The International Journal of Advanced Manufacturing Technology (2023) 128:2885–2897
https://doi.org/10.1007/s00170-023-11602-y

ORIGINAL ARTICLE

Manufacturing industry based on dynamic soft sensors in integrated with feature representation and classification using fuzzy logic and deep learning architecture

Shakir Khan[1,2] · Tamanna Siddiqui[3] · Azrour Mourade[4] · Bayan Ibrahimm Alabduallah[5] · Saad Abdullah Alajlan[1] · Abrar almjally[1] · Bader M. Albahlal[1] · Amani Alfaifi[1]

Received: 1 March 2023 / Accepted: 12 May 2023 / Published online: 6 June 2023

Abstract

Soft sensors are data-driven devices that allow for estimates of quantities that are either impossible to measure or prohibitively expensive to do so. DL (deep learning) is a relatively new feature representation method for data with complex structures that has a lot of promise for soft sensing of industrial processes. One of the most important aspects of building accurate soft sensors is feature representation. This research proposed novel technique in automation of manufacturing industry where dynamic soft sensors are used in feature representation and classification of the data. Here the input will be data collected from virtual sensors and their automation-based historical data. This data has been pre-processed to recognize the missing value and usual problems like hardware failures, communication errors, incorrect readings, and process working conditions. After this process, feature representation has been done using fuzzy logic-based stacked data-driven auto-encoder (FL_SDDAE). Using the fuzzy rules, the features of input data have been identified with general automation problems. Then, for this represented features, classification process has been carried out using least square error backpropagation neural network (LSEBPNN) in which the mean square error while classification will be minimized with loss function of the data. The experimental results have been carried out for various datasets in automation of manufacturing industry in terms of computational time of 34%, QoS of 64%, RMSE of 41%, MAE of 35%, prediction performance of 94%, and measurement accuracy of 85% by proposed technique.

Keywords Soft sensors · Deep learning · Automation · FL_SDDAE · Classification · LSEBPNN

✉ Shakir Khan
sgkhan@imamu.edu.sa

✉ Bayan Ibrahimm Alabduallah
Bialabdullah@pnu.edu.sa

Tamanna Siddiqui
tsiddiqui.cs@amu.ac.in

Azrour Mourade
mo.azrour@umi.ac.ma

Saad Abdullah Alajlan
saalajlan@imamu.edu.sa

Bader M. Albahlal
bmalbahlal@imamu.edu.sa

Amani Alfaifi
Amani0004@hotmail.com

[1] College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia

[2] Department of Computer Science and Engineering, University Centre for Research and Development, Chandigarh University, Mohali 140413, India

[3] Department of Computer Science, Aligarh Muslim University, Aligarh, UP, India

[4] Computer Sciences Department, Faculty of Sciences and Technics, Moulay Ismail University, Meknes, Morocco

[5] Department of Information Systems, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia

1 Introduction

Soft sensor is a virtual inferential prediction method that uses easily measured variables to forecast process variables that are difficult to measure directly due to technological, economic constraints as well as a complex environment. Soft sensor attempts to construct a regression prediction method between easily measured variables as well as difficultly measured variables, which is used to address issue that hinders measurements from being used as feedback signals in quality control methods [1]. For at least 10 years, there has been a growing trend in use of data-driven AI (artificial intelligence) approaches to enhance machines, processes, and products across several industrial domains [2]. In recent years, reducing emissions as a result of stronger environmental restrictions has also been a major motivator [3]. However, gathering the data required for such approaches is fraught with difficulties, one of which is the long life of industrial gear. Official depreciation estimates range from (rarely) 6 to more than 30 years, depending on the country, type of machinery, and industrial sector [4]. Experience suggests that, particularly in small and medium-sized businesses, resilient equipment can last even longer in regular usage. Soft sensor approaches are used more widely in industrial processes, and they have become a key emerging trend in both academics as well as industry [5]. Early academics proposed model predictive control like generalized predictive control, dynamic matrix predictive control and model control method, in light of model prediction in industrial production process [6]. However, these soft sensor prediction approaches have several flaws. ANN (Artificial neural networks), rough set, SVM (support vector machine) and hybrid techniques are some AI and ML methods based on data-driven technologies that have been proposed to solve issues where it is difficult to measure key processes as well as quality variables for soft sensor methods as a result of DL in soft sensor control method as well as continuous progress in engineering technology [7].

The contribution of this research is as follows:

- To design novel techniques in automation of manufacturing industry where the dynamic soft sensors are used in feature representation and classification of the data
- To collect the data cloud storage and create the virtual sensors dataset based on gear fault detection, spindle fault detection, and bearing fault detection in automation industry
- To represent the feature using fuzzy logic-based stacked data driven auto-encoder (FL_SDDAE) where the features of input data have been identified with general automation problems.
- Then, the features have been classified using least square error backpropagation neural network (LSEBPNN) in which the mean square error while classification will be minimized with loss function of the data
- Here the experimental results have been carried out in terms of QoS, measurement accuracy, RMSE, MAE, prediction performance, and computational time.

Research organization is as follows. In Section 2, related works are described. Section 3 gives details of proposed method. proposed method performance, and the results are present in Section 4. Finally, Section 5 concludes the work.

2 Related works

DL-based techniques are recently exhibited solid representation competency and success in a variety of computer science domains, including image processing, computer vision, NLP, and more [8]. Stack autoencoder (SAE) [9], DBN (deep belief network) [10], CNN [11], and LSTM [12] are some of widely utilized deep network architectures. Greedy layer-wise unsupervised pre-training, as well as supervised fine-tuning, are highly important for DL architectures like SAE. The SAE weights evaluated during unsupervised pre-training step are used in supervised fine-tuning stage, which is a more significant method than random weight initialization [13]. As a result, various industrial applications of soft sensors based on SAE [14] are presented. Same authors improved this result significantly by utilizing a TDNN in [15].Mean error dropped to just 1.14 to 1.32% and 1.65° to 3.08°, in the same conditions utilized in [16, 17]. As a result, the type of network used in these two papers had a significant impact on the algorithms' performance. In [18], an RNN is presented that collects information regarding air–fuel ratio λ, ignition angle, and turbocharger boost pressure in addition to rotational speed signal. Focus was on neural network design, which had a significant impact on the algorithm's performance. To estimate cylinder pressure curves, [19] uses a NN with RBF (radial basis functions) as well as consequently no recurrence. Authors of [20] presents a novel convolutional, BiGRU, and Capsule network-based deep learning model, HCovBi-Caps, to classify the hate speech and authors of [21] introduce BiCHAT: a novel BiLSTM with deep CNN and hierarchical attention-based deep learning model for tweet representation learning toward hate speech detection. Authors of [15] do not use raw rotational speed signal, but instead translate it into frequency domain as well as process only first 20 harmonics, to earlier research are used an RBF network. They also employ structure-borne sound signal's 21st–50th harmonics. As a result, the preparation of the given data is the most important aspect of this project. The typical errors for pMax as well as its position

in crank angle range are 3.4% and 1.5°, respectively. Using a multi-layer perceptron, [22] predicts combustion parameters directly from crankshaft's rotational speed as well as acceleration data, in contrast to the previous studies (MLP). The mean error lies between 1.38° and 9.1°, with a range of 4.1 to 8.0%. A deep learning-based R2DCNNMC model is proposed for detection and classification of COVID-19 employed chest X-ray images data [23]. Privacy of data driven uses on the k-anonymity and l-diversity supervised models classifies the healthcare data [24]. Virtualization for dynamics on cloud for network operation and management is discussed in [25] and proposed hybrid model on cloud ensures the maximum benefits from virtualization.

The effective implementations of SAE-based DL listed above reveal a significant capacity to extract features. Deep structures exceed typical soft-sensing prediction performance thanks to unsupervised layer-wise pre-training as well as supervised fine-tuning processes. Proposed industrial soft sensors are static methods based on notion of a static process as well as steady-state. However, the inherently dynamic nature of industrial processes cannot be neglected. Chemical processes, for example, are highly dynamic, with current state being linked to earlier ones. As a result, time-related characteristics of time-series recorded data are important.

3 System model

This section discusses the proposed design in automation of manufacturing industry based on dynamic soft sensors. Here the data has been processed to recognize the missing value and usual problems like hardware failures, incorrect readings, communication errors, process working conditions. Then their features have been represented in module 1 and the represented feature has been classified in module 2 using deep learning techniques. The overall research architecture is given in Fig. 1.

