Process Safety and Environmental Protection 150 (2021) 510–521

Contents lists available at ScienceDirect

Process Safety and Environmental Protection

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

Deeppipe: A semi-supervised learning for operating condition recognition of multi-product pipelines

Jianqin Zheng[a], Jian Du[a], Yongtu Liang[a,*], Qi Liao[a,*], Zhengbing Li[a], Haoran Zhang[b], Yi Wu[c]

[a] National Engineering Laboratory for Pipeline Safety/MOE Key Laboratory of Petroleum Engineering/Beijing Key Laboratory of Urban Oil and Gas Distribution Technology, China University of Petroleum-Beijing, Fuxue Road No. 18, Changping District, Beijing, 102249, PR China
[b] Center for Spatial Information Science, The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa-shi, Chiba, 277-8568, Japan
[c] University of Edinburgh Business School, 29 Buccleuch Place, Edinburgh, EH8 9JS, UK

ARTICLE INFO

Article history:
Received 5 January 2021
Received in revised form 20 April 2021
Accepted 21 April 2021
Available online 24 April 2021

Keywords:
Pipeline
Operating condition recognition
Semi-supervised learning
Sensitivity analysis

ABSTRACT

Intelligent operating monitoring of pipelines helps to detect anomalies in time to ensure pipeline safe, reducing potential risk. However, the operating conditions of the multi-product pipeline change frequently, and the recognition and monitoring by on-site personnel are easy to cause misjudgment, so the operating conditions of the pipeline cannot be accurately recognized. Noticeably, operating condition recognition is an important part of pipeline safety and risk management. Although ample operating data are stored in SCADA system, these data are lack of corresponding condition labels, making it hard to be mined. In this work, a semi-supervised learning for operating condition recognition is proposed to overcome aforementioned issues. Firstly, the operating parameters of each station are preprocessed and collected to construct into data matrices to overcome transient disturbance considering the pipeline space characteristics and time series of the operating data. Then stacked autoencoder (SAE) is used to pre-train the network parameters of multi-layer neural network (MLNN) based on a large amount of unlabeled operating data. After that, MLNN is fine-tuned based on a small amount of labeled data annotated by referring to the operation log. To verify the effectiveness of the semi-supervised learning, a real multi-product pipeline is taken as an example for operating condition recognition. The accuracy, precision, recall and F1 score is 95 %, 95 %, 80 % and 80 %, respectively. Results show that the condition recognition accuracy of the proposed model is better than other machine learning models. Finally, the sensitivity analysis is conducted to illustrate the importance of SAE in this classification model.

1. Introduction

1.1. Background

The multi-product pipeline has been the main mode (Cui et al., 2020) to transport refined products from refineries to consumer markets (Zhang et al., 2018). The typical characteristic of multi-product pipelines is that different types of refined oil are transported in the same pipeline based on market demand, resulting in troublesome daily operations such as batch tracking, contamination cut, and following the operating schedule. Consequently, there are many complex operating conditions in multi-product pipeline (Zheng et al., 2020). And the actual status of pipeline is reflected by operating data, which is the reason that the accurate operating condition recognition model is required to ensure safe operation. The rapid development of multi-product pipelines brings economic benefits and convenience but meanwhile increases risk and accidents (Halim et al., 2020) with the more complex and large-scale process (Pontiggia et al., 2020). As an important safety problem, studies on anomaly detection have been widely conducted within different fields of the industrial process (Chetouani, 2014). For the pipeline industry, intelligence is not only needed in production but also process monitoring. Process monitoring plays a significant role in pipeline safety management, and

Abbreviations: AE, autoencoder; BP, Back-propagation; DT, decision tree; GB, gradient boosting; GMM, gaussian mixture model; KS, Kolmogorov-Smirnov; KNN, K-Nearest neighbor; LSTM, long short-term memory; MLNN, multilayer neural network; MRI, Magnetic Resonance Imaging; RF, random forest; SVM, support vector machine; SCADA, Supervisory Control and Data Acquisition; SAE, stacked autoencoder; XGB, extreme gradient boosting.

* Corresponding authors.
E-mail addresses: zhengjianqin_cup@163.com (J. Zheng), liangyt21st@163.com (Y. Liang), qliao_cup@outlook.com (Q. Liao).

https://doi.org/10.1016/j.psep.2021.04.031
0957-5820/

Nomenclature

A	initial station
B	middle station
b	bias of network
C	final station
f	The nonlinear transfer function of encoder
g	The nonlinear transfer function of decoder
h	The hidden representation of encoder
k	The total number of class i
m	The number of training samples
N_{hl}	The number of hidden layers
$N_n^{(i)}$	The number of neurons in each layer
P	pressure
Q	flowrate
$R^{(i)}$	The number range of nodes in each layer
w	weight of network
x_i	The input value, representing pipeline operating condition data matrices, the origin output values of each neuron
x_{input}	the normalized values
x_{min}	the minimum values of the origin data
x_{max}	the maximum values of the origin data
y	The output value, representing the output of stacked autoencoder, the output of ReLU function
z_i	The output, representing the reconstructed result of autoencoder, the output of SoftMax function
ρ	density of refined oil
θ	The parameters of each layer in autoencoder

operating condition recognition is a vital step of it. The recognition of the pipeline operating conditions can enable the operator to have a more comprehensive understanding of the pipeline operating status, and provide a guarantee for the safe operation of the pipeline. Intelligent condition recognition is a promising way to release the contribution from human labor and automatically recognize the operating state of the pipelines. Especially, when the recognized condition is not the expected one, it is likely that the pipeline safety is threatened, and emergency investigation is needed immediately (Zheng et al., 2021).

At present, the operating management of the long-distance oil pipeline mainly relies on the experience of the operators and combines the data trend of the real-time data to analyze and recognize the operating condition. However, due to the exponential growth of monitoring points, human monitoring of pipelines becomes more and more difficult. Besides, the condition switch is frequent and the hydraulic change is complicated for the pipeline. Inaccurate identification can not dynamically track the operation status of the pipeline, which leads to the failure to timely and effectively discover the potential risks of the pipeline. When performing on-site operations, if the human operating conditions are inconsistent with the model recognition, it may be that the model recognition is inaccurate, or there may be other potential operating conditions. When the model recognition accuracy is high, it is more likely to be other potential operating conditions like leakage, or other abnormal conditions.

Thanks to complete computer hardware equipment and databases with strong storage capacity, pipeline operating data can be extracted and saved in real-time. However, the operating data have complex characteristics such as nonlinearity, high dimensionality, and time-series. Furthermore, there are no corresponding condition labels for these pipeline operating data. Therefore, although a large amount of pipeline operating data is stored, there are few studies on using these data for operating condition recognition.

The difficulty of operating condition recognition for multi-product pipeline can be concluded as follows:

(1) The complexity of the pipeline topology and the complicated hydraulic caused by multi-batch transportation make it more difficult to recognize the operating conditions.
(2) Due to the nonlinearity and high dimensionality, it is difficult to effectively mine the pipeline operation data.
(3) Operating condition recognition is a classification problem, but the unlabeled pipeline data cannot be used for classification model training effectively.

1.2. Literature review

At present, studies on pipeline operating condition recognition were conducted by some scholars. Zhang et al. (2009) used the momentum term gradient descent algorithm and adaptive learning rate optimized back-propagation (BP) algorithm to identify the pipeline operating condition. In order to overcome the problem of false leak detection, Mandal et al.(2012) combined rough set theory and support vector machine (SVM) to propose a novel leak detection scheme. In their work, supervised learning models such as BP and SVM are utilized for condition recognition, which is based on insufficient labeled operating data. Therefore, the precision of classifier is limited as well as the time-consuming and ineffective data labeling. For this reason, studies on unsupervised condition recognition based on plenty of unlabeled operating data were conducted. Ye et al. (2009) denoised pipeline pressure data by wavelet transform, extracted time-domain features, and then identified pipeline operating conditions based on the fuzzy c-means algorithm. Rai and Kim (2021) proposed a health index-oriented approach based on multiscale analysis, Kolmogorov-Smirnov (KS) test, and Gaussian mixture model (GMM) to determine the leakage situation in pipelines. With regards to the complicated changes of operating conditions on oil pipelines, a simulated annealing K-means algorithm was proposed by Ye et al. (2009) to cluster oil pipeline operating conditions. Feature extraction and feature compressing were conducted by Liang and Zhang (2012) based on wavelet packet analysis and principal component analysis respectively, the gradient and slope turn rejection was presented for oil pipeline leakage detection. da Silva et al. (2005) used fuzzy system to classify running mode and identify transients process, then a methodology based on a combination of clustering and classification tools was proposed to detect pipeline leakage accurately.

