Research paper

A self-adaptive deep learning algorithm for intelligent natural gas pipeline control

Tao Zhang, Hua Bai, Shuyu Sun

Abstract

Natural gas has been recognized as a promising energy supply for modern society due to its relatively less air pollution in consumption, while pipeline transportation is preferred especially for long-distance transmissions. A simplified pipeline control scenario is proposed in this paper to deeply accelerate the management and decision process in pipeline dispatch, in which a direct relevance between compressor operations and the inlet flux at certain stations is established as the main dispatch logic. A deep neural network is designed with specific input and output features for this scenario and the hyper-parameters are carefully tuned for a better adaptability of this problem. The realistic operation data of two pipelines have been obtained and prepared for learning and testing. The proposed algorithm with the optimized network structure is proved to be effective and reliable in predicting the pipeline operation status, under both the normal operation conditions and abnormal situations. The successful definition of “ghost compressors” make this algorithm to be the first self-adaptive deep learning algorithm to assist natural gas pipeline intelligent control.

© 2021 Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

The world’s economic growth is mainly driven by the depletion of conventional fossil energies, but the consequent environmental problems have arisen the increasing public concerns. Natural gas, as a relatively clean resource in fossil energies, has been more preferred at the current circumstance that we are still unable to fully get rid of the reliance on fossil energy. It has been announced that the global consumption of primary energy (also known as natural energy, referring to the existing energy in nature, including coal, oil, natural gas, water energy, wind energy, etc (Ahmad and Zhang, 2020)) grows by 1.3% in 2019, only half the increase compared with 2018, but the proportion of natural gas consumption reaches about 3.98 trillion cubic meters, with a year-on-year increase of 3.5%, and a rapid growth has been witnessed in the world's major economies. The consumption in the United States was 848.1 billion cubic meters, with an increase of 3.8% over the previous year and the consumption in Europe was about 566.5 billion cubic meters, with an increase of 3.2% over the previous year. China's natural gas consumption in 2019 was 302.5 billion cubic meters, with a significant increase of 9.0%, including 170.5 billion cubic meters of natural gas, a year-on-year increase of 9.1%, and the import volume of 133 billion cubic meters, with a year-on-year increase of 7.1%. Pipeline transportation is preferred in long-distance natural gas transmission from the production and storage facilities to the marketing and consumption regions, due to economic and safety reasons (Wen et al., 2019; Gas). In 2019, the global gas trade volume through pipelines has reached about 859.6 billion cubic meters, with an increase of 6.7% over the previous year. In China, the total mileage of the natural gas national main pipeline network is nearly 81000 km, and the annual gas transmission capacity is more than 350 billion cubic meters. Thus, a safe and reliable operation on natural gas pipelines is essentially required for the stable energy supply to our modern society.

In the conventional industrial practice of natural gas pipeline control, manual regulations by pipeline dispatchers based on empirical decisions have always been the main approach (Szoplík, 2016; Chaczykowski, 2010). The rapid development of artificial intelligence (AI) technologies has promoted the deeper integration of traditional industrial manufacturing together with automation and informatization (Zhang et al., 2020a,b). Upgrading to intelligent pipeline control and dispatch has become the focus of natural gas pipeline operation and management, in which the remote monitoring and decision capabilities are expected to be
leveled up. Several pipeline intelligent dispatch and control systems were proposed in Hao et al. (2018), Kyriakides and Polycarpou (2014), Nianzhong et al. (2005) and here we propose a generalized structure. As illustrated in Fig. 1, the framework starts with the construction of an enterprise-level central database combining various systems including Supervisory Control And Data Acquisition (SCADA) System (Kruzt, 2005), Pipeline Production System (PPS) (Gas), Plant Information System (PIS) (Oz et al., 2019) and Pipeline Integrity Management System (PIMS) (Oviedo and Malpartida Moya, 2019) to support further decision and execution. The meteorological data was also suggested to take into consideration in Viljanen et al. (2006). The virtual brain of intelligent dispatch and control is working with professional process analysis and advanced pipeline simulation to assist the decision making of pipeline schedulers and dispatchers on various dispatch and control scenarios. AI techniques are applied to transform the traditional empirical decision to scientific decision, to promote the working efficiency and capabilities of pipeline staffs and to accelerate the progress of pipeline digitalization, promptness, automation and informatization.