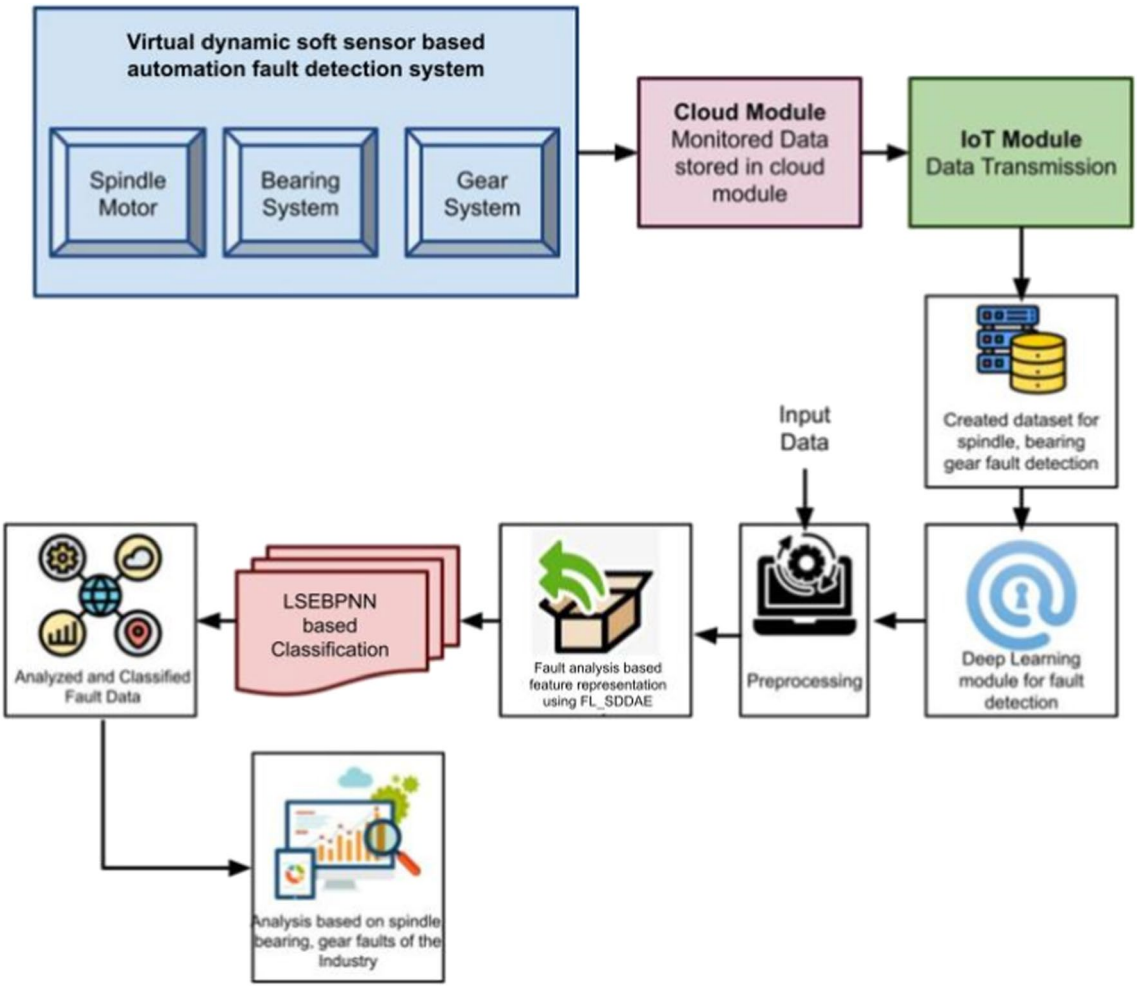

Fig. 1 Overall Proposed diagram for virtual sensor-based fault detection in automation industry

3.1 Feature representation using fuzzy logic based stacked data driven auto-encoder (FL_SDDAE)

ENotice is replaced Eq. (2) represents overall input–output transfer function about general autoencoder (AE) structure. The input $\left(x^{[\alpha]} \in \mathcal{R}^d\right)$ is supplied to hidden layer, whose output is utilized to reconstruct $\left(\overset{\backprime}{x}^{[\alpha]}\right)$ input through output layer (y) as shown in Eq. (1).

$$\overset{\backprime}{x}^{[\alpha]} = y_{(W',b')}\left(h_{(W,b)}\left(x^{[\alpha]}\right)\right) \equiv x^{[\alpha]} \tag{1}$$

An encoder or recognition model is another name for this approach. Optimize variational parameters φ such that as shown in Eq. (2):

$$q_\phi(\mathbf{z}|\mathbf{x}) \approx p_\theta(\mathbf{z}|\mathbf{x}) \tag{2}$$

As stated in Eq. (3), inference models are any directed graphical model.

$$q_\phi(\mathbf{z}|\mathbf{x})\left(\mathbf{z}_1, \ldots, \mathbf{z}_M|\mathbf{x}\right) = \prod_{j=1}^{M} q_\phi\left(\mathbf{z}_j\middle|Pa\left(\mathbf{z}_j\right), \mathbf{x}\right) \tag{3}$$

In the directed graph, $Pa\left(\mathbf{z}_j\right)$ is set of parent variables of variable $\mathbf{z}_j$.

$$\begin{aligned}\log p_\theta(\mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x})\right] = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_\theta(\mathbf{x},\mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})}\right]\right] \\ &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})}\right]\right] = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right]\right]}_{=\mathcal{L}_{\theta,\phi}(\mathbf{x})} \\ &\quad + \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})}\right]\right]}_{=D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\middle\|p_\theta(\mathbf{z}|\mathbf{x})\right)}\end{aligned} \tag{4}$$

The non-negative Kullback–Leibler (KL) divergence between $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{z}|\mathbf{x})$ is the second term in Eq. (5):

$$D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\middle\|p_\theta(\mathbf{z}|\mathbf{x})\right) \geq 0 \tag{5}$$

The variational lower bound, commonly known as ELBO, is the first term in Eq. (6):

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right] \tag{6}$$

Because the KL divergence is non-negative, ELBO shows a lower bound on data's log-likelihood, as demonstrated in Eq. (7).

$$\begin{aligned}\mathcal{L}_{\theta,\phi}(\mathbf{x}) &= \log p_\theta(\mathbf{x}) - D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\middle\|p_\theta(\mathbf{z}|\mathbf{x})\right) \\ &\leq \log p_\theta(\mathbf{x})\end{aligned} \tag{7}$$

$$\begin{aligned}\nabla_\theta\mathcal{L}_{\theta,\phi}(\mathbf{x}) &= \nabla_\theta\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right] \\ &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\nabla_\theta\left(\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right)\right] \\ &\simeq \nabla_\theta\left(\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right) = \nabla_\theta\left(\log p_\theta(\mathbf{x},\mathbf{z})\right)\end{aligned} \tag{8}$$

Because the ELBO's expectation is taken $q_\phi(\mathbf{z}|\mathbf{x})$, which is a function of φby Eq. (9):

$$\begin{aligned}\nabla_\phi\mathcal{L}_{\theta,\phi}(\mathbf{x}) &= \nabla_\phi\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right] \\ &\neq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\nabla_\phi\left(\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right)\right]\end{aligned} \tag{9}$$

Apply a reparameterization approach to compute unbiased estimates of $\nabla_\phi\mathcal{L}_{\theta,\phi}(\mathbf{x})$, in case of continuous latent variables.

Replace an expectation w.r.t. $q_\phi(\mathbf{z}|\mathbf{x})$ with one w.r.t. p_θ via reparameterization given by Eq. (10).

$$\begin{aligned}\mathcal{L}_{\theta,\phi}(log\mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right] \\ &= \mathbb{E}_{p(\epsilon)}\left[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right]\end{aligned} \tag{10}$$

$$\begin{aligned}\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})\mathbf{z} = \mathbf{g}(\boldsymbol{\phi},\mathbf{x},\boldsymbol{\epsilon})\overset{\backprime}{\mathcal{L}}_{\theta,\boldsymbol{\phi}}(\mathbf{x}) &= \log p_\theta(\mathbf{x},\mathbf{z}) \\ &\quad - \log q_\phi(\mathbf{z}|\mathbf{x})\mathbb{E}_{p(\epsilon)}\left[\nabla_{\theta,\phi}\overset{\backprime}{\mathcal{L}}_{\theta,\boldsymbol{\phi}}(\mathbf{x};\boldsymbol{\epsilon})\right] \\ &= \mathbb{E}_{p(\epsilon)}\left[\nabla_{\theta,\boldsymbol{\phi}}\left(\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\right)\right] \\ &= \nabla_{\theta,\boldsymbol{\phi}}\left(\mathbb{E}_{p(\epsilon)}\left[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\right]\right) \\ &= \nabla_{\theta,\boldsymbol{\phi}}\mathcal{L}_{\theta,\boldsymbol{\phi}}(\mathbf{x})\end{aligned} \tag{11}$$

A simple factorized vGaussian encoder by Eq. (12)

$$\begin{aligned}q_\phi(\mathbf{z}|\mathbf{x}) &= \mathcal{N}\left(\mathbf{z};\boldsymbol{\mu},\ \mathrm{diag}\left(\boldsymbol{\sigma}^2\right)\right) : (\boldsymbol{\mu},\ \log\boldsymbol{\sigma}) \\ &= \mathrm{EncoderNeuralNet}_{\boldsymbol{\phi}}(\mathbf{x})q_\phi(\mathbf{z}|\mathbf{x}) \\ &= \prod_i q_\phi(\mathbf{z}|\mathbf{x}) = \prod_i \mathcal{N}\left(z_i;\mu_i,\sigma_i^2\right)\end{aligned}$$

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon} \tag{12}$$

The log determinant of the Jacobian is given by Eq. (13):

$$\log d_\phi(\mathbf{x},\boldsymbol{\epsilon}) = \log\left|\det\left(\frac{\partial\mathbf{z}}{\partial\boldsymbol{\epsilon}}\right)\right| = \sum_i \log\sigma_i \tag{13}$$

and the posterior density is given by Eq. (14):

$$\begin{aligned}\log q_\phi(\mathbf{z}|\mathbf{x}) &= \log p(\boldsymbol{\epsilon}) - \log d_\phi(\mathbf{x},\boldsymbol{\epsilon}) = \sum_i \log\mathcal{N}\left(\epsilon_i;0,1\right) \\ &\quad - \log\sigma_i \text{when } z = g(\boldsymbol{\epsilon},\boldsymbol{\phi},\mathbf{x})\end{aligned} \tag{14}$$

From Eq. (15)

$$\boldsymbol{\Sigma} = \mathbb{E}\left[(\mathbf{z} - \mathbb{E}[\mathbf{z}])(\mathbf{z} - \mathbb{E}[\mathbf{z}])^T\right] = \mathbb{E}\left[\mathbf{L}\boldsymbol{\epsilon}(\mathbf{L}\boldsymbol{\epsilon})^T\right] = \mathbf{L}\mathbb{E}\left[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T\right]\mathbf{L}^T = \mathbf{L}\mathbf{L}^T \tag{15}$$

Let Gx be defined: $X \subset Rn \rightarrow R$, that is, a function on compact set $X = \alpha 1, 1 \times \ldots \times [\alpha n, \beta n]$ and analytic formula of Gx be unknown.