Although the above unsupervised recognition methods perform well, the feature extraction capability of these models still needs to be improved. Furthermore, some other methods such as the two-stage decision scheme and dynamic pressure transmitter were utilized to detect pipeline leakage. Based on the chaotic characteristics, Zhang et al. (2009) proposed a method to identify the normal state and leakage state of the pipeline by using the pipeline signal of the dynamic pressure transmitter. Liu et al. (2019) introduced Markov feature and the two-stage decision scheme to construct a novel leakage detection model. The short-term and long-term detection models were correctly selected according to switching rule to recognize pipeline conditions precisely. Zhang et al. (2014) designed a dynamic pressure transmitter to obtain dynamic pressure signals along the pipeline and then detected pipeline leakage based on extracted wavelet packet entropy of the signals.

Though the above-mentioned methods make great progress in pipeline operating condition recognition, the limitations of the existing studies can be concluded as follows.

(1) The recognition of pipeline operating conditions is mainly carried out for leaking conditions, and there are few studies on establishing pipeline operating condition recognition models.
(2) The supervised learning-based methods can not achieve satisfactory accuracy due to the lack of a large amount of labeled data.
(3) The unsupervised learning-based methods can realize data clustering, but the corresponding category is unknown.

During pipeline transportation, operating data will be continuously generated. It is impossible to label these operating data manually, which severely reduces work efficiency and has lag. Considering that there is a large amount of historical operating data in the Supervisory Control and Data Acquisition (SCADA) system, but it lacks annotations. It is acceptable that a small amount of operating data is marked based on the operation log, and a sample library of the common operating condition is established. Due to the insufficient number of samples for each operating condition in the sample library, if supervised learning is directly used for classification training, the recognition accuracy is not satisfied. Thereby, how to make full use of the few labeled data and the plenty of unlabeled data for pipeline operating condition recognition is the key research of this work.

With the development of artificial intelligence algorithms and data mining technologies, deep learning enjoys its popularization in state monitoring. Machine learning methods can be mainly partitioned into supervised ones and unsupervised ones. Supervised learning requires a large amount of labeled data for training, and it is inappropriate for pipeline industrial applications due to the lack of labeled historical data in the real situation. Annotating pipeline data are extremely expensive and time-consuming. Due to the drawbacks of supervised learning, unsupervised learning which can cope with unlabeled data should be considered to mine the pipeline operating data. In terms of supervised learning, the system tries to construct a mapping relationship between input and output via learning given inputs and corresponding outputs and targets. As a typical supervised algorithm, multi-layer neural network (MLNN) shows excellent performance in classification problems, such as image recognition (Yang and Ding, 2020). Conversely, unsupervised learning acquires unlabeled data and tries to extract the representative features among each set of data samples. As an unsupervised feature extraction algorithm, stacked autoencoder (SAE) outperforms in feature extraction (Zheng and Zhao, 2020). Li et al. (2021) constructed SAE to extract the deep features of the variables in each operating unit to represent the essential structure of the unit, then two constructed statistics were utilized to detect the faults in local units.

Considering the physical space characteristics of the pipeline, this work analyzes and collects the operating parameters of each station. At the same time, considering the time series characteristics of the pipeline operating, operating data matrices are constructed based on the SCADA system to overcome transient disturbance of a single moment. Then based on powerful feature compression capabilities during the encoding process, SAE is introduced to pre-train the network parameters of MLNN. After pre-training, MLNN is adjusted based on the labeled sample library to complete the training of the operating condition classification model. Finally, the hybrid model, namely SAE-MLNN is established to recognize the operating conditions of the multi-product pipeline, as shown in Fig. 1.

1.3. Contributions of this work

The contributions of this work can be listed as follows:

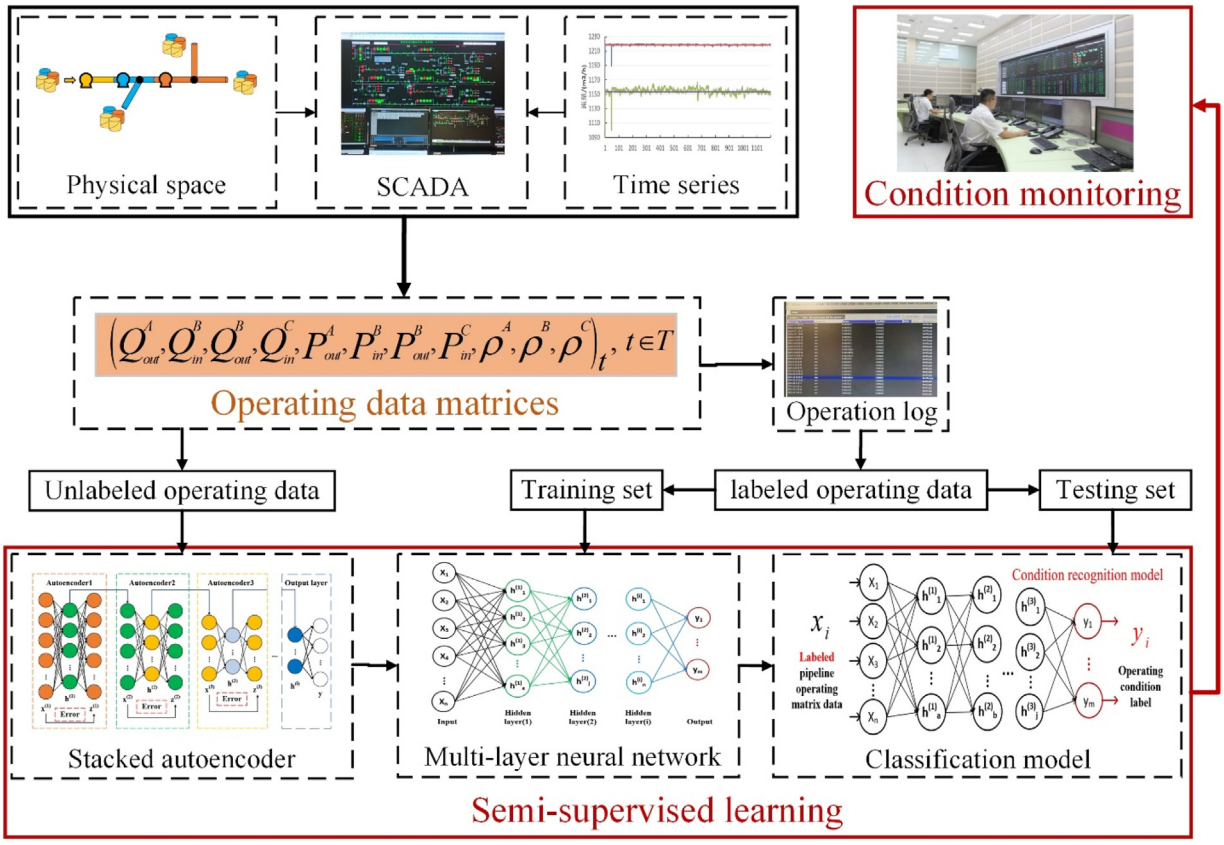


Fig. 1. The main framework of this research.

