Machine Intelligence Research
www.mi-research.net

22(1), February 2025, 117-130
DOI: 10.1007/s11633-024-1523-3

End-to-end Identification of Autoregressive with Exogenous Input (ARX) Models Using Neural Networks

Aoxiang Dong Andrew Starr Yifan Zhao

School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield MK43 0AL, UK

Abstract: Traditional parametric system identification methods usually rely on apriori knowledge of the targeted system, which may not always be available, especially for complex systems. Although neural networks (NNs) have been increasingly adopted in system identification, most studies have failed to derive interpretable parametric models for further analysis. In this paper, we propose a novel end-to-end autoregressive with exogenous input (ARX) model identification framework using NNs. An order-wise neural network structure is introduced and trained using a multitask learning approach to simultaneously identify both the model terms and coefficients of the ARX model. Through testing with various neural network backbones and training data sizes in different scenarios, we empirically demonstrate that the proposed framework can effectively identify an arbitrary stable ARX model with finite simulation training data. This study opens up a new research opportunity for parametric system identification by harnessing the power of deep learning.

Keywords: Linear system identification and estimation, learning systems, model structure determination, multivariable systems, deep learning.

Citation: A. Dong, A. Starr, Y. Zhao. End-to-end identification of autoregressive with exogenous input (ARX) models using neural networks. *Machine Intelligence Research*, vol.22, no.1, pp.117–130, 2025. http://doi.org/10.1007/s11633-024-1523-3

1 Introduction

Parametric system identification, which is the process to uncover the inherent dynamics of a system based on the model built with the observed inputs and outputs data, can be used to control, analyse, or design a complex system[1]. For linear system identification, the autoregressive with exogenous input (ARX) model is widely adopted due to its transparency for causality analysis, pole-zero analysis, and spectrum analysis[2–4]. The derivation of an ARX model identification typically involves two parts: 1) identification of the model order, and 2) estimation of the model coefficients. Various statistical approaches have been proposed in the past few decades to identify ARX models. The parameters of an ARX model are commonly estimated with ordinary least squares offline for the linear time-invariant (LTI) system, and recursive least squares, Kalman filter online for the linear time-varying (LTV) system[5–7]. Then the model order can be selected according to likelihood-based methods such as the Akaike information criterion (AIC), and Bayesian information criterion (BIC)[8–11]. To avoid exhaustive searching and the tendency of overfitting, sparse regression methods like the least absolute shrinkage and selection operator (LASSO) and error reduction ratio-based orthogonal forward regression (ERR-OFR) have been proposed and widely adopted[12–15]. However, this process usually relies on apriori knowledge of the targeted system, which may not always be available and sometimes are unlikely, especially for complex systems.

With the rapid increase of computing power, deep learning has drawn more attention in the area of time series modelling and system identification due to its less reliance on prior knowledge and prominent universal approximation ability. Various neural network structures such as the vanilla recurrent neural network (RNN), long short-term memory (LSTM), gated recurrent unit (GRU) and temporal convolutional neural network (TCN) have been developed to learn the system input-output mapping[16–20]. However, due to the complexity of the neural network structure, it is difficult to carry out the causality analysis, pole-zero analysis, and spectrum analysis based on a neural network model.

Although the majority of the research focuses on improving the prediction accuracy of the model instead of the model′s interpretability, the derivation of parametric mathematic models from neural networks has been investigated in some research. In [21], the system input-output mapping can be learned by a single hidden feed-forward neural network (FNN) or an output-feedback (Jordan) RNN, and then the generalized frequency response function (GFRF) has been derived through truncated Volterra series expansion of the activation function.

Research Article
Manuscript received on April 3, 2024; accepted on August 10, 2024
Recommended by Associate Editor Huiyu Zhou
Colored figures are available in the online version at https://link.springer.com/journal/11633

Similarly, in [22], the ARX model and linear transfer function have been derived by approximating the activation function of multi-layer perceptron (MLP) with the first two terms of Taylor series expansion to avoid nonlinearity. One limitation of this kind of method is that it is almost impossible to obtain a parametric mathematic model from neural networks with more complex structures that learn input-output mapping better, such as LSTM, GRU and TCN, using a series of expansion.