Define $N_j (j = 1, 2, \ldots, n)$ fuzzy sets $A_j^1, A_j^2, \ldots, A_j^{N_j} \in [\alpha_j, \beta_j]$, which are normal, consistent, and complete with triangular MFs $\mu_{A_j^1}\left(x_j; a_j^1, b_j^1, c_j^1\right), \ldots, \mu_{A_j{}^{N_j}}\left(x_j; a_j^{N_j}, b_j{}^{N_j}, c_j{}^{N_j}\right)$, and $A_j^1 < A_j^2 < \cdots < A_j^{N_j}$ with $a_j^1 = b_j^1 = \alpha_j$ and $b_j^{N_j} = c_j^{N_j} = \beta_j$, which,

- $e_1^1 = \alpha_1, e_1^{N_1} = \beta_1$, and $e_1^j = b_1^j$ for $j = 2, 3, \ldots, N_1 - 1$,
- $e_2^1 = \alpha_2, e_2^{N_2} = \beta_2$, and $e_1^j = b_2^j$ for $j = 2, 3, \ldots, N_2 - 1$
- $e_n^1 = \alpha_n, e_n^{N_n} = \beta_n$, and $e_1^j = b_1^{'}$ for $j = 2, 3, \ldots, N_n - 1$.
- Construct $I = N_1 \times N_2 \times \cdots \times N_n$ fuzzy if–then rules in following form:
- $R_X^{j_1 - j_n}$: IF x_1 is $A_1^{j_1}$ and x_2 is $A_2^{j_2}$ and … and x_n is $A_n^{j_n}$ Then y is $B^{j_1 - j_n}$, where $j_1 = 1, 2, \ldots, N_1, j_2 = 1, 2, \ldots, N_2, \ldots, j_n = 1, 2, \ldots, N_n$, and center of the fuzzy set $B^{j_1 \cdots /_n}$, denoted by $\overleftarrow{y}^{'}{}^{j_1 \cdots /_n}$, is chosen as Eq. (16):
- $\overleftarrow{y}^{j_1 \cdots /_n} = G\left(e_1^{j_1}, \ldots, e_n^{j_n}\right)$

$$\vartheta_l = \tau\left(\mu_{A_1^{1-j-in,i}}(x_1), \mu_{A_2^{1-jn,i}}(x_2), \ldots, \mu_{A_n^{j1-jn,i}}(x_n)\right) \tag{16}$$

Therefore, from $\mu_{\overline{B^4}}(y) = t\left(\vartheta_i, \mu_{B^i}(y)\right), \forall y \in R$, fuzzy inference produces fuzzy set of output by: $\mu_{\bar{B}/1-jn,A}(y) = t\left(\vartheta_i, \mu_{B^{i1-jn,i}}(y)\right) \forall y \in R.\ \mu_{\overleftarrow{B}/1-/n}(y) = s\left(\mu_{B/1-1n,1}(y), \mu_{B/1-jn,2}(y), \ldots, \mu_{B/1-jn,}(y)\right)$. $f_i(x_1, x_2, \ldots, x_n) = a_1^{\dagger} x_1 + a_2^{\natural} x_2 + \cdots + a_n^i x_n + a_{n+1}^l$, where a_j^i are parameters, and are evaluated by LSM.

$$\left(\mu_{Q_{IM}}(x, y) = \min\left[\mu_{A_1}(x), \mu_{A_2}(y)\right], Q_{IM} \in X \times Y\right)$$

$$\mu_{B'}(y) = \max_{\forall i}\left[\sup_{x \in X} \min\left(\mu_{A'}(x), \mu_{A_1'}(x_1), \ldots, \mu_{A_n'}(x_n), \mu_{B^i}(y)\right)\right]$$

$$\mu_{A'}(x) = \begin{cases} 1 & \text{if } x = x^* \\ 0 & \text{otherwise} \end{cases} \quad y^* = \frac{\sum_{i=1}^{l} \overleftarrow{y}^{i} w_l}{\sum_{i=1}^{l} w_l} \tag{17}$$

Since the fuzzy sets $A_j^1, \ldots, A_j^{N_j}$ are complete at every $x \in X$, then there exist $j_1, j_2, \ldots, j_n$ such that: $\min\left(\mu_{A_1{}'}(x_1), \mu_{A_2'^2}(x_2), \ldots, \mu_{A_n{}^{n}}(x_n)\right) \neq 0$. Let $f(x)$ be fuzzy system in (13) and $G(x)$ be unknown function in (18). If $G(x)$ is continuously differentiable on $X = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times \cdots \times [\alpha_n, \beta_n]$, then:

$$||G - f||_{\infty} \leq \left\|\frac{\partial G}{\partial x_1}\right\|_{\infty} h_1 + \left\|\frac{\partial G}{\partial x_2}\right\|_{\infty} h_2 + \cdots + \left\|\frac{\partial G}{\partial x_n}\right\|_{\infty} h_n. \tag{18}$$

where infinite norm $||.||_{\infty}$ is given as: $||d(x)||_{\infty} = \sup_{x \in X} |d(x)|$ and $h_j = \max_{1 \leq k \leq N_j} \left|e_j^{k+1} - e_j^k\right|, (j = 1, 2, \ldots, n$

. Let $X^{j1} \ldots /_n = \left[e_1^{j_1}, e_1^{j_1+1}\right] \times \left[e_2^{j_2}, e_2^{j_2+1}\right] \times \cdots \times \left[e_n^{j_n}, e_n^{j_n+1}\right]$, where $j_1 = 1, 2, \ldots, N_1 - 1, j_2 = 1, 2, \ldots, N_2 - 1, \ldots, \quad j_n = 1, 2, \ldots, N_n - 1$. Since $[\alpha_j, \beta_j] = \left[e_j^1, e_j^2\right] \cup \left[e_j^2, e_j^3\right] \cup \cdots \cup \left[e_j^{N_j - 1}, e_j^{N_j}\right], j = 1, 2, \ldots, n$. From Eq. (19):

$$f(x) = \frac{\sum_{k_1=j_1}^{j_1+1} \cdots \sum_{k_n=j_n}^{j_n+1} \overleftarrow{y}^{k_1 \ldots k_n} \left(\mathrm{m}\left(\mu_{A_1^{k_1}}(x_1), \mu_{A_2^{k_2}}(x_2), \ldots, \mu_{A_n^{k_n}}(x_n)\right)\right)}{\sum_{k_1=j_1}^{j_1+1} \cdots \sum_{k_n=j_n}^{j_n+1} \mathrm{m}\left(\mu_{A_1^{k_1}}(x_1), \mu_{A_2^{k_2}}(x_2), \ldots, \mu_{A_n^{k_n}}(x_n)\right)} \tag{19}$$

From (20), (21), (22), we obtain:

$$f(x) = \sum_{k_1=j_1}^{j_1+1} \cdots \sum_{k_n=j_n}^{j_n+1} \left[\frac{\mathrm{m}\left(\mu_{A_1^{k_1}}(x_1), \ldots, \mu_{A_n^{k_n}}(x_n)\right)}{\sum_{k_1=j_1}^{j_1+1} \cdots \sum_{k_n=j_n}^{j_n+1} \mathrm{m}\left(\mu_{A_1{}^{k_1}}(x_1), \ldots, \mu_{A_n^{k_n}}(x_n)\right)}\right] * G\left(e_1^{k_1}, \ldots, e_n^{k_n}\right) \tag{20}$$

$$\sum_{k_1=j_1}^{j_1+1} \cdots \sum_{k_n=j_n}^{j_n+1} \left[\frac{\mathrm{m}\left(\mu_{A_1^{k_1}}(x_1), \ldots, \mu_{A_n^{k_n}}(x_n)\right)}{\sum_{k_1=j_1}^{j_1+1} \cdots \sum_{k_n=j_n}^{j_n+1} \mathrm{m}\left(\mu_{A_1^{k_1}}(x_1), \ldots, \mu_{A_n^{k_n}}(x_n)\right)}\right] = 1 \tag{21}$$

$$|G(x) - fx| \leq \sum_{k_1=j_1}^{j_1+1} \cdots \sum_{k_n=j_n}^{j_n+1} \left[\frac{\mathrm{m}\left(\mu_{A_1^{k_1}}(x_1), \ldots, \mu_{A_n{}^{k_n}}(x_n)\right)}{\sum_{k_1=j_1}^{j_1+1} \cdots \sum_{k_n=j_n}^{j_n+1} \mathrm{m}\left(\mu_{A_1^{k_1}}(x_1), \ldots, \mu_{A_n{}^{k_n}}(x_n)\right)}\right] \max_{k_1=j_1+1} \left|G(x) - G\left(e_1^{k_1}, \ldots, e_n^{k_n}\right)\right| * \left|G(x) - G\left(e_1^{k_1}, \ldots, e_n^{k_n}\right)\right| \tag{22}$$