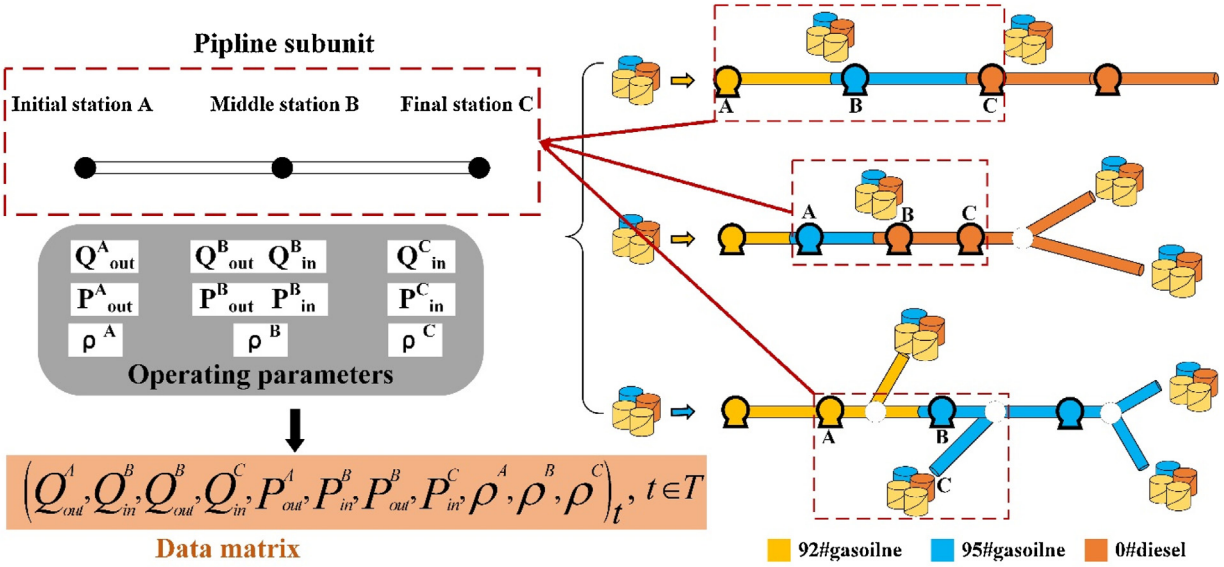


Fig. 2. The construction process of pipeline operating data matrix.

(1) Considering the physical space characteristics of the pipeline and the operating time series characteristics, data matrices of operating conditions are constructed based on the SCADA system.
(2) Based on supervised learning and unsupervised learning, a hybrid method, namely, semi-supervised learning is proposed to recognize the operating conditions of the multi-product pipeline.
(3) Taking a multi-product pipeline in China as an example, the effectiveness and superiority of the proposed method are verified.
(4) Sensitivity analysis is conducted on the network structure of the proposed method, illustrating that coupling SAE can improve recognition accuracy.

The remainder of this paper is organized as follows. In section 2, the basic operation process of the multi-product pipeline is introduced. The semi-supervised learning combined SAE and MLNN will be elaborated in detail in section 3. In section 4, a case study is taken as an example to verify the effectiveness of the proposed model, and sensitivity analysis of the model network is conducted. Section 5 provides a conclusion and future work.

2. Problem description

Refined oil is strategic energy that influences the national economy and social stability. As the most significant way for oil transportation, the normal operational pipeline has great importance in supplying oil safely and reliably (Zhou et al., 2020). Because of the kinds of regulating commands, the operation of the multi-product pipeline can occasionally generate a series of conditions consisting of stoppage, restart, distribution, increment, internal pump regulation of single station or multiple stations, as well as the switch of oil types. These operating commands are checked and executed by dispatchers, bringing challenges to workload and caution day after day. Also, these normal conditions can easily cause false alarms, misleading leakage detection measures and maintenance. As a result, it is significant to recognize the operating condition to monitor the pipeline state.

The operating data sets of the multi-product pipeline can be obtained from SCADA system. Considering that the pipeline operating data possesses physical space characteristics, regardless of the topology of the pipeline network, it can be divided into several subunits and each subunit contains three stations, namely, initial station A, middle station B, and final station C, as shown in Fig. 2. There are a total of 11 operating parameters of each subunit that should be sorted out, including outlet pressure and flowrate of station A, inlet and outlet pressure and flowrate of station B, inlet pressure and flowrate of station C, and density of the refined product in each station (Zhou et al., 2019a). To overcome the influence of transient disturbance, the time series characteristics of pipeline operating are considered, and the operating data in an assumed period (1 min in this work) is constructed as a two-dimension matrix. Every 5 s of operating data were recorded by SCADA system, so the shape of these matrices is 12×11. In fact, if the period is short, the data matrix can not reflect the sequence correlation; if the period is long, the data matrix can not meet the needs of real-time dynamic monitoring of operating conditions.

With these constructed matrices, how to recognize the pipeline operating condition at any moment? Obviously, this can be treated as a multi-classification task due to multiple operating conditions. Besides, the training of this multi-classification task is only with input (operating data matrices) but lack of output (corresponding condition). With the help of on-site operation logs, a small amount of operating data can be annotated with corresponding conditions, which is acceptable with limited time-consuming. This work is to mine these few labeled and a large amount of unlabeled operating data for operating condition recognition. This multi-classification task will be solved by semi-supervised learning mentioned in section 3.

3. Semi-supervised learning model

In this section, the semi-supervised learning model will be constructed, in which the features of pipeline operating condition data matrices are extracted to achieve condition recognition accurately. As a traditional machine learning algorithm, MLNN requires ample data to acquire satisfied recognition results (Jiang et al., 2020). However, there are few labeled pipeline operating data on-site. Considering that SAE performs well in extracting representative features only with unlabeled pipeline operating data which are very sufficient in SCADA system, semi-supervised learning is leveraged to overcome this drawback. In brief, SAE is utilized to pre-train the parameters of MLNN by extracting the represen-

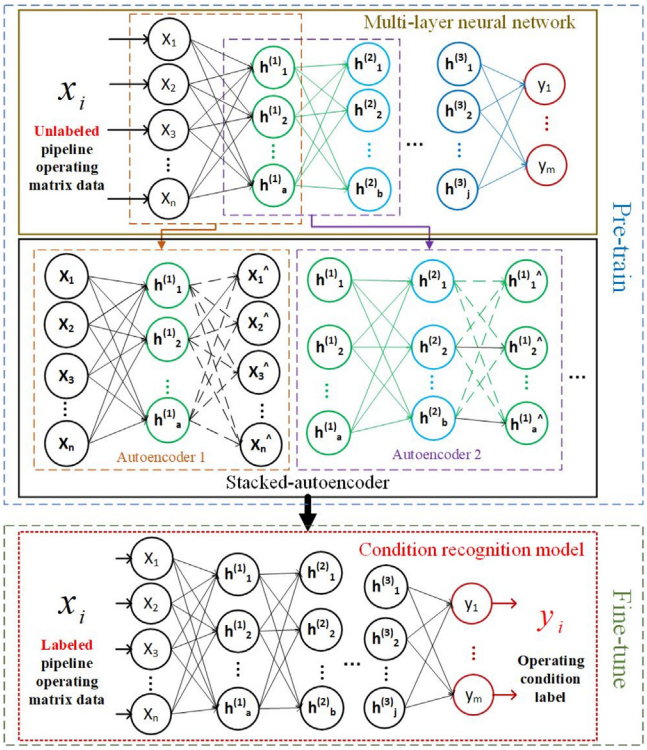


Fig. 3. The basic architecture of the semi-supervised learning model.

tative features of the pipeline operating data based on a large amount of unlabeled data. Then a small amount of labeled data is used to fine-tune the network parameters to converge the training error in the cause of better recognition performance. Finally, the semi-supervised learning model for condition recognition of multi-pipelines is established.

