_i$  are real. The rules for solving higher-order differential equations with constant coefficients can then be used to find the required  $n$  linearly independent solutions: If  $r$  is a real root that appears  $k$  times, then Eq. (46) represents the solutions:

$$y = e^{rt}, y = t \cdot e^{r-t}, y = t^2 \cdot e^{r-t}, \dots, y = t^{k-1} \cdot e^{r-t} \quad (46)$$

If  $r = \alpha \pm \beta i$  are complex conjugate roots appearing  $k$  times, then results are given by Eq. (47):

$$\begin{aligned} &e^{\Delta t \cos(\beta \cdot t)}, e^{\Delta t \sin(\beta \cdot t)} t \cdot e^{a-t \cos(\beta \cdot t)}, \\ &t \cdot e^{\Delta t \sin(\beta \cdot t)} t^2 \cdot e^{-t \cos(\beta \cdot t)}, t^2 \cdot e^{a-t \sin(\beta \cdot t)}, \mathbf{U}^{(l)} \\ &= \begin{bmatrix} u_{11}^{(l)} & \dots & u_{1N}^{(l)} \\ \vdots & \ddots & \vdots \\ u_{c1}^{(l)} & \dots & u_{eN}^{(l)} \end{bmatrix} \end{aligned} \quad (47)$$

$$t^{k-1} \cdot e^{a-t \cos(\beta \cdot t)}, t^{k-1} \cdot e^{a-t \sin(\beta \cdot t)}$$

The proposed optimization algorithm is discussed below (Fig. 2).

Stage 1: Assign number of clusters to  $c=2$  and  $c_{\max}$ , and use formula to obtain the right number of clusters (8). Set settings for ACO technique, initialize solution  $S_i, I=1, \dots, T$ , and replace  $S_i$  with  $2i=S_i$  in fitness function (5).

Stage 2: Sort answers by fitness function and then organize them in ascending order. Probability value for renewing results as  $S_i I=1, \dots, T$  using (9) and (10).

Stage 3: Evaluate mean  $\mu$  utilizing probability value produced by Step 2 and roulette technique and set SD  $\sigma$  i v

utilizing (11). If condition  $|f^{(l+1)}(\mu^i) - f^{(l+1)}(S_i)| < 0$  is satisfied, then set  $S_i = \mu^i$ ; otherwise, keep original  $S_i$ .

Stage 4: If condition  $|f^{(l+1)}(S_i) - f^{(l)}(S_i)| < \delta$  is satisfied, then define  $S_i \equiv 2i$ ; otherwise set  $l=l+1$  and return to Step 2.

Stage 5: Examine for less fitness value utilizing (5) until condition  $\|\mathbf{U}^{(d+1)} - \mathbf{U}^{(l)}\| < \epsilon$  is satisfied, where

$$\mathbf{U}^{(l)} = \begin{bmatrix} u_{11}^{(l)} & \dots & u_{1N}^{(l)} \\ \vdots & \ddots & \vdots \\ u_{c1}^{(l)} & \dots & u_{eN}^{(l)} \end{bmatrix}$$

The center of bell-shaped MF  $\alpha_i$  and SD of bell-shaped MF  $\beta_q^i$  is given by following formulas given by Eqs. (48) and (49):

$$\alpha_q^i = \frac{\sum_{k=1}^n u_{ik} x_{qk}}{\sum_{k=1}^n u_{ik}} \quad (48)$$

$$\beta_q^i = \sqrt{\frac{2 \sum_{k=1}^n u_{ik} (x_{qk} - \alpha_q^i)^2}{\sum_{k=1}^n u_{ik}}} \quad (49)$$

The grade of MF  $A_g^i(x_q(k))$  is given by (3) weight  $w_i(k)$  and output  $y^{\wedge}$  is given as  $w_i(k) = \prod_{q=1}^n A_q^i(x_q(k))$ , and  $\hat{y}(k) = \frac{\sum_{i=1}^r m_i(k) y^i(k)}{\sum_{i=1}^r m_i(k)}$