From the Mean Value $^{k_n = f_n :/_n + 1}$

From the Mean Value model is given (23) as:

$$|G(x) - f(x)| \leq \max_{k_1=1, j_1+1} \left(\left\|\frac{\partial G}{\partial x_1}\right\|_{\infty} \left|x_1 - e_1^{k_1}\right| + \left\|\frac{\partial G}{\partial x_2}\right\|_{\infty} \left|x_2 - e_2^{k_2}\right| + \cdots + \left\|\frac{\partial G}{\partial x_n}\right\|_{\infty} \left|x_n - e_n^{k_n}\right|\right) \tag{23}$$

Since $x \in X^{j_1 - j_n}$, means that $x_1 \in \left[e_1^{j1}, e_1^{j1+1}\right]$, $x_2 \in \left[e_2^{j2}, e_2^{j2+1}\right] \ldots x_n \in \left[e_n^{jn}, e_n^{jn+1}\right]$, have by Eq. (24),

$$\left|x_1 - e_1^{k_1}\right| \leq \left|e_1^{j_1+1} - e_1^{j_1}\right|, \left|x_2 - e_2^{k_2}\right| \leq \left|e_2^{j_2+1} - e_2^{j_2}\right| \ldots,$$
$$\text{and } \left|x_n - e_n^{k_n}\right| \leq \left|e_n^{j_n+1} - e_n^{j_n}\right| \text{ for } k_1 = j_1, j_1 + 1, k_2 = j_2, j_2 + 1, \ldots, \text{and } k_n = j_n, j_n + 1 \tag{24}$$

Then, (25) becomes:

$$|G(x) - f(x)| \leq \left\|\frac{\partial G}{\partial x_1}\right\|_{\infty} \left|e_1^{j_1+1} - e_1^{j_1}\right| + \left\|\frac{\partial G}{\partial x_2}\right\|_{\infty} \left|e_2^{j_2+1} - e_2^{j_2}\right| + \cdots + \left\|\frac{\partial G}{\partial x_n}\right\|_{\infty} \left|e_n^{j_n+1} - e_n^{j_n}\right|$$
$$\text{Since} ||d(x)||_{\infty} = \sup_{x \in X} |d(x)| \text{then} ||G - f||_{\infty} = \sup_{x \in X} |G - f|, \text{we get:}$$
$$||G - f||_{\infty} \leq \left\|\frac{\partial G}{\partial x_1}\right\|_{\infty} \sum \max_{1 \leq 1 \leq N_1 - 1} \left|e_1^{j_1+1} - e_1^{j_1}\right| + \cdots + \left\|\frac{\partial G}{\partial x_n}\right\|_{\infty} 1 \leq \max_{n \leq \mathcal{I}_n - 1} \left|e_n^{j_n+1} - e_n^{j_n}\right|$$
$$\therefore ||G - f||_{\infty} \leq \left\|\frac{\partial G}{\partial x_1}\right\|_{\infty} h_1 + \left\|\frac{\partial G}{\partial x_2}\right\|_{\infty} h_2 + \cdots + \left\|\frac{\partial G}{\partial x_n}\right\|_{\infty} h_n \tag{25}$$

From (26), conclude that fuzzy systems in form. $\left\|\frac{\partial G}{\partial x_1}\right\|_\infty, \left\|\frac{\partial G}{\partial x_2}\right\|_\infty, \ldots, \left\|\frac{\partial G}{\partial x_n}\right\|_\infty$ are finite numbers for any given $\varepsilon > 0$, select $h_1, h_2, \ldots, h_n$ small enough such that $\left\|\frac{\partial G}{\partial x_1}\right\|_\infty h_1 + \left\|\frac{\partial G}{\partial x_2}\right\|_\infty h_2 + \cdots + \left\|\frac{\partial G}{\partial x_n}\right\|_\infty h_n < \varepsilon$. Hence from (27):

$$\sup_{x\in X}|G-f| = ||G-f||_\infty < \varepsilon \tag{27}$$

We can see from (28) that we need to know the boundaries of the derivatives of G(x) about $x_1, x_2, \ldots, x_n$ to represent a fuzzy system with a pre-specified accuracy.

$$\left\|\frac{\partial G}{\partial x_1}\right\|_\infty, \left\|\frac{\partial G}{\partial x_2}\right\|_\infty, \ldots, \left\|\frac{\partial G}{\partial x_n}\right\|_\infty \tag{28}$$

Select a fuzzy method with a MIS, an SF, a CADand a Triangular MF, which then derive using Eq. (29).

$$\begin{aligned} f(x) &= \frac{\sum_{i=1}^{l} \overleftarrow{y}^{i}\left(\min_{\forall j}\mu_{A_j}(x_j)\right)}{\sum_{i=1}^{l}\left(min_{\forall j}\mu_{A_j'}(x_j)\right)} \\ &= \frac{\sum_{i=1}^{l} \overleftarrow{y}^{i}\left[\min_{\forall j}\left(\max\left(\min_{vj}\left(\frac{x_j-a_j^i}{b_j^i-a_j^i}, \frac{c_j^i-x_j}{c_j^i-b_j^i}\right), 0\right)\right)\right]}{\sum_{i=1}^{l}\left[\min_{vj}\left(\max\left(\min_{\forall j}\left(\frac{x_j-a_j^i}{b_j^i-a_j^i}, \frac{c_j^i-x_j}{c_j^i-b_j^i}\right), 0\right)\right)\right]} \end{aligned} \tag{29}$$

The more rules you have, the more parameters you will have and the more computation you will have to do, but you will get better accuracy. When initial parameters *yi* (0), *aji* (0), *bji* (0), *cji* (0) are specified, the fuzzy system becomes by Eq. (30).

$$f(x) = \frac{\sum_{j_1=1}^{N_1}\cdots\sum_{j_n}^{N_n} \overleftarrow{y}^{j_1-j_n}(0)\left[\mathrm{m}\left(\mathrm{m}\left(\min_{\forall k}\left(\frac{x_{k0}^p-a_k^{j_1-j_n}(0)}{b_k^{j_1j_2j_3}(0)-a_k^{j_1j_2j_3}(0)'}, \frac{c_k^{j_1-j_n}(0)-x_{k0}^p}{c_k^{j_1-j_n}(0)-b_k^{j_1-j_n}(0)}\right), 0\right)\right)\right]}{\sum_{j_1=1}^{N_1}\cdots\sum_{j_n}^{N_n}\left[\mathrm{m}\left(\mathrm{m}\left(\min_{\forall k}\left(\frac{x_{k0}^p-a_k^{j_1-j_n}(0)}{b_k^{j_1j_n}(0)-a_k^{j_1-j_n}(0)'}, \frac{c_k^{j_1-j_n}(0)-x_{k0}^p}{c_k^{j_1-j_n}(0)-b_k^{j_1j_n}(0)}\right), 0\right)\right)\right]} \tag{30}$$

for a sigmoid activation function, it gives by Eq. (31):

$$\begin{aligned} &h_l^{[\gamma](t)} = \frac{1}{1+exp-\left(\sum_{u=1}^{d'} w'_{l,u} x_{\varphi(u)}^{(t)} + b_l^{[\gamma]}(t)\right)} w'_{l,u} \\ &= \sum_{f=1}^{N_f}\sum_{p=1}^{P}\sum_{q=1}^{Q} K_{p,q}^f w_{l,u}^{[\gamma]} \\ &Y_u^{(t)} \equiv Y_{i,j}^{(t)} = \sum_{f=1}^{N_f}\sum_{p=1}^{P}\sum_{q=1}^{Q} K_{p,q}^f x_{\varphi(u)}^{(t)} h_l^{[\gamma](t)} \\ &1 = \sigma\left(\sum_{u=1}^{d'} w_{l,u}^{[\gamma]} Y_u^{(t)} + b_l^{[\gamma](t)}\right), l \\ &\in \{1,\ldots,s\} y_k^T = \Psi_k^T\left(h^{[\gamma](1)},\ldots,h^{[\gamma](T)}\right),\ k \in \{1,\ldots,r\} \\ &h_l^{[\rho]} = \sigma\left(\sum_{k=1}^{r} w_{l,k}^{[\rho]} y_k^T + b_l^{[\rho]}\right), l \in \left\{1,\ldots,r'\right\} \end{aligned} \tag{31}$$

Thus, if consider $\overleftarrow{X} = (0,\ldots,0)', b_l^{[\gamma](t)} = 0\forall t \in \{1,\ldots,T\}$, the Taylor series expansion of $h_l^{[\gamma](t)}$ is given by Eq. (32):

$$\begin{aligned} h_l^{[\gamma](t)} \approx& h_l^{[\gamma](t)}\left(\overleftarrow{X}\right) + \nabla h_l^{[\gamma](t)} X^{(t)} \\ &= \frac{1}{2} + \sum_{u=1}^{d'} \frac{\partial h_l^{[\gamma](t)}\left(\overleftarrow{X}\right)}{\partial x_{\varphi(u)}^{(t)}} x_{\varphi(u)}^{(t)} = \frac{1}{2} + \sum_{u=1}^{d'} w'_{l,u} x_{\varphi(u)}^{(t)} \end{aligned} \tag{32}$$

with w′ l u, given by Eq. (33), and $X^{(t)} = \left(x_{\varphi(1)}^{(t)},\ldots,x_{\varphi(d,)}^{(t)}\right)'$ being a column vector of the input at time t. Let $H = \left(h^{[\gamma](1)},\ldots,h^{[\gamma](T)}\right)'$, for $X = X$ then:

$$\overleftarrow{H} = H\left(\overleftarrow{X}\right) = \left\{\left\{\frac{1}{2}\right\}^s,\ldots,\left\{\frac{1}{2}\right\}^s\right\}^T \tag{33}$$

where s is number of hidden neurons. Taylor series expansion of Ψk T isgiven by Eq. (34):

$$\begin{aligned} y_k^T =& \Psi_k^T(H) \approx \Psi_k^T\left(\overleftarrow{H}\right)\left(H-\overleftarrow{H}\right) = \Psi_k^T\left(\overleftarrow{H}\right) \\ &+ \sum_{t=1}^{T}\sum_{u=1}^{s} \frac{\partial \Psi_K^T\left(\overleftarrow{H}\right)}{\partial h_u^{[\gamma(k)}}\left(h_u^{[\gamma](t)} - \frac{1}{2}\right) \end{aligned} \tag{34}$$

