Multi-mode industrial soft sensor method based on mixture Laplace variational auto-encoder

Zhang, Tianming; Yan, Gaowei; Li, Rong; Xiao, Shuyi; Pang, Yusong

DOI
10.1016/j.measurement.2024.114435

Publication date
2024

Document Version
Final published version

Published in
Measurement: Journal of the International Measurement Confederation

Citation (APA)
Zhang, T., Yan, G., Li, R., Xiao, S., & Pang, Y. (2024). Multi-mode industrial soft sensor method based on mixture Laplace variational auto-encoder. *Measurement: Journal of the International Measurement Confederation*, *229*, Article 114435. https://doi.org/10.1016/j.measurement.2024.114435

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

Multi-mode industrial soft sensor method based on mixture Laplace variational auto-encoder

Tianming Zhang [a], Gaowei Yan [a,b,*], Rong Li [a], Shuyi Xiao [a], Yusong Pang [c]

[a] *College of Electrical and Power Engineering, Taiyuan University of Technology, Taiyuan, 030024, China*
[b] *Shanxi Research Institute of Huairou Laboratory, Taiyuan, 030032, China*
[c] *Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, Delft, 2628CD, Holland, Netherlands*

ARTICLE INFO

Keywords:
Soft sensor
Mixture Laplace
Variational auto-encoder
Multi-mode
Heavy tail

ABSTRACT

The industrially collected process data usually exhibit non-Gaussian and multi-mode characteristics. Due to sensor failures, irregular disturbances, and transmission problems, there are unavoidable outliers that make the data exhibit heavy-tailed characteristics. To this end, a variational auto-encoder regression method based on the mixture Laplacian distribution (MLVAER) is proposed, by introducing a type-II multivariate Laplacian distribution in the latent variable space for robust modeling, and further extending it to the mixture form to accommodate multi-mode processes, the corresponding reparameterization trick is finally proposed for the mixture form of this distribution for neural network gradient descent training. The model based on this distribution assumption has higher degrees of freedom than the model based on the traditional multivariate Laplace distribution assumption when the network structure is the same. Numerical simulation and experiments on two industrial examples demonstrate that the proposed algorithm reduces the root mean square error by over 15% compared to other algorithms.

1. Introduction

Some key variables in industrial processes are impossible to monitor directly with typical hardware sensors due to high measurement costs, physical constraints, and so on. In response to these challenges, soft sensor technology has advanced quickly in recent decades, allowing it to forecast key variables by mapping auxiliary variables for difficult-to-measure variables. Soft sensor technology based on mechanism modeling is costly to model because it requires prior knowledge of the processes and is considerably more difficult to model due to the process's complexity. As information technology improves, soft sensor technology has introduced data-driven methods for modeling, such as statistical machine learning or deep learning, that are more efficient, less expensive, and easier to deploy. Many traditional linear methods, for instance, Principal components regression (PCR) [1], Partial least square (PLS) [2–4] and non-linear methods such as Support vector regression (SVR) [5,6], Kernel methods [7], Artificial neural networks (ANN) [8] have been widely used in the field of soft sensors. Data-driven methods are also generally used in other industrial research fields, including fault diagnosis, defect detection, remaining useful life estimation, etc [9–11]. In this paper, we focus on the research progress of these methods in the field of soft sensors.

The above deterministic modeling method requires objects with deterministic quantitative relationships, while most process data are inevitably subject to noise interference and are essentially random variables. Random variables are inherently uncertain. In contrast, probabilistic models, which use methods from probability theory, stochastic processes, and mathematical statistics to model objects with contingent and random properties, are more suitable for characterizing the behavior of random variables in processes. For modeling data with process noise, Ge et al. [12] proposed a probabilistic PCR model (PPCR) based on PCR. Gustafsson [13] conducted probability derivation based on the traditional nonlinear iterative partial least squares algorithm. Li et al. [14] proposed a new probabilistic PLS (PPLS) model based on the PLS method, probabilistic principal component analysis (PPCA), and probability curve fitting ideas for quantitative analysis of Raman spectral data. However, some of the traditional PLS model's features are not well described in the above PPLS, so Zheng et al. [15] introduced two types of latent variables, the first controlling the relationship between the model's input and output variables and the second relating only to the input data, so only the first latent variable was used to explain the output data. Compared to the non-deep methods mentioned above, deep learning techniques have demonstrated powerful data modeling capabilities by extracting deeper abstract features from

[*] Corresponding author at: College of Electrical and Power Engineering, Taiyuan University of Technology, Taiyuan, 030024, China.
E-mail addresses: zhangtianming0408@link.tyut.edu.cn (T. Zhang), yangaowei@tyut.edu.cn (G. Yan).

https://doi.org/10.1016/j.measurement.2024.114435
Received 8 July 2023; Received in revised form 25 December 2023; Accepted 3 March 2024
Available online 5 March 2024

data through multi-layer nonlinear mapping, which is an irreplaceable advantage in the increasingly complex modern industrial processes [8]. As a deep probabilistic latent variable model, variational auto-encoder (VAE) blends deep learning and Bayesian variational inference, which not only has strong feature extraction ability but also can model process uncertainty and data noise [16]. It has a promising future in the field of soft sensors. Shen et al. [17] first applied VAE to soft sensor modeling in industrial processes and suggested a new nonlinear form of probabilistic latent variable model by stacking VAE to extract deeper nonlinear information. Chai et al. [18] integrated transfer learning into the VAE framework and proposed a deep probabilistic transfer regression to address the issue of the target domain's lack of labeling while utilizing the model's generation and reconstruction capabilities to handle cases with missing data.

Because of the requirements of varied product grades or operating conditions, most industrial processes have many modes of operation, resulting in data with multi-mode characteristics. Probabilistic models with a single Gaussian assumption cannot handle multi-mode data successfully. Gaussian mixture regression unifies operational pattern recognition and variable regression into a single model via finite Gaussian mixture, avoiding the creation and switching of numerous models, and is widely used in soft sensors [19–21]. The probability model for a single distribution can be easily extended to the case of mixed distributions. Ge et al. [12] extended PPCR to a mixture form and created a mixture PPCR (MPPCR) for quality prediction in multiple operating mode processes based on the proposed PPCR. Zheng et al. [15] developed a mixture form of the PPLS (MPPLS) to cope with more intricate process data information. Cui et al. [22] described the distribution of latent variables in VAE using a Gaussian mixture model and presented a mixture variational auto-encoder regression (MVAER) model, which was used for soft sensor modeling of complicated multi-mode industrial processes. Zhang et al. [23] further proposed a deep Gaussian mixture adaptive network (DGMAN) with multi-mode modeling, fast calibration, and distribution alignment capabilities to address process drift issues, bridging the gap between laboratory output and industrial practice.

The above probabilistic models generally assume that the noise obeys Gaussian distribution or Gaussian mixture distribution, but due to sensor failure, irregular interference, and transmission problems, the data inevitably have outliers and thus exhibit heavy-tailed characteristics. The presence of outliers can lead to skewed distributions and seriously affect the mean and variance of the data, so models based on the Gaussian distribution assumption can be poorly modeled due to the influence of outliers. In recent years, many studies have introduced heavy-tailed distributions such as Student's-t distribution and Laplace distribution for robust modeling [24,25] to overcome the effects of data outliers. Peel and McLachlan [26] proposed a more robust clustering method by modeling the data with a mixed Student's-t distribution. Zhu et al. [27] proposed a robust modeling strategy with a mixture PPCA, which can handle both outliers and missing data and applies to multi-mode data, also using multivariate Student's-t distribution to reduce the negative impact of outliers. Considering the supervised case, Wang et al. [28] proposed a robust soft sensor approach based on Variational Bayesian Student's-t mixture regression (VBSMR), which explicitly considered the dependence of quality variables on process variables and introduced the Student's-t distribution to handle outliers. Yan et al. [29] proposed a robust stochastic configuration network method based on Student's-t mixture distribution (SM-RSC), aiming to alleviate the impact of outliers or noise on data-driven modeling. However, the degree of freedom parameter in the Student's-t distribution requires a numerical optimization algorithm to solve the differential equation. For the Laplace distribution, a closed-form solution with unknown parameters can be used, making the computational procedure and expression simpler. Zhu et al. [30] constructed a robust principal component regression model (MRPPCR-L1) with multiple modes of operation using multivariate Laplace distribution. Yang et al. [31] developed a soft sensor algorithm based on a robust mixture probabilistic partial least squares model (RMPPLS). The multivariate Laplace distribution is used for robust modeling and the mixture form of the probabilistic partial least squares model is used for multi-mode description.

Currently, the literature primarily utilizes the type I multivariate Laplace distribution, as proposed by Eltoft et al. [32]. This distribution assumes that the mixture variables corresponding to each component of the random vector share a common factor, resulting in identical heavy tails for each marginal distribution. Consequently, even when the covariance matrix is diagonal, the components of the random vector remain correlated. This lack of flexibility may lead to redundancy when using this distribution as a prior for latent variables. To address these limitations, Zhang et al. [33] introduced a more flexible alternative called the type II multivariate Laplace distribution. In this distribution, the common factor of the type I distribution is replaced with multiple independent and identically distributed standard exponential random variables. As a result, the correlation between components of the random vector now depends solely on the covariance matrix's structure. Although this modification improves flexibility, the type II multivariate Laplace distribution still assumes a single-peaked distribution, making it unsuitable for modeling multi-mode data.

To overcome this limitation, this paper proposes a mixture type II multivariate Laplace distribution by introducing random variables that follow a multinomial distribution. Furthermore, this distribution is combined with a deep network to develop a variational auto-encoder regression method based on the mixture Laplace distribution.

Table 1 shows the differences between the recently published work and the proposed algorithm in terms of whether to consider uncertainty, outliers, multi-mode, nonlinearity, and multiple outputs. It can be seen that the proposed algorithm provides solutions for a wider range of industrial problems compared to other algorithms. It is important to note that the term "nonlinearity" in the table refers to the nonlinearity of the model in a single mode. The models GMR, MPPCR, MPPLS, VBSMR, MRPPCR-L1, and RMPPLS are essentially a combination of multiple linear models, with each mode corresponding to a linear model. On the other hand, MVAER, DGMAN, and MLVAER are a combination of multiple nonlinear models, and each mode corresponds to a nonlinear model. The main contributions of this paper are summarized below:

(1) A variational autoencoder regression method based on the mixture Laplace distribution is proposed. By introducing the mixture Laplace distribution, our method offers a flexible and robust framework for modeling multi-mode processes contaminated with outliers. This method breaks the limitations of single-peaked distributions and effectively mitigates the interference caused by outliers, resulting in improved algorithmic robustness and accuracy.

(2) A mixture type II multivariate Laplace distribution combined with a deep network is proposed. Due to its marginal distribution can have different heavy tails, the model based on this distribution assumption exhibits higher generality compared to models based on the traditional multivariate Laplace distribution assumption for the same network structure. This higher generality enables the model to handle more complex industrial scenarios. Additionally, this method offers an effective solution to enhance model compactness.

(3) A resampling strategy suitable for multivariate Laplace distribution is studied for gradient descent training of neural networks.