3.1. Model components

The basic process of the semi-supervised learning model is depicted in Fig. 3. It mainly consists of two parts, namely, pre-training and fine-tuning. In this work, MLNN is employed in the operating condition recognition of multi-product pipelines due to its excellent nonlinear approximation ability. MLNN is known as the "computational models" and "universal approximators" with special characteristics including self-learning or self-adaptation and self-organization which is essentially an artificial neural network (ANN) with multiple hidden layers (Szoplik, 2015). Each layer has a certain number of neurons that are employed to obtain the summation of the weighted inputs and then activate the summation to generate the outputs, separately (Jiang et al., 2020). The input value is forward transferred and each layer accepts signals from neurons in the previous layer and passes its outputs to the next consecutive layer.

MLNN is composed of an input layer, an output layer, and several hidden layers. In order to make the model universally applicable, specific network parameters need to be determined through specific case debugging. Based on the pipeline operating condition data matrix dimensions, the number of the neurons in the input layer is determined as 132 (matrix shape was 12×11). The neuron number of the output layer is determined by the types of pipeline operating conditions. In this work, 12 condition types are considered to test the proposed model. As shown in Fig. 3, how to pre-training the MLNN by SAE will be elaborated in detail.

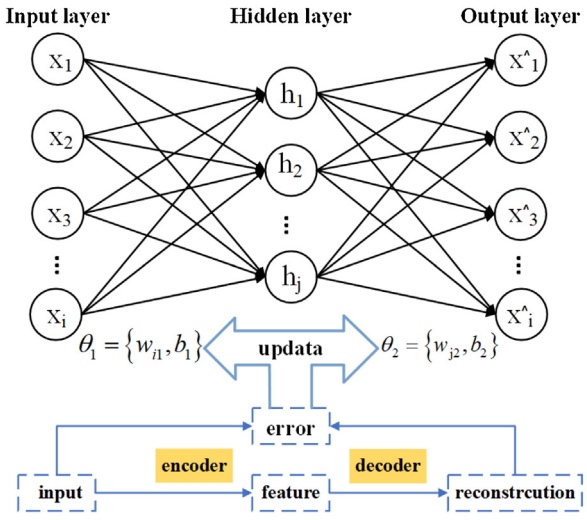


Fig. 4. The basic structure of autoencoder.

3.1.1. Constructing an AE model based on the network architecture of MLNN

To obtain the better parameters between the input layer and the first hidden layer of MLNN, assigning the neurons of these two layers to the input layer and compressed representation of SAE, respectively. In other words, the connection of the input layer and the first hidden layer of MLNN is treated as the encoder of AE 1.

3.1.2. Training the AE model

As an unsupervised learning algorithm, AE enjoys its popularization in the reduction of dimensionality and obtaining intricate features (Law and Ghosh, 2019). As depicted in Fig. 4, AE is an architecture belonging to the neural network, including basic input, hidden, and output layers. The input layers and the hidden layers are combined into an encoder, and similarly, the hidden layers and the output layers are combined into a decoder. Given the training samples of pipeline operating condition data matrices x_i, the encoder compresses x_i to a hidden representation h through nonlinear transformation. After that, the decoder aims to map h to a reconstructed output z_i via utilizing a similar transformation. The formulations of the encoding and decoding process are as follows.

$$h(x_i;\theta_1) = f(w_{i1}x_i + b_1) \tag{1}$$

$$z_i(h;\theta_2) = g[w_{j2}h + b_2] \tag{2}$$

where f and g are the nonlinear transfer function of encoder and decoder, w is the weight, and $b$is the bias (Yu and Zhang, 2020), i and j are dimension size of input layer and hidden layer, respectively.

AE attempts to recreate the output $x_i^{\wedge}$ from the compressed representation h, trying to make the output value the same as the pipeline operating condition data matrices x_i. The reconstruction error (Eq. 3) is formulated to backpropagate and update the weights w and bias b by training iteratively until *MSE* reaches the convergence condition (Saha et al., 2020). Finally, AE shows good performance in extracting representative features from data matrices x_i.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(x_i^{\wedge} - x_i)^2 \tag{3}$$

3.1.3. Updating the network parameters of MLNN by AE

After the training of AE is completed, the parameters of the encoder are assigned to the parameters between the input layer and the first hidden layer of MLNN conversely. In a word, the par-

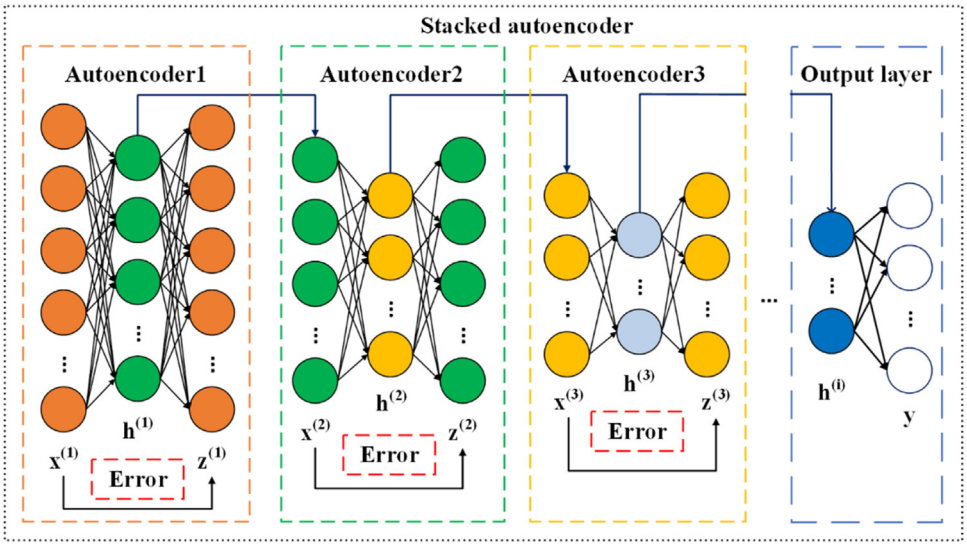


Fig. 5. The architecture of SAE.

tial initial parameters of MLNN are determined by a trained AE with customized structure.

3.1.4. Pre-training the entire network architecture except output layer of MLNN by AE based on (1)-(3)

SAE constructs deep neural architecture by successively stacking multiple single-layered autoencoders as shown in Fig. 5. In its deep architecture, the intricate features of the pipeline operating condition data x_i can be extracted by making the compressed representation of each AE act as the input of the next AE. To avoid converging to the local minimum, greedy layer-wise unsupervised pre-training is adopted to obtain the weights and bias. Similarly, AE 2 is employed to update the parameters between the first hidden layer and the second one of MLNN by assigning the connection relationship to the encoder of AE 2. After training AE 2, the corresponding network parameters of MLNN is updated again. One by one, the network parameters except those between the last hidden layer and the output layer of MLNN are pre-trained by these AEs utilizing plenty of unlabeled pipeline operating data successively. In the pre-training stage, each AE is trained individually. As a tool of feature extraction, SAE extracts the characteristic parameters, which contain significant information about time-series characteristics and physical space characteristics of pipeline operating data matrix. After the pre-training procedure is finished, the supervised fine-tune is introduced to adjust the parameters of all layers simultaneously based on the forward propagation algorithm.

3.1.5. Fine-tuning the semi-supervised learning model

Finally, MLNN is fine-tuned by supervised learning which merely acquires a small of labeled pipeline data (x_i, y_i). The parameters in each layer are adjusted in the meantime by supervised training to obtain the precise results of operating condition recognition y_i. Softmax function is used to nonlinear the final output to classify the operating condition with probability (Eq. 4) (Wang et al., 2020). In this work, we optimize the cross-entropy error of the proposed model with Adam optimizer (Kingma and Ba, 2014). The cross-entropy error J of operating condition recognition is formulated as follows.

$$p_i = \text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{4}$$

Table 1
Classification results.

	Predict as condition i	Predict as other condition
Condition i	True positive (TP)	False negative (FN)
Other condition	False positive (FP)	True negative (TN)

$$J = -\sum_{i=1}^{k} B_i \log(p_i) \tag{5}$$

Here, z_i and p_i is the origin output values of each neuron and the corresponding probability of condition i respectively, k is the number of the pipeline operating condition types, and B_i equals 1 if the condition is i.