By replacing $h_u^{[\gamma](t)}$ of Eq. (35) with $h_l^{[\gamma](t)}$ of Eq. (36):

$$\Psi_k^T(H) \approx \Psi_k^T\left(\overleftarrow{H}\right) + \frac{1}{4}\sum_{t=1}^{T}\sum_{u=1}^{s}\sum_{v=1}^{d'} \frac{\partial \Psi_k^T\left(\overleftarrow{H}\right)}{\partial h_u^{[\gamma]^{(t)}}} w'_{l,v} x_{\varphi(v)}^{(t)} \tag{35}$$

Finally, by substituting

$$
\begin{aligned}
h_l^{[\rho]} &= \sigma\left(\sum_{k=1}^{r} w_{l,k}^{[e]}\left[\Psi_k^T\left(\overleftarrow{H}\right) + \frac{1}{4}\sum_{t=1}^{T}\sum_{u=1}^{s}\sum_{v=1}^{d'}\frac{\partial\Psi_k^T\left(\overleftarrow{H}\right)}{\partial h_u^{[\gamma]}(t)} w'_{l,v} x_{\varphi(v)}^{(t)}\right] + b_l^{[\rho]}\right)\\
h_l^{[\rho]} &= \sigma\left(\sum_{k=1}^{r} w_{l,k}^{[\rho]}\Psi_k^T\left(\overleftarrow{H}\right) + \frac{1}{4}\sum_{k=1}^{r}\left(\sum_{v=1}^{d'}\underbrace{\sum_{u=1}^{s}\frac{\partial\Psi_K^T(H)}{\partial h_u^{(Y)(1)}} w'_{l,v} w_{l,k}^{[\rho]} x_{\varphi(v)}^{(1)}}_{w_{l,v}^{(1)}} + \cdots\right.\right.\\
&\quad \left.\left. + \sum_{v=1}^{d'}\underbrace{\sum_{u=1}^{s}\frac{\partial\Psi\Psi_(\left(\overleftarrow{H}\right)}{\partial h_u^{[](T)}} w'_{l,v} w_{l,k}^{[\rho]} x_{\varphi(v)}^{(T)}}_{w_{l,v}^{(T)}} + b_l^{[\rho]}\right)
\end{aligned}
\tag{36}
$$

$$
w_{l,v}^{''(t)} = \sum_{u=1}^{s}\sum_{f=1}^{N_f}\sum_{p=1}^{P}\sum_{q=1}^{Q}\frac{\partial\Psi_k^T\left(\overleftarrow{X}\right)}{\partial h_u^{[\gamma](t)}} K_{p,q}^{f} w_{l,v}^{[\gamma]} w_{l,k}^{[\rho]} \tag{37}
$$

derived features $w_{l,v}^{''(t)}$ through summations on indexes f, p, and q combine features $\left(\left(w_{l,v}^{[\gamma]}\right.\right.$ and $\left.\left.w_{l,k}^{[\rho]}\right)\right)$ extracted from both Fuzzy based SAEs and gives compact representation of input over time.

3.2 Least square error back propagation neural network (LSEBPNN)

Let, training set in a C-class issue contains vector pairs $\left\{\left(x_1, y_1\right), \left(\boldsymbol{x}_2, \boldsymbol{y}_2\right), \ldots, \left(\boldsymbol{x}_P, \boldsymbol{y}_P\right)\right\}$ where $\boldsymbol{x}_p \in \mathbb{R}^N$ refers to pth input pattern and $\boldsymbol{y}_p \in \left\{\boldsymbol{t}_c\right|, c = 1, 2, \ldots, C; \boldsymbol{t}_c \in \mathbb{R}^c\}$ refers to target output of c network corresponding to this input.

All weights and bias terms are included in LSEBPNN's adaptive parameters. The training phase's main aim is to establish the best weights and bias terms for minimizing difference between network output as well as target output. The difference is referred regarded as the network's training error. MSE for pth input pattern in the traditional BP technique is $E_p = \frac{1}{2}\sum_{k=1}^{C}\left(t_{pk} - o_{pk}^{\circ}\right)^2$. It shows that an input pattern's target value could be several. To put it another way, any input pattern can have any target value with any membership value. To put it another way, the training problem can be thought of as a fuzzy constraint fulfillment problem.Suggested network modifies parameters throughout training phase to ensure that these limitations are overcome as efficiently as possible. The constraints for pth input pattern are stated mathematically as fuzzy MSE term, which is given by Eq. (38)

$$
\dot{}^{\mathrm{f}} = \frac{1}{2}\sum_{k=1}^{C}\sum_{c=1}^{C}\mu_c^q\left(\boldsymbol{x}_p\right)\left(t_{ck} - o_{pk}^{\mathrm{o}}\right)^2 \tag{38}
$$

The learning laws for networks are derived using same approach as traditional BP technique. Suppose that the weight update, Dw, happens after each input pattern has been presented. Assuming that all weight changes in network are made with same learning-rate parameter h, weight changes applied to weights w and w are k j ji determined according to the gradient-descent rules by Eq. (39), (40):

$$
\Delta w_{kj}^{\mathrm{o}} = -\eta\frac{\partial E_p^{\mathrm{f}}}{\partial w_{kj}^{\mathrm{o}}} \text{ and } \Delta w_{ji}^{\mathrm{h}} = -\eta\frac{\partial E_p^{\mathrm{f}}}{\partial w_{ji}^{\mathrm{h}}} \tag{39}
$$

$$
\begin{aligned}
\Delta w_{kj}^{\mathrm{o}} &= \eta\left[\mu_k^q\left(\boldsymbol{x}_p\right) - \sum_{c=1}^{C}\mu_c^q\left(\boldsymbol{x}_p\right)o_{pk}^{\mathrm{o}}\right]\\
&\quad \times o_{pk}^{\mathrm{o}}\left(1 - o_{pk}^{\mathrm{o}}\right)o_{pj}^{\mathrm{h}}\\
&= \eta\delta_{pk}^{\mathrm{o}}o_{pj}^{\mathrm{h}}
\end{aligned}
\tag{40}
$$

where by Eq. (41)

$$
\delta_{pk}^{o} = \left[\mu_k^q\left(\boldsymbol{x}_p\right) - \sum_{c=1}^{C}\mu_c^q\left(\boldsymbol{x}_p\right)o_{pk}^{\mathrm{o}}\right]o_{pk}^{\mathrm{o}}\left(1 - o_{pk}^{\mathrm{o}}\right) \tag{41}
$$

Again, from Eq. (42),

$$
\begin{aligned}
\Delta w_{ji}^{\mathrm{h}} &= \eta f_j^{\mathrm{h}}\left(\mathrm{net}_{pj}^{\mathrm{h}}\right)x_{pi}\sum_{k=1}^{C}\\
&\quad \left[\mu_k^q\left(\left(\boldsymbol{x}_p\right) - \sum_{c=1}^{C}\mu_c^q\left(\left(\boldsymbol{x}_p\right)o_{pk}^{\mathrm{o}}\right]o_{pk}^{\mathrm{o}}\left(1 - o_{pk}^{\mathrm{o}}\right)w_{kj}^{\mathrm{o}}\right.\right.\\
&= \eta f_j^{\mathrm{h}}\left(\mathrm{net}_{pj}^{\mathrm{h}}\right)x_{pi}\sum_{k=1}^{C}\delta_{pk}^{\mathrm{o}}w_{kj}^{\mathrm{o}} = \eta\delta_{pj}^{\mathrm{h}}x_{pi},
\end{aligned}
\tag{42}
$$

where by Eq. (43)

$$\delta_{pj}^{\mathrm{h}} = f_j^{\mathrm{h}}\left(\mathrm{net}_{pj}^{\mathrm{h}}\right)\sum_{k=1}^{C}\delta_{pk}^{\mathrm{o}} w_{kj}^{\mathrm{o}}. \tag{43}$$

In many circumstances, the traditional BP technique may not converge quickly, when classes overlap. Because ambiguous vectors are assigned full weightage in one class, this is case. In suggested version, error to be back propagated is given more weight in the case of nodes with higher membership values.

