A Pipeline Leak Detection and Localization Approach Based on Ensemble TL1DCNN

MENGFEI ZHOU, YANHUI YANG, YINZE XU, YINCHAO HU, YIJUN CAI, JUNJIE LIN, AND HAI TIAN PAN
Department of Chemical Engineering, Zhejiang University of Technology, Hangzhou 310027, China
Corresponding author: Haitian Pan (htpan@zjut.edu.cn)
This work was supported in part by the Application Research of Public Welfare Technology in Zhejiang Province, China, under Grant LGF20E090005, and in part by the National Natural Science Foundation of China under Grant 21676251.

ABSTRACT
There is an increasing need for timely pipeline leak detection and localization methods, pipeline leak could lead to not only the loss of the goods but also considerable environmental and economic problems. With the rapid development of hardware and software, the pipeline leak detection and localization algorithms have been widely researched and applied in many fields. However, traditional methods are usually limited by extracting features manually, which is inefficient and time-consuming. Convolutional neuron network is an effective method to extract features automatically. In this paper, a novel ensemble transfer learning one-dimension convolutional neural network (TL1DCNN) for the pipeline leak detection and localization is proposed. The TL1DCNN plays the role of base learner. The results of a set of obtained base learners are integrated to achieve the task of pipeline leak detection and localization. Firstly, one-dimension convolutional neural network (1DCNN) models with different parameters are pretrained with source domain data. A small learning rate is set to retrain the above 1DCNN models for target task with target domain data in order to obtain TL1DCNN base learners. Then, the four ensemble strategies with different number base learners whose ensemble weights are optimized by particle swarm optimization algorithm are obtained by minimizing the sum of similarity. The dataset simulated by pipeline network model is used to evaluate the effectiveness of the proposed approach. The indicators such as classification accuracy, precision, recall, F_score and confusion matrix are used to compare the proposed approach with traditional methods and other deep learning methods. The experimental results show that the performance of the proposed approach is superior to other compared methods.

INDEX TERMS
Leak detection, transfer learning, one-dimension convolutional neural network, ensemble, machine learning.

I. INTRODUCTION
Because of the advantage of large transport volume, safety, reliability and strong continuity, pipeline transportation is widely used in the delivering of oil, gas and urban running water. There were over 3,800 long distance oil and gas pipelines with a distance approximately 1.95 million kilometers by 2017 [1]. However, with long-term use, pipeline would usually be damaged due to corrosion, aging, construction damage, etc., which may cause huge environmental and economic loss. It was reported that there was a total of 12,310 pipeline accidents in America between 2000 to 2019, which made 307 deaths and 1,221 injuries, and economic losses of 10,020,251,321 dollars [2]. Thus, early detection and identification of pipeline abnormalities and faults is very important to minimize performance degradation and avoid dangerous situation [3].

The technology of pipeline leak detection has been extensively studied and many corresponding algorithms has been presented. According to the published literature research, the algorithms could be divided into direct detection and indirect detection [4]. More often, these methods have been divided into hardware-based methods and software-based methods. The detailed classification is shown in Figure 1 [5]. The hardware detection methods rely on hardware devices, such as hydrophone, acoustic method, cable sensor, optical...
However, when the hardware-based methods are applied on a large scale, expensive equipment needs to be installed, which results in high economic costs. Since some measured variables (such as pressure, volume, mass) are usually needed to be measured for the software-based methods, pipeline hydraulic dynamics or machine learning methods are adopted to obtain the models to detect and locate pipeline leaks [11]–[15]. Although many pipeline leak detection and location tasks can be solved by traditional software-based methods, there are still many limitations. For example, the extraction of suitable features usually relies on strong professional knowledge and experience, which can be considered as an important factor for the best performance. Moreover, only shallow information, which is difficult to be used in complex systems, can be obtained by the manually extracted features. Therefore, the disadvantages of traditional software-based methods limit the classification and regression performance.

More recently, deep learning brings breakthroughs in computer vision, natural language processing and fault diagnosis [16]. The convolutional neural network (CNN) has been largely followed with interest when the AlexNet model [17] won the ImageNet competition champion in 2012. Deep features from raw data without prior knowledge can be automatically extracted with the CNN which generally consists of convolutional layers, pooling layers and fully connected (FC) layers. Some researchers converted one-dimension signal to two-dimension image and then selected two-dimension convolutional neural network (2DCNN) for fault diagnosis in the early stage [18]–[21]. Compared with the direct use of one-dimension convolutional neural network (1DCNN) model, some useful information may be lost and the computational cost would be increased after the conversion of one-dimensional signals to two-dimensional signals. As a result, some scholars have recently been paying more attention to 1DCNN. Ince et al. [22] proposed a timely and accurate motor condition monitoring and incipient fault detection system based on 1DCNN trained from raw data directly. Wang et al. [23] employed a multi-attention 1DCNN supported by attention mechanism for wheelset bearing fault diagnosis. Kang et al. [24] used 1DCNN to extract the features from original time series signals collected by accelerometers and fused a multilayer perceptron and a support vector machine, the latter obtained the results from the extracted features for leak detection. However, there are some shortcomings in the application of CNN. For example, training data and testing data should be in the same feature space and the distribution need to be assumed to be the same. Moreover, a large labeled data is required. These assumptions are often difficult to be satisfied in the real world. Recently, transfer learning [25], [26] has been utilized to avoid these problems and has been applied to fault diagnosis in many fields [27]–[30].

Nevertheless, transfer learning sometimes is difficult to achieve outstanding performance for the tasks with a small amount of data and many feature dimensions yet. The risk of poor generalization performance and falling into a local minimum by combining several learners can be reduced with ensemble learning [31]. As a result, the ensemble deep learning algorithms were applied in fault diagnosis by some literatures [32], [33]. In addition, many satisfactory results can be obtained by the ensemble methods which combine several base learners with different training data, parameters or structures [24], [34].