3.2. Evaluation metrics

Before the model training, data normalization is conducted to avoid severe numerical round-off effects when there are huge differences of numerical range between input and output variables. The input and output data are normalized to values within the range of 0–1 by,

$$x_{input} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{6}$$

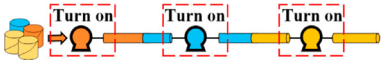
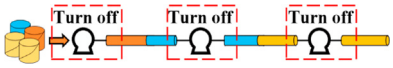
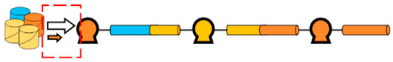
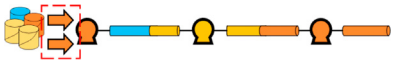
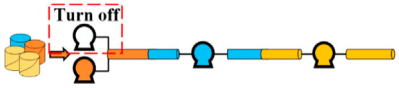
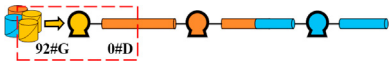
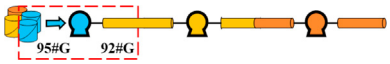
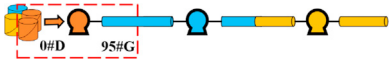
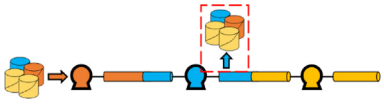
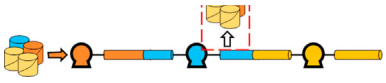
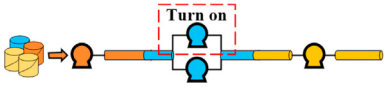
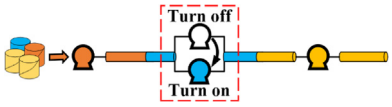
where x_{input} is the normalized value, the x_i is the value of origin data, and the $x_{\max}$, $x_{\min}$ are the maximum and minimum values of the original data, respectively.

To represent the performance of these models, the accuracy, precision, recall, and F1 score (Sokolova and Lapalme, 2009) are selected as the evaluation metrics to analyze and identify the best performing models. The evaluation metrics are introduced as Table 1.

(1) Accuracy: representing the overall effect of the classification model.

$$Accuracy = \frac{\sum_{i=1}^{k} \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}}{k} \tag{7}$$

Table 2
Pipeline operating conditions.

Label	Operating conditions	Diagrams
1	Pipeline startup	
2	Pipeline shutdown	
3	Throughput degradation at initial station	
4	Throughput increment at initial station	
5	Pump stoppage at initial station	
6	Change of oil being injected at initial station (0#D→92#G)	
7	Change of oil being injected at initial station (92#G→95#G)	
8	Change of oil being injected at initial station (95#G→0#G)	
9	delivery at middle station	
10	Middle station without distribution	
11	Pump startup at middle station	
12	Switching pump at middle station	

(2) Precision: providing information about how many the data matrices labeled as condition i which are actually true (Messner et al., 2020).

$$Precision = \frac{\sum_{i=1}^{k} \frac{TP_i}{TP_i+FP_i}}{k} \tag{8}$$

(3) Recall: indicating how many data matrices of condition i are truly recognized (He et al., 2020).

$$Recall = \frac{\sum_{i=1}^{k} \frac{TP_i}{TP_i+FN_i}}{k} \tag{9}$$

(4) F1 score: representing the effect of the classification model to recognize positive class(Zhou et al., 2019b) and being obtained by the harmonic mean of precision (Eq. 7) and recall (Eq. 8) (Chen et al., 2020).

$$F1score = \frac{2 * precision * recall}{precision + recall} \tag{10}$$

In Eqs. (7)–(10),k represents the total number of conditions i.

4. Results and discussion

This section evaluates the performance of the proposed semi-supervised learning by using a real-world case. The accuracy of the hybrid model is verified in comparison with other traditional machine learning models. Meanwhile, the sensitivity analysis of the MLNN network architecture is also conducted to evaluate the effectiveness of introducing SAE to pre-train MLNN.

4.1. Case study

In this work, a multi-product pipeline in China is taken as an example, and pipeline operating condition data are obtained based on the SCADA system from April 25, 2020 to May 8, 2020. The dataset contains flow rate, pressure and oil density of each station. In order to analyze pipeline operating conditions as many as possible, 12 kinds of basic operating conditions are employed for the case study. In a total of 12 operating conditions, every condition contains 100 sets of data. In order to train the hybrid model, 10 sets of data are selected from each operating condition for data annotation manually. In these 120 sets of data samples, 100 sets of data are selected evenly from each condition to fine-tune the hybrid model,

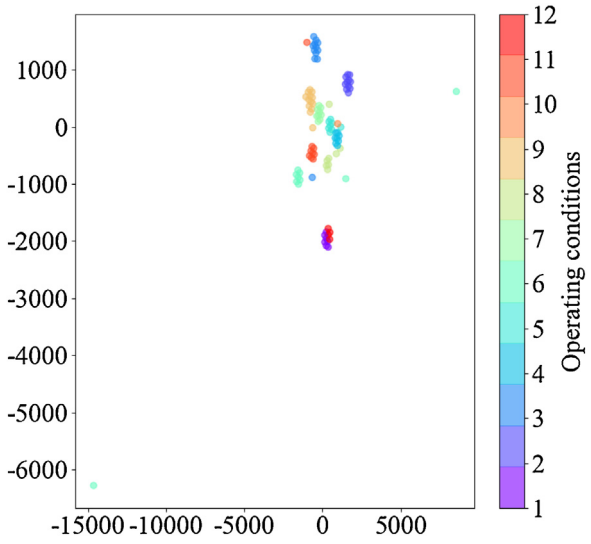


Fig. 6. The classification results of labeled data based on t-SNE.

Table 3
Recognition accuracy of each condition for SAE-MLNN.

Experiments	1	2	3	4	5
Accuracy/%	95	90	95	95	100
Average accuracy/%	95				

and other 20 sets of data are utilized to test the performance of these classification models. The remaining 1080 sets of unlabeled data are used to pre-train the hybrid model. The training and testing procedure of the proposed model is conducted based on the operating data matrices and the hybrid model which are described in sections 2 and 3, respectively. Besides, the other classification models are applied for comparison to embody the superiority of semi-supervised learning in pipeline operating condition recognition.

The extracted pipeline operating conditions are depicted in Table 2. Data splitting for cross-validation is necessary to be conducted to ensure that models are robust. Before data splitting, the data matrices are shuffled firstly. In order to visualize the classification results of labeled data, t-SNE is utilized for data clustering. As shown in Fig. 6, samples that do not belong to the same category are clustered together, which indicate that the clustering effect of t-SNE is unsatisfied. Consequently, classification results of the unlabeled data will not meet the need of accurate recognition due to the lack of the label information as well.

The proposed hybrid model and other classification models are executed on TensorFlow, and the programming environment is Python. Based on the "trial and error method" and the "empirical formula method"(Zhang et al., 2021) as well as the practical problem of pipeline operating condition recognition, the number of the hidden layers is assumed as 3, and their neuron number is set as 200, 100, and 50 respectively. The setting of convergence condition depends on the specific problem, the network will stop training when the MSE of AE and MLNN tend to be stable in this paper. The hybrid model is trained and its recognition effect is verified. Recognition results of 20 sets of testing data matrices are shown in Table 3. Several experiments are conducted, the train data and test data are random sampling to conduct cross validation. The average accuracy of the hybrid model reaches 95 %.

Subsequently, to illustrate the superiority of the hybrid model in operating condition recognition, the proposed model has been compared with six other traditional machine learning models such as MLNN, decision tree (DT), random forest (RF), etc. In this work, the hybrid model and MLNN are constructed with the same network architecture to reflect the superiority of using SAE to pre-train MLNN. DT divides the feature space and calculates the output value based on feature importance (Sabah et al., 2019). RF determines the output value by constructing multiple decision trees and comprehensively considering multiple results (Ao et al., 2019). K-Nearest neighbor (KNN) specifies the class of each test data according to the main class of the k points closest to the training data (Arian et al., 2020). Gradient boosting (GB) is a classification model based on the principle of combining multiple classification trees (Jian and Huaguang, 2004). Extreme gradient boosting (XGB) uses grid search cross-validation method to optimize model parameters (Pathy and Balasubramanian, 2020).