2. Variational auto-encoder

Consider the data set $\mathbf{X}$ composed of N independent samples of the same distribution, and the sample $\mathbf{x}$ in $\mathbf{X}$ is obtained from the sampling of continuous or discrete variables. Suppose there is an unobservable variable $\mathbf{z}$, which is generated by some prior distribution $\mathbf{z}$, and the value of $\mathbf{x}$ is generated by some conditional distribution

Table 1
Comparison table of recently published work.

Model	Uncertainties	Outliers	Multi-mode	Nonlinear	Multi-output
PCR and PLS based [1–4]	×	×	×	×	✓
SVR based [5,6]	×	×	×	✓	✓
PPCR and PPLS based [12–15]	✓	×	×	×	✓
VAE based [17,18]	✓	×	×	✓	✓
GMR, MPPCR and MPPLS [12,15,19]	✓	×	✓	×	✓
MVAER and DGMAN [22,23]	✓	×	✓	✓	✓
VBSMR, MRPPCR-L1 and RMPPLS [28,30,31]	✓	✓	✓	×	✓
SM-RSC [29]	✓	✓	✓	✓	×
MLVAER (Ours)	✓	✓	✓	✓	✓

$p_\theta(\mathbf{x}\,|\mathbf{z})$. Given a sample $\mathbf{x}$, its logarithmic marginal likelihood $p_\theta(\mathbf{x})$ can be decomposed into:

$$\begin{aligned}\log p_\theta(\mathbf{x}) &= \sum_{\mathbf{z}} q_\phi(\mathbf{z})\log p_\theta(\mathbf{x})\\ &= \sum_{\mathbf{z}} q_\phi(\mathbf{z})\left(\log p_\theta(\mathbf{x},\mathbf{z}) - \log p_\theta(\mathbf{z}\,|\mathbf{x})\right)\\ &= \sum_{\mathbf{z}} q_\phi(\mathbf{z})\log\frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z})} - \sum_{\mathbf{z}} q_\phi(\mathbf{z})\log\frac{p_\theta(\mathbf{z}\,|\mathbf{x})}{q_\phi(\mathbf{z})}\\ &= ELBO + KL(q_\phi(\mathbf{z})\,\|p_\theta(\mathbf{z}\,|\mathbf{x}))\end{aligned} \tag{1}$$

where $q_\phi(\mathbf{z})$ is the additionally introduced variational density function, $KL(q_\phi(\mathbf{z})\,\|p_\theta(\mathbf{z}\,|\mathbf{x}))$ is the KL divergence of distributions $q_\phi(\mathbf{z})$ and $p_\theta(\mathbf{z}\,|\mathbf{x})$, ϕ and θ are their parameters, respectively, and $ELBO$ is the lower bound of evidence. According to Bayes' theorem, the posterior distribution of $\mathbf{z}$ can be obtained as follows:

$$p_\theta(\mathbf{z}\,|\mathbf{x}) = \frac{p_\theta(\mathbf{x}\,|\mathbf{z})p_\theta(\mathbf{z})}{\int_{\mathbf{z}} p_\theta(\mathbf{x}\,|\mathbf{z})p_\theta(\mathbf{z})d\mathbf{z}} \tag{2}$$

It can be seen from Eq. (2) that the solution of $p_\theta(\mathbf{z}\,|\mathbf{x})$ involves the integral problem and is not easy to calculate. When the distribution of $p_\theta(\mathbf{z}\,|\mathbf{x})$ is more complex, the effect of using simple distribution $q_\phi(\mathbf{z})$ to approximate $p_\theta(\mathbf{z}\,|\mathbf{x})$ is also poor. $p_\theta(\mathbf{x}\,|\mathbf{z})$ is also generally difficult to model directly using known distribution family functions because of its complexity. As a deep generative model, the variational autoencoder method uses neural networks to model $q_\phi(\mathbf{z})$ and $p_\theta(\mathbf{x}\,|\mathbf{z})$, which are called inferential networks and generative networks, respectively. Since the goal of $q_\phi(\mathbf{z})$ is to approximate $p_\theta(\mathbf{z}\,|\mathbf{x})$, which is related to $\mathbf{x}$. It is often written as $q_\phi(\mathbf{z}\,|\mathbf{x})$, and generally assumed that $q_\phi(\mathbf{z}\,|\mathbf{x})$ obeys the multivariate Gaussian distribution of a diagonal matrix with a covariance matrix. The overall goal of variational autoencoders is to maximize the evidence of the lower bound $ELBO$, as follows:

$$\begin{aligned}ELBO &= E_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left(\log\frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}\,|\mathbf{x})}\right)\\ &= E_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left(\log p_\theta(\mathbf{x}\,|\mathbf{z})\right) - KL\left(q_\phi(\mathbf{z}\,|\mathbf{x})\,\|p_\theta(\mathbf{z})\right)\end{aligned} \tag{3}$$

where $p_\theta(\mathbf{z})$ is the prior distribution, generally taken as the standard Gaussian distribution, θ and ϕ are the corresponding parameters of the generated network and the inferred network, respectively.

3. Mixture Laplace distribution

To solve the analysis problem of continuous data with outliers, constructing more flexible and heavy-tailed distributions has become the research content of many scholars. The Laplace distribution is a heavy-tailed distribution. The definition of the traditional multivariate Laplace distribution, also known as the Type I multivariate Laplace distribution, is as follows:

$$\mathbf{z} = \boldsymbol{\mu} + \sqrt{2}U^{1/2}\mathbf{t} \sim L_d(\boldsymbol{\mu},\boldsymbol{\Sigma}) \tag{4}$$

where $\mathbf{t} = [T_1,\ldots,T_d]^{\mathrm{T}} \sim N_d(\mathbf{0},\boldsymbol{\Sigma})$, $U \sim Exp(1)$, U and $\mathbf{t}$ are independent of each other, d is the number of variables, $\boldsymbol{\mu}$ is the mean vector, and $\boldsymbol{\Sigma}$ is the scale parameter matrix. It can be seen that all components in Eq. (4) have the same value for the mixture variable, i.e. $U \sim Exp(1)$, so each marginal distribution can only have the same heavy-tailed, which also leads to a necessary connection between the different components, making the degrees of freedom of the type I multivariate Laplace distribution smaller than d.

To obtain a more flexible multivariate Laplace distribution, Zhang et al. [33] proposed a type II multivariate Laplace distribution by replacing the common factor U with variables $\{U_i\}_{i=1}^{d} \overset{\text{i.i.d}}{\sim} Exp(1)$ that follow a standard exponential distribution with independent identical distribution. The different components of the distribution no longer share a common factor, leading to increased degrees of freedom compared to the type I distribution. Higher degrees of freedom imply greater generality of the model with the same network structure. Furthermore, it signifies higher model compactness with the same network complexity. The type II multivariate Laplace distribution is defined as follows:

$$\mathbf{z} = \boldsymbol{\mu} + \sqrt{2}\mathbf{U}^{1/2}\mathbf{t} \sim L_d(\boldsymbol{\mu},\boldsymbol{\Sigma}) \tag{5}$$

where $\mathbf{u} = [U_1,\ldots,U_d]^{\mathrm{T}}$, $\mathbf{U}^{1/2} = diag\left(\sqrt{\mathbf{u}}\right)$, $\mathbf{u}$ and $\mathbf{t}$ are independent of each other. Given $\mathbf{u}$, the conditional distribution corresponding to $\mathbf{z}$ is:

$$\mathbf{z}\,|\mathbf{u} \sim N_d\left(\boldsymbol{\mu}, 2\mathbf{U}^{1/2}\boldsymbol{\Sigma}\mathbf{U}^{1/2}\right) \tag{6}$$

When $\boldsymbol{\Sigma}$ is a diagonal matrix, the type II Laplace distribution will degenerate into the product of multiple independent unary Laplace distributions. In this case, the Eq. (6) is equivalent to the following form:

$$\mathbf{z}\,|\mathbf{u} \sim N_d(\boldsymbol{\mu}, 2\mathbf{U}\boldsymbol{\Sigma}) \tag{7}$$

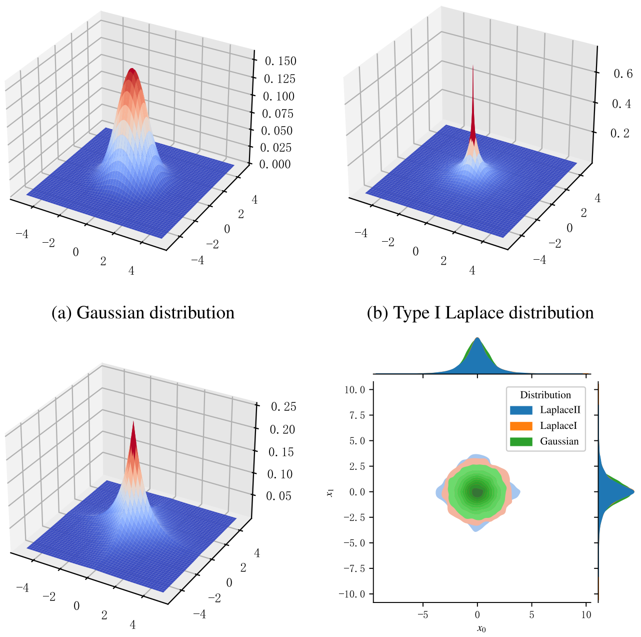
Fig. 1 shows the probability density plots of the Gaussian distribution, Type I Laplace distribution, and Type II Laplace distribution, as well as their joint probability density and edge probability density distribution comparison plots. It can be seen from the figure that the Laplace distribution has a higher probability at the edge than the Gaussian distribution. The type II Laplace distribution and type I Laplace distribution have the same edge distribution, but there are differences in the joint distribution.

Given the diverse product grades and operating conditions in industrial processes, it is common for these processes to exhibit multi-mode characteristics in the collected data. Single-peaked probability models are inadequate in effectively capturing and representing such complex, multi-mode data. To address this limitation, this paper further introduces a random variable, denoted as $c \in \{1,\ldots,K\}$, which follows a multinomial distribution. This variable is utilized to indicate the specific Laplace distribution from which the samples originate. A novel framework based on a mixture of type II multivariate Laplace distributions is proposed. The multinomial distribution is defined as follows:

$$p(c = k) = \pi_k, 1 \leqslant k \leqslant K \tag{8}$$

Let $\boldsymbol{\pi} = [\pi_1,\ldots,\pi_K]$ and satisfy $\pi_k \geqslant 0$, $\sum_{k=1}^{K}\pi_k = 1$, where π_k denotes the probability. Sample $\mathbf{x}$ is generated by the kth Laplace distribution. For convenience, $p(c = k)$ is abbreviated to $p(c_k)$. From this, a mixture model with k Laplace components can be obtained as follows:

$$ML_d\left(\boldsymbol{\pi}, \{\boldsymbol{\mu}_k\}_{k=1}^{K}, \{\boldsymbol{\Sigma}_k\}_{k=1}^{K}\right) = \sum_{k=1}^{K}\pi_k L_d(\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) \tag{9}$$



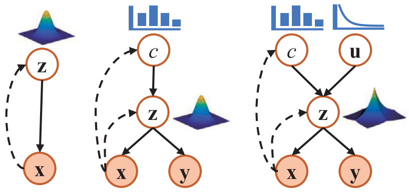
(a) Gaussian distribution

(b) Type I Laplace distribution

(c) Type II Laplace distribution

(d) Distribution comparison

Fig. 1. Distribution visualization.