The general natural pipeline control process involves the compressor and valve openess, which further regulates the pressure and flux at the inlet and outlet of each station along the pipeline. During pipeline scheduling process, such parameters are also the main index considered for medium-and-long-term pipeline operations (Wen et al., 2018). In the conventional steady-state pipeline management, all the compressors along the pipelines are assumed to keep the normal working conditions in the process and the optimized pipeline control schedule is established based on the minimized energy consumption and maximized economic earnings (Liu et al., 2019; Altiparmak et al., 2009). However, the changing compressor variables representing the unsteady working conditions challenges these management plans, while special measurements are needed to take place to bring a new steady state of the pipeline operation indexes (Chaczykowski and Zarodkiewicz, 2017). Modeling of natural gas pipelines has been recognized as a reliable approach to predict the transient pipeline operation situations with the unsteady conditions including compressor failure or pipe leakage (Ke and Ti, 2000; Baumrucker and Biegler, 2010; Madoliat et al., 2016a). The mathematical scheme is always complex for describing the long-distance main pipeline network, and various techniques have been successfully applied to reduce the complexity and computational costs. The radial basis function surrogates and proper orthogonal decomposition reduction techniques were introduced in Grundel et al. (2013), the electronic circuit concepts was proposed in Ke and Ti (2000), a linear state-space description using Taylor approximations was developed in Wen et al. (2018), the third-order Hermite polynomials were used in to spatially discretize the pipeline network (Baumrucker and Biegler, 2010), and a fully-implicit method was derived in Madoliat et al. (2016b) with particle–swarm optimization. However, these mathematical schemes are still generated based on the hydraulic formulations, while the computation cost is often expensive due to the inevitable iterations required in general algorithms including Finite Different Methods (Chaczykowski and Zarodkiewicz, 2017), Finite Volume Methods (Guandalini et al., 2017), Finite Element Methods (Behbahani-nejad et al., 2019) and Spectral Element Methods (Dorao and Fernandino, 2011).

In this paper, an exploratory investigation is presented to deeply accelerate the natural gas pipeline control processes with the application of deep learning algorithms. A simplified pipeline control scenario is proposed first in Section 2, concluded based on the author’s working experience in pipeline controlling centers. The deep neural network is designed in Section 3 with the input and output features selected specifically aiming at the scenario, and a two-network structure is constructed to uniform the number of effective compressors along the pipelines. The network designed in this paper is a deep neural network because a large number of hidden layers are constructed between the input and output layer, which is believed to significantly improve the prediction reliability. The number of hidden layers is carefully tuned to achieve a better training and prediction performance. We have added this statement in the Introduction section. Data is obtained from realistic pipeline operation records and certain deep learning techniques are involved to overcome the overfitting problem caused by limited number of data. The network hyper-parameters are carefully tuned in Section 4, and the optimized network is used for further learning and predictions. The network performance is good for both cases with or without padding the “ghost compressors” in testing data, while the failure in padding the training data is explained and discussed in Section 5.

2. Physical model

The pipeline control process is simplified based on the consumption that the inlet flux at the certain station in one certain pipeline is solely determined by the operations on the compressors along such pipeline under the condition of keeping the initial flux constant in the process. In practice, the dispatcher in the controlling center indeed monitors the inlet flux at the key stations and operates on the compressors to maintain a stable pipeline transportation. Thus, the input and output features in our deep neural network are determined as the compressor operations and inlet flux respectively, and the aim of our deep learning algorithm is to predict the inlet flux at certain station as a consequence of compressor operations using the trained model. In previous studies, a complex hydraulic model needs constructing first and numerical simulations are performed to predict the flux, while the numerical error is hard to avoid in both the model derivation and algorithm convergence. As an effective approach approximating directly the production data, numerical errors introduced in conventional modeling and simulation can be eliminated in the deep learning algorithms, while the complex relationships between compressor operations and inlet flux are represented now by the trained model.