To achieve the task of accurately detecting and locating the pipeline leak from small data, an approach that 1DCNN played a role of base learner and the results of several base learners were combined and optimized to achieve the task.
of pipeline leak detection and localization is proposed in the paper. Additionally, the proposed approach can eliminate the limitation of obtaining sufficient fault sample data and improve the accuracy of leak detection.

The contributions of this paper are summarized as follows:

1. Transfer learning one-dimension convolutional neural network (TL1DCNN) for the task of pipeline leak detection and location is proposed. Then, several classification probability vectors of different TL1DCNN are combined with different optimized weights to improve the performance of pipeline leak detection and location.

2. In order to verify the effectiveness, this proposed approach is tested in a simplified water distribution network to detect leakage and location, where the TL1DCNNs are trained using different combination strategies of hyperparameters. The result shows that the performance of the TL1DCNN model is greatly affected by hyperparameters. In addition, some traditional leakage detection approaches and other deep learning approaches are selected to compare with the proposed approach.

3. The proposed method can be used to deal with the signals collected by sensors to identify a leak and to locate the leak section roughly.

4. The double descent curve of deep learning model is verified with the leakage data.

The remainder of this paper is organized as follows: In the next section, the TL1DCNN for feature extraction and ensemble strategy is described in detail. Section 3 analyzes and discusses the results of the research, and compares our method with other methods. Finally, the conclusion is given in Section 4.

II. METHODS

A. TRANSFER LEARNING

In many fields such as fault detection, it is hard to obtain enough sample data to build models. Transfer learning is employed to overcome this problem. Transfer learning aims to solve one problem and applies the project to another different but related problem. The transfer learning techniques can be divided as inductive transfer learning, transductive transfer learning and unsupervised transfer learning. Pan and Yang [26] gave a definition of transfer learning: Given a source domain $D_S$ and learning task $T_S$, a target domain $D_T$ and learning task $T_T$, inductive transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $D_T$ using the knowledge in $D_S$ and $T_S$, where $T_S \neq T_T$. For example, if there is a task of identifying pictures of cats, but only few pictures of cats could be collected and many dog pictures could be obtained. In this case, when the transfer learning technique is employed, the dog pictures can be set as source data and the cat pictures can be set as target data. Moreover, identifying the dogs and cats are set as source task and target task respectively. Many machine learning algorithms are used for transfer learning. In particular, deep learning models such as CNN and RNN are widely applied to the transfer learning system by the scholars in recent years [35], [36].

In the work, the transfer learning is used to learn some useful parameters, which can be transferred to the target task with small target domain data under other working conditions from source domain data. The left part in Figure 2 shows the transfer learning process. The training process is expresses by:

- **Step 1**: The best performance pretrained CNN model is obtained by source domain data and optimized by adjusting hyperparameters.
- **Step 2**: The parameters of target model are initialized by the pretrained CNN model. The neurons numbers of output layer are changed from source label category numbers to target label category numbers. Then, target model is retrained to fine tune the model parameters with little learning rates until the value of loss function keeps stable.
- **Step 3**: Several TL1DCNNs for ensemble with different parameters are obtained by repeating Step 1 and Step 2.

B. TL1DCNN MODEL

The CNN is used as a feature extraction module in pipeline leak detection and localization, because the CNN has been proven to perform well in computer vision, object recognize, fault diagnosis, etc. Different from 2DCNNs that usually process two-dimensional pictures, one-dimensional time series signals are input into the 1DCNN model for feature extraction in the study.

Local features from input data are extracted by the convolutional layers. Some experience has shown that bottom layers usually obtain general features such as edge, shape, etc., and upper layers usually get higher-level features based on previous local features [16]. The function of convolution is described by the following expression:

$$S_i = b_i + \sum_n (W_{in} \ast X_n)$$  \hspace{1cm} (1)

where $i, n, W$ denote the index, size and parameters of the convolutional kernel, respectively. $X$ and $S$ represent the input and output of current convolutional kernel are represented by $X$ and $S$, and the bias of the $i$-th kernel is denoted by $b_i$.

Krizhevsky et al. [17] regarded ReLU activation function as the key factor for the success of the AlexNet model. The mathematical representation of ReLU is:

$$\text{ReLU} (x) = \max (0, x)$$  \hspace{1cm} (2)

The eigenvalue distribution of each layer will be close to activation function saturation interval of output range as network depth increasing, and it would lead to vanishing gradient. Batch Normalization (BN) makes current eigenvalue distribution return to standard normal distribution, and the eigenvalue would fall in the interval where the activation function is more sensitive to the input.
The feature maps output by the convolution layer can be downscaled by the pooling layers, which generally includes maximum pooling and average pooling. Thus, the size of data and the parameters to be calculated can be reduced exponentially.

The FC layers are set at the top layer of CNN model after several alternate convolutional layers and pooling layers. The key features extracted before FC layer are calculated by softmax function and the predicted probability vector is expressed as \( \hat{y} \). The predicted error \( E \) can be calculated by the cost function using cross entropy, expressed by:

\[
E = - \sum_m \sum_n y_n^{(m)} \log \hat{y}_n^{(m)}
\]  

where \( y_n^{(m)} \) and \( \hat{y}_n^{(m)} \) denote label and predictive output, respectively. \( n \) is category index. \( m \) is sample index. The purpose of the training is to update the weights and biases of the network through the learning rate to minimize the prediction error. The updating direction and number size of each epoch are calculated by Adam optimizer.

C. ENSEMBLE TL1DCNN APPROACH

Better performance can be achieved by using multiple base learners with the ensemble approach. The key point of ensemble method is a difference between base learners, which means each base learner is good but different. Generally, to make ensemble method more valid, data, parameters, structure and model type should be set different. The parameters’ difference means that the different base learners are trained from different model parameters combination, even with the same training set [34].