The classification performances of models are evaluated based on the four proposed evaluation metrics, which have been explained in Section 3.4. It is can be observed from Fig. 7 that the four metrics of SAE-MLNN are higher than any other classifiers. For the neural networks, the accuracy and precision of SAE-MLNN can reach 95 %, showing better performance than MLNN. The reason is that the classification ability of SAE-MLNN is optimized by introducing SAE to pre-train MLNN firstly. For the typical classification algorithms (KNN and DT), the testing results of KNN show better effectiveness with 85 % accuracy and precision while DT has the worst performance in recall with 33.3 %. For the ensemble models (RF, GB, XGB), the four metrics of RF are higher than those of all rival methods. The difference between the performance of GB and XGB is not quite. GB has a higher 55 % accuracy and precision than XGB. Conversely, XGB performs better in recall and F1 score with 43.3 % and 45 % than GB, respectively. The aforementioned analysis proves that the proposed hybrid model possesses the strongest ability of pipeline operating condition recognition on-site.

4.2. Sensitivity analysis

Generally, the number of hidden layers and the number of neurons in each hidden layer affect the fitting ability of the model for complex nonlinear classification problems. Although the hybrid model performs better on certain network architecture, the changes in network structure are out of consideration. As a consequence, the sensitivity analysis for the number of hidden layers (N_{hl}) and the number of neurons ($N_n^{(i)}$) is conducted in this section to verify that the proposed model is robust.

Increasing the number of hidden layers can achieve better feature extraction for the complex nonlinear pipeline operating matrix data, but at the same time, it easily leads to overfitting and increase the training time of the models. In the meanwhile, the number of neurons in each layer can also affect the model performance. To prove that the proposed semi-supervised learning model performs better than MLNN regardless of how the neural network structure transforms, different network architectures with different numbers of hidden layers and different numbers of neurons are constructed for the proposed model and MLNN.

The process of sensitivity analysis is shown in Table 4. Generally, an increase of hidden layers is able to improve recognition performance. However, when the number of hidden layers comes to a certain threshold value (N), increasing the number of hidden layers no longer makes any sense to the performance improvement of these models. In this work, it is found that the performance of the proposed network is not improved anymore when the number of hidden layers is more than 4 through multiple experiments. Hence, the threshold value N is set as 4. Furthermore, in order to reflect the higher robustness and stability of semi-supervised model compared to MLNN, multiple groups (T) with random neurons in each hidden layer ($R^{(i)}$) are conducted. The number range ($R^{(i)}$) of nodes in each layer was determined based on experience interval and the number of groups is set as 10.

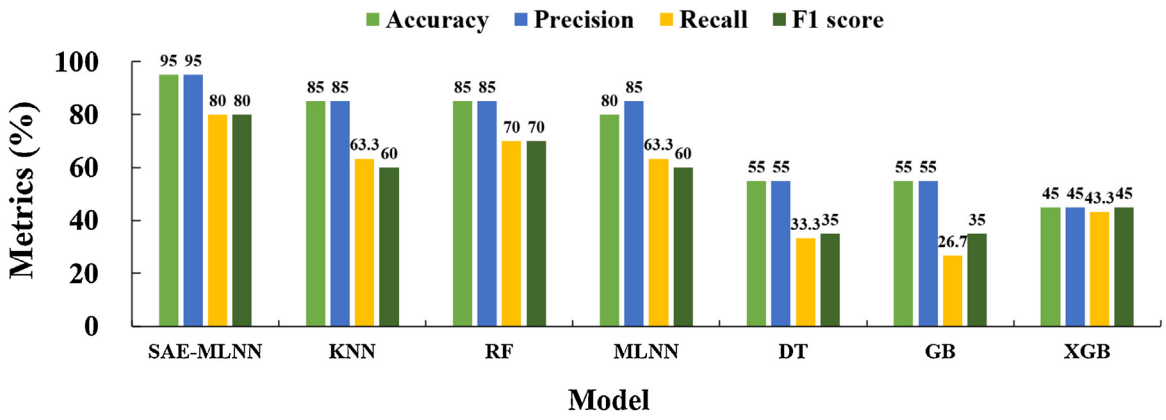


Fig. 7. Comparison results of these different models.

Table 4
The process of sensitivity analysis.

Training process: choosing Adaptive moment estimation as the training algorithm of the hybrid model and MLNN

n=1

while n≤N:

n=n+1

t=1

Determine the number range ($R^{(i)}$) of nodes in each layer

while t≤T:

t=t+1

The number of hidden layers (N_{hl}) is determined as n

For each layer, set the number of neurons as $N_n^{(i)}$=random($R^{(i)}$)

Sample a batch of the input data matrices $\{x_1, x_2, \ldots, x_m\}$ from the labeled samples

Obtain the corresponding labels $\{y_1, y_2, \ldots, y_m\}$ recognized by the proposed model and MLNN

Calculate the evaluation metrics (accuracy, precision, recall, and F1 score) of MLNN and the proposed model

end while

end while

The comparison results of different network structures are presented in Appendix A, and the best values are highlighted in bold. It is obvious that the semi-supervised learning model possesses better recognition performance and robustness than MLNN. Moreover, MLNN represents intense volatility and instability during multiple experiments. The instability of MLNN is most severe when the number of hidden layers is 4 particularly.

Fig. 8 shows the holistic recognition effect of semi-supervised learning (SAE-MLNN) and MLNN. It can be seen that SAE-MLNN outperforms MLNN in each evaluation metric. At the same time, no matter which metric, the increase in the number of hidden layers leads to a decrease in model performance for both SAE-MLNN and MLNN. The MLNN model merely obtains 75 % accuracy when there are four hidden layers for neural networks, indicating that the increasingly complex network architecture increases the difficulty

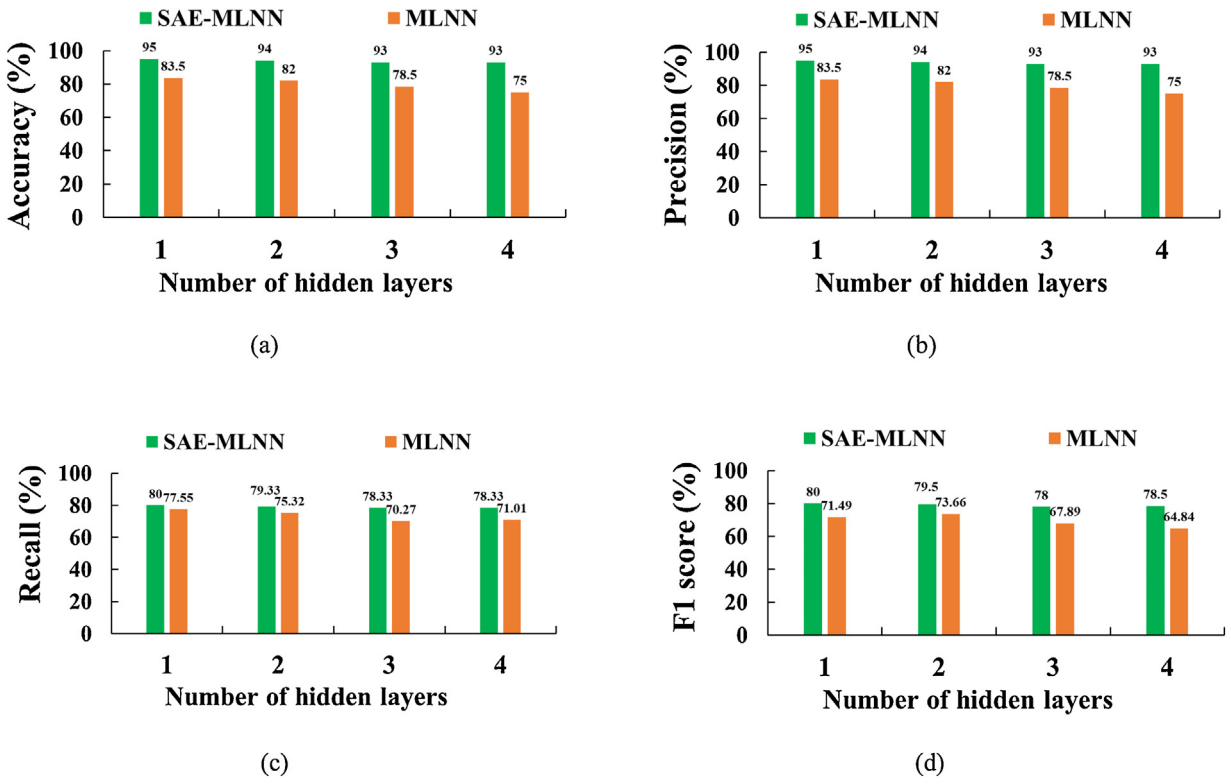


Fig. 8. Comparison results of SAE-MLNN and MLNN.