(a) VAE (b) MVAER (c) MLVAER

Fig. 2. Probabilistic graphical model.

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ represent the mean vector and scale parameter matrix corresponding to the kth Laplace component, respectively, while $\left\{\boldsymbol{\mu}_k\right\}_{k=1}^{K}$ and $\left\{\boldsymbol{\Sigma}_k\right\}_{k=1}^{K}$ represent the set of $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, respectively.

4. Mixture Laplace variational autoencoder

The mixture type II multivariate Laplace distribution still has a limited ability to fit complex nonlinear data, for this reason, this paper extends it to deep networks and proposes a variational auto-encoder regression algorithm based on the mixture Laplace distribution, called MLVAER, where the latent variable $\mathbf{z} \sim ML_d\left(\boldsymbol{\pi}, \left\{\boldsymbol{\mu}_k\right\}_{k=1}^{K}, \left\{\boldsymbol{\Sigma}_k\right\}_{k=1}^{K}\right)$, and specifically, $\boldsymbol{\Sigma}_k$ is the diagonal matrix. From Section 3, the Laplace distribution can be regarded as a mixture of Gaussian distributions under multiple exponential distributions, so the latent variable $\mathbf{u}$ is introduced to assist the construction of the Laplace distribution, and the mixture Laplace distribution is in turn a mixture of Laplace distributions, so the category indicator variable c is introduced to assist the construction of the mixture Laplace distribution. The variables $\mathbf{u}$ and c are defined in the same way as in Section 3. Fig. 2 shows the probabilistic graphical models of VAE, MVAER, and MLVAER, where the solid line indicates the generative model and the dashed line indicates the variational approximation.

Further, the framework of MLVAER is shown in Fig. 3, where the trapezoidal boxes represent the neural network, the light blue background boxes represent additional descriptions of the variables, and the arrows represent the flow of data. It can be seen from Fig. 3 that MLVAER is mainly composed of five parts: inference network 1, inference network 2, prior network, generative network, and regression network. Its theoretical basis and loss function of training will be given in Section 4.1.

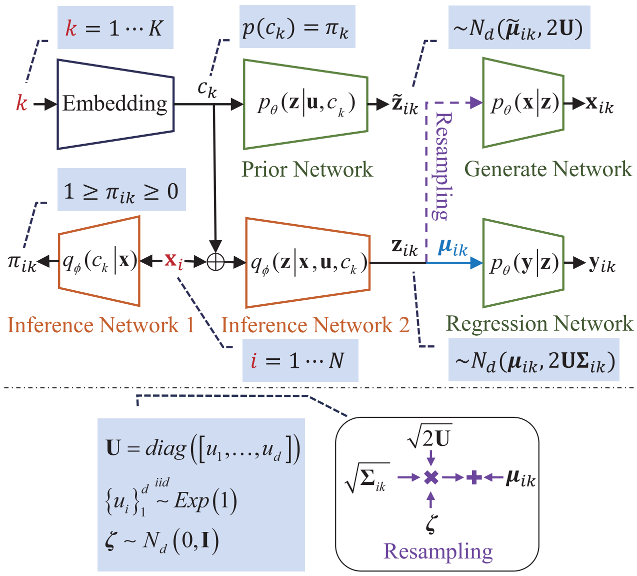

$k = 1 \cdots K$
$p(c_k) = \pi_k$
$\sim N_d(\tilde{\boldsymbol{\mu}}_{ik}, 2\mathbf{U})$
k
Embedding
c_k
$p_\theta(\mathbf{z}|\mathbf{u}, c_k)$
$\tilde{\mathbf{z}}_{ik}$
Resampling
$p_\theta(\mathbf{x}|\mathbf{z})$
$\mathbf{x}_{ik}$
Prior Network
Generate Network
$1 \geq \pi_{ik} \geq 0$
π_{ik}
$q_\phi(c_k|\mathbf{x})$
$\mathbf{x}_i$
$q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{u}, c_k)$
$\mathbf{z}_{ik}$
$\boldsymbol{\mu}_{ik}$
$p_\theta(\mathbf{y}|\mathbf{z})$
$\mathbf{y}_{ik}$
Inference Network 1
Inference Network 2
Regression Network
$i = 1 \cdots N$
$\sim N_d(\boldsymbol{\mu}_{ik}, 2\mathbf{U}\boldsymbol{\Sigma}_{ik})$
$\mathbf{U} = diag([u_1, \ldots, u_d])$
$\{u_i\}_1^d \overset{iid}{\sim} Exp(1)$
$\zeta \sim N_d(0, \mathbf{I})$
$\sqrt{2\mathbf{U}}$
$\sqrt{\boldsymbol{\Sigma}_{ik}}$
$\boldsymbol{\mu}_{ik}$
ζ
Resampling

Fig. 3. Mixture Laplace auto-encoder framework diagram.

4.1. Variational lower bound

The mixture Laplace variational autoencoder introduces three latent variables $\mathbf{z}$, $\mathbf{u}$, and c. According to Eq. (3), its variational lower bound is:

$$\begin{aligned} ELBO = & E_{\mathbf{z},\mathbf{u},c_k \sim q_\phi(\mathbf{z},\mathbf{u},c_k|\mathbf{x})}\left(\log p_\theta(\mathbf{x},\mathbf{y}\,|\mathbf{z},\mathbf{u},c_k)\right) \\ & - KL\left(q_\phi(\mathbf{z},\mathbf{u},c_k\,|\mathbf{x})\,\|p_\theta(\mathbf{z},\mathbf{u},c_k)\right) \end{aligned} \tag{10}$$

According to the generation process, the generation of $\mathbf{x}$ and $\mathbf{y}$ is only directly related to $\mathbf{z}$. Eq. (10) can be further expressed in the following form:

$$\begin{aligned} ELBO = & \sum_{k=1}^{K} q_\phi(c_k\,|\mathbf{x}) E_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x},c_k)}\left(\log p_\theta(\mathbf{x}\,|\mathbf{z})\right) \\ & + \sum_{k=1}^{K} q_\phi(c_k\,|\mathbf{x}) E_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x},c_k)}\left(\log p_\theta(\mathbf{y}\,|\mathbf{z})\right) \\ & - KL\left(q_\phi(\mathbf{z},\mathbf{u},c_k\,|\mathbf{x})\,\|p_\theta(\mathbf{z},\mathbf{u},c_k)\right) \end{aligned} \tag{11}$$

The first term can be regarded as the minimum reconstruction error, and its expectation can be approximated according to the sampling, but since the sampling relation is not derivable, the corresponding reparameterization technique is proposed for calculation in Section 4.2. Assuming that $p_\theta(\mathbf{x}\,|\mathbf{z})$ obeys Laplace distribution with mean $\hat{\mathbf{x}}_k$ and scale parameter 1, the first term can be simplified to absolute value loss $-\sum_{k=1}^{K} q_\phi(c_k\,|\mathbf{x})\,|\mathbf{x} - \hat{\mathbf{x}}_k|$.

The second term can adopt a similar strategy as the first one, however, considering the practical situation of soft sensor modeling, the predicted value should be more accurate and stable close to the real value without the need for generation ability. Meanwhile, the variance part of the latent variable $\mathbf{z}$ can be regarded as noise independent of the predicted value $\mathbf{y}$, so only the mean part of z needs to be input to the regressor, that is, the input to the regression network is $\boldsymbol{\mu}_k$. Assuming that $p_\theta(\mathbf{y}\,|\mathbf{z})$ obeys a Laplace distribution with mean $\hat{\mathbf{y}}_k$ and scale parameter 1, the second term can be simplified to absolute value loss $-\sum_{k=1}^{K} q_\phi(c_k\,|\mathbf{x})\,|\mathbf{y} - \hat{\mathbf{y}}_k|$.

The third term can be further broken down as follows:

$$\begin{aligned}
&KL\left(q_{\phi}(\mathbf{z},\mathbf{u},c_k\,|\mathbf{x})\,\|p_{\theta}(\mathbf{z},\mathbf{u},c_k)\right)\\
&=\int_{\mathbf{z}}\sum_{k=1}^{K}\int_{\mathbf{u}}\left(q_{\phi}(\mathbf{z},\mathbf{u},c_k\,|\mathbf{x})\log\frac{q_{\phi}(\mathbf{z},\mathbf{u},c_k\,|\mathbf{x})}{p_{\theta}(\mathbf{z},\mathbf{u},c_k)}\right)\\
&=\int_{\mathbf{z}}\sum_{k=1}^{K}\int_{\mathbf{u}}\left(q_{\phi}(\mathbf{z},\mathbf{u},c_k\,|\mathbf{x})\log\frac{q_{\phi}(\mathbf{z}\,|\mathbf{x}\,,\mathbf{u},c_k)}{p_{\theta}(\mathbf{z}\,|\mathbf{u},c_k\,)}\right)\\
&\quad+\int_{\mathbf{z}}\sum_{k=1}^{K}\int_{\mathbf{u}}\left(q_{\phi}(\mathbf{z},\mathbf{u},c_k\,|\mathbf{x})\log\frac{q_{\phi}(\mathbf{u})}{p_{\theta}(\mathbf{u})}\right)\\
&\quad+\int_{\mathbf{z}}\sum_{k=1}^{K}\int_{\mathbf{u}}\left(q_{\phi}(\mathbf{z},\mathbf{u},c_k\,|\mathbf{x})\log\frac{q_{\phi}(c_k\,|\mathbf{x})}{p_{\theta}(c_k)}\right)\\
&=\sum_{k=1}^{K}q_{\phi}(c_k\,|\mathbf{x})\int_{\mathbf{u}}q_{\phi}(\mathbf{u})KL\left(q_{\phi}(\mathbf{z}\,|\mathbf{x}\,,\mathbf{u},c_k)\,\|p_{\theta}(\mathbf{z}\,|\mathbf{u},c_k\,)\right)\\
&\quad+KL\left(q_{\phi}(c_k\,|\mathbf{x})\,\|p_{\theta}(c_k)\right)
\end{aligned}\tag{12}$$

where $p_{\theta}\left(\mathbf{z}\,|c_k\right)=L_d\left(\mathbf{0},\mathbf{I}\right)$, $p_{\theta}\left(\mathbf{z}\,|\mathbf{u},c_k\right)=N_d\left(\tilde{\boldsymbol{\mu}}_k,2\mathbf{U}\right)$, $p_{\theta}(c_k)=\frac{1}{K}$, $\tilde{\boldsymbol{\mu}}_k$ is the parameter to be learned by the neural network.

$$KL\left(q_{\phi}(c_k\,|\mathbf{x})\,\|p_{\theta}(c_k)\right)=\sum_{k=1}^{K}q_{\phi}(c_k\,|\mathbf{x})\log\left(Kq_{\phi}(c_k\,|\mathbf{x})\right)\tag{13}$$

$$\begin{aligned}
&KL\left(q_{\phi}(\mathbf{z}\,|\mathbf{x}\,,\mathbf{u},c_k)\,\|p_{\theta}(\mathbf{z}\,|\mathbf{u},c_k\,)\right)\\
&=\frac{1}{2}\left(\operatorname{tr}\left(\boldsymbol{\Sigma}_k\right)-d-\log\left(\left|\boldsymbol{\Sigma}_k\right|\right)+\frac{\left(\boldsymbol{\mu}_k-\tilde{\boldsymbol{\mu}}_k\right)^{\mathrm{T}}\mathbf{U}^{-1}\left(\boldsymbol{\mu}_k-\tilde{\boldsymbol{\mu}}_k\right)}{2}\right)
\end{aligned}\tag{14}$$