For a two-category classification, \( y \in \{-1, +1\} \) and real function \( f(x) \), it is assumed the error rate of base learner is \( e \). Each base learner \( h_i(x) \) meets:

\[
P(h_i(x) \neq f(x)) = e
\]

If \( N \) base learners are integrated by voting, the result can be obtained by:

\[
V(x) = \text{sign}(\sum_{i=1}^{N} h_i(x))
\]

Assuming that the error rates of the base classifiers are independent of each other, according to the Hoeffding inequality, the integrated error rate is:

\[
P(H(x) \neq f(x)) = \sum_{j=0}^{N} \left( \begin{array}{c} N \\ j \end{array} \right) (1 - e)^j e^{N-j} \leq \exp\left(-\frac{1}{2}N(1-2e)^2\right)
\]  

As shown in (6), the error rate of ensemble would decrease rapidly as the number of the base learners increases. The probability vectors from \( n \) TL1DCNNs with different parameters multiply with \( n \) weights are optimized by particle swarm optimization (PSO) respectively, and the output of the proposed method is obtained by adding \( n \) multiplications.
equation (7) is represented in the lower part and equation (8) is displayed in the upper part. The function is shown as following:

$$Y = \sum_{i=1}^{n} w_i \times P_i$$  \hspace{1cm} (7)

$$\text{Result} = \arg\max (Y)$$  \hspace{1cm} (8)

where $Y$ denotes the probability vector calculated by ensemble method, $w_i$ and $P_i$ represent $i$-th weight parameter and probability from $i$-th base learner. $n$ is obtained by balancing the computing cost and the model performance. Result is the category output by the proposed method.

III. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed method is implemented in Python 3.7 with TensorFlow 2.1 and runs on Windows 7 with an Intel i5 CPU.

The following metrics are calculated to quantify the classification performance of different methods:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$  \hspace{1cm} (9)

$$\text{Precision} = \frac{TP}{TP + FP}$$  \hspace{1cm} (10)

$$\text{Recall} = \frac{TP}{TP + FN}$$  \hspace{1cm} (11)

$$\text{F-score} = \frac{2TP}{2TP + FP + FN}$$  \hspace{1cm} (12)

where $TP$, $TN$, $FP$, $FN$ are the abbreviation of True Positive, True Negative, False Positive and False Negative, respectively.

A. EXPERIMENTAL DATA

The simulated data set used in the proposed ensemble TL1DCNN method comes from Flowmaster V7, which is a well-known pipeline simulation software. Flowmaster is used as a general 1D computational fluid dynamics code for modeling and analyzing fluid mechanics in complex piping systems of any scale. Moreover, a method of characteristics solver is used, which can accurately handle fluid transients in complex systems and systems-of-systems [37]. The structure of the pipeline is shown in Figure 3. In the case, the leak problem which is a form of leak characterized by short duration but usually large flow caused by pipe burst is considered. The pipeline system is composed by a pump providing the water pressure with 10 ± 0.5 bar, a main pipeline with a length of 1 kilometer and a diameter of 400 millimeters, two branch pipes with a length of 500 meters and a diameter of 200 millimeters and 250 millimeters respectively. The roughness of the pipeline wall is 0.07 mm. The fluid in the pipeline is water with a density of 1000 kg/m$^3$ and a temperature of 20 $^\circ$C. The flow rate in the main pipe is about 0.19 m$^3$/s, and the flow rate in the two branch pipes are 0.037 m$^3$/s and 0.043 m$^3$/s, respectively. The maximum flow rate at the leak area is 0.013 m$^3$/s. the sampling length of the four pressure sensors is 10 seconds with a sampling frequency set to 50 Hz to collect the negative pressure wave signals.
FIGURE 4. Pressure signals of each category. No leak denotes normal condition; Small leak and Large leak are distinguished by the leak flow less than 5% of pipeline flow or not.

Usually, it is assumed that the velocity of the negative pressure wave is 1000 m/s in many published paper [38], [39]. As a result, the negative pressure wave transmission speed is set at 1000 m/s in the simulation. The opening of ball valves is used to simulate the size of leakage. The flowmeter is set between the ball valve and pipe to measure the leak flow. The pressure signals are collected at the pipeline by changing the water pressure from pump, leak aperture diameter, leak position and the ball valve opening time. Thus, 500 points pressure data are collected from each sensor. The points of a sample are 2,000 by connecting the signals from the four sensors end to end. Generally, the large leak is defined that the ratio of leak flow to pipe flow is bigger than 5% [40]. Because the Flowmaster software generates noiseless data, by setting the leak flow rate, the collected data can easily be labeled as large or small leak. However, under actual conditions, there will always be some noise in the collected signal, such as measurement noise and environmental noise. In order to simulate real working conditions and improve the robustness of the model, the zero-average standard normal noise distribution is applied to the collected pressure data. There are 9 leak sections which are divided into large leak and small leak according to the leak flow bigger than 5% of the pipeline flow or not, and each section is 100 meters long. In the case, there are 19 categories with one normal condition, 9 large leak conditions (Large S1-S9) and 9 small leak conditions (Small S1-S9). Random Gaussian noise between −0.5 bar and 0.5 bar is added to the data to simulate the real environment. There are 3,144 samples that 2,358 samples as source training set and 786 samples as source validation set with the labels from S1 to S3 and no leak. The target dataset with 1,137 samples is composed by 717 samples with the labels from S4 to S9 and 420 samples extracted from source data set randomly. The target data set is divided into target training set with 730 samples, target validation set with 122 samples and target test set with 285 samples. Figure 4 shows the pressure signals of each category. The information about datasets is given in Table 1.