On the other hand, neural networks have been adopted to extract useful information from system inputs and outputs in the process of learning input-output mapping, which improves its interpretability. In [23], the Granger causality between inputs and outputs has been detected by spatial attention threshold and permutation importance validation based on a depth-wise TCN which is trained with system inputs and outputs. In [24], the Granger causality has been derived by applying the group LASSO, hierarchical LASSO and sparse group LASSO penalty on the first hidden layer weights of the component-wise MLP, RNN and LSTM. The maximum time delay and specific time delay which reveal the system order can only be detected with the component-wise MLP since different time steps share the same weights in RNN and LSTM. However, parametric mathematic models have not been derived in this research, which makes it impossible to carry out pole-zero analysis and spectrum analysis.

In other studies, neural networks have been utilised to determine the model order to facilitate the derivation of parametric models. In [25], the residual convolutional neural network (ResNet) has been trained to select the autoregressive integrated moving average (ARIMA) model order based on a low-resolution image of the autocorrelation function (ACF) and partial autocorrelation function (PACF) of the historical system output. In [26], the ResNet has been trained to select the autoregressive moving average (ARMA) model order based on the fixed length system output data directly instead of images. In these studies, neural networks are pre-trained with simulated data generated from systems under a selected maximum order with randomly generated parameters. After training, the neural network can be applied directly to select the system order with unknown parameters. However, the estimation of coefficients still relies on conventional regression approaches. Furthermore, the system with observable excitation (exogenous input) has not been considered.

Using neural networks directly to model systems has been widely used for predicting the system output, but this approach usually lacks interpretability for further analysis such as pole-zero analysis or spectrum analysis. Such an analysis can be crucial in control and reverse engineering. This paper is motivated by the lack of neural network-based identification approaches for ARX models, which can estimate model terms and coefficients simultaneously without relying on detailed prior knowledge about the studied system. With the training and validation data generated by simulation, a novel end-to-end ARX model identification framework was proposed to identify the ARX model of any linear stable SISO systems. In this framework, an order-wise neural network structure is introduced and can be trained with a multi-task learning approach so that both model terms and coefficients of the ARX model can be identified in an end-to-end manner. Finally, the framework is tested with various ARX models. In testing, RNN and TCN are adopted as backbones of proposed order-wise neural networks. Their performance is evaluated and compared in different scenarios.

2 Methodology

The general ARX model representing a linear single-input single-output (SISO) system is shown in (1).

$$y(k) = \sum_{i=1}^{n_y} a_i y(k-i) + \sum_{j=0}^{n_u} b_j u(k-j) + \mathrm{e}(k) \tag{1}$$

where n_y and n_u are the maximum time delay of the system output y and exogenous input u respectively; a_i and b_j are the coefficients of the output and input delay terms; $e(k)$ is the white noise at the time k. According to the poles and zeros of the underlying linear SISO system, the output and input delay terms could be sparse. An effective identification approach should be able to detect correct delay terms and estimate proper coefficients even when the signal-to-noise ratio (SNR) is low.

To identify the linear SISO system, the flowchart of the proposed end-to-end ARX model identification framework using neural networks is presented in Fig. 1. It includes three main steps: producing training and validation data via simulation, training and fine-tuning neural networks with simulated data, and deploying trained neural networks to identify the ARX model from the input and output data of linear SISO systems directly. The performance of the proposed framework will be assessed and compared in testing. The detail of each step will be introduced in the following sections.

2.1 Data generation via simulation

This end-to-end ARX model identification framework starts with data generation. The data used for training and validation are generated from simulations. To ensure the neural networks can be trained to identify the corresponding ARX model of the actual system with simulated data, the hypotheses of data generation are described as follows:

Assumption 1. Let Y be an ARX function space of simulated models in the training set. The bases of Y are