The learning algorithm's purpose is to reduce the squared error cost function, which is given by Eq. (44)

$$j_i^{(s)} = \frac{1}{2}\sum_{q=1}^{m}\left(d_{i,q}^{(s)} - v_i(s)\right)^2 \tag{44}$$

Equation (45), where m is total number of vectors in training data set given by

$$j_i^{(s)} = \frac{1}{2}\sum_{q=1}^{m}\left(d_{i,q}^{(s)} - w_i(s)t.x_{out.q}^{(s-1)}\right)^2 \tag{45}$$

Partial derivative about w i (s) and equate it to zero to determine weight vector that minimizes cost function given by Eq. (46).

$$\frac{\partial j_i^{(s)}}{\partial w_i(s)} = \sum_{q=1}^{m}\left(-d_{i,q}^{(s)}x_{\mathrm{out}.q^{(s-1)}} + x_{\mathrm{out}.q^{(s-1)}}x_{\mathrm{out}.q^{(s-1)}}t_i(s)\right) = 0 \tag{46}$$

$$c_i^{(s)} = \sum_{q=1}^{m}x_{\mathrm{out}.q^{(s-1)}}x_{\mathrm{out}.q^{(s-1)}}tp_i^{(s)} = d_{i,q}^{(s)}x_{\mathrm{out.q}^{(s-1)}} \tag{47}$$

In vector matrix form, Eq. (48) are rearranged as

$$c_i^{(s)}w_i^{(s)} = p_i^{(s)} \tag{48}$$

$w_i^{(s)}$ is weight vector to ith linear combiner in sth layer, Eq. (49) is given as deterministic normal equation

$$w_{i^{(s)}} = \left[c^{(s)}\right]^{-1}p_i^{(s)} \tag{49}$$

By equating partial derivative of performance index $\mathbf{w}_i^{(k)}(n)$ and setting it equal to zero, the performance index is minimised (50)

$$\frac{\partial J(n)}{\partial \mathbf{w}_i^{(k)}(n)} = \&2\sum_{t=1}^{n}\lambda^{n-t}\times\sum_{j=1}^{N_L}\left[\epsilon_{j,R}^{(L)}(t)\frac{\partial\epsilon_{j,R}^{(L)}(t)}{\partial\mathbf{w}_i^{(k)}(n)} + \epsilon_{j,I}^{(L)}(t)\frac{\partial\epsilon_{j,I}^{(L)}(t)}{\partial\mathbf{w}_i^{(k)}(n)}\right]$$
$$= \& - 2\sum_{t=1}^{n}\lambda^{n-t}\times\sum_{j=1}^{N_L}\left[\zeta_{j,R}^{(L)}(t)\epsilon_{j,R}^{(L)}(t)\frac{\partial y_{j,R}^{(L)}(t)}{\partial\mathbf{w}_i^{(k)}(n)} + \zeta_{j,I}^{(L)}(t)\epsilon_{j,I}^{(L)}(t)\frac{\partial y_{j,I}^{(L)}(t)}{\partial\mathbf{w}_i^{(k)}(n)}\right] = 0 \tag{50}$$

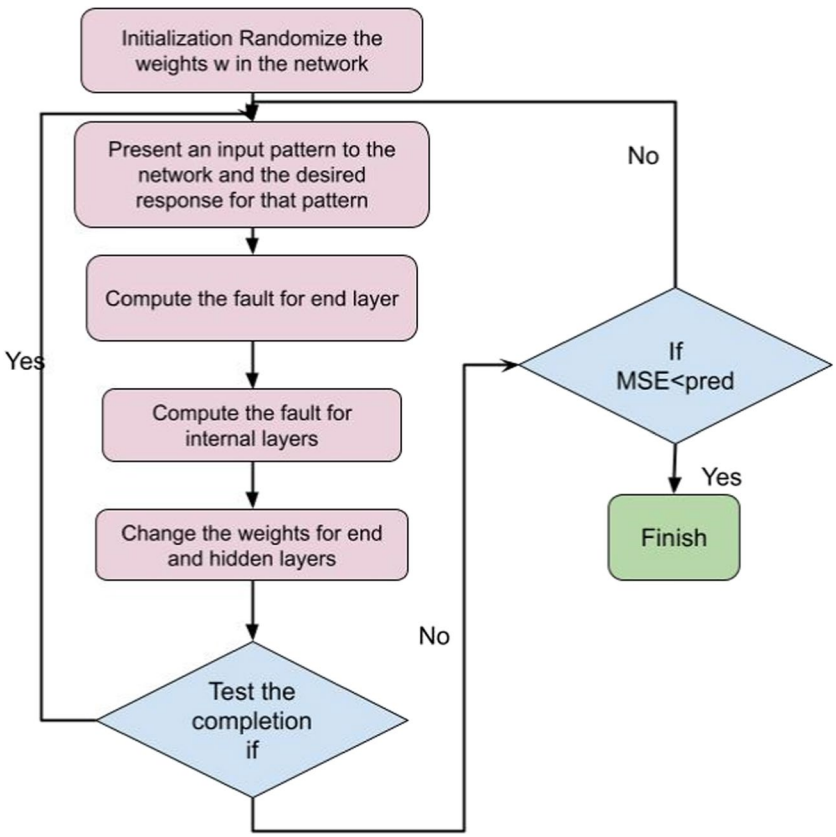

Fig. 2 The flow chart for LSEBPNN

Table 1 comparative analysis for various fault situation for proposed and existing technique

Virtual sensor-based datasets of automation industry	Techniques	Computational rate	QoS	RMSE	MAE	Prediction performance	Measurement accuracy
Spindle-based dataset	CNN	41	59	47	43	91	76
	RBF	36	61	45	40	93	79
	FL_SDDAE-LSEBPNN	34	64	41	35	94	85
Gear-based dataset	CNN	50	62	51	51	73	79
	RBF	46	63	48	45	76	81
	FL_SDDAE-LSEBPNN	43	67	43	39	79	85
Bearing-based dataset	CNN	59	63	53	49	79	73
	RBF	53	65	49	45	83	77
	FL_SDDAE-LSEBPNN	49	68	45	41	86	84

Equation (51), (52) is set to the following

$$\begin{aligned}&\sum_{t=1}^{n}\lambda^{n-t}\left[\left\{\psi_{i,R}^{(k)}(t)-y_{i,R}^{(k)}(t)\right\}\zeta_{i,R}^{(k)}(t)f'\left(s_{i,R}^{(k)}(t)\right)\right.\\&\left.+\&\left\{\psi_{i,I}^{(k)}(t)-y_{i,I}^{(k)}(t)\right\}\zeta_{i,I}^{(k)}(t)f'\left(s_{i,I}^{(k)}(t)\right)\right]\times \mathrm{x}^{(k)*}(t)=0.\end{aligned}\tag{51}$$

$$\mathbf{r}_i^{(k)}(n)=\mathbf{R}_i^{(k)}(n)\mathbf{w}_i^{(k)}(n)\tag{52}$$

where by Eq. (53)

$$\begin{aligned}\mathbf{r}_i^{(k)}(n)=&\sum_{t=1}^{n}\lambda^{n-t}\\&\times\left[\zeta_{i,R}^{(k)}(t)\psi_{i,R}^{(k)}(t)f'\left(s_{i,R}^{(k)}(n)\right)\right.\\&\left.+\jmath\zeta_{i,I}^{(k)}(t)\psi_{i,I}^{(k)}(t)f'\left(s_{i,I}^{(k)}(n)\right)\right]\\&\times\mathbf{x}^{(k)*}(t)\\\mathbf{R}_i^{(k)}(n)=&\sum_{t=1}^{n}\lambda^{n-t}\mathbf{x}^{(k)*}(t)\\&\times\left[\zeta_{i,R}^{(k)}(t)y_{i,R}^{(k)}(t)f'\left(s_{i,R}^{(k)}(t)\right)\right.\\&\left.+\jmath\zeta_{i,I}^{(k)}(t)y_{i,I}^{(k)}(t)f'\left(s_{i,I}^{(k)}(t)\right)\right]\\&\times s_i^{(k)^{-1}}(t)\mathbf{x}^{(k)^T}(t).\end{aligned}\tag{53}$$

Now, define a matrix operation for simplicity $A\odot B\doteq A_RB_R+\jmath A_IB_I$. The flow chart for LSEBPNN is represented in Fig. 2.

4 Performance analysis

Proposed method is implemented into a prototype software system utilizing Python 3.7 to evaluate and assess potential contribution of proposed strategy for future real-world applications. Resources utilized to combine proposed method were an Intel i7 processor (Intel(R) Core(TM) i7-3770 CPU @3.40 GHz 3.80 Ghz) and an eight (8) gigabyte RAM (Intel, Santa Clara, CA, USA) (Samsung, Seoul, Korea). Microsoft Windows 10 was the operating system on which the suggested system was hosted and tested.

Table 1 shows comparative analysis for various fault situations for proposed and existing techniques. Here the fault situation has been detected by virtual sensor-based datasets of automation industry. The parametric analysis has been carried out in terms of QoS, measurement accuracy, RMSE, MAE, prediction performance, and computational time.

Figures 3, 4, and 5 show comparative analysis for various virtual sensor-based datasets from automation industry. The dataset collected from cloud is based on spindle fault detection-based data, gear fault detection-based data, and bearing fault detection-based data. For spindle fault detection data, the proposed technique obtained computational time of 34%, QoS of 64%, RMSE of 41%, MAE of 35%, prediction performance of 94%, measurement accuracy of 85%.The proposed technique obtained computational time of 43%, QoS of 67%, RMSE of 43%, MAE of 39%, prediction performance of 79%, and measurement accuracy of 85% by gear-based fault detection dataset. For bearing fault detection data, the proposed technique obtained computational time of 49%, QoS of 68%, RMSE of 45%, MAE of 41%, prediction performance of 86%, and measurement accuracy of 84%. From the above analysis, proposed technique obtained optimal results for all the fault detection based on automation industry data.

The fundamental challenge in dealing with soft sensor principles is a lack of understanding due to their novelty and, as a result, a lack of typical mathematical descriptions or structure. On the other hand, it allows for more creative expression. In general, vast arrays of statistics for calculations are required when working with soft sensors. It is vital to have a thorough understanding of the controlled process's principles, physical characteristics, and the parameters' relationships.