### TABLE 1. Information about datasets.

|                   | Source domain | Target domain |
|-------------------|---------------|---------------|
| **Training set**  |               |               |
| Labels            | No leak, Small S1-S3, Large S1-S3 | No leak, Small S1-S9, Large S1-S9 |
| Size              | 2358          | 717           |
| **Validation set**|               |               |
| Labels            | No leak, Small S1-S3, Large S1-S3 | No leak, Small S1-S9, Large S1-S9 |
| Size              | 786           | 122           |
| **Test set**      |               |               |
| Labels            | None          | No leak, Small S1-S9, Large S1-S9 |
| Size              | None          | 285           |

**B. EXPERIMENTAL RESULTS**

The source domain task (TS) is to establish a classification model of the source domain data (DS) with 7 categories, the target domain task (TT) is to establish a classification model of the target domain data (DT) with 19 categories. The
knowledge of source task model could help to train the target task because the data volume of $D_S$ is much larger than that of $D_T$.

Before establishing the model, the data should be preprocessed. Firstly, the data from each sensor are spliced end to end and then the standard normal distribution noises are added to the splicing data. Finally, to speed up the convergence of weight parameters, the data are preprocessed through standardization before inputting them to the neural network.

1) MODEL CAPACITY
Traditional computational learning theory believes that for a certain algorithm, when the amount of data is fixed, the performance of the model first becomes better and then becomes worse as the number of parameters increases, whose test risk is a U-shaped curve. But in the field of deep learning, this rule is not entirely correct. There are lots of literatures which mentioned over-parameterization is necessary [41]–[43]. Goodfellow et al. [44] pointed out that for general non-convex optimization problems of deep learning, there is only a few theoretical analyses. The AlexNet model [17], which won the 2012 ImageNet champion, trained 60 million parameters by only 1.2 million images, so the calculation theory of deep learning should be re-understood.

Belkin et al. [45] proposed a double descent risk curve to modify the U-shaped curve, and gave a formula for calculating the peak value of the curve, which is the product of the number of samples and the number of categories. The authors explained that the performance of the model becomes better with over-parameterization after the peak because more options are provided by more parameters for the model to choose smoother and simpler functions. Therefore, the performance of the classifier can be improved by increasing the model capacity. Ba and Caruana [46] also proposed that the flexible model can be simulated by simple model, in other words, the inner function of a flexible model may be not really flexible and the model expressiveness is not the same as the model complexity.

The pipeline leakage data are used to verify the conclusions of the above literature. The average loss of the sample decreases first, and then increases as the number of parameters increase. After reaching the peak, it decreases and is even better than the sweet spot of the U-shaped curve. The detail is shown in Figure 5. Therefore, the magnitude of the subsequent models’ parameter is $10^5$.

2) SELECTION OF THE NUMBER OF PRESSURE SENSORS
In the study, it is founded that the number and location of pressure sensors are very important for balancing the computational consumption and model accuracy. Thus, sensor layout plans with different numbers and location are compared. Table 2 presents 5 cases of the sensor layout.

The TL1DCNNs are built for Case 1 to Case 5 with the same structure and hyperparameters. The confusion matrices of these five cases are shown in Figure 6. It can be found that the accuracies of Case 1 to Case 4 are 74.04%, 78.60%, 81.75% and 84.91%, respectively, while the accuracy of case 5 reached 90.18%, which means that the accuracy of case 5 is higher than 5% in each of the previous four cases. The reason for the above phenomenon may be that if there is no sensor installed on a certain pipeline, and the signal of the pipeline is similar with that of other pipelines nearby, and it would cause the misjudge of the model prediction. Therefore, four sensors are selected for achieving an acceptable accuracy and saving computational resources as much as possible.

3) SELECTION OF OPTIMAL TL1DCNN PARAMETERS
In order to get the model parameters with best performance in validation set and test set, the layer number, learning rate, epoch and batch size are selected to compare the prediction accuracy. The combination of these parameters is listed in Table 3. It can be found that the model using the second set of parameters achieved the largest accuracy of the test set when the training cost was small, and is similar to the accuracy of the training set, which indicates the overfitting phenomenon and insufficient model expression ability can be overcome to a certain extent. Thus, the parameters of the No.2 are used for subsequent analysis.

4) SELECTION OF ENSEMBLE MODELS
Similarity is defined as the ratio of the same predictive label number of two models and the total validation sample number. The output results of each of the five TL1DCNN models whose structure and accuracy are shown in Table 4 is...
probability vector, which is denoted by \( P_1 \) to \( P_5 \). Moreover, the epochs of five TL1DCNN models are shown in Figure 7. The first index points to the label of no leak. The second index points to small \( S1 \) and the third index points to large \( S1 \). As the same law, the fourth index to the nineteenth index point to small \( S2 \) to large \( S9 \) respectively. For a probability vector, if one of the indexes gets the maximum, the predictive label of this sample is the pointed label of the index which is represented as \( I \). The predictive results and similarity of two models can be defined as:

\[
I = \{ x \mid \forall y: P[y] \leq P[x] \}
\]

\[
\text{Similarity}_{i,j} = \frac{1}{n} \sum_{i} (I_i = I_j)
\]

where \( P[y] \) and \( n \) denote the \( y \)th element in the \( P \) vector and the number of samples respectively. The similarity matrices of the five models in validation dataset are shown in Table 5.