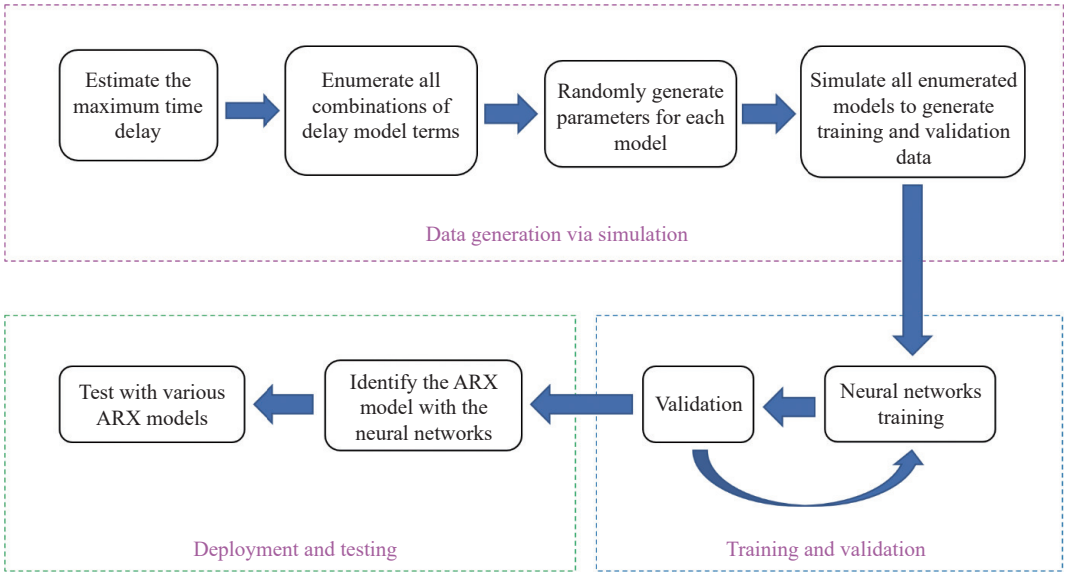


Fig. 1 Flowchart of the end-to-end ARX model identification framework

$\{y(k-1), \cdots, y(k-\widehat{n_y}), u(k), \cdots, u(k-\widehat{n_u})\}$, where $\widehat{n_u}$ and $\widehat{n_y}$ are estimated possible maximum time delays of the input u and output y of an SISO system, respectively. Any element in Y can be represented by the linear combination of bases. Let $y(k)$ be the corresponding ARX model of the actual linear SISO system to be identified whose maximum time delay of the input and the output are n_u and n_y. It is assumed that $\forall \widehat{n_y} \geq n_y$, $\widehat{n_u} \geq n_u, \exists Y =$

$$\begin{aligned}&\{(y(k-1), \cdots, y(k-\widehat{n_y}), u(k), \cdots,\\ &u(k-\widehat{n_u}))\,|k \in \mathbf{N}_{>0}\},\\ &\text{s.t.}, y(k) \in Y.\end{aligned}$$

Assumption 2. The neural network model trained from the simulation data consisting of a finite set of simulated linear ARX models in Assumption 1 can be used to identify an arbitrary stable linear ARX model with the maximum time delays of n_u and n_y.

The first step of the simulation is to estimate the possible maximum time delay $\widehat{n_u}$ and $\widehat{n_y}$ of the input and output of the system to be identified, then all combinations of input and output delay terms should be enumerated to produce Y. The number of enumerated ARX model structure N_e can be described as

$$N_e = \sum_{i=1}^{\widehat{n_u}} \sum_{j=1}^{\widehat{n_y}} C_{\widehat{n_u}}^{i} C_{\widehat{n_y}}^{j}. \tag{2}$$

For each enumerated model structure, n sets of coefficients are generated randomly with a uniform distribution, meanwhile, the stability condition requiring characteristic roots to be inside the unit cycle is imposed. Then the total number of models to be simulated N_m will be

$$N_m = nN_e. \tag{3}$$

To generate simulated output data, a finite number of time steps of observed input excitation of the system to be identified will serve as the inputs for all simulated models. Then both inputs and outputs of the simulated models become the input features for the neural network. The corresponding label for each pair of input and output of a simulated ARX model is a vector. To create the label vector, the first step is to generate a multi-hot vector where each bit of vector indicates whether a delay term exists. The length of the label vector L can be expressed as

$$N_L = \widehat{n_u} + \widehat{n_y}. \tag{4}$$

If a delay term exists, the bit that is one will be replaced by the corresponding coefficient. As a result, the label vector can represent both model coefficients and delay terms. The detail of the data generation is illustrated in Fig. 2.