However, the KL divergence between $q_{\phi}(\mathbf{z}\,|\mathbf{x}\,,\mathbf{u},c_k)$ and $p_{\theta}(\mathbf{z}\,|\mathbf{u},c_k\,)$ tends to infinity when the elements in $\mathbf{u}$ tend to zero, and the KL divergence fails to measure the difference between the two distributions. Therefore, this paper uses the Wasserstein distance to measure the difference between $q_{\phi}(\mathbf{z}\,|\mathbf{x}\,,\mathbf{u},c_k)$ and $p_{\theta}(\mathbf{z}\,|\mathbf{u},c_k\,)$.

$$\begin{aligned}
&\mathcal{W}[q_{\phi}(\mathbf{z}\,|\mathbf{x}\,,\mathbf{u},c_k),p_{\theta}(\mathbf{z}\,|\mathbf{u},c_k\,)]\\
&=\left\|\boldsymbol{\mu}_k-\tilde{\boldsymbol{\mu}}_k\right\|^2+2\operatorname{Tr}\left(\mathbf{U}\right)+2\operatorname{Tr}\left(\boldsymbol{\Sigma}_k\mathbf{U}\right)-4\operatorname{Tr}\left(\left(\boldsymbol{\Sigma}_k\mathbf{U}^2\right)^{1/2}\right)
\end{aligned}\tag{15}$$

Further obtainable:

$$\begin{aligned}
&\int_{\mathbf{u}}q_{\phi}(\mathbf{u})\mathcal{W}[q_{\phi}(\mathbf{z}\,|\mathbf{x}\,,\mathbf{u},c_k),p_{\theta}(\mathbf{z}\,|\mathbf{u},c_k\,)]\\
&=\left\|\boldsymbol{\mu}_k-\tilde{\boldsymbol{\mu}}_k\right\|^2+2\operatorname{Tr}\left(\mathbf{I}\right)+2\operatorname{Tr}\left(\boldsymbol{\Sigma}_k\right)-4\operatorname{Tr}\left(\boldsymbol{\Sigma}_k^{1/2}\right)
\end{aligned}\tag{16}$$

Combining the above analysis, the final variational lower bound is expressed as:

$$\begin{aligned}
ELBO=&-\sum_{k=1}^{K}q_{\phi}(c_k\,|\mathbf{x})\left|\mathbf{x}-\hat{\mathbf{x}}_k\right|-\sum_{k=1}^{K}q_{\phi}(c_k\,|\mathbf{x})\left|\mathbf{y}-\hat{\mathbf{y}}_k\right|\\
&-\sum_{k=1}^{K}q_{\phi}(c_k\,|\mathbf{x})Div
\end{aligned}\tag{17}$$

where $Div=\log\left(Kq_{\phi}(c_k\,|\mathbf{x})\right)+\left\|\boldsymbol{\mu}_k-\tilde{\boldsymbol{\mu}}_k\right\|^2+2\operatorname{Tr}\left(\mathbf{I}\right)+2\operatorname{Tr}\left(\boldsymbol{\Sigma}_k\right)-4\operatorname{Tr}\left(\boldsymbol{\Sigma}_k^{1/2}\right)$.

4.2. Reparameterization trick

In the framework of the algorithm proposed in this paper, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the outputs of the neural network containing gradient information. To ensure that the gradient descent training of the neural network proceeds smoothly, the resampling equation for the Laplace distribution only needs to include $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$. In Eq. (4), since $\mathbf{u}$ and $\mathbf{t}$ are independent of each other, they can be sampled separately to obtain $\hat{\mathbf{u}}$ and $\hat{\mathbf{t}}$. When sampling $\mathbf{t}$, considering that $\mathbf{t}$ obeys a Gaussian distribution, the standard Gaussian distribution can be sampled first to obtain $\boldsymbol{\zeta}_k$, and according to the nature of the Gaussian distribution, the sampling of $\mathbf{t}$ can be further obtained as $\hat{\mathbf{t}}=\boldsymbol{\zeta}_k\odot\sqrt{\boldsymbol{\Sigma}_k}$. The detailed process is:

(1) Generate K-independent samples $\left\{\boldsymbol{\zeta}_k\right\}_{k=1}^{K}$ that obey the standard Gaussian distribution;

(2) Generate d independent samples $\hat{\mathbf{u}}=\left[u_1,\dots,u_d\right]^{\mathrm{T}}$, $\hat{\mathbf{U}}^{1/2}=diag\left(\sqrt{\hat{\mathbf{u}}}\right)$ that obey an exponential distribution with $\lambda=1$;

(3) Repeat step 2 to generate a total of K sets of $\left\{\hat{\mathbf{U}}_k^{1/2}\right\}_{k=1}^{K}$;

(4) The sampling results for the Laplace distribution are as follows:

$$\hat{\mathbf{z}}_k=\boldsymbol{\mu}_k+\sqrt{2}\hat{\mathbf{U}}_k^{1/2}\left(\boldsymbol{\zeta}_k\odot\sqrt{\boldsymbol{\Sigma}_k}\right).$$

4.3. Procedure of MLVAER modeling

For convenience, the inferred network 1 $q_{\phi}(c_k\,|\mathbf{x})$, inferred network 2 $q_{\phi}(\mathbf{z}\,|\mathbf{x}\,,\mathbf{u},c_k)$, prior network $p_{\theta}(\mathbf{z}\,|\mathbf{u},c_k\,)$, generative network $p_{\theta}(\mathbf{x}\,|\mathbf{z})$, and regression network $p_{\theta}(\mathbf{y}\,|\mathbf{z})$ are denoted as $\mathit{\Pi}_{\phi}$, Q_{ϕ}, P_{θ}, G_{θ} and R_{θ}. The pseudo-code of MLVAER is shown in Algorithm 1.

Algorithm 1 MLVAER

```
Input: Data $\left\{\mathbf{x}_i\right\}_{i=1}^{n}$, $k=1,\dots,K$
Output: $\mathit{\Pi}_{\phi}$, $Q_{\phi}$, $P_{\theta}$, $G_{\theta}$ and $R_{\theta}$
Initialization parameters $\phi$ and $\theta$;
while $(\phi,\theta)$ not converged do
    $c_k=\mathrm{Embedding}\left(k\right)$
    $\boldsymbol{\pi}_i=\mathit{\Pi}_{\phi}\left(\mathbf{x}_i\right)$
    $\tilde{\boldsymbol{\mu}}_{ik}=P_{\theta}\left(c_k\right)$
    $\left\{\boldsymbol{\mu}_{ik}\right\}_{k=1}^{K},\left\{\boldsymbol{\Sigma}_{ik}\right\}_{k=1}^{K}=\left\{Q_{\phi}\left(\mathbf{x}_i\oplus c_k\right)\right\}_{k=1}^{K}$
    According to Section 4.2, we get $\left\{\hat{\mathbf{z}}_{ik}\right\}_{k=1}^{K}$
    $\hat{\mathbf{x}}_{ik}=G_{\theta}\left(\hat{\mathbf{z}}_{ik}\right)$
    $\hat{\mathbf{y}}_{ik}=R_{\theta}\left(\boldsymbol{\mu}_{ik}\right)$
    $Div_{ik}=\log\left(K\pi_{ik}\right)+\left\|\boldsymbol{\mu}_{ik}-\tilde{\boldsymbol{\mu}}_{ik}\right\|^2+2\operatorname{Tr}\left(\mathbf{I}\right)+2\operatorname{Tr}\left(\boldsymbol{\Sigma}_{ik}\right)-4\operatorname{Tr}\left(\boldsymbol{\Sigma}_{ik}^{1/2}\right)$
    $ELBO_i=-\sum_{k=1}^{K}\pi_i\left|\mathbf{x}_i-\hat{\mathbf{x}}_{ik}\right|-\sum_{k=1}^{K}\pi_i\left|\mathbf{y}_i-\hat{\mathbf{y}}_{ik}\right|-\sum_{k=1}^{K}\pi_iDiv_{ik}$
    The parameters $(\phi,\theta)$ are updated according to the loss $Loss=-\sum_{i=1}^{n}ELBO_i$ using the gradient descent method.
end while
```

5. Experiment

In this section, the proposed model is experimentally validated by numerical simulation experiments, Tennessee Eastman (TE) simulation experiments [34], and laboratory small wet ball mill experiments, and compared with the five methods. The GMR algorithm is a common non-deep multi-mode regression algorithm; AE is a representative algorithm for deep learning; VAE can be regarded as a probabilistic form of AE, which is a traditional algorithm for deep probability learning; MVAER extends the latent variables obeying Gaussian distribution to Gaussian mixture distribution based on VAE; LVAER replaces Gaussian distribution with Laplace distribution, and LVAER is also a special form of the proposed method when the mode number is set to 1. In summary, this experiment contains a comparison of the deep learning method and non-deep learning method, the comparison of the non-probability model and probability model, the comparison of the Gaussian distribution hypothesis and Laplace distribution hypothesis, the comparison of the single distribution hypothesis and mixture distribution hypothesis, which can make the effect of the model fully verified.

To evaluate the performance of the model, two metrics (root mean squared error (RMSE) and R^2 coefficient) are used to quantify the prediction effect of the model, which are defined in Eqs. (18) and (19), respectively.

$$\mathrm{RMSE}=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i-\hat{y}_i\right)^2}\tag{18}$$

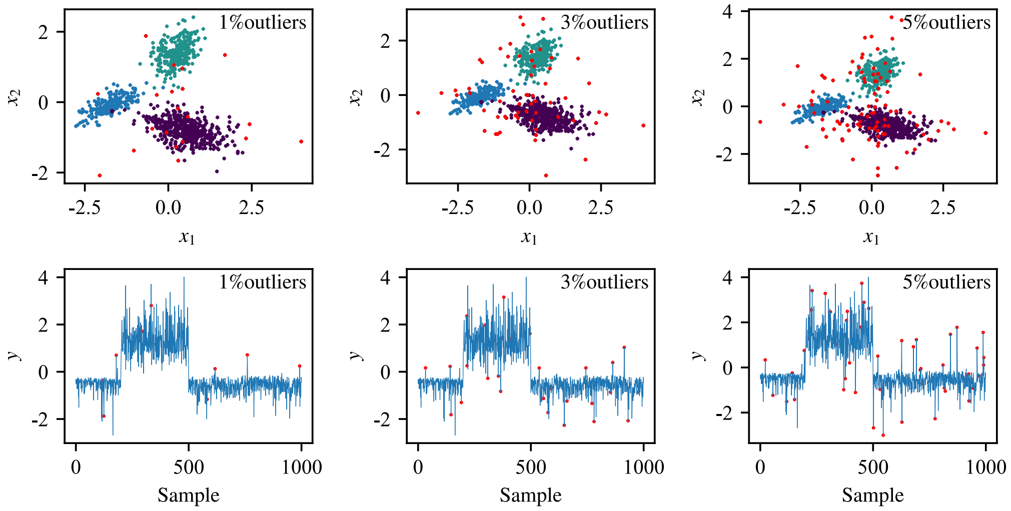


Fig. 4. Schematic diagram of the distribution of the training set data in numerical simulation.