### Table 3. List of model parameters and corresponding accuracy results.

| No. | Layers | Learning rate | Epoch | Batch size | Training accuracy | Validation accuracy | Testing accuracy |
|-----|--------|---------------|-------|------------|-------------------|--------------------|-----------------|
| 1   | 2      | 0.0005        | 40    | 20         | 98.83%            | 91.23%             | 87.37%          |
| 2   | 3      | **0.0005**    | 40    | 20         | **96.48%**        | **91.81%**         | **90.18%**      |
| 3   | 4      | 0.0005        | 40    | 20         | 90.32%            | 87.13%             | 85.61%          |
| 4   | 5      | 0.0005        | 40    | 20         | 88.86%            | 84.21%             | 83.16%          |
| 5   | 3      | 0.0003        | 40    | 20         | 96.77%            | 88.30%             | 87.72%          |
| 6   | 3      | 0.001         | 40    | 20         | 95.89%            | 88.30%             | 87.72%          |
| 7   | 3      | 0.0005        | 40    | 10         | 87.68%            | 78.36%             | 81.40%          |
| 8   | 3      | 0.0005        | 40    | 30         | 86.66%            | 78.95%             | 81.05%          |
| 9   | 3      | 0.0005        | 30    | 20         | 96.19%            | 88.89%             | 88.42%          |
| 10  | 3      | 0.0005        | 50    | 20         | 98.83%            | 91.23%             | 87.37%          |

### FIGURE 6. Confusion matrices of five cases. (a) Case 1. (b) Case 2. (c) Case 3. (d) Case 4. (e) Case 5.
From all strategies with the same base learner number, the combined strategy with the smallest sum of similarity in the validation dataset is selected to form an ensemble case in the work. According to this algorithm, four cases can be obtained, namely, Case 1: TL1DCNN 3, TL1DCNN 4; Case 2: TL1DCNN 3, TL1DCNN 4, TL1DCNN 5; Case 3:
TL1DCNN 2, TL1DCNN 3, TL1DCNN 4, TL1DCNN 5; Case 4: all the TL1DCNNs. Thus, the ensemble result can be obtained by adding the matrices of each base learner probability matrix with corresponding weights. The weights are obtained by using PSO algorithm with 50 particles and 40 iterations. The accuracy of each case is shown in Figure 8. The strategy of Case 4 gets the best performance with a weight matrix [0.3186, 0.1044, 0.1798, 0.3444, 0.0527]. As a result, all the five models with the weights 0.3186, 0.1044, 0.1798, 0.3444, 0.0527 respectively are used for analysis.

C. RESULT ANALYSIS

In order to evaluate the performance of the proposed ensemble TL1DCNN model, the traditional methods and the ensemble TL1DCNN method with same weight parameters and a single TL1DCNN method are selected to compare. Usually, empirical mode decomposition (EMD) and wavelet decomposition (WT) are employed to denoise the original signal. Then, some features can be extracted from the preprocessed data, such as mean, variance, standard deviation, root mean square, skewness, kurtosis, form factor, crest factor, impulse factor and margin factor [47], [48] which are defined as (15)-(24). Finally, the support vector machine (SVM) and back propagation neural network (BP) are selected in order to compare with the performance of the proposed method for leak detection. As a result, four traditional methods involving EMD-SVM, EMD-BP, WT-SVM and WT-BP are considered for comparison with the proposed method.

\[
X_{\text{mea}} = \frac{1}{n} \sum_{j=1}^{n} x_j \tag{15}
\]

\[
X_{\text{var}} = \frac{1}{n} \sum_{j=1}^{n} (x_j - X_{\text{mea}})^2 \tag{16}
\]

\[
X_{\text{std}} = \sqrt{X_{\text{var}}} \tag{17}
\]

\[
X_{\text{rms}} = \sqrt{\frac{1}{n} \sum_{j=1}^{n} x_j^2} \tag{18}
\]

\[
X_{\text{ske}} = \left( \frac{1}{n} \sum_{j=1}^{n} x_j^3 / (X_{\text{rms}}^3) \right) \tag{19}
\]

\[
X_{\text{kur}} = \left( \frac{1}{n} \sum_{j=1}^{n} x_j^4 \right)^{1/2} \tag{20}
\]

\[
X_{\text{fof}} = X_{\text{rms}} / \left( \frac{1}{n} \sum_{j=1}^{n} |x_j| \right) \tag{21}
\]

\[
X_{\text{crf}} = \max \left( \frac{|x_j|}{X_{\text{rms}}} \right) \tag{22}
\]

\[
X_{\text{imf}} = \max \left( \frac{|x_j|}{\left( \frac{1}{n} \sum_{j=1}^{n} |x_j| \right)} \right) \tag{23}
\]

\[
X_{\text{maf}} = \max \left( \frac{|x_j|}{\left( \frac{1}{n} \sum_{j=1}^{n} |x_j|^{1/2} \right)^2} \right) \tag{24}
\]

where \( n \) denotes the sample numbers and \( X_{\text{mea}}, X_{\text{var}}, X_{\text{std}}, X_{\text{rms}}, X_{\text{ske}}, X_{\text{kur}}, X_{\text{fof}}, X_{\text{crf}}, X_{\text{imf}} \) and \( X_{\text{maf}} \) denote mean, variance, standard deviation, root mean square, skewness, kurtosis, form factor, crest factor, impulse factor and margin factor.

The denoising effect are shown in Figure 9 and the extracted features are shown in Table 6. The features are extracted via the above formulas from the denoised data and then these features are used as the input of the SVM and BP classifiers. For the unbalanced data, BP would predict the labels of less number samples as the labels of larger number samples. In order to overcome this shortcoming, the no leak data are resampled to balance the categories. The compared results between the proposed method and the other methods are shown in Table 7. It should be noted that the 2DCNN method [19] in Table 7 was applied to a long straight pipeline, which is a different type of pipelines from the pipeline network for the proposed method, and the 2DCNN was also
TABLE 6. The extracted features.