In this work, the training, validation and test set are generated separately following the steps described above, so that the neural network is tested with ARX models consisting of all combinations of delay terms with random coefficients not shown in the training and validation set. The training set contains 60% of the data, while the validation set contains 20% of the data and the test set contains 20% of the data.

2.2 Training and validation of the order-wise neural networks

In this paper, an order-wise neural network structure is proposed to identify each delay term and estimate the corresponding coefficient independently, because learning

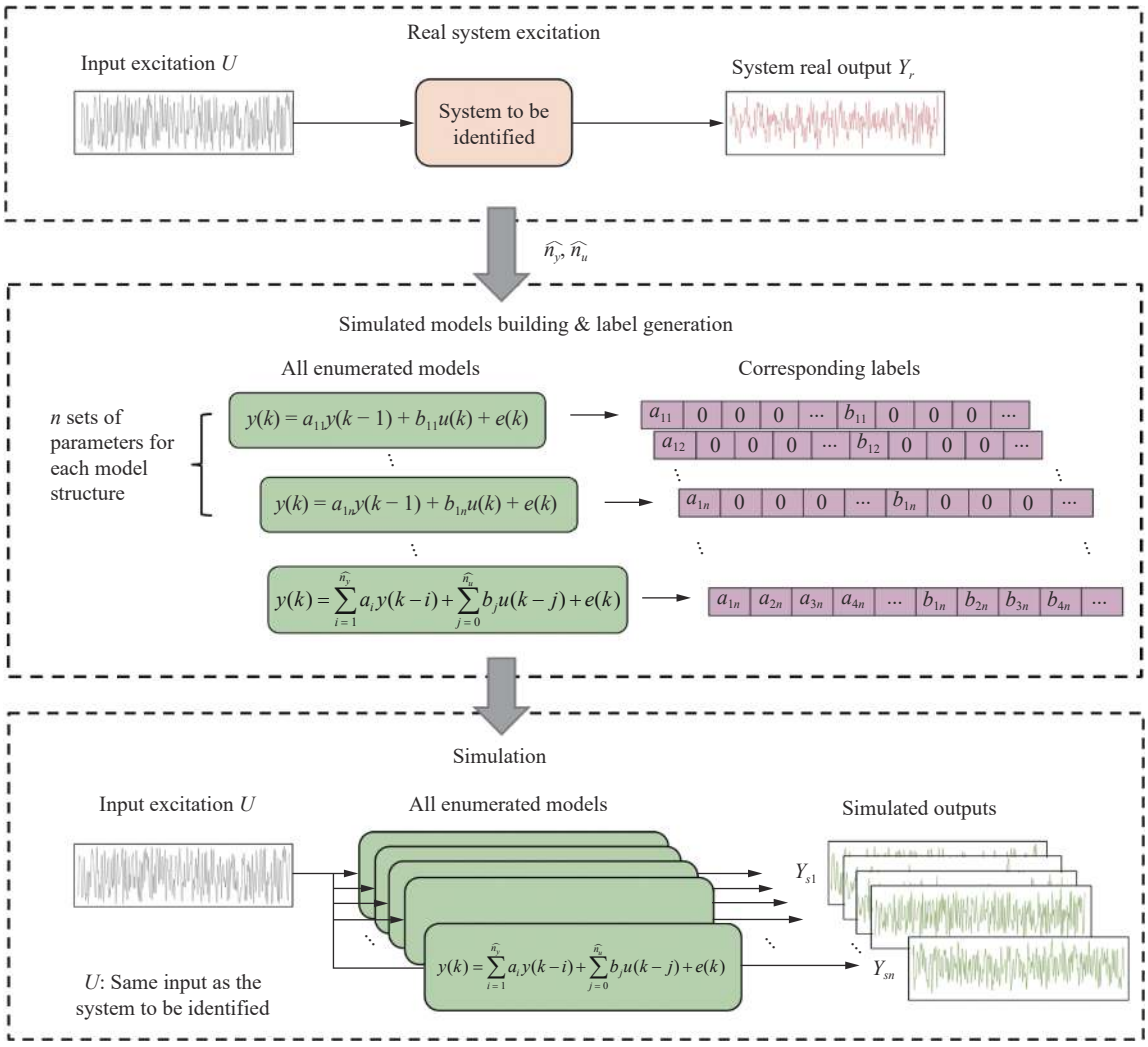


Fig. 2 Detail of simulation data generation. The system to be identified is excited with an input frequency-rich signal U generating the output Y_r. The first step of data generation is to estimate the possible maximum time delay $\widehat{n_u}$ and $\widehat{n_y}$ of the input and output of the system to be identified. Then all combinations of input and output delay terms are enumerated to produce all possible ARX model structure. For each model structure, n sets of coefficients are generated randomly with a uniform distribution. There are $\widehat{n_u} + \widehat{n_y}$ bits in the label vectors. The existing delay terms in the model are represented by the non-zero bits in the corresponding labels. The specific number in the non-zero bits are the corresponding coefficients of the existing delay terms. In simulation, input excitation of the system to be identified will serve as the inputs for all simulated models. Then both inputs and outputs of the simulated models become the input features for the neural network.