Two pipelines are considered in this paper, namely Pipeline 1 and Pipeline 2, with 9 stations and 7 stations respectively, and the number of compressors at each station as well as the operation status in different periods are listed in Tables 1 and 2. It can be referred from the two tables that there was one station working in the “offline” mode in both the two pipelines during the period 2017–2018, while three stations have been working in the “offline” mode from 2019 until now (2020). The label “offline” recorded in the data collected from the pipeline data acquisition system may refer to the maintenance in such stations or network errors, thus no effective information can be obtained in these stations. An implicit condition in the deep learning algorithm designed in this paper is the constant fluid and thermodynamic properties of the transported gas. This condition is accepted by our industrial partners because generally a long-distance natural gas pipeline will only transport certain gas from one certain resource. If the gas properties have been changed, the current dispatch systems should also be adjusted because a different hydrodynamic formulation should be developed, which is not preferred. Thus, the simplified scenario and the corresponding deep learning algorithm is meaningful to the natural gas pipeline controllers.
Table 1

<table>
<thead>
<tr>
<th>Pipeline</th>
<th>Station</th>
<th>Number of compressors</th>
<th>Operation status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pipeline 1</td>
<td>A</td>
<td>4</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>4</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>3</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>4</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>4</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>3</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>G</td>
<td>4</td>
<td>Offline</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>4</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>3</td>
<td>Normal</td>
</tr>
<tr>
<td>Pipeline 2</td>
<td>J</td>
<td>4</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>K</td>
<td>4</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>3</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>4</td>
<td>Offline</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>4</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>O</td>
<td>3</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>4</td>
<td>Normal</td>
</tr>
</tbody>
</table>

Table 2

<table>
<thead>
<tr>
<th>Pipeline</th>
<th>Station</th>
<th>Number of compressors</th>
<th>Operation status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pipeline 1</td>
<td>A</td>
<td>4</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>4</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>3</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>4</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>4</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>3</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>G</td>
<td>4</td>
<td>Offline</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>4</td>
<td>Offline</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>3</td>
<td>Offline</td>
</tr>
<tr>
<td>Pipeline 2</td>
<td>J</td>
<td>4</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>K</td>
<td>4</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>3</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>4</td>
<td>Offline</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>4</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>O</td>
<td>3</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>4</td>
<td>Normal</td>
</tr>
</tbody>
</table>

3. Deep learning algorithm

In this section, the network structure designed for the prediction of inlet flux is presented in details as well as the deep learning techniques applied for a better prediction performance. The self-adaptive network scheme can be used for handling the unexpected shut down or network error in some of the stations, while the dimension of input features may change as well as the nodes connections. The pipeline operations in an abnormal status will be recorded and the size of such digital data will be automatically uniformed with that of the training data. Ghost stations are defined to replace the offline stations at the same location in the data padding process, and the compressors are set to be always closed in these stations. This design is verified in Section 4 on the reliability and efficiency, and certain selected techniques developed for deep learning algorithm are also applied in our study. Meanwhile, the obtained operation data need to be processed first to be better prepared for training and testing.

3.1. Network construction

The designed artificial neural networks are expected to be feasible of extracting the key features of input data and discovering the underneath relationship between the output and input. In order to accomplish the pipeline control task described in Section 2, the input features and output target are determined as compressor operations and inlet flux at certain station respectively. The two-network structure proposed in Zhang et al. (2020a) are introduced to handle the varieties between the size of input data during the operation period 2017–2018 and 2019–2020. As shown in Fig. 2, the input features are determined to be the N compressor Boolean indexes, defined as “0” for compressor stopping and “1” for compressor running, also written as CB. The subscript denotes the compressor location and the output of the first padding network is of dimension M. The parameters M and N denote the number of compressors with effective operation record (not offline) in the training and target period respectively. The output prediction is the flux at the ith certain key station, denoted by Vi. For example, in the case with the production data during the period 1 (2017–2018) used as training data in this paper, M equals 29 for Pipeline 1 and 22 for Pipeline 2. If the trained model is used to predict the flux during the same period, N equals 29 for Pipeline 1 and 22 for Pipeline 2 as well. However, if the trained model is used to predict the flux during period 2 (2019–2020), N equals 22 for Pipeline 1, in which M − N = 7 “ghost compressors” are needed to pad the testing data for the sake of uniforming the data dimensions. The “ghost compressors”
are defined at the locations same as in the offline stations, and set as always stopped during the operations. In another word, the intelligent pipeline control is assumed to be operating on the effective compressors only, which is in fact the real case. The self-adaptive network designed in this paper is capable of automatically padding the testing data if needed after reading the training and testing data.