of the solution and weakens the recognition ability of the model in the meantime. Therefore, the five-hidden-layer network is out of consideration. But for SAE-MLNN, with the increase in the number of hidden layers, although its performances are slightly worse, it shows strong robustness. Based on the aforementioned analysis, it can be concluded that SAE-MLNN possesses superior performance and stronger robustness in recognition of pipeline operating conditions on-site in comparison to MLNN.

5. Conclusion

Due to the rapid development of the multi-product pipelines, as a mode of process monitoring, operating condition recognition is the basic support for pipeline safety and risk management. In this paper, a semi-supervised learning based on SAE and MLNN is proposed. Firstly, the operating data are preprocessed and constructed into data matrices considering the space characteristics and time series of pipeline operating. SAE is utilized to pre-train MLNN via unsupervised learning based on a large amount of unlabeled operating data. Subsequently, MLNN is fine-tuned based on a small amount of labeled data. To verify the effectiveness of the hybrid model, a real-world multi-product pipeline is taken as an example to validate the accuracy, precision, recall, and F1 score. Compared to MLNN, DT, RF, KNN, GB, and XGB, the hybrid model performs better in operating condition recognition, with accuracy, precision, recall, and F1 score being 95 %, 95 %, 80 % and 80 % respectively. Furthermore, the sensitivity analysis of MLNN network construction is conducted to illustrate the importance of SAE in the hybrid model. The results prove that the semi-supervised learning possesses stronger robustness and superior performance regardless of how the neural network is established. The proposed method can be used as a tool for decision support in monitoring and recognizing multi-product pipeline operating conditions.

The limitation of this work is that the fault detection of pipeline operating is not conducted. To recognize the fault condition, a sufficient number of fault data are required. However, there are few faults in practical engineering, resulting in the lack of fault data. As a consequence, the classification model cannot satisfy the demand for accurate fault condition recognition. Our future work will focus on establishing a classification model for normal and abnormal operating condition recognition.

Declaration of Competing Interest

The authors report no declarations of interest.

Acknowledgment

This work was part of the Program of “Study on Optimization and Supply-side Reliability of Oil Product Supply Chain Logistics System” funded under the National Natural Science Foundation of China, grant number 51874325. The authors are grateful to all study participants.

Appendix A

(a) Comparison results of one hidden layer

Number of hidden layers	Number of nodes	Accuracy		Precision		Recall		F1 score	
	Hidden layer1	SAE-MLNN	MLNN	SAE-MLNN	MLNN	SAE-MLNN	MLNN	SAE-MLNN	MLNN
1	61	**95**	85	**95**	85	**80**	73.3	**80**	75
	76	**95**	85	**95**	85	**80**	73.3	**80**	75
	71	**95**	85	**95**	85	80	**88.9**	80	**82.2**
	86	**95**	80	**95**	80	**80**	70	**80**	70
	52	**95**	75	**95**	75	**80**	63.3	**80**	60
	89	**95**	85	**95**	85	**80**	70	**80**	65
	83	**95**	80	**95**	80	**80**	70	**80**	70
	72	**95**	85	**95**	85	80	**88.9**	80	**82.2**
	66	**95**	90	**95**	90	80	**88.9**	80	**82.2**
	58	**95**	85	**95**	85	80	**88.9**	80	**82.2**

(b) Comparison results of two hidden layers.

Number of hidden layers	Number of nodes		Accuracy		Precision		Recall		F1 score	
	Hidden layer1	Hidden layer2	SAE-MLNN	MLNN	SAE-MLNN	MLNN	SAE-MLNN	MLNN	SAE-MLNN	MLNN
2	205	57	**95**	85	**95**	85	**80**	73.3	**80**	75
	226	94	**95**	85	**95**	85	**80**	73.3	**80**	75
	233	57	**95**	85	**95**	85	80	**88.9**	80	**82.2**
	210	87	**95**	80	**95**	80	**80**	70	**80**	70
	249	42	**95**	75	**95**	75	**80**	63.3	**80**	60
	209	70	**95**	85	**95**	85	**80**	73.3	**80**	75
	200	67	**85**	85	**85**	85	**73.3**	**73.3**	**75**	**75**
	226	85	**95**	85	**95**	85	80	**88.9**	80	**82.2**
	198	85	**95**	70	**95**	70	**80**	60	**80**	60
	197	79	**95**	85	**95**	85	80	**88.9**	80	**82.2**

(c) Comparison results of three hidden layers.

Number of hidden layers	Number of nodes			Accuracy		Precision		Recall		F1 score	
	Hidden layer1	Hidden layer2	Hidden layer3	SAE-MLNN	MLNN	SAE-MLNN	MLNN	SAE-MLNN	MLNN	SAE-MLNN	MLNN
3	248	106	31	**95**	85	**95**	85	**80**	73.3	**80**	75
	203	116	50	**85**	75	**85**	75	**70**	**70**	**65**	61.7
	185	102	46	**95**	70	**95**	70	**80**	60	**80**	55
	208	104	54	**95**	75	**95**	75	**80**	63.3	**80**	65
	193	119	55	**95**	85	**95**	85	**80**	73.3	**80**	75
	191	108	55	**95**	75	**95**	75	**80**	65	**80**	66.7
	221	90	44	**95**	85	**95**	85	80	**88.9**	80	**82.2**
	208	126	39	**85**	75	**85**	75	**73.3**	60	**75**	60
	183	109	48	**95**	70	**95**	70	**80**	60	**80**	55
	185	117	41	**95**	90	**95**	90	80	**88.9**	80	**83.3**

(d) Comparison results of four hidden layers.

Number of hidden layers	Number of nodes				Accuracy		Precision		Recall		F1 score	
	Hidden layer1	Hidden layer2	Hidden layer3	Hidden layer4	SAE-MLNN	MLNN	SAE-MLNN	MLNN	SAE-MLNN	MLNN	SAE-MLNN	MLNN
4	254	163	96	32	**85**	75	**85**	75	73.3	**77.8**	**73.3**	65.6
	248	159	90	41	**95**	75	**95**	75	**80**	63	**80**	60
	232	159	108	43	**95**	60	**95**	60	**80**	50	**80**	42.9
	254	175	107	45	**95**	90	**95**	90	80	**92.6**	80	**88.9**
	264	155	96	33	**85**	80	**85**	80	**70**	**70**	70	**70**
	248	168	108	50	**95**	75	**95**	75	**80**	63.3	**80**	60
	239	167	95	33	**95**	85	**95**	85	80	**88.9**	80	**82.2**
	241	155	97	44	**95**	70	**95**	70	**80**	60	**80**	60
	245	177	93	49	**95**	65	**95**	65	**80**	66.7	**80**	53.2
	263	153	84	36	**95**	75	**95**	75	**80**	77.8	**80**	65.6

References

Ao, Yile, Li, Hongqi, Zhu, L.iping, Ali, Sikandar, Yang, Z.hongguo, 2019. The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling. J. Pet. Sci. Eng. 174, 776–789.