the combination of multiple temporal patterns of different orders is difficult for a single neural network[26]. The order-wise neural network structure is shown in Fig. 3. To identify a system with the possible maximum input and output delay $\widehat{n_u}$ and $\widehat{n_y}$, $\widehat{n_u} + \widehat{n_y}$ separate neural networks backbones are concatenated together. Each neural network is designed to identify the delay term and estimate the corresponding coefficient based on the multi-task learning strategy[27–29]. Based on the assumption that the model delay terms and the related coefficient can be identified with the same learnt low-level features, these two tasks share the same neural network backbone. Thus, the loss function l_n of the n neural network is given by

$$l_n = w_{dn} \times l_{dn} + w_{pn} \times l_{pn} \tag{5}$$

where l_{dn} and l_{pn} are the loss functions of the delay term identification and corresponding coefficients estimation, respectively; w_{dn} and w_{pn} are the weights of the delay term identification loss and corresponding coefficients estimation loss, respectively.

Since multiple delay terms may exist at the same time in an ARX model, the task of model term identification is regarded as multi-label classification. The label vectors containing coefficients information are required to be converted back to the multi-hot vector for classification. In this study, the binary cross entropy (BCE) loss is deployed and can be expressed as

$$l_{dn} = -w_n[y_{dn} \times \log\sigma(x_{dn}) + (1-y_{dn}) \times \log(1-\sigma(x_{dn}))] \tag{6}$$