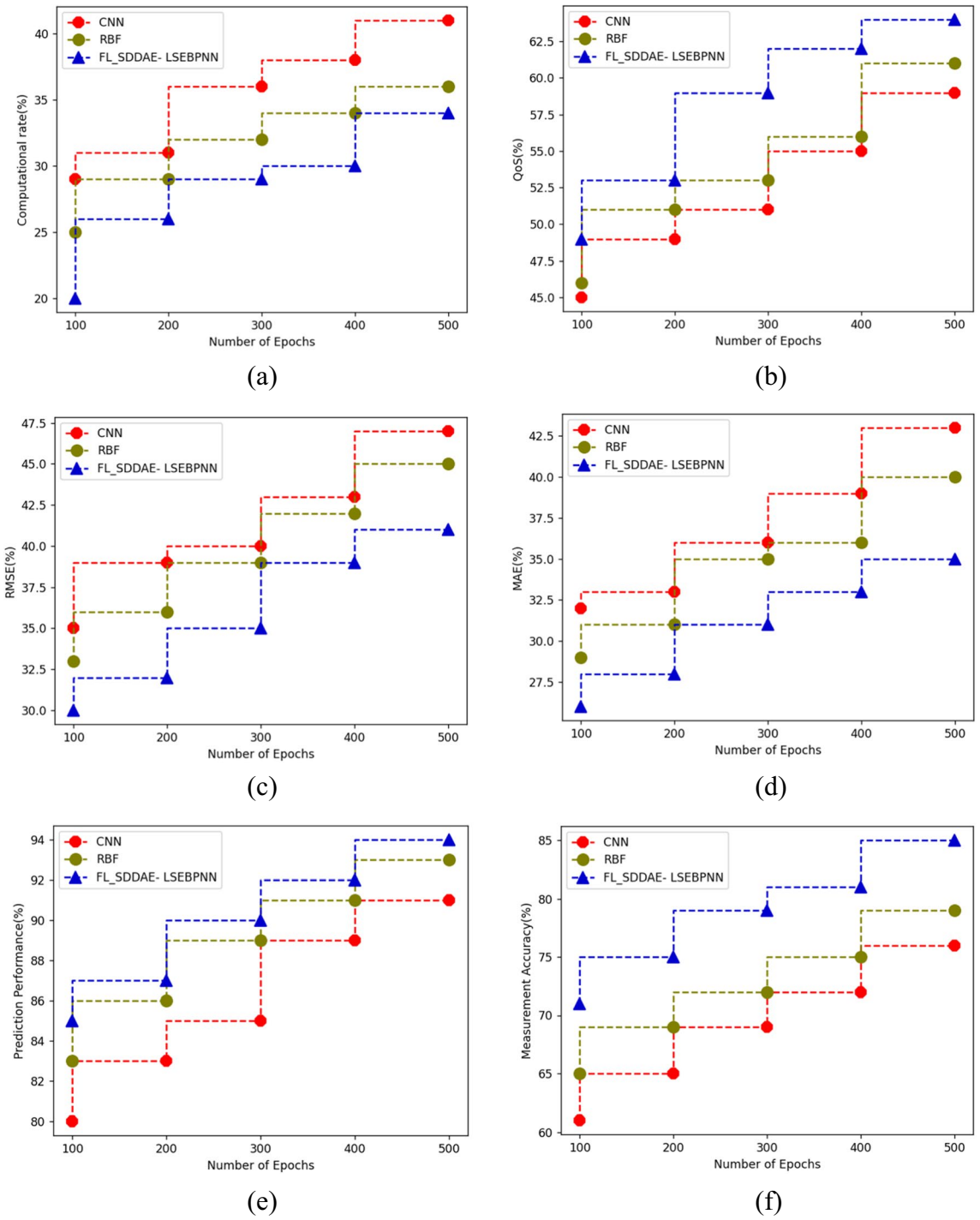

Fig. 3 Comparative analysis of spindle-based dataset in terms of **a** computational time, **b** QoS, **c** RMSE, **d** MAE, **e** prediction performance, **f** measurement accuracy

5 Conclusion

This research propose novel technique in virtual soft sensor-based fault detection in automation industry using deep learning technique integrated with cloud module. Here the aim is to design novel techniques in automation of manufacturing industry where the dynamic soft sensors are used in feature representation and classification of the data. The data has been collected from cloud storage and created the virtual sensors dataset based on gear fault detection, spindle fault detection, and bearing fault detection in automation industry. Then to represent the feature using fuzzy

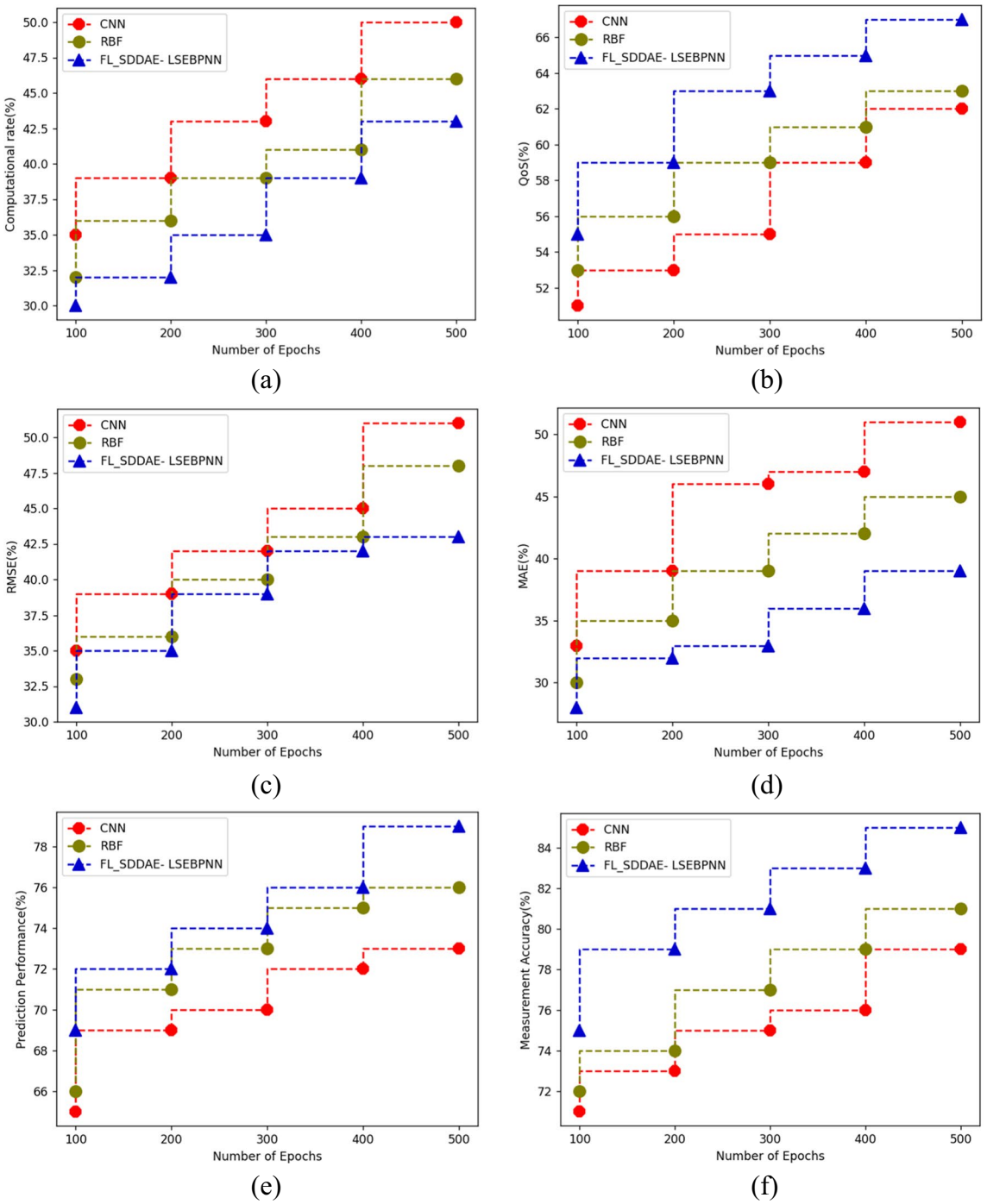

Fig. 4 Comparative analysis of gear-based dataset in terms of **a** computational time, **b** QoS, **c** RMSE, **d** MAE, **e** prediction performance, **f** measurement accuracy

logic-based stacked data-driven auto- encoder (FL_SDDAE) where the features of input data have been identified with general automation problems. Then the features have been classified using least square error back propagation neural network (LSEBPNN) in which the mean square error while classification will be minimized with loss function of the data. Here the experimental results have been carried out in terms of computational time of 34%, QoS of 64%, RMSE of 41%, MAE of 35%, prediction performance of 94%, and measurement accuracy of 85% has been obtained