|            | EMD         | WT          |
|------------|-------------|-------------|
|            | No leak     | Small leak  | Large leak  |
|            | No leak     | Small leak  | Large leak  |
| \(X_{mea}\) | 6.4834      | 6.4402      | 6.4739      | 6.4612      | 6.5162      | 6.3937      |
| \(X_{var}\) | 0.0998      | 0.4498      | 0.2187      | 0.0983      | 0.1100      | 0.4102      |
| \(X_{std}\) | 0.3160      | 0.6707      | 0.4676      | 0.3135      | 0.3317      | 0.6405      |
| \(X_{rms}\) | 6.4911      | 6.4750      | 6.4908      | 6.4688      | 6.5246      | 6.4257      |
| \(X_{exc}\) | 0.4735      | -0.1180     | 0.0895      | 0.4275      | 0.4565      | -1.9583     |
| \(X_{kal}\) | -1.2993     | 1.0460      | 0.3941      | -1.3424     | -1.1314     | 12.9483     |
| \(X_{tof}\) | 1.0012      | 1.0054      | 1.0026      | 1.0012      | 1.0013      | 1.0050      |
| \(X_{crf}\) | 1.0853      | 1.3257      | 1.2100      | 1.0830      | 1.1053      | 1.3861      |
| \(X_{omf}\) | 1.0866      | 1.3328      | 1.2131      | 1.0843      | 1.1067      | 1.3931      |
| \(X_{mof}\) | 1.0872      | 1.3365      | 1.2147      | 1.0849      | 1.1074      | 1.3972      |

TABLE 7. The compared results of the proposed model and the other methods.

|            | No leak     | Small leak  | Large leak  |
|------------|-------------|-------------|-------------|
|            | Precision   | Recall      | Accuracy    |
| EMD-SVM    | 86.7%       | 88.9%       | 90.4%       |
| EMD-BP     | 51.7%       | 85.6%       | 89.5%       |
| WT-SVM     | 93.3%       | 91.1%       | 95.6%       |
| WT-BP      | 87.5%       | 94.6%       | 92.8%       |
| 2DCNN      | 84.9%       | 89%         | 92.6%       |
| Ensemble   | Recall      | Precision   | Accuracy    |
| TL1DCNN    | 100%        | 99.3%       | 95.6%       |

The base learners of the proposed method have good classification ability and there are differences between them. The weights for each base learner are different and optimized, moreover, the one with a great contribution to classification is given a greater weight.
TABLE 8. Performance of models.

| Label  | Ensemble TL1DCNN | Ensemble TL1DCNN with Same Weights | TL1DCNN |
|--------|------------------|------------------------------------|---------|
|        | Precision | Recall | F_score | Precision | Recall | F_score | Precision | Recall | F_score |
| No leak| 100%      | 100%    | 100%    | 100%      | 100%    | 100%    | 100%      | 100%    | 100%    |
| Small S1| 86.7%    | 76.5%   | 81.2%   | 86.7%    | 76.5%   | 81.2%   | 86.7%    | 72.2%   | 78.8%   |
| Large S1| 80.0%    | 93.8%   | 88.9%   | 80.0%    | 100%    | 88.9%   | 73.3%    | 100%    | 84.6%   |
| Small S2| 93.3%    | 93.3%   | 93.3%   | 93.3%    | 93.3%   | 93.3%   | 93.3%    | 93.3%   | 93.3%   |
| Large S2| 100%     | 100%    | 100%    | 93.3%    | 100%    | 96.6%   | 100%     | 100%    | 100%    |
| Small S3| 93.3%    | 77.8%   | 84.8%   | 93.3%    | 82.4%   | 87.5%   | 93.3%    | 73.7%   | 82.4%   |
| Large S3| 100%     | 88.2%   | 93.8%   | 100%     | 88.2%   | 93.8%   | 100%     | 83.3%   | 90.9%   |
| Small S4| 100%     | 100%    | 100%    | 93.3%    | 100%    | 96.6%   | 93.3%    | 100%    | 96.6%   |
| Large S4| 93.3%    | 100%    | 96.6%   | 93.3%    | 93.3%   | 93.3%   | 93.3%    | 87.5%   | 90.3%   |
| Small S5| 100%     | 93.8%   | 96.8%   | 100%     | 93.8%   | 96.8%   | 100%     | 100%    | 100%    |
| Large S5| 100%     | 100%    | 100%    | 100%     | 100%    | 100%    | 86.7%    | 92.9%   | 89.7%   |
| Small S6| 73.3%    | 100%    | 84.6%   | 80.0%    | 92.3%   | 85.7%   | 66.7%    | 83.3%   | 74.1%   |
| Large S6| 86.7%    | 100%    | 92.9%   | 80.0%    | 100%    | 88.9%   | 66.7%    | 100%    | 80.0%   |
| Small S7| 66.7%    | 90.9%   | 76.9%   | 73.3%    | 78.6%   | 75.9%   | 73.3%    | 100%    | 84.6%   |
| Large S7| 93.3%    | 100%    | 96.6%   | 93.3%    | 100%    | 96.6%   | 93.3%    | 100%    | 96.6%   |
| Small S8| 93.3%    | 66.7%   | 77.8%   | 80.0%    | 66.7%   | 72.7%   | 100%     | 75.0%   | 85.7%   |
| Large S8| 100%     | 93.8%   | 96.8%   | 100%     | 93.8%   | 96.8%   | 100%     | 83.3%   | 90.9%   |
| Small S9| 93.3%    | 87.5%   | 90.3%   | 93.3%    | 87.5%   | 90.3%   | 93.3%    | 93.3%   | 93.3%   |
| Large S9| 93.3%    | 100%    | 96.6%   | 93.3%    | 93.3%   | 93.3%   | 100%     | 100%    | 100%    |

IV. CONCLUSION
A pipeline leak detection and localization method based on the ensemble TL1DCNN is proposed in this study. The main contribution of this research is that the transfer learning is integrated into 1DCNN, and the PSO is used to optimize the weights of several selected TL1DCNN base learners, and an ensemble TL1DCNN is finally obtained for pipeline leak detection and localization. Compared to the traditional data-driven methods, the features can be automatically extracted rather than manually, and the performance of the model is improved by the ensemble approach. Moreover, the need for dataset size is reduced by using transfer learning technology. Due to the above advantages, the proposed TL1DCNN model trained on a small dataset achieves a good performance for pipeline leak detection and localization.