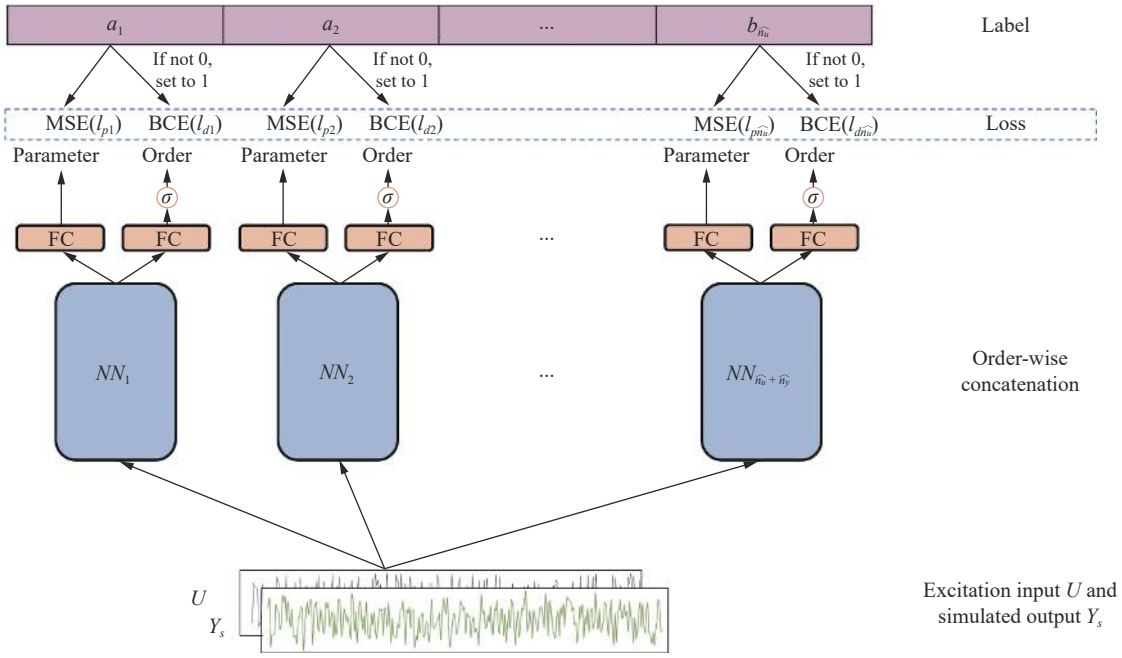


Fig. 3 Training process of the order-wise neural networks. The system excitation U and simulation output Y_s are allocated in two channels as the input of the neural network. There are $\widehat{n_u} + \widehat{n_y}$ backbones concatenated in the order-wise neural network (NN). Each backbone identifies the delay term and corresponding coefficient independently. Each concatenated backbone receives the same input. Each backbone is connected to two independent fully connected (FC) layers. The identified coefficient is the output of the first FC layer. The second FC layer is connected to the sigmoid function to identify whether the delay term exists. In the label vector, if a model delay term exists, its coefficient is located in the corresponding bit. Therefore, to train the NN to identify the existence of delay terms with the binary cross entropy (BCE) loss, the bit that is not 0 should be converted back to 1, while the coefficient estimation can be trained with the MSE loss directly. In this work, TCN and four types of modern RNNs including LSTM, Bi-LSTM, GRU and Bi-GRU are adopted as concatenated backbones of the order-wise neural network separately.

where l_{bn} is the BCE loss between y_{dn} representing the n bits of the multi-hot vector label and the n neural network output x_{dn}; σ is the sigmoid function converting x_{dn} into a value between 0 and 1; w_n is the weight of each class in the loss function and can be tuned to adjust the imbalanced training set. Since the training data is generated via simulation and always keep balanced, the value of w_n is always set to 1.

The coefficients estimation is regarded as a regression task. Each bit in the label vectors containing coefficients information is regarded as the label for each neural network. The mean square error (MSE) loss is adopted and expressed as

$$l_{pn} = \frac{1}{N}\sum_{i=1}^{N}\left(Y_{pi} - \widehat{Y}_{pi}\right)^2 \tag{7}$$

where l_{pn} is the MSE loss between the output of the neural network $\widehat{Y}_{pi}$ and the corresponding label Y_{pi}; N denotes the batch size. In training, the Adam optimizer is selected to optimise the loss function. The training of the order-wise neural networks with multi-task learning strategy is outlined in algorithm 1.

Algorithm 1. Multi-task learning of order-wise neural networks

Let $\widehat{n_y}$, $\widehat{n_u}$ be the possible maximum output and input time delay of actual system.

Let θ_n be all parameters of the n-th backbone of order-wise neural networks, $\forall n \in [1, \widehat{n_y} + \widehat{n_u}]$.

Input: The input excitation U of the actual system and the simulation output Y_s

Output: Label $T = [a_1, a_2, \cdots, a_{\widehat{n_y}}, b_1, b_2, \cdots, b_{\widehat{n_u}}]$ representing the ARX model structure and corresponding parameters, where T_n is the n-th element in T.

Initialize all parameters θ_n

for the number of training iteration **do**

  for $n = 1$ to $\widehat{n_y} + \widehat{n_u}$ **do**

    Compute MSE loss l_{pn} with T_n according to (7)

    Binarize the label $T \rightarrow T_b$

    Compute BCE loss l_{pn} with T_{bn} according to (6)

    $l_n \leftarrow w_{dn} \times l_{dn} + w_{pn} \times l_{pn}$

    for each parameter $\vartheta_n \in \theta_n$ **do**

      Compute all $\nabla l_n(\vartheta_n)$

      Update $\vartheta_n \leftarrow \vartheta_n - \eta \nabla l_n(\vartheta_n)$

    end for

  end for

end for

return all parameters θ_n, $n = 1, \cdots, \widehat{n_y} + \widehat{n_u}$.