It is also interesting to investigate the other case, in which the dimension of training data is less than that of testing data. For example, if we want to predict the flux during Period 1 using the model trained with production data in Period 2, 7 “ghost compressors” are needed to pad the training data. However, it should be noted that the pipeline control scenario is quite different with the phase equilibrium scenario studied in Zhang et al. (2020a), as the selected input features are determined based on a simplified physical reality. In another word, the calculated weights for the “ghost compressors” cannot describe accurately the work of the online compressors in the normal operations. A validation is performed in Section 4.3.

The following equation models the simulation in each node in the hidden layers, as shown in Fig. 3, y_{i} = f_{3}(W_{1} \ast CB_{i} + b_{i}), \quad (1)

where y_{i} and CB_{i} denotes the output and input of this layer respectively, f denotes the activation function applied in this layer and b_{i} denotes the bias introduced for a better performance. W_{1} is the trained parameter evaluating the work of each compressor done to maintain the flux in the pipeline. If multiple hidden layers are included in the network structure, the input of the next layer is the output of the previous one, and the following equation is formulated to describe the scheme (using three activation layers as an example):

v = f_{3}(W_{1} \ast f_{2}(W_{2} \ast f_{1}(W_{1} \ast CB_{i} + b_{1}) + b_{2}) + b_{3}), \quad (2)

where CB_{i} denotes the initial input features, e.g. compressor boolean indexes, W_{1}, i = 1, 2, 3 denotes the weights in each layer i, b_{i}, i = 1, 2, 3 denotes the bias in each layer and v denotes the final output, e.g. the inlet value at certain station.

The self-adaptive deep learning algorithm is designed to automatically adjust its structure to pad the “ghost compressors” in the training or testing data as needed. For example, if operation in period 2 is used as training and operation in period 1 is used as testing, M − N = 7 “ghost compressors” will be padded into the training data, modifying the CB_{i} in Eq. (2) for training to be:

CB_{i} = \{CB_{1}, CB_{2}, CB_{3}, \ldots, CB_{N}, CB_{N+1}, CB_{N+2}, \ldots, CB_{M}\}, \quad (3)

and the final output v in the regression layer remains as v_{a}. On the other hand, if operation in period 1 is used as training and operation in period 2 is used as testing, 7 “ghost compressors” will be padded into the testing data, requiring 7 more compressor boolean values added into CB_{i} in Eq. (2) for the testing process. All the CB_{N+1}, CB_{N+2}, \ldots, CB_{M} are always set to be zero, indicating non-active for the “ghost compressors” in the pipeline control.

As the amount of realistic operation data is always limited, overfitting is hard to avoid in our studies. This common-seem issue often occurs when too many parameters are involved in the model training, and the data supported for training and testing is not enough to obtain a reliable weight vector for the input features representing the underneath physical rules. Usually, a poor performance is resulted in the model validation of testing data, while a perfect performance can be achieved in the training process after tuning the hyper-parameters. This problem is also known as over-parameterized model, and an additional constraint is a common solution to reduce the damage caused by overfitting to our final trained model. In practice, an extremely large norm can be obtained for the weight parameters of certain input features, especially for the boolean values used in this study, which is the main cause of overfitting. In order to resolve this issue, the loss function can be constructed with a penalization term for the large weights, as shown in Eq. (4)

\begin{equation}
L = \frac{1}{N} \sum_{n=1}^{N} \|v - \hat{v}\|^2 + \lambda \|W\|^2. \quad (4)
\end{equation}

where \(v\) and \(\hat{v}\) denote the realistic production data and model prediction of the inlet flux respectively, \(N\) denotes the number of training data and \(L\) is the loss function with a L2 penalization term of weights \(W\) with the regularization coefficient \(\lambda\).