Table 2
Numerical simulation parameter settings.

Parameters	input $[x_1 \quad x_2]$			output y
	π	μ	Σ	
k=1	0.2	$[-4 \quad 0]$	$\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$	$y = 1.5x_1e^{x_2}$
k=2	0.3	$[2 \quad 6]$	$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 2 \end{bmatrix}$	$y = x_1 + x_2^2$
k=3	0.5	$[3 \quad -3]$	$\begin{bmatrix} 3 & -1 \\ -1 & 1.5 \end{bmatrix}$	$y = x_1x_2$

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{N}\left(y_i - \bar{y}\right)^2} \tag{19}$$

where N is the number of samples, $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, $\hat{y}_i$ is the predicted value of the ith sample.

5.1. Numerical simulation experiments

In this section, a numerical simulation with three modes is designed, where each mode follows a different Gaussian distribution. Each Gaussian component and its functional relationship with the output are shown in Table 2.

According to the parameter settings in Table 2, 1000 samples were generated as the training set, and 500 samples were generated as the test set. The training set data are normalized to zero mean unit variance, and the test set is similarly processed according to the mean and variance of the training set. To simulate data polluted by outliers, outliers uniformly distributed in [−2, 2] are added along each feature and label dimension for the normalized training set. The percentages of abnormal values were 1%, 3% and 5% of the total, respectively, and a total of one set of normal data and three sets of polluted data were generated. Fig. 4 plots the distribution of the training set samples contaminated with outliers, where the red dots represent samples or labels with outliers. From the figure, we can see that the generated data are multi-mode and heavy-tailed.

According to the trial-and-error method, the inferred networks 1 and 2 of the proposed method are set as a single hidden layer, the number of hidden layer units is set to 8 and 16, the latent variable dimension is set to 2, the number of modes is set to 3, the regression network is set as a fully connected layer, the generated network is symmetric with the inferred network, the epoch is set to 200, the batch size is set to 32, and the model with the best result in the training set 200 times is taken as the final model. The corresponding parameter settings of other comparison methods are consistent with those of the method proposed in this paper. The experimental results, RMSE and R^2 are shown in Table 3.

For a more visual comparison of the models, histograms and line plots of the evaluation index results for different models with different proportional outlier disturbances are plotted in Fig. 5.

From the experimental results, it can be seen that MLVAER achieves the best results in all cases, and LVAER is the next best. The slope of the line in Fig. 5 shows that MLVAER and LVAER are less affected by the outliers, and MLVAER is more robust than LVAER. The VAE algorithm based on a Gaussian distribution of latent variables and the MVAER algorithm based on a Gaussian mixture distribution are more susceptible to outliers. As a deterministic modeling method, AE is less accurate than probabilistic modeling and even inferior to the non-deep GMR algorithm when the percentage of outliers is small. The performance of the GMR algorithm deteriorates dramatically when the outliers are accounted for 5%. Due to the powerful feature extraction capability of the neural network, the upward trend of the error or the downward trend of the R^2 of the AE algorithm does not change greatly.

Fig. 6 shows the predicted curve versus the true curve when the outliers are accounted for 3% in the total. Methods constructed based on Gaussian or a mixture of Gaussian distributions are disturbed by outliers that increase the variance of the Gaussian distribution, which enhances the robustness and may significantly reduce the sensitivity of the model to the sample, making it difficult to fit the output to the peak data. The AE algorithm does not take into account the randomness of the data, and the fit to the peak data is also poor. MLVAER and LVAER algorithms introduce the Laplace distribution to model the heavy-tailed data, which effectively reduces the interference of outliers. MLVAER also takes into account the multi-mode situation of the data and further improves the accuracy compared with LVAER.

5.2. TE dataset experiments

The TE process simulation platform is a common industrial process simulation platform in the field of soft sensors. The TE process is shown in Fig. 7. A detailed description is available in [34].

The reactor pressure and reactor level in the process have a significant impact on the production cost and are often adjusted according to the production requirements. In this paper, the reactor pressure and level are set according to the parameters in Table 4, and the data are obtained for four conditions, each containing 1000 samples. From each condition, 70% of the samples were randomly selected as the training set, and the rest of the samples were used as the test set. The TE

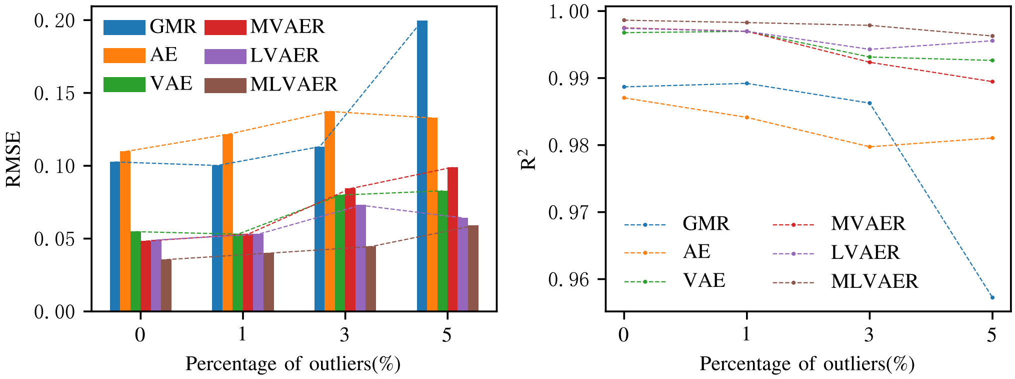


Fig. 5. Visualization comparison of algorithm performance in numerical simulation experiments.

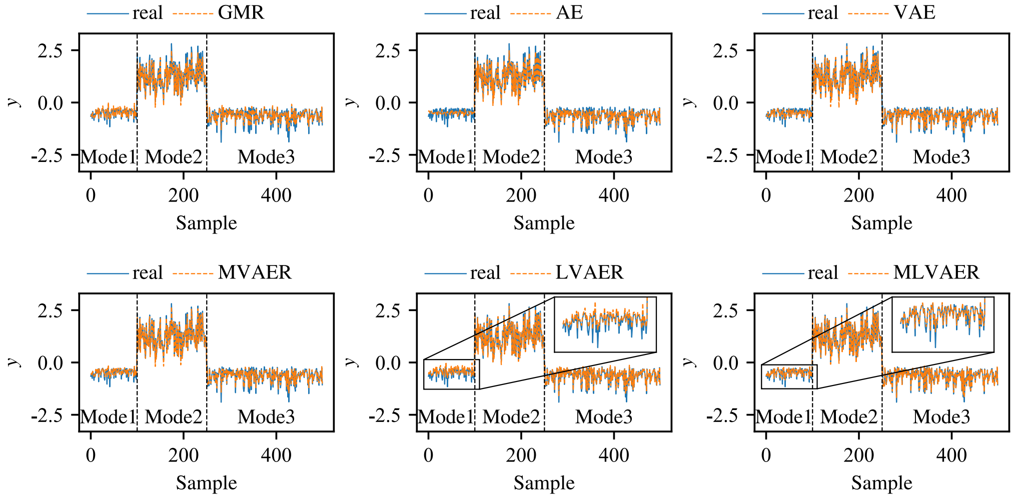


Fig. 6. Comparison of prediction curves using different algorithms in numerical simulation experiments.

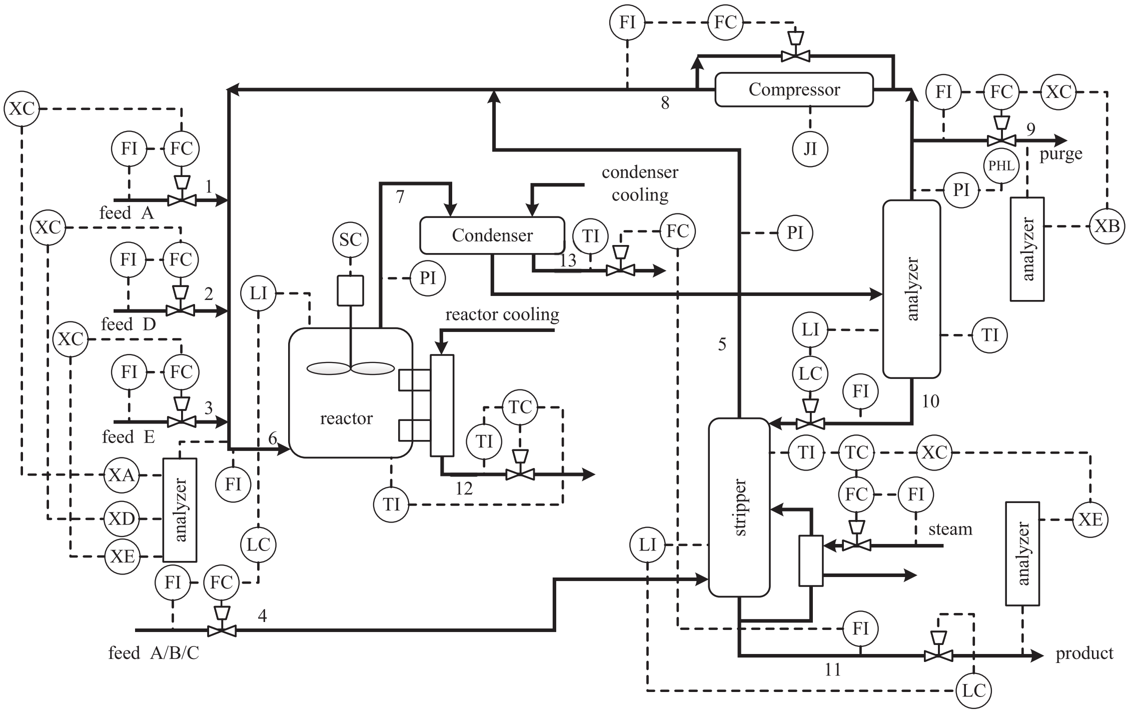


Fig. 7. Schematic diagram of the TE process.

Table 3
Algorithm performance evaluation indicators for numerical simulation experiments.

Percentage of outliers	Evaluation indicators	Soft sensor algorithms					
		GMR	AE	VAE	MVAER	LVAER	MLVAER
0%	RMSE	0.1026	0.1097	0.0548	0.0483	0.0490	**0.0356**
	R^2	0.9887	0.9871	0.9968	0.9975	0.9974	**0.9986**
1%	RMSE	0.1002	0.1215	0.0530	0.0531	0.0530	**0.0340**
	R^2	0.9892	0.9841	0.9970	0.9970	0.9970	**0.9983**
3%	RMSE	0.1331	0.1373	0.0799	0.0843	0.0729	**0.0445**
	R^2	0.9863	0.9797	0.9931	0.9924	0.9943	**0.9978**
5%	RMSE	0.1995	0.1327	0.0828	0.0990	0.0642	**0.0588**
	R^2	0.9572	0.9811	0.9926	0.9895	0.9956	**0.9963**

Table 4
Pressure and level setting values of TE reactor.