The concatenated backbone of the order-wise neural networks is designed to extract the temporal features of the input time series for each order. As the backbone, recurrent neural networks (RNN) and temporal convolutional neural networks (TCN), commonly used for time

series modelling, are deployed and trained. The performance of these methods was then evaluated in testing.

Four order-wise neural networks consisting of modern RNNs backbones including LSTM, bidirectional LSTM (Bi-LSTM), GRU and bidirectional GRU (Bi-GRU) are developed respectively in this study. The LSTM and GRU are constructed with 2 hidden layers, and the Bi-LSTM and Bi-GRU are constructed with 1 hidden layer. Then, the output features of these RNNs pass through fully connected FC layers for coefficient estimation and delay term identification. The input windows of all RNNs are set to 60 with 50% overlap. In training, the learning rate is set as 0.001 and weight decay is set as 0.000 1. It should be noted that the hyperparameters of these models are tuned by the validation set.

In the order-wise neural networks with concatenated TCN backbone, some modifications were made to improve their performance for the tasks in this study. The hyperparameters were tuned on the validation set to achieve lower loss. For each TCN backbone, five temporal blocks were constructed and connected as the residual connection to avoid gradient vanishing. In each temporal block, the dropout layer after the activation function was abandoned. The parametric rectified linear unit (PReLU) is used as the activation function. The output of TCNs is flattened and passes through fully connected layers for coefficient estimation and delay term identification. The batch size for both RNN and TCN-based order-wise neural networks is determined by the size of the training set. For the training set generated from simple models containing only single input and output delay terms, the batch size is set as 128, while for the training set of the multiple delay terms, the batch size is set as 8 192. The neural networks in this study were developed with the deep learning framework PyTorch 1.11 and Python 3.8. The GPU is NVIDIA Tesla A100 with 80 GB of graphic memory.

2.3 Deployment and testing

After training and validation, the order-wise neural networks can be deployed directly to identify the ARX model of the linear SISO system. The detail of deploying the order-wise neural networks is shown in Fig. 4. To identify the linear SISO system, the input features of the order-wise neural networks are the sampled input excitation U and output Y_r of the system to be identified. The input data length of neural networks during deployment is identical to that during training. The outputs of the order-wise neural networks are two vectors including the vector of identified delay terms and the vector of identified coefficients. Finally, the vector representing the identified model can be derived by the element-wise product of the identified coefficients vector and identified delay terms vector.

To evaluate the performance of delay term identification of the order-wise neural network with different backbones, this study uses the macro-average area under the receiver operating characteristic curve (AUC) and accuracy, given by

$$Macro-averagedAUC=\frac{1}{\widehat{n_u}+\widehat{n_y}}\sum_{i=1}^{\widehat{n_u}+\widehat{n_y}}\mathrm{AUC}_i \tag{8}$$