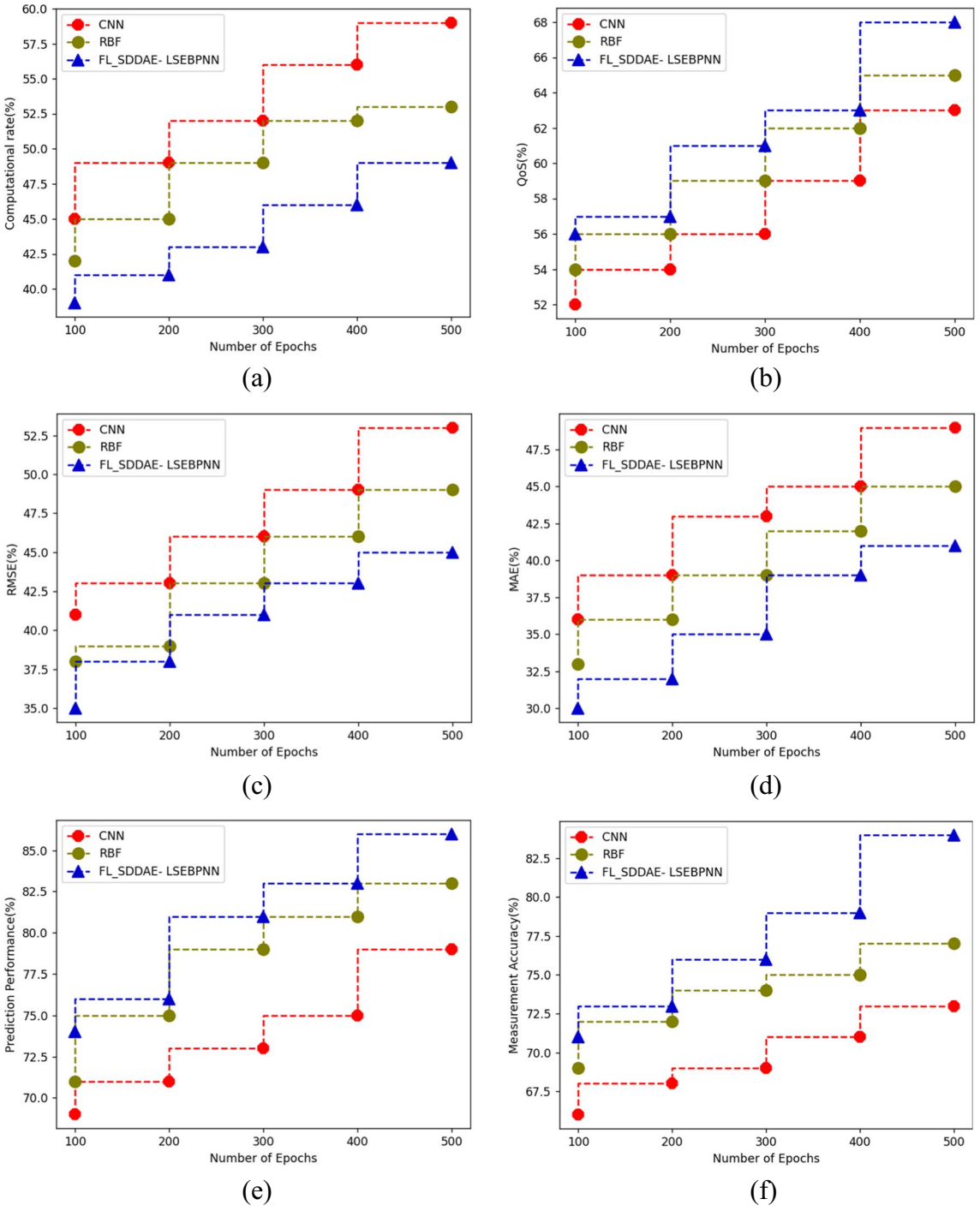

Fig. 5 Comparative analysis of bearing-based dataset in terms of **a** computational time, **b** QoS, **c** RMSE, **d** MAE, **e** prediction performance, **f** measurement accuracy

by proposed technique. One is that nonlinear systems' predictive control cannot be solved successfully. Another issue is that stability as well as resilience of multivariable predictive control algorithms must be addressed, and accurate principle models for complex systems are extremely difficult to construct. Despite the contributions made so far, there are still areas where future work might be improved. On the loss function, targeted-output regularizes would extract even better features, improving the suggested work. Another future intervention would be to use approaches on the unsupervised pre-training to identify dynamic-related aspects. In addition, industrial research scenarios were used

to apply the proposed method, however developing a soft sensor proposal for a real-world industrial scenario could be challenging. Non-linearities, abnormalities, and highly complex ecosystems must all be taken into account. The industrial study cases have shown to be suitable and widely used in the implementation and evaluation of models, and they serve as the foundation for many contributions in this field of research.

Funding The authors extend their appreciation to the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) for funding and supporting this work through Research Partnership Program no. RP-21-07-06. The authors acknowledge the support from Princess Nourah Bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R440), Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia.

Declarations

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

Conflict of interest The authors declare no competing interests.

References

1. Savytskyi O, Tymoshenko M, Hramm O, Romanov S (2020) Application of soft sensors in the automated process control of different industries. In E3S Web of Conferences (Vol. 166, p 05003). EDP Sciences. https://doi.org/10.1051/e3sconf/202016605003
2. Sun Q, Ge Z (2021) A survey on deep learning for data-driven soft sensors. IEEE Transactions on Industrial Informatics 17(9):5853–5866
3. Andronie M, Lăzăroiu G, Iatagan M, Uță C, Ștefănescu R, CocoȘatu M (2021) Artificial intelligence-based decision-making algorithms, internet of things sensing networks, and deep learning-assisted smart process management in cyber-physical production systems. Electronics 10(20):2497
4. Kovacova M, Lăzăroiu G (2021) Sustainable organizational performance, cyber-physical production networks, and deep learning-assisted smart process planning in Industry 4.0-based manufacturing systems. Econ Manag Fin Markets 16(3):41–54
5. Bibi R, Saeed Y, Zeb A, Ghazal TM, Rahman T, Said RA, ... Khan MA (2021) Edge AI-based automated detection and classification of road anomalies in VANET using deep learning. Comput Intell Neurosci 2021:1–16
6. Valaskova K, Ward P, Svabova L (2021) Deep learning-assisted smart process planning, cognitive automation, and industrial big data analytics in sustainable cyber-physical production systems. J Self-Governance Manag Econ 9(2):9–20
7. Hndoosh RW, Kumar S, Saroa MS (2014) Fuzzy mathematical models for the analysis of fuzzy systems with application to liver disorders. IOSR J Comput Eng 16(5):71–85
8. Kingma DP, Welling M (2019) An introduction to variational autoencoders. Foundations and Trends® in Machine Learning 12(4):307–392
9. D'Angelo G, Palmieri F (2021) A stacked autoencoder-based convolutional and recurrent deep neural network for detecting cyber-attacks in interconnected power control systems. Int J Intell Syst 36(12):7080–7102
10. Abrar S, Zerguine A, Bettayeb M (2002) Recursive least-squares backpropagation algorithm for stop-and-go decision-directed blind equalization. IEEE Trans Neural Netw 13(6):1472–1481
11. Bargellesi N, Beghi A, Rampazzo M, Susto GA (2021) AutoSS: a deep learning-based soft sensor for handling time-series input data. IEEE Robot Autom Lett 6(3):6100–6107
12. Thomopoulos SC (2021) Risk assessment and automated anomaly detection using a deep learning architecture. IntechOpen. https://doi.org/10.5772/intechopen.96209
13. Senthilkumar P, Rajesh K (2021) Design of a model based engineering deep learning scheduler in cloud computing environment using Industrial Internet of Things (IIOT). J Ambient Intell Humanized Comput 1–9
14. Yao L, Ge Z (2021) Dynamic features incorporated locally weighted deep learning model for soft sensor development. IEEE Trans Instrum Meas 70:1–11
15. Moreira de Lima JM, Ugulino de Araújo FM (2021) Industrial semi-supervised dynamic soft-sensor modeling approach based on deep relevant representation learning. Sensors 21(10):3430
16. Sahoo SR, Gupta BB (2021) Multiple features based approach for automatic fake news detection on social networks using deep learning. Appl Soft Comput 100:106983
17. Fazil M, Khan S, Albahlal BM, Alotaibi RM, Siddiqui T, Shah MA (2023) Attentional multi-channel convolution with bidirectional LSTM cell toward hate speech prediction. IEEE Access 11:16801–16811. https://doi.org/10.1109/ACCESS.2023.3246388
18. Zhang J, Gao RX (2021) Deep learning-driven data curation and model interpretation for smart manufacturing. Chin J Mech Eng 34(1):1–21
19. Xia K, Sacco C, Kirkpatrick M, Saidy C, Nguyen L, Kircaliali A, Harik R (2021) A digital twin to train deep reinforcement learning agent for smart manufacturing plants: environment, interfaces and intelligence. J Manuf Syst 58:210–230
20. Khan S et al (2022) HCovBi-caps: hate speech detection using convolutional and Bi-directional gated recurrent unit with Capsule network. IEEE Access 10:7881–7894
21. Khan S et al (2022) BiCHAT: BiLSTM with deep CNN and hierarchical attention for hate speech detection. J King Saud Univ-Comput Inform Sci 34(7):4335–4344
22. Maschler B, Ganssloser S, Hablizel A, Weyrich M (2021) Deep learning based soft sensors for industrial machinery. Procedia CIRP 99:662–667
23. Khan S, Saravanan V, Lakshmi TJ, Deb N, Othman NA (2022) Privacy protection of healthcare data over social networks using machine learning algorithms. Comput Intell Neurosci 2022(9985933):8
24. Haq AU, Li JP, Ahmad S, Khan S, Alshara MA, Alotaibi RM (2021) Diagnostic approach for accurate diagnosis of COVID-19 employing deep learning and transfer learning techniques through chest X-ray images clinical data in E-healthcare. Sensors 21(24):8219
25. Khan S, Al-Mogren AS, AlAjmi MF (2015) Using cloud computing to improve network operations and management. 2015 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW), Riyadh, Saudi Arabia, 2015, pp. 1–6. https://doi.org/10.1109/NSITNSW.2015.7176418