Stage 6: Replace input–output data and  $y^{\wedge}(k)$  by Eq. (50).

$$\mathbf{y} = \hat{\Phi} \hat{\Theta}^T + \mathbf{e} \quad (50)$$

where  $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$ , real model  $\mathbf{y} = [y(1) \dots y(N)]^T$  the established model  $\hat{\mathbf{y}} = [\hat{y}(1) \dots \hat{y}(N)]^T$ ,  $\hat{\Phi}$  is shown in Eq. (51)

$$\hat{\Phi} = \begin{bmatrix} \Phi(1) \\ \vdots \\ \Phi(N) \end{bmatrix} = \begin{bmatrix} x_1^T \\ \vdots \\ x_r^T \end{bmatrix}^T = \begin{bmatrix} \Phi_1 & \Phi_{1,x_1(1)} & \dots & \Phi_{1,x_n(1)} & \dots & \Phi_c & \Phi_{c,x_1(1)} & \dots & \Phi_{c,x_n(1)} \\ \Phi_1 & \Phi_{1,x_1(2)} & \dots & \Phi_{1,x_n(2)} & \dots & \Phi_c & \Phi_{c,x_1(2)} & \dots & \Phi_{c,x_n(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \Phi_1 & \Phi_{1,x_n(N)} & \dots & \Phi_{1,x_n(N)} & \dots & \Phi_c & \Phi_{c,x_1(N)} & \dots & \Phi_{c,x_n(N)} \end{bmatrix} \quad (51)$$

$$\begin{aligned} \Phi_i &= w_i / \sum_{i=1}^c w_i, \hat{\Theta} = [\theta_1 \dots \theta_c] \\ &= [a_0^1 \dots a_n^1 \mid \dots \mid a_0^c \dots a_n^c] \end{aligned}$$

Step 7: Convert set  $\chi_i$ , where  $i=1, \dots, r$ , produced from Step 6 as orthogonal basis vectors, utilizing subsequent process:

1. Set  $\hat{m} = 1$
2.  $\Gamma_1 = \chi_1$  and  $q_1 = \frac{\langle \Gamma_1, y_2 \rangle}{\langle \Gamma_1, \Gamma_1 \rangle}$
3. For  $\hat{m} = 2$  to  $r$

$$\gamma_{iim} = \frac{\langle \Gamma_{1,xa} \rangle}{\langle \Gamma_{i,\Gamma} \rangle}, 1 \leq i < m$$

$q_i = \frac{\langle \Gamma_{i,y} \rangle}{\langle \Gamma_{i,\Gamma} \rangle}$  3) Figure out  $\hat{\Theta}$  by subsequent equation:

$$A\Theta^T = \mathbf{q}$$

where

$$A = \begin{bmatrix} 1 & \gamma_{12} & \gamma_{13} & \dots & \dots & \gamma_{1r} \\ 0 & 1 & \gamma_{23} & \dots & \dots & \gamma_{2r} \\ 0 & 0 & 1 & \dots & \dots & \gamma_{3r} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & \gamma_{(r-1)r} \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

$$\mathbf{q} = [q_1 \ q_2 \ \dots \ q_r]$$

### 4 Results and discussion

The Python programming language was used to implement the strategy. The proposed solution is implemented using the Theano library, which features dynamic C code generation, robust and rapid optimization techniques, and integration with mathematical NumPy library. A learning method plus a real-time module make up implementation. Learning method learns the parameters for both spatial pooling as well as temporal inference continuously and saves them in a database that is shared with real-time module. Real-time method uses parameters contained in shared database to execute real-time FDI. Module is simply concerned with execution of technique with already learned parameters and does not perform any learning. This operation of separating the learning and execution processes is required to provide real-time functioning, which would otherwise be impossible to achieve. The learning method is executed on a dedicated server, and deployed method is made available as a service. System begins by obtaining multiple data samples from an SPC database. Database stores a lot of signals created by manufacturing methods as they occur throughout time. This information is recorded in a database as textual data as well as imported into computer memory by learning method as a list of string objects.