Condition	Reactor pressure /Pa	Reactor level /%
1	2750	65
2	2250	65
3	2250	75
4	2750	75

Table 5
Algorithm performance evaluation indicators for TE simulation experiments.

	GMR	AE	VAE	MVAER	LVAER	MLVAER
RMSE	0.0890	0.1400	0.0777	0.0744	0.0737	**0.0555**
R^2	0.9922	0.9808	0.9941	0.9946	0.9947	**0.9970**

process has 12 manipulated variables (XMV(1-12)) and 41 measured variables (XMEAS(1-41)), of which the measured variables contain 22 process-measured variables and 19 component-measured variables. In this paper, the process measurement variables and manipulation variables are used as auxiliary variables of the data-driven soft sensor to predict component B (XMEAS(30)) to be measured.

According to the trial-and-error method, the inferred networks 1 and 2 of the proposed method are set as a single hidden layer, the number of hidden layer cells is set to 16 and 32, the latent variable dimension is set to 8, the number of modes is set to 4, the batch size is set to 128, and the rest of the settings are the same as the numerical simulation experiments.

The experimental results (RMSE and R^2 coefficients) are shown in Table 5. For a more visual comparison of the model performance, Fig. 8 shows the histograms of the performance metrics of different algorithms. From the figure, it can be seen that the MLVAER with the latent variable introduced into the mixture Laplace distribution performs significantly better than the other algorithms, and the LVAER with the single Laplace distribution is the second best. Similarly, the MVAER algorithm with latent variables introduced into the mixture Gaussian distribution outperforms the VAE algorithm with a single Gaussian distribution. GMR, as a non-deep method, cannot extract the nonlinear relationships of the data well, and GMR uses all the features in the prediction process and is more susceptible to outlier interference. The AE algorithm does not consider the uncertainty of the data, and its modeling effect is the worst.

Fig. 9 shows the plot of predicted and true values for different algorithms, while Fig. 10 shows the scatter plot of predicted and true values for different algorithms. From these two plots, it can be seen that the prediction results of MLVAER are closest to the true values. From Fig. 9, it can be seen that LVAER has a significant drift in the third mode prediction. The prediction curves of other algorithms have more noise fluctuations. Among them, GMR also has a large outlier in the first mode.

5.3. Ball mill dataset experiments

Ball mills are essential crushing equipment in the process industry. If they cannot operate at the optimal load point, it can cause a large amount of energy waste, and in severe cases, lead to serious production accidents. Its process has the characteristics of high nonlinearity, multiple outputs, and strong coupling. At the same time, its internal environment is harsh, making it difficult to directly detect load parameters using physical sensors. Due to the widespread occurrence of multiple operating conditions in industrial processes, there is an urgent need for high-performance multi-mode soft sensor algorithms to predict mill load parameters.

The key load parameters commonly used in the industry for ball mills include charge volume ratio (CVR), material-to-ball volume ratio (MBVR), pulp density (PD) and ball charge volume ratio (BCVR). In this paper, a small laboratory wet ball mill with a drum diameter of 60 cm, length of 70 cm, volume of 200L, and maximum steel ball loading of 0.6t was used for simulated operation. The mill was driven by a three-phase motor with a power of 4 kW through a QYD-type speed reducer, and its speed was controlled by a frequency converter. A steel ball with a diameter of 30 mm was selected as the grinding medium to grind the iron powder in the test, and an acceleration sensor model ULT2003V was used to detect the vibration signal of the bearing seat. The signals were synchronously acquired by NI's cDAQ9184 with a sampling frequency of 51.2 kHz. To facilitate the storage and monitoring of data, Labview is used to build a data acquisition platform, and high-speed real-time communication with cDAQ9184 is carried out through USB. The experimental process of the ball mill is shown in Fig. 11.

To simulate the multi-working condition process in the actual industry, the experiment changed the BCVR of the mill and got five different working conditions. After Fourier transformed the collected vibration signals, the experimental data of the ball mill's multi-condition process was obtained. In this paper, the Fourier-transformed features of the bearing vibration signals are used as input to model and predict the key load parameters (CVR, MBVR, and PD) of the mill.

According to the trial-and-error method, the inferred network 1 was set as two hidden layers with the number of hidden layer cells set to 64 and 32, the inferred network 2 was set as two hidden layers with the number of hidden layer cells set to 128 and 64, the latent variable dimension set to 64, the number of modes set to 5, the batch size set to 512, and the rest of the settings were the same as the numerical simulation experiments.

The experimental results, RMSE and R^2 are shown in Table 6. Fig. 12 shows the histograms of the performance evaluation for a more visual comparison of the models. The scatter plots for the three major load parameters of CVR, MBVR, and PD are shown in Fig. 13, Fig. 14, and Fig. 15. It can be seen that MLVAER performs best among all the algorithms, with LVAER being the next best. The scatter plot shows that the Laplace distribution-based algorithm does not have significant mean drift and the predictions are less noisy and closer to the true values than the other algorithms. GMR is the worst performer in dealing with high-dimensional nonlinear data such as ball mill data because it does not have feature extraction and nonlinear modeling capabilities.

Based on the above experimental analysis, the proposed method MLVAER and its special form of single Laplace distribution LVAER are significantly better than other comparison algorithms in dealing with nonlinear and outliers. Among them, the accuracy of MLVAER is further improved based on LVAER by considering the multi-mode of the data.

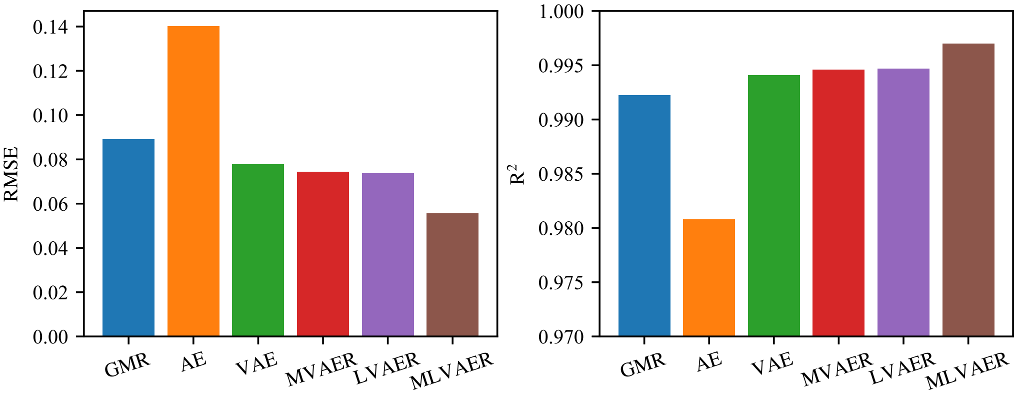


Fig. 8. Histograms of performance indicators for different algorithms in TE experiments.

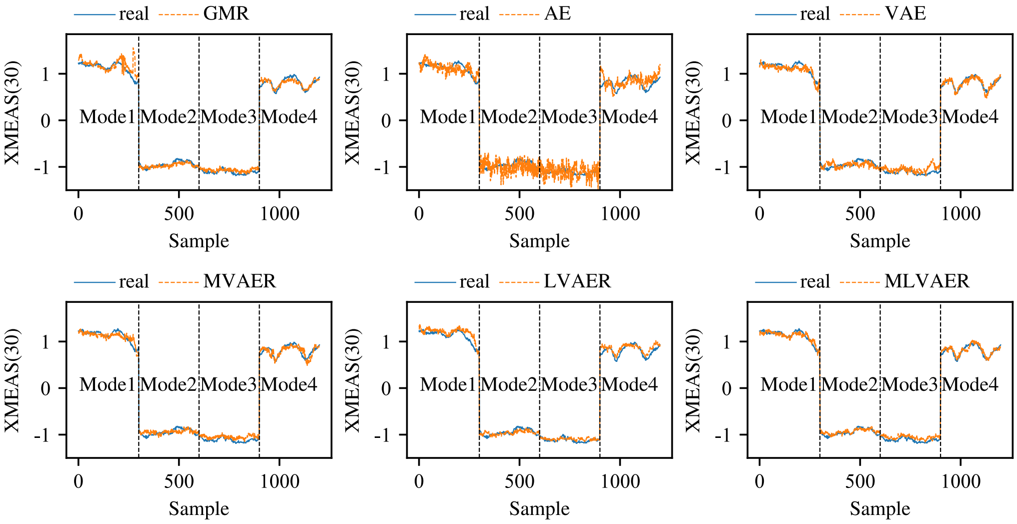


Fig. 9. Comparison of prediction curves using different algorithms in TE simulation experiments.

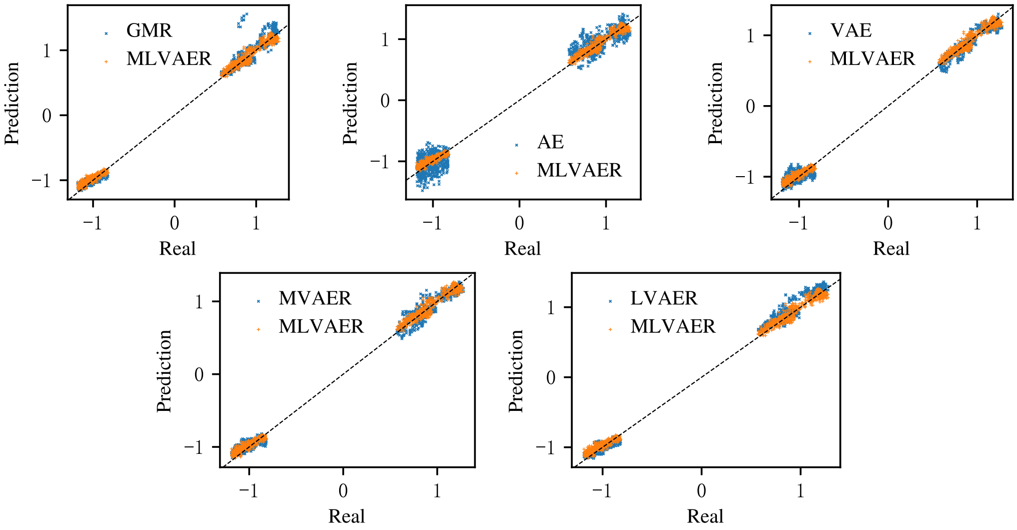
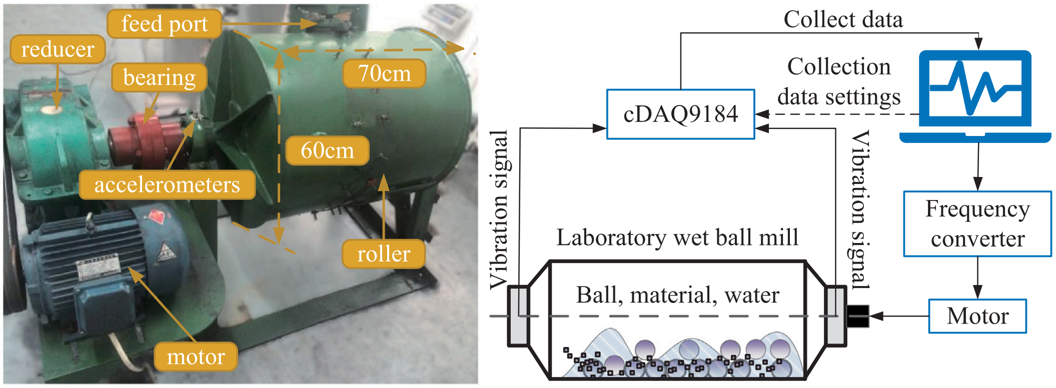


Fig. 10. Scatter plot comparison of prediction results of different algorithms in the TE simulation experiment.