In this study, the different combination strategies of model structure, learning rate, epoch, batch size are selected to obtain optimal parameters. One of the four ensemble strategy that achieves the best performance is selected. The effectiveness of the proposed method is evaluated by simulated data which achieves 90.5%, 91.8% and 90.4% in the precision, recall and F_score, respectively. On the other hand, the proposed method is better than the enumerated traditional approaches, the ensemble TL1DCNN with same weights method and a single TL1DCNN method. These results show that the proposed method has certain prospects in the field of pipeline leak detection and location.

REFERENCES
[1] H. Lu, L. Guo, M. Azimi, and K. Huang, “Oil and gas 4.0 era: A systematic review and outlook,” Comput. Ind., vol. 111, pp. 68–90, Oct. 2019, doi: 10.1016/j.compind.2019.06.007.
[2] PHMSA. (Jun. 11, 2020). Pipeline Incident 20 Year Trends. [Online]. Available: https://www.phmsa.dot.gov/data-and-statistics/pipeline/pipeline-incident-20-year-trends
[3] Z. Guo, C. Cecati, and S. X. Ding, “A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches,” IEEE Trans. Ind. Electron., vol. 62, no. 6, pp. 3757–3767, Jun. 2015, doi: 10.1109/TIE.2015.2417501.
[4] H. F. Lu, T. Iseley, S. Behbahani, and L. D. Fu, “Leakage detection techniques for oil and gas pipelines: State-of-the-art,” Tunn. Undergr. Space Technol., vol. 98, p. 15, Apr. 2020, doi: 10.1016/j.tust.2019.103249.
[5] P.-S. Murvay and I. Silea, “A survey on gas leak detection and localization techniques,” J. Loss Prevention Process Industries, vol. 25, no. 6, pp. 966–973, Nov. 2012, doi: 10.1016/j.jlp.2012.05.010.
[6] Y. A. Khulief, A. Khalifa, R. Ben Mansour, and M. A. Abib, “Acoustic detection of leaks in water pipelines using measurements inside pipe,” J. Pipeline Syst. Eng. Pract., vol. 3, no. 2, pp. 47–54, 2012, doi: 10.1061/(ASCE)PS.1949-1304.0000089.
[7] A. M. Sadeghioon, N. Metje, D. Chapman, and C. Anthony, “Water pipeline failure detection using distributed relative pressure and temperature measurements and anomaly detection algorithms,” Urban Water J., vol. 15, no. 4, pp. 287–295, Apr. 2018, doi: 10.1080/1573062X.2018.1424213.
[8] L. Wong, R. Deo, S. Rathnayaka, B. Shannon, C. Zhang, W. K. Chu, J. Kodikara, and H. Widyastuti, “Leak detection in water pipes using submersible optical optic-based pressure sensor,” Sensors, vol. 18, no. 17, Dec. 2018, doi: 10.3390/s18124192.
[9] F. Wang, W. Lin, Z. Liu, S. Wu, and X. Qiu, “Pipeline leak detection by using time-domain statistical features,” IEEE Sensors J., vol. 17, no. 19, pp. 6431–6442, Oct. 2017, doi: 10.1109/JSEN.2017.2740220.
[10] T. T. N. Luong and J. M. Kim, “The enhancement of leak detection performance for water pipelines through the renovation of training data,” Sensors, vol. 20, no. 9, p. 16, May 2020, doi: 10.3390/s20092542.
Z. Chen, K. Gryllias, and W. Li, “Intelligent fault diagnosis for rotary machinery using transferable convolutional neural network,” IEEE Trans. Ind. Informat., vol. 16, no. 1, pp. 339–349, Jan. 2020, doi: 10.1109/TII.2019.2917233.

T. G. Dietterich, “Ensemble methods in machine learning,” in Proc. Int. Workshop Multiple Classifier Syst., Cagliari, Italy: Springer, 2000, pp. 1–15.

F. Jia, S. Li, H. Zuo, and J. Shen, “Deep neural network ensemble for the intelligent fault diagnosis of machines under imbalanced data,” IEEE Access, vol. 8, pp. 129074–129082, 2020, doi: 10.1109/ACCESS.2020.3006895.

S. Ma and F. Chu, “Ensemble deep learning-based fault diagnosis of rotor bearing systems,” Comput. Ind., vol. 105, pp. 143–152, Feb. 2019, doi: 10.1016/j.compind.2018.12.012.

Y. Ren, L. Zhang, and P. N. Suganthan, “Ensemble classification and regression-recent developments, applications and future directions [review article],” IEEE Comput. Intell. Mag., vol. 11, no. 1, pp. 41–53, Feb. 2016, doi: 10.1109/MCI.2015.2471235.

G. Vrbanic and V. Podgorelec, “Transfer learning with adaptive fine-tuning.” IEEE Access, vol. 8, pp. 196197–196211, 2020, doi: 10.1109/ACCESS.2020.3034343.

A. A. Abbasi, L. Hussain, I. A. Awan, I. Abbasi, A. Majid, M. S. A. Nadeem, and Q.-A. Chaudhary, “Detecting prostate cancer using deep learning convolution neural network with transfer learning approach,” Cognit. Neurodyn., vol. 14, no. 4, pp. 523–533, Aug. 2020, doi: 10.1007/s11571-020-09587-5.

G. Tondello, W. Boruszewski, F. Meneghe, M. Assato, S. Shimizu, and S. Ziegler, “Coupled simulation of the secondary air flow, heat transfer, and structural deflection of a gas turbine engine,” in ASME Turbo Expo. Copenhagen, Denmark: AMER SOC Mechanical Engineers, Jun. 2012, pp. 2293–2305.