Table 1 shows the parametric comparison for fault situation 1, fault situation 2, and fault situation 3. The parameters considered are computational rate, QoS, RMSE, fault detection rate, and control optimization. The techniques compared are PCA and LDA with proposed technique. The graph for above comparison table is given below.

Figures 3, 4, and 5 show parametric analysis in terms of computational rate, QoS, RMSE, fault detection rate, control

optimization. The proposed technique has been compared with existing PCA and LDA. In terms of computational rate of 50%, QoS of 80%, RMSE of 57%, fault detection rate of 89%, and control optimization of 92% for fault situation 1 by proposed technique. Computational rate of 52%, QoS of 79.9%, RMSE of 50%, fault detection rate of 89.9%, and control optimization of 92% for fault situation 2 calculated based on fault detection. Based on this comparison, proposed technique obtained higher QoS and fault detection rate in fault location. In terms of computational rate of 40%, QoS of 78%, RMSE of 45%, fault detection rate of 90%, and control optimization of 93% for fault situation 3 by proposed technique.

More crucially, an inferential model like this might be used to forecast paperboard qualities like flat crush strength and compression strength directly. Controlling the refining parameters directly based on feedback from the finished product (board or paper) quality could be a modern control method for refining. Overall, the findings of this study showed that machine learning-based solutions have a lot of promise in the pulp and paper industry, as long as the constraints of data-driven solutions are understood and significant process expertise is used when constructing predictive models.

### 5 Conclusion

This research proposed novel design in monitoring and control optimization of soft sensors in automation industry for fault detection. The aim is to collect the historical data from soft sensors in designing fault detection systems for monitoring and controlling with optimization. Then, to pre-process the collected data in removing null values and missing data. For detection and diagnosis of the faults of processed data using probabilistic multi-layer Fourier transform perceptron (PMLFTP). Then, optimization and control of the data of soft sensors have been done using auto-regression-based ant colony optimization (AR\_ACO) which has effect in increasing the production of industry automatically. The experimental results have been carried out in terms of computational rate of 40%, QoS of 78%, RMSE of 45%, fault detection rate of 90%, and control optimization of 93% which been obtained for various historical data-based evaluations.

**Acknowledgements** Funding: No

**Author contribution** Wongchai A: conceived and design the analysis. Mohammed A.S. Abourehab: editing and figure design, investigation. Mohammed Altaf Ahmed: methodology, writing—original draft preparation, collecting the data. Saibal Dutta: contributed data and analysis stools, performed and analysis, software, validation. Koduganti Venkatrao: visualization, conception and design of study,

conceptualization, wrote the paper. Kashif Irshad: funding acquisition, investigation, project administration, supervision, writing—review and editing.

**Data availability** N/A.

**Code availability** N/A.

#### Declarations

We confirm that we have read, understand, and agreed to the submission guidelines, policies, and submission declaration of the journal.