6. Model complexity analysis

In this section, MACs (Multiply-Accumulate Operations) and Params (Parameter Quantity) are used to measure the time complexity and space complexity of the model, respectively. Where 1 MACs contains a Multiply-Accumulate Operation and an Additive Operation, while Params denote the parameter quantity of the model. When calculating MACs and Params, the structure of the model adopts the same settings as the experiments on the TE dataset, and the size of the input matrix is set to 100 × 30. In addition, to better compare the computation time



(a) Ball mill structure

(b) Ball mill data collection process

Fig. 11. Schematic diagram of the experimental process of the ball mill.

Table 6
Algorithm performance evaluation indicators for ball mill experiments.

Load parameters	Evaluation indicators	Soft sensor algorithms					
		GMR	AE	VAE	MVAER	LVAER	MLVAER
CVR	RMSE	0.2454	0.1091	0.1075	0.1306	0.0934	**0.0837**
	R^2	0.9398	0.9881	0.9884	0.9829	0.9913	**0.9930**
MBVR	RMSE	0.2990	0.0981	0.0959	0.0916	0.0795	**0.0737**
	R^2	0.9106	0.9904	0.9908	0.9916	0.9937	**0.9946**
PD	RMSE	0.1922	0.1192	0.1216	0.1309	0.1150	**0.0987**
	R^2	0.9630	0.9858	0.9852	0.9829	0.9868	**0.9903**

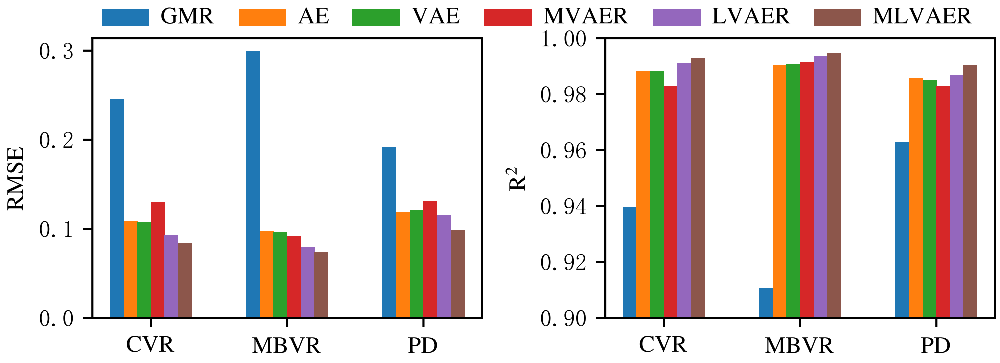


Fig. 12. Histograms of performance indicators for different algorithms in ball mill experiments.

Table 7
Complexity and accuracy of the model.

Metrics	AE	VAE	MVAER	LVAER	MLVAER
MACs	2.4400e+05	2.6960e+05	8.1419e+06	3.2940e+05	1.1739e+06
Params	2.5430e+03	2.8070e+03	8.8160e+03	3.4400e+03	3.4910e+03
Train Time	26.5488	33.4180	95.1792	53.8783	109.1532
Test Time	0.0073	0.0094	0.0193	0.0149	0.0175
RMSE	0.1400	0.0777	0.0744	0.0737	0.0555

of the model in the real case, this section compares the running time of different models in the training and testing phases using the TE dataset as an example, and the RMSE of the model is also given as a reference. The specific results are shown in Table 7.

To compare the differences between different models more intuitively, this section normalizes the different metrics and plots them in the form of radar charts, as shown in Fig. 16. The closer the model metrics are to the center point location that their corresponding performance is better. From the figure, it can be seen that the mixture models MLVAER and MVAER take longer time to train and test compared to other models, and the time complexity and space complexity of the models is also higher. However, the time complexity and space complexity of MLVAER is much smaller than that of MVAER. This is mainly because MVAER builds a separate neural network for each distribution, which results in redundancy of the network, while MLVAER simplifies the network by changing the inputs of different category variables to realize different distributions corresponding to different outputs. Studies have shown that the shallow layers of neural networks are mainly used to extract some generalized features, and the last layers are related to the task. The structure of the proposed algorithm and the comparison algorithm in this paper can be regarded as the last layers of the network that are related to the task, and thus do not bring excessive computational burden due to the structural problem when the data dimensions are higher and the model is more complex.

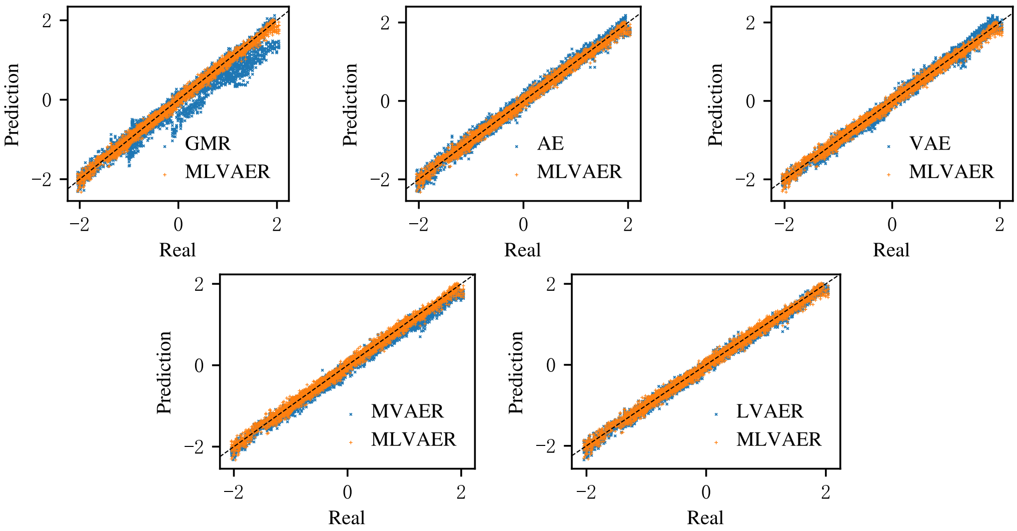


Fig. 13. Prediction results of CVR in ball mill experiments.

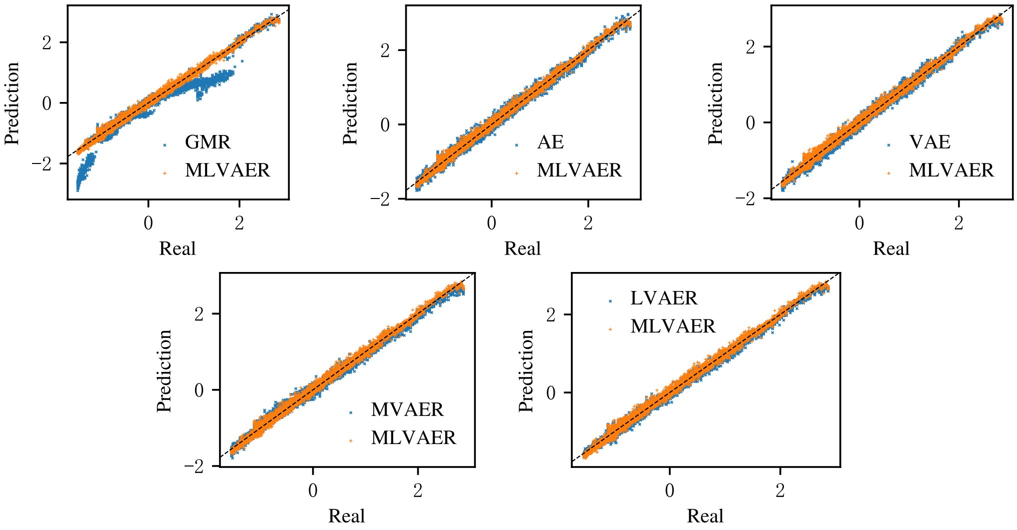


Fig. 14. Prediction results of MBVR in ball mill experiments.

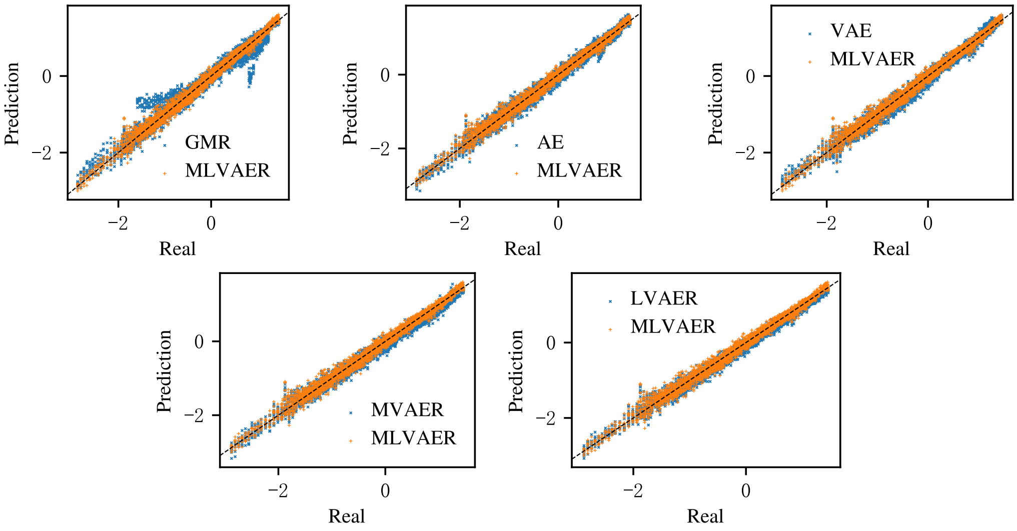


Fig. 15. Prediction results of PD in ball mill experiments.

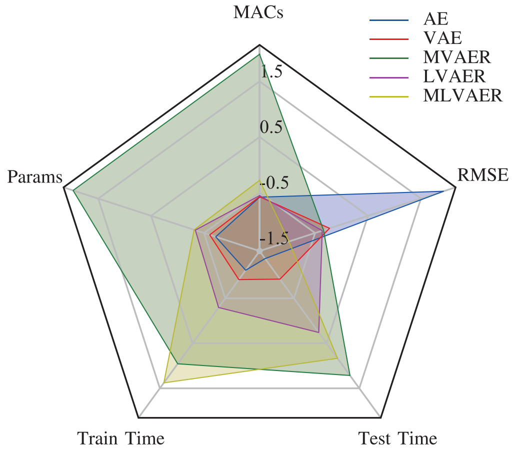


Fig. 16. Histograms of performance indicators for different algorithms in TE experiments.

In terms of testing time, the mixture algorithm, although slower than the other algorithms, runs within a reasonable range. In terms of model accuracy, the proposed algorithm has the smallest RMSE, which is significantly better than other algorithms.