3.2. Data preparation

Operation data of the compressors along the pipelines and the inlet flux at certain station during the period of 2017–2020 are collected from the pipeline data acquisition system for training and testing in this paper. The operation of all the compressors and the averaged hourly flux in one single day are organized in one data point. The compressor status “running” and “stopping” are translated to be the boolean value “1” and “0” respectively, and the translated digital matrix containing operation and flux information is evaluated first to eliminate outliers. For example, there remains certain compressor status as “not connected”, maybe due to the problem in data transmission or network connection in the corresponding station, while we have no idea of whether the compressor is working normally or not. All the operation and flux data in these dates involving such outliers are eliminated from the ground-truth data set to avoid unexpected variance. It should also be noted that the inlet flux generally lies in the range 8000 m³/h - 13 000 m³/h, in a much larger order of magnitude compared with the operation boolean data. Thus, the unit of flux is transferred to be ×10⁴ m³/h to avoid the possible too large weights. Totally, we have 1810 data points available as the ground truth after above preparations. Usually, thousands of parameters are involved in the deep neural network, which requires a large of samples to feed the training and testing. The current limited number of realistic production data is easy to cause overfitting in the training, thus certain advanced deep learning techniques should be applied to reduce the damage of overfitting, as introduced in Section 3.3.

3.3. Deep learning techniques

In the designed deep neural network, the weight parameters are determining the work of each compressor done to control the pipeline operation and to maintain the inlet flux at certain stations. As a result, the final trained weight values are playing a critical role in obtaining a reliable trained model to accurately simulate the pipeline control process, and a proper initialization is expected to accelerate the training convergence (loss function decay) to construct a robust and efficient deep neural network structure. In details, the input variance may rapidly increase if an overestimated weight initialization is conducted when constructing the neural network, which the training process may be failed due to no convergence as the consequence of gradient vanishing or explosion. On the contrary, if an underestimated initialization is conducted for the weight parameters, the input variance may rapidly drop to a very small value, damaging the model complexity and causing the unsatisfactory performance of the trained model. Xavier initialization is a common-used approach to control the signal variance following the Gaussian distribution, by which the output and input variance of one layer
should keep the same, as modeled in the following formulation for the process described in Fig. 3:

\[ \text{var}(y) = \text{var}(w_1 \cdot CB_1 + w_2 \cdot CB_2 + \ldots + w_M \cdot CB_M + b) \]
\[ = \text{var}(w_1) \cdot \text{var}(CB_1) + \text{var}(w_2) \cdot \text{var}(CB_2) + \ldots \]
\[ + \text{var}(w_M) \cdot \text{var}(CB_M) \]
\[ (\ast) \quad M \cdot \text{var}(w_i) = \text{var}(CB_i), \]

where the equivalence (5) is meaningful only if \( w_i \) and \( CB_i \) are identity distributed, and the following constraint is needed:

\[ M \cdot \text{var}(w_i) = 1. \]  

(6)

It can be easily referred from Eq. (6) that the weight parameters are following a Gaussian distribution with the variance \( 1/M \), which could be a good trial in the initialization.

As explained in Sections 3.1 and 3.2, overfitting is hard to avoid in the deep learning algorithm developed for intelligent pipeline control, while dropout and batch normalization are two approaches solving this issue. The idea of dropout is to randomly discard certain nodes in the network to reduce the high complexity and freedom, and the technique of batch normalization can be described as normalizing the data in input and hidden layers. Both the two approaches have been successfully applied in the fast prediction of phase equilibrium in, where the complex thermodynamic correlations of the selected input features are simulated efficiently and accurately. It is expected that these techniques can also be applied in this study to improve the performance of our trained model in predicting the pipeline control situations.

4. Results

Hyper-parameters are the key information describing a specific deep neural network, including the selection of activation function, the number of nodes in one hidden layer and many others. Tuning the hyper-parameter is essentially needed to construct a good network efficiently training the model with a high convergence rate and a small validation error. After the network hyper-parameters are determined, the optimized trained model using operation data during period 1 (2017–2018) is applied in two cases: to predict the inlet flux during period 1 and to predict during period 2. The dimensions of training and testing are various in the second case, thus the padding network is effective to transform the size of testing data. It is also worth trying the prediction of the inlet flux in Period 1 using the data in Period 2, in which the training data is padded.