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

**Consent to participate** Not applicable

**Consent for publication** Not applicable

**Conflict of interest** The authors declare no competing interests.

## References

- Iqbal R, Maniak T, Doctor F, Karyotis C (2019) Fault detection and isolation in industrial processes using deep learning approaches. *IEEE Trans Industr Inf* 15(5):3077–3084
- Villalba-Diez J, Schmidt D, Gevers R, Ordieres-Meré J, Buchwitz M, Wellbrock W (2019) Deep learning for industrial computer vision quality control in the printing industry 4.0. *Sensors* 19(18):3987
- Lyu Y, Chen J, Song Z (2019) Image-based process monitoring using deep learning framework. *Chemom Intell Lab Syst* 189:8–17
- Sun Q, Ge Z (2021) A survey on deep learning for data-driven soft sensors. *IEEE Trans Industr Inf* 8:465–471
- Lomov I, Lyubimov M, Makarov I, Zhukov LE (2021) Fault detection in Tennessee Eastman process with temporal deep learning models. *J Ind Inf Integr* 23:100216
- Rai R, Tiwari MK, Ivanov D, Dolgui A (2021) Machine learning in manufacturing and industry 4.0 applications. *Int J Prod Res* 59(16):4773–4778
- Zheng D, Zhou L, Song Z (2021) Kernel generalization of multi-rate probabilistic principal component analysis for fault detection in nonlinear process. *IEEE/CAA Journal of Automatica Sinica* 8(8):1465–1476
- Cecconi F, Rosso D (2021) Soft sensing for on-line fault detection of ammonium sensors in water resource recovery facilities. *Environ Sci Technol* 55(14):10067–10076
- Ullrich T (2021) On the autoregressive time series model using real and complex analysis. *Forecasting* 3(4):716–728
- Tsai SH, Chen YW (2016) A novel fuzzy identification method based on ant colony optimization algorithm. *IEEE Access* 4:3747–3756
- Bouktif S, Hanna EM, Zaki N, Khousa EA (2014) Ant colony optimization algorithm for interpretable Bayesian classifiers combination: application to medical predictions. *PLoS One* 9(2):e86456
- Rao BN, Chowdhury R (2008) Probabilistic analysis using high dimensional model representation and fast Fourier transform. *Int J Comput Methods Eng Sci Mech* 9(6):342–357
- Ribeiro MV, Barbedo JGA, Romano JMT, Lopes A (2005) Fourier-lapped multilayer perceptron method for speech quality assessment. *EURASIP J Adv Signal Process* 2005(9):1–10
- Fedotov A, Fedotov E, Bahteev K (2017) Application of local Fourier transform to mathematical simulation of synchronous machines with valve excitation systems. *Latv J Phys Tech Sci* 54(1):31
- Cai Q, Zhang D, Zheng W, Leung SC (2015) A new fuzzy time series forecasting model combined with ant colony optimization and auto-regression. *Knowl-Based Syst* 74:61–68
- Taqvi SAA, Zabiri H, Tufa LD, Uddin F, Fatima SA, Maulud AS (2021) A review on data-driven learning approaches for fault detection and diagnosis in chemical processes. *Chem Bio Eng Rev*
- Bernardi E, Adam EJ (2020) Observer-based fault detection and diagnosis strategy for industrial processes. *J Frankl Inst*
- Liu B, Chai Y, Liu Y, Huang C, Wang Y, Tang Q (2021) Industrial process fault detection based on deep highly-sensitive feature capture. *J Process Control* 102:54–65
- Huang K, Wu S, Li F, Yang C, Gui W (2021) Fault Diagnosis of Hydraulic Systems Based on Deep Learning Model With Multirate Data Samples. *IEEE Transactions on Neural Networks and Learning Systems*
- Kazemi P, Bengoa C, Steyer JP, Giralt J (2021) Data-driven techniques for fault detection in anaerobic digestion process. *Process Saf Environ Prot* 146:905–915
- Yella J, Zhang C, Petrov S, Huang Y, Qian X., Minai AA, Bom S (2021) Soft-sensing conformer: a curriculum learning-based convolutional transformer. In 2021 IEEE International Conference on Big Data (Big Data) (pp. 1990–1998). IEEE
- Zhe L, Wang KS (2017) Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario. *Adv Manuf* 5:377–387
- Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, Hasan M, Van Essen BC, Awwal AAS, Asari VK (2019) A State-of-the-art survey on deep learning theory and architectures. *Electronics*
- Neupane D, Seok J (2020) Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: a review. *IEEE Access* 8:93155–93178
- Deutsch J, He D (2018) Using deep learning-based approach to predict remaining useful life of rotating components. *IEEE Trans Syst Man Cybern Syst* 48:11–20

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.