Arian, R., Hariri, A., Mehridehnavi, A., Fassihi, A., Ghasemi, F., 2020. Protein kinase inhibitors' classification using K-Nearest neighbor algorithm. Comput. Biol. Chem. 86, 107269.

Chen, X., Liu, Y., Achuthan, K., Zhang, X., 2020. A ship movement classification based on Automatic Identification System (AIS) data using Convolutional Neural Network. Ocean. Eng., 218.

Chetouani, Y., 2014. A sequential probability ratio test (SPRT) to detect changes and process safety monitoring. Process. Saf. Environ. Prot. 92, 206–214.

Cui, Y., Quddus, N., Mashuga, C.V., 2020. Bayesian network and game theory risk assessment model for third-party damage to oil and gas pipelines. Process. Saf. Environ. Prot. 134, 178–188.

da Silva, H.V., Morooka, C.K., Guilherme, I.R., da Fonseca, T.C., Mendes, J.R.P., 2005. Leak detection in petroleum pipelines using a fuzzy system. J. Pet. Sci. Eng. 49, 223–238.

Halim, S.Z., Yu, M., Escobar, H., Quddus, N., 2020. Towards a causal model from pipeline incident data analysis. Process. Saf. Environ. Prot. 143, 348–360.

He, X., Wang, T., Wu, K., Liu, H., 2020. Automatic defects detection and classification of low carbon steel WAAM products using improved remanence/magneto-optical imaging and cost-sensitive convolutional neural network. Measurement, 108633.

Jian, F., Huaguang, Z., 2004. Oil pipeline leak detection and location using double sensors pressure gradient method. In: Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No. 04EX788), IEEE, pp. 3134–3137.

Jiang, H., Xi, Z., Rahman, A.A., Zhang, X., 2020. Prediction of output power with artificial neural network using extended datasets for Stirling engines. Appl. Energy 271, 115123.

Kingma, D., Ba, J., 2014. Adam: A Method for Stochastic Optimization. Computer ence.
Law, A., Ghosh, A., 2019. Multi-label classification using a cascade of stacked autoencoder and extreme learning machines. Neurocomputing, 358.
Li, Z., Tian, L., Jiang, Q., Yan, X., 2021. Distributed-ensemble stacked autoencoder model for non-linear process monitoring. Inf. Sci. (Ny) 542, 302–316.
Liang, W., Zhang, L., 2012. A wave change analysis (WCA) method for pipeline leak detection using Gaussian mixture model. J. Loss Prev. Process Ind. 25, 60–69.
Liu, J., Zang, D., Liu, C., Ma, Y., Fu, M., 2019. A leak detection method for oil pipeline based on markov feature and two-stage decision scheme. Measurement 138, 433–445.
Mandal, S.K., Chan, F.T.S., Tiwari, M.K., 2012. Leak detection of pipeline: an integrated approach of rough set theory and artificial bee colony trained SVM. Expert Syst. Appl. 39, 3071–3080.
Messner, E., Fediuk, M., Swatek, P., Scheidl, S., Pernkopf, F., 2020. Multi-channel lung sound classification with convolutional recurrent neural networks. Comput. Biol. Med., 103831.
Pathy, A., Balasubramanian, P., 2020. Predicting algal biochar yield using eXtreme Gradient Boosting (XGB) algorithm of machine learning methods. Algal Res., 50.
Pontiggia, M., Vairo, T., Fabiano, B., 2020. Critical aspects of natural gas pipelines risk assessments. A case-study application on buried layout. In: Process Safety and Environmental Protection.
Rai, A., Kim, J.-M., 2021. A novel pipeline leak detection approach independent of prior failure information. Measurement 167, 108284.
Sabah, M., Talebkeikhah, M., Agin, F., Telebkeikhah, F., Hasheminasab, E., 2019. Application of decision tree, artificial neural networks, and adaptive neuro-fuzzy inference system on predicting lost circulation: a case study from Marun oil field. J. Pet. Sci. Eng.
Saha, M., Santara, A., Mitra, P., Chakraborty, A., Nanjundiah, R.S., 2020. Prediction of the Indian summer monsoon using a stacked autoencoder and ensemble regression model. Int. J. Forecast.
Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. Inf. Process. Manag. 45, 427–437.
Szoplik, J., 2015. Forecasting of natural gas consumption with artificial neural networks. Energy 85, 208–220.
Wang, N., Zhang, D., Chang, H., Li, H., 2020. Deep learning of subsurface flow via theory-guided neural network. J. Hydrol. (Amst) 584, 124700.
Yang, G., Ding, F., 2020. Associative memory optimized method on deep neural networks for image classification. Inf. Sci. (Ny) 533, 108–119.
Ye, Y., Zhang, L., Liang, W., Wang, Z., 2009a. Fuzzy C-means algorithm in work condition recognition of oil pipeline. 2009 4th IEEE Conference on Industrial Electronics and Applications, 682–686.
Ye, Y., Zhang, L., Wei, L., Dongliang, Y., Zhaohui, W., 2009b. Oil pipeline work conditions clustering based on simulated annealing K-means algorithm. 2009 WRI World Congress on Computer Science and Information Engineering, 646–650.
Yu, J., Zhang, C., 2020. Manifold regularized stacked autoencoders-based feature learning for fault detection in industrial processes. J. Process Control, 92.
Zhang, L., Ye, Y., Wei, L., Dongliang, Y., Zhaohui, W., 2009a. A novel BP algorithm for pipeline condition recognition. 2009 WRI World Congress on Computer Science and Information Engineering, 220–224.
Zhang, Y., Li, J., Chen, S.L., Zhang, J.C., Jin, S.J., 2009b. Recognition method for oil pipeline leak based on chaotic characteristics. Nami Jishu yu Jingmi Gongcheng/Nanotechnology and Precision Engineering 7, 337–341.
Zhang, Y., Chen, S., Li, J., Jin, S., 2014. Leak detection monitoring system of long distance oil pipeline based on dynamic pressure transmitter. Measurement 49, 382–389.
Zhang, C., Wu, J., Hu, X., Ni, S., 2018. A probabilistic analysis model of oil pipeline accidents based on an integrated Event-Evolution-Bayesian (EEB) model. Process. Saf. Environ. Prot. 117, 694–703.
Zhang, Y., Zhang, Y., He, K., Li, D., Xu, X., Gong, Y., 2021. Intelligent feature recognition for STEP-NC-compliant manufacturing based on artificial bee colony algorithm and back propagation neural network. J. Manuf. Syst.
Zheng, J., Dai, Y., Liang, Y., Liao, Q., Zhang, H., 2020. An online real-time estimation tool of leakage parameters for hazardous liquid pipelines. International Journal of Critical Infrastructure Protection.
Zheng, J., Liang, Y., Xu, N., Wang, B., Zheng, T., Li, Z., Liao, Q., Zhang, H., 2021. Deeppipe: a customized generative model for estimations of liquid pipeline leakage parameters. Computers & Chemical Engineering.
Zheng, S., Zhao, J., 2020. A new unsupervised data mining method based on the stacked autoencoder for chemical process fault diagnosis. Comput. Chem. Eng. 135, 106755.
Zhou, X., Zhang, H., Qiu, R., Liang, Y., Wu, G., Xiang, C., Yan, X., 2019a. A hybrid time MILP model for the pump scheduling of multi-product pipelines based on the rigorous description of the pipeline hydraulic loss changes. Comput. Chem. Eng., 121.
Zhou, Y., Daamen, W., Vellinga, T., Hoogendoorn, S.P., 2019b. Ship classification based on ship behavior clustering from AIS data. Ocean. Eng., 175.
Zhou, X., Gelder, P.H.A.J.Mv., Liang, Y., Zhang, H., 2020. An integrated methodology for the supply reliability analysis of multi-product pipeline systems under pumps failure. Reliab. Eng. Syst. Saf., 204.