4.1. Hyper-parameter tuning

The number of nodes in each layer is directly relevant with the complexity of the network and the training efficiency. Generally speaking, more nodes in one layer can accelerate the decrease of loss function, and less prediction error is expected using the trained model. As shown in Fig. 4, a relatively slower decrease can be detected in the loss function curve changing with training steps if 100 nodes are placed in the hidden layers, while the performance of 200 nodes and 300 nodes seem to be similar. A significant enhancement is illustrated if 500 nodes are used, which indicates that indeed the network structure with more nodes in the hidden layers can ensure a better model performance. However, it can be referred from Fig. 5 that the mean relative error of inlet flux prediction with different numbers of nodes in the hidden layers does not change much, which indicates a severe overfitting problem occurred in the training with a large number of nodes. In all, 200 nodes in one hidden layer is the best choice considering the impact on both loss convergence rate and estimation errors.

Activation function, the main contributor in the network to simulate the non-linearity underneath the pipeline control process, is another key hyper-parameter needing to be tuned carefully for a better training and prediction performance. The performance of deep neural networks with various activation functions
depend on the problem characteristics and the choice of input features and output results. As shown in Fig. 6, the loss function decreases significantly faster if the Softsign activation function is used, while the network with ReLU and Softplus show similar performance in loss convergence and the network with Sigmoid seems to be the worst. However, as overfitting is serious in this case, the prediction error in testing is more convincing to determine the appropriate activation function. It can be referred from Fig. 7 that the performance of Softsign function is surprisingly bad, while the prediction accuracy of Sigmoid is as well as that of the Softplus function. In all, the Softplus function is the best choice due to the relatively faster loss convergence and lower estimation errors.

Based on the above analysis, 200 nodes in one hidden layer and the Softplus function are selected in the optimized network structure for further investigations.

4.2. No-padding deep learning

During Period 1 (2017–2018), 29 compressors are in normal operation condition along Pipeline 1 and 22 compressors are in normal operation condition along Pipeline 2. For Pipeline 1, the inlet flux at Station A is the main index considered in practical pipeline control, and the inlet flux at Station P is the main index for Pipeline 2. If the deep learning model trained with the operation data during Period 1 is used to predict the pipeline control index during the same period, the input dimension of the training and testing data keep the same so that no padding is needed. Namely, the first padding network in Fig. 2 is skipped and we directly move to the training network. Operations on the compressors along the two pipelines from March 11th, 2017 to March 14th, 2017 are illustrated in the left of Fig. 8 and Fig. 9, and the corresponding inlet flux at Station A and P are illustrated in the right of the two figures respectively. The compressor boolean value is plotted to represent the compressor operations, where the value 1 means “compressor open” and 2 means “compressor stop”. It can be seen that the inlet flux at certain stations changes as a consequence of pipeline controls on the compressors, and the prediction of our trained deep learning model meet well with the practical operation data. The reliability of our proposed self-adaptive deep learning algorithm is verified, and the simplified pipeline control scenario, e.g. operation on the compressors to affect station flux, has been verified.

4.3. Padding deep learning

During Period 2 (2019–2020), 22 compressors have been working in normal conditions due to the offline of three stations. If the model trained during Period 1 is used to predict the pipeline control index during Period 2, e.g., the inlet flux at Station A, padding is needed in the testing data, which means that the padding network in Fig. 2 is playing a role in this case. 7 “ghost compressors” are added at the locations where offline ones are, and set to be always stopped. The loss function changing with training steps and the prediction error in testing are illustrated in Figs. 10 and 11 respectively, showing a rapid convergence rate and a small testing error, which are both acceptable. The trained model is then applied to predict the inlet flux of Station A from January 1st to January 4th, 2019 in Pipeline 1, and the operations on compressors are plotted in the left of Fig. 12 and
the corresponding flux are plotted in the right. It can be stated that the model trained with the padding deep learning algorithm works well with the prediction of pipeline control inlet, e.g. the specific inlet flux at certain station with a relatively very small error, and the result is meaningful for pipeline controllers to determine the compressor operations.