S. Meniconi, B. Brunone, M. Ferrante, C. Capponi, C. A. Carretti, C. Chiesa, D. Segalini, and E. A. Lanfranchi, “Anomaly pre-localization in distribution–transmission mains by pump trip: Preliminary field tests in the milan pipe system,” J. Hydroinform., vol. 17, no. 3, pp. 377–389, May 2015, doi: 10.2166/hydro.2014.038.

Q. Chen, G. Shen, J. Jiang, X. Diao, Z. Wang, L. Ni, and Z. Dou, “Effect of rubber washers on leak location for assembled pressurized liquid pipeline based on negative pressure wave method,” Process Saf. Environ. Protec., vol. 119, pp. 181–190, Oct. 2018, doi: 10.1016/j.prosafe.2018.07.023.

L. Billmann and R. Isermann, “Leak detection methods for pipelines,” Automatica, vol. 23, no. 3, pp. 381–385, May 1987, doi: 10.1016/0005-1098(87)90011-2.

E. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, “Exploiting linear structure within convolutional networks for efficient evaluation,” in Proc. 28th Conf. Neural Inf. Process. Syst. (NIPS), Montreal, QC, Canada, Dec. 2014, pp. 1267–1277.

B. Neyshabur, R. Tomioka, and N. Srebro, “In search of the real inductive bias: On the role of implicit regularization in deep learning,” in Proc. 3rd Int. Conf. Learn. Represent. (ICLR), San Diego, CA, USA: Elsevier, May 2015, pp. 2–8.

M. Geiger, A. Jacot, S. Spigler, F. Gabriel, L. Sagun, S. d’Ascoli, G. Biroli, C. Hongler, and M. Wyart, “Scaling description of generalization with number of parameters in deep learning,” J. Stat. Mech., Theory Exp., vol. 2020, no. 2, p. 23, Feb. 2020, doi: 10.1088/1742-5468/ab633c.

I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: The MIT Press, 2016, pp. 110–120.

M. Belkin, D. Hsu, S. Ma, and S. Mandal, “Reconciling modern machine-learning practice and the classical bias–variance trade-off,” Proc. Nat. Acad. Sci. USA, vol. 116, no. 32, pp. 15849–15854, Aug. 2019, doi: 10.1073/pnas.1903701116.

L. J. Ba and R. Caruana, “Do deep nets really need to be deep?” in Proc. 28th Conf. Neural Inf. Process. Syst. (NIPS), Montreal, QC, Canada, Dec. 2014, pp. 2564–2652.

X. Lang, P. Li, Z. Hu, H. Ren, and Y. Li, “Leak detection and location of pipelines based on LMD and least squares twin support vector machine,” IEEE Access, vol. 5, pp. 8659–8668, 2017, doi: 10.1109/ACCESS.2017.2703122.

X. Diao, Z. Chi, J. Jiang, A. Mebarik, L. Ni, Z. Wang, and Y. Hao, “Leak detection and location of flanged pipes: An integrated approach of principle component analysis and guided wave mode,” Saf. Sci., vol. 129, Sep. 2020, Art. no. 104809, doi: 10.1016/j.ssci.2020.104809.
MENGFEI ZHOU received the M.S. degree in chemical engineering from the Zhejiang University of Technology, Hangzhou, China, in 2004, and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, in 2010. He is currently an Associate Professor with the Zhejiang University of Technology. His research interests include process control, machine learning, fault detection and diagnose, especially leak detection and location in the pipeline.

YANHUI YANG was born in Hebei, China, in 1995. He received the B.S. degree in chemical engineering and technology from the Zhejiang University of Technology, Hangzhou, China, in 2018, where he is currently pursuing the master’s degree in chemical engineering and technology. His research interests include machine learning, deep learning, and leak detection and location in the pipeline.

YINZE XU was born in Zhejiang, China, in 1997. He received the B.S. degree in chemical engineering and technology from the Zhejiang University of Technology, Hangzhou, China, in 2019, where he is currently pursuing the master’s degree in chemical engineering and technology. His research interests include artificial intelligence, machine learning, and leak detection and location in the pipeline.

YINCHAO HU was born in Zhejiang, China, in 1998. He received the B.S. degree in chemical engineering and technology from Zhejiang Ocean University, Zhoushan, China, in 2020. He is currently pursuing the master’s degree in chemical engineering and technology with the Zhejiang University of Technology. His research interests include machine learning, and leak detection and location in the pipeline.

YIJUN CAI received the B.S. degree in industrial automation from Zhejiang University, Hangzhou, China, in 1986, and the M.S. degree in chemical engineering from the Zhejiang University of Technology, Hangzhou, in 2006. He is currently a Senior Engineer with the Zhejiang University of Technology. His research interests include integration of control systems, and fault detection and diagnose.

YANHUI YANG was born in Hebei, China, in 1995. He received the B.S. degree in chemical engineering and technology from the Zhejiang University of Technology, Hangzhou, China, in 2018, where he is currently pursuing the master’s degree in chemical engineering and technology. His research interests include machine learning, deep learning, and leak detection and location in the pipeline.

JUNJIE LIN was born in Zhejiang, China, in 1996. He received the B.S. degree in chemical engineering and technology from the Zhejiang University of Technology, Hangzhou, China, in 2018, where he is currently pursuing the master’s degree in chemical engineering and technology. His research interests include machine learning, and leak detection and location in the pipeline.

HAITIAN PAN received the B.S. degree in industrial automation and the Ph.D. degree in chemical engineering from Zhejiang University, Hangzhou, China, in 1982 and 2003, respectively. He is currently a Full Professor with the Zhejiang University of Technology. His research interests include artificial intelligence, machine learning, and fault detection and diagnose.