7. Conclusion

Aiming at nonlinearity, multi-mode, and the data heavy-tailed problem caused by outliers contamination widely existing in the industry, this paper proposed a multi-mode industrial soft sensor method based on a mixture Laplace variational auto-encoder. The method introduces a type-II multivariate Laplace distribution for robust modeling of process noise containing outliers. Each marginal distribution of the type-II distribution can have different heavy tails so that the model based on the assumption of this distribution has higher degrees of freedom than that based on the assumption of the traditional multivariate Laplace distribution with the same network structure. Extending it to the mixture form can cope with more complex data distribution scenarios in the industry. Compared with the method based on Gaussian distribution, the proposed method not only breaks the limitation of single-peak distribution, but also is less susceptible to the interference of outliers, and can more effectively extract the potential features of complex multi-mode data affected by outliers. The experiments show that the proposed method can provide better prediction results compared with other methods.

As a deep probabilistic learning method, the proposed method still has much room for research in dealing with dynamic problems, missing data problems, and semi-supervised problems with insufficient labels, which are common in industry. In addition, the method proposed in this paper is a universal framework, and its application is not limited to the field of soft sensors. For example, changing the neural network section to Convolutional Neural Network (CNN) can be used for processing computer vision or image recognition tasks, while changing it to Long Short Term Memory Network (LSTM) can be used for time series prediction tasks. Due to the fact that the method proposed belongs to the generative model, it can theoretically also be used for generative tasks. Therefore, the method still needs further exploration and research in a wider range of applications.

CRediT authorship contribution statement

Tianming Zhang: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Gaowei Yan:** Writing – review & editing, Supervision, Resources, Funding acquisition, Conceptualization. **Rong Li:** Visualization, Investigation. **Shuyi Xiao:** Supervision, Resources. **Yusong Pang:** Validation, Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

The authors are grateful for the financial support from the National Natural Science Foundation of China (61973226, 62003233), the Gemeng Science and Technology Innovation Foundation Project (2022-05), the Shanxi Province Major Special Program of Science and Technology "Unveiling and Commanding" Project (202201090301013). and Shanxi Province Science Foundation for Youths, China (2022030212 22101).

References

[1] L. Zhou, Y. Wang, Z. Ge, Multi-rate principal component regression model for soft sensor application in industrial processes, Sci. China (Inf. Sci.) 63 (230–232) (2020) http://dx.doi.org/10.1007/s11432-018-9624-8.

[2] Z.X. Wang, Q.P. He, J. Wang, Comparison of variable selection methods for PLS-based soft sensor modeling, J. Process Control 26 (2015) 56–72, http://dx.doi.org/10.1016/j.jprocont.2015.01.003.

[3] T. Zhang, G. Yan, M. Ren, L. Cheng, R. Li, G. Xie, Dynamic transfer soft sensor for concept drift adaptation, J. Process Control 123 (2023) 50–63, http://dx.doi.org/10.1016/j.jprocont.2023.01.012.

[4] Z. Zhao, G. Yan, M. Ren, L. Cheng, Z. Zhu, Y. Pang, Dynamic transfer partial least squares for domain adaptive regression, J. Process Control 118 (2022) 55–68, http://dx.doi.org/10.1016/j.jprocont.2022.08.011.

[5] H. Kaneko, K. Funatsu, Adaptive soft sensor based on online support vector regression and Bayesian ensemble learning for various states in chemical plants, Chemometr. Intell. Lab. Syst. 137 (2014) 57–66, http://dx.doi.org/10.1016/j.chemolab.2014.06.008.

[6] Z. Li, H. Jin, S. Dong, B. Qian, B. Yang, X. Chen, Semi-supervised ensemble support vector regression based soft sensor for key quality variable estimation of nonlinear industrial processes with limited labeled data, Chem. Eng. Res. Des. 179 (2022) 510–526, http://dx.doi.org/10.1016/j.cherd.2022.01.026.

[7] T. Zhang, G. Yan, R. Li, S. Xiao, M. Ren, L. Cheng, An online transfer kernel recursive algorithm for soft sensor modeling with variable working conditions, Control Eng. Pract. 141 (2023) 105726, http://dx.doi.org/10.1016/j.conengprac.2023.105726.

[8] Q. Sun, Z. Ge, A survey on deep learning for data-driven soft sensors, IEEE Trans. Ind. Inform. 17 (9) (2021) 5853–5866, http://dx.doi.org/10.1109/TII.2021.3053128.

[9] A. Rohani Bastami, A. Aasi, H.A. Arghand, Estimation of remaining useful life of rolling element bearings using wavelet packet decomposition and artificial neural network, Iran. J. Sci. Technol. Trans. Electr. Eng. 43 (1) (2019) 233–245, http://dx.doi.org/10.1007/s40998-018-0108-y.

[10] R. Tabatabaei, A. Aasi, S.M. Jafari, Experimental investigation of the diagnosis of angular contact ball bearings using acoustic emission method and empirical mode decomposition, Adv. Tribol. 2020 (2020) 8231752, http://dx.doi.org/10.1155/2020/8231752.

[11] A. Aasi, R. Tabatabaei, E. Aasi, S.M. Jafari, Experimental investigation on time-domain features in the diagnosis of rolling element bearings by acoustic emission, J. Vib. Control 28 (19–20) (2022) 2585–2595, http://dx.doi.org/10.1177/10775463211016130.

[12] Z. Ge, F. Gao, Z. Song, Mixture probabilistic PCR model for soft sensing of multimode processes, Chemometr. Intell. Lab. Syst. 105 (1) (2011) 91–105, http://dx.doi.org/10.1016/j.chemolab.2010.11.004.

[13] M.G. Gustafsson, A probabilistic derivation of the partial least-squares algorithm, J. Chem. Inf. Comput. Sci. 41 (2) (2001) 288–294, http://dx.doi.org/10.1021/ci0003909.

[14] S. Li, J. Gao, J.O. Nyagilo, D.P. Dave, Probabilistic partial least square regression: A robust model for quantitative analysis of Raman spectroscopy data, in: 2011 IEEE International Conference on Bioinformatics and Biomedicine, 2011, pp. 526–531, http://dx.doi.org/10.1109/BIBM.2011.94.

[15] J. Zheng, Z. Song, Z. Ge, Probabilistic learning of partial least squares regression model: Theory and industrial applications, Chemometr. Intell. Lab. Syst. 158 (2016) 80–90, http://dx.doi.org/10.1016/j.chemolab.2016.08.014.

[16] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, CoRR abs/1312.6114v11, 2022, http://dx.doi.org/10.48550/arXiv.1312.6114.

[17] B. Shen, L. Yao, Z. Ge, Nonlinear probabilistic latent variable regression models for soft sensor application: From shallow to deep structure, Control Eng. Pract. 94 (2020) 104198, http://dx.doi.org/10.1016/j.conengprac.2019.104198.

[18] Z. Chai, C. Zhao, B. Huang, H. Chen, A deep probabilistic transfer learning framework for soft sensor modeling with missing data, IEEE Trans. Neural Netw. Learn. Syst. 33 (12) (2022) 7598–7609, http://dx.doi.org/10.1109/TNNLS.2021.3085869.

[19] X. Yuan, Z. Ge, Z. Song, Soft sensor model development in multiphase/multimode processes based on Gaussian mixture regression, Chemometr. Intell. Lab. Syst. 138 (2014) 97–109, http://dx.doi.org/10.1016/j.chemolab.2014.07.013.

[20] X. Zhang, C. Song, J. Zhao, D. Xia, Gaussian mixture continuously adaptive regression for multimode processes soft sensing under time-varying virtual drift, J. Process Control 124 (2023) 1–13, http://dx.doi.org/10.1016/j.jprocont.2023.02.003.

[21] D. Li, Z. Song, A novel incremental Gaussian mixture regression and its application for time-varying multimodal process quality prediction, in: 2020 IEEE 9th Data Driven Control and Learning Systems Conference, DDCLS, 2020, pp. 645–650, http://dx.doi.org/10.1109/DDCLS49620.2020.9275082.

[22] L. Cui, B. Shen, Z. Ge, A mixture variational autoencoder regression model for soft sensor application, Acta Automat. Sinica 48 (2) (2022) 398–407, http://dx.doi.org/10.16383/j.aas.c210035.

[23] X. Zhang, C. Song, J. Zhao, Z. Xu, Deep Gaussian mixture adaptive network for robust soft sensor modeling with a closed-loop calibration mechanism, Eng. Appl. Artif. Intell. 122 (2023) 106124, http://dx.doi.org/10.1016/j.engappai.2023.106124.

[24] H. Kodamana, B. Huang, R. Ranjan, Y. Zhao, R. Tan, N. Sammaknejad, Approaches to robust process identification: A review and tutorial of probabilistic methods, J. Process Control 66 (2018) 68–83, http://dx.doi.org/10.1016/j.jprocont.2018.02.011.

[25] J. Zhu, Z. Ge, Z. Song, F. Gao, Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data, Annu. Rev. Control 46 (2018) 107–133, http://dx.doi.org/10.1016/j.arcontrol.2018.09.003.

[26] D. Peel, G.J. McLachlan, Robust mixture modelling using the t distribution, Stat. Comput. 10 (4) (2000) 339–348, http://dx.doi.org/10.1023/A:1008981510081.

[27] J. Zhu, Z. Ge, Z. Song, Robust modeling of mixture probabilistic principal component analysis and process monitoring application, AIChE J. 60 (6) (2014) 2143–2157, http://dx.doi.org/10.1002/aic.14419.

[28] J. Wang, W. Shao, Z. Song, Robust inferential sensor development based on variational Bayesian Student's-t mixture regression, Neurocomputing 369 (2019) 11–28, http://dx.doi.org/10.1016/j.neucom.2019.08.039.

[29] A. Yan, J. Guo, D. Wang, Robust stochastic configuration networks for industrial data modelling with Student's-t mixture distribution, Inform. Sci. 607 (2022) 493–505, http://dx.doi.org/10.1016/j.ins.2022.05.105.

[30] P. Zhu, X. Yang, H. Zhang, Mixture robust L1 probabilistic principal component regression and soft sensor application, Can. J. Chem. Eng. 98 (8) (2020) 1741–1756, http://dx.doi.org/10.1002/cjce.23739.

[31] X. Yang, X. Liu, C. Xu, Robust mixture probabilistic partial least squares model for soft sensing with multivariate Laplace distribution, IEEE Trans. Instrum. Meas. 70 (2021) 1–9, http://dx.doi.org/10.1109/TIM.2020.3009354.

[32] T. Eltoft, T. Kim, T.-W. Lee, On the multivariate Laplace distribution, IEEE Signal Process. Lett. 13 (5) (2006) 300–303, http://dx.doi.org/10.1109/LSP.2006.870353.

[33] C. Zhang, M.-L. Tang, T. Li, Y. Sun, G.-L. Tian, A new multivariate Laplace distribution based on the mixture of normal distributions, Sci. Sin. Math. 50 (5) (2020) 711–728, http://dx.doi.org/10.1360/N012019-00141.

[34] N. Lawrence Ricker, Decentralized control of the Tennessee eastman challenge process, J. Process Control 6 (4) (1996) 205–221, http://dx.doi.org/10.1016/0959-1524(96)00031-5.