It is also interesting to investigate the performance of our proposed self-adaptive deep learning algorithm if the training data is padded, for example, to predict the pipeline control index during Period 1 using the model trained using the data during Period 2. The testing error is plotted in Fig. 13, which indicates that the trained model fails to describe the underneath correlations between compressor operations and station flux. This failure is reasonable as the added “ghost compressors” in the training data damages the realistic hydraulic rules along the pipeline. In another word, the work of the compressors in the testing data corresponding to the “ghost compressors” in training data is not correctly modeled by the calculated weight parameters.
5. Conclusion and remarks

Natural gas has been recognized as a relatively clean supplement to the world’s energy market, and pipeline transportation is preferred in practice due to economic and safety reasons. In this paper, a simplified scenario is designed for natural gas pipeline control, in which the inlet flux at certain station is defined as the control index, and as a consequence of operations on the compressors along the pipelines. A self-adaptive deep learning algorithm is proposed aiming at this physical problem, and a two-network structure is constructed. Compressor operations, represented by compressor boolean values, are used as network input features, station flux is used as network output result and a padding network is introduced to uniform the dimension of training and testing data. The operation data of two realistic pipelines are obtained and processed to be prepared for further learning, and two operation periods are determined with different numbers of compressors in normal working conditions. For Pipeline 1, more compressors are effective in control during Period 1 (2017–2018) compared with Period 2 (2019–2020). The key hyper-parameters in the deep neural network, including the number of nodes in one hidden layer and the activation function, are tuned first, and the optimized network is then applied in both padding and non-padding deep learning tasks. The reliability of our proposed deep learning algorithm has been verified by the very small prediction errors for all the cases, the robustness has been verified by the good performance for both the two pipelines and the self-adaption has been verified by the learning in which the dimensions of input features in training and testing data are different. This reliable, robust and self-adaptive deep learning algorithm is promising in intelligent pipeline control, and the proposed simplified intelligent control scenario can significantly accelerate the decision process of pipeline controllers by quickly providing a reliable estimation of the inlet flux at certain stations as a consequence of operations on alongside compressors. The intelligence of this algorithm is raised based on the automatic prediction of key gas flux at certain station as a consequence of compressor operations, and the automatic network structure adjustment for different pipelines. The application of deep learning techniques helps us achieve this intelligence.

In practical pipeline dispatch process, the controller need to operate on the compressors to ensure certain essential and safe flux at the key station if unexpected malfunction or interruption occurred along the pipelines. The simplified scenario designed in this paper aims to develop a quick prediction system that is capable of handling various operating conditions including compressor failure, planned maintenance and interruption in the data transmission. The corresponding deep learning algorithm is then developed to provide a quick prediction under the unexpected conditions. The wide adaptability has been verified by the various numbers of compressors along the pipeline and sudden change in the compressor operating conditions (corresponding to compressor Boolean value changing from 1 to 0). It has been illustrated in Section 4.3 that the performance of our deep neural network is bad if the training data is padded with “ghost compressors”. The failure is reasonable as the target pipeline control problem is a simplified scenario, and the simple compressor operations cannot fully describe the work of offline stations to determine the proper weight parameters. A potential direction in future researches is to investigate deeper the thermal-hydraulic processes (Li et al., 2021) controlling the flow and transport in natural gas pipelines, so as to extract more input and output features representing the underneath physical rule more comprehensively. Besides, currently we assume that the transient properties caused by special operations like compressor shutdown or failure is reflected by the zero compressor Boolean value. In the future, we can classify further the full shutdown or partial failure to improve the adaptability of our deep learning algorithm.

CRediT authorship contribution statement

Tao Zhang: Conceptualization, Methodology, Writing. Hua Bai: Data, Methodology. Shuyu Sun: Supervision, Writing – review & editing.
Declaration of competing interest

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

Acknowledgments

The work of Tao Zhang and Shuyu Sun was supported by funding from the National Natural Scientific Foundation of China (Grants No. 51874262) and King Abdullah University of Science and Technology (KAUST), Saudi Arabia through the grants BAS/1/1351-01-01.

References