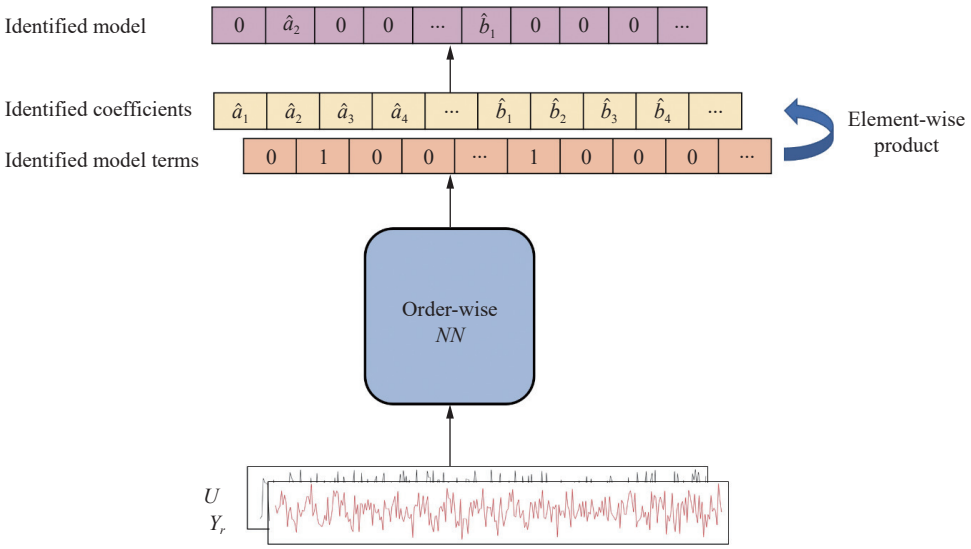


Fig. 4 Deployment of the order-wise neural networks. After training, the input of the neural network is the excitation U and the output Y_r of the system to be identified. The outputs of the order-wise neural networks are two vectors including the vector of identified delay terms and the vector of identified coefficients. The identified model vector is derived by the element-wise product of the delay terms vector and the coefficients vector. If a bit in the identified model vector is not 0, it indicates the existence of the corresponding delay term, and the specific number is its coefficient.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

where TP = true positive; FN = false negative; TN = true negative; and FP = false positive. The accuracy measures whether the neural network accurately identified all delay terms. On the other hand, since the delay terms identification is regarded as multi-label classification and the generated dataset is balanced in terms of classes, the AUC of each delay term is calculated independently and given equal weights. Consequently, the macro-average AUC is used to evaluate the classification of each delay term. To evaluate the performance of coefficient estimation, root mean square error (RMSE) is adopted and given by (10).

$$\mathrm{RMSE} = \sqrt{\frac{1}{\widehat{n_y}} \sum_{i=1}^{\widehat{n_y}} (a_i - \widehat{a}_i)^2 + \frac{1}{\widehat{n_u}} \sum_{j=0}^{\widehat{n_u}} \left(b_j - \widehat{b}_j\right)^2} \tag{10}$$

where a_i and $\widehat{a}_i$ are the actual coefficients and estimated coefficients of the output delay terms. b_j and $\widehat{b}_j$ are the actual coefficients and estimated coefficients of the input delay terms.

In testing, both RNN and TCN-based order-wise neural networks were trained and tested in multiple scenarios to determine the amount of data required for training a proper model. Furthermore, the neural network which is considered to be more suitable for the system identification task should perform better with fewer training data and low SNR. Two types of models are tested in this study. The simple model containing the single input and output delay terms shown in (11) is tested first.

$$\begin{aligned} &y(k) = a_i y(k-i) + b_j u(k-j) + \mathrm{e}(k) \\ &i \in [1, \widehat{n_y}], j \in [0, \widehat{n_u}], i, j \in \mathbf{Z}. \end{aligned} \tag{11}$$

Then the robustness test for data with different SNR levels is carried out based on the determined amount of training data. Finally, a complicated model containing multiple input and output delay terms shown in (12) is tested.

$$y(k) = \sum_{i=1}^{\widehat{n_y}} a_i y(k-i) + \sum_{j=0}^{\widehat{n_u}} b_j u(k-j) + \mathrm{e}(k). \tag{12}$$

3 Results

3.1 Deployment and testing

3.1.1 The number of coefficient sets

First, the sensitivity tests for the amount of training data required to identify simple ARX models with a single input and output delay term are conducted. Since the amount of training data depends on the number of coefficient sets generated for each model structure and the data length of the time series generated by each model, the first test was run with the length of the time series fixed and the number of coefficient sets varied. As for the testing model, the maximum time delays of both input and output delay terms, $\widehat{n_u}$ and $\widehat{n_y}$, were set to 10, resulting in a total of 100 model structure variations.

To generate the training set via simulation, for each model structure, a certain number of coefficient sets are generated from a uniform distribution, meanwhile, the stability condition requiring characteristic roots to be inside the unit cycle is imposed. The number of coefficient sets varies from 50 to 1 000 and the length of the input time series is set to 300, as shown in Table 1. In this study, we have tested different data lengths and observed that the accuracy of structure identification converged after a data length of 300. Therefore, 300 is selected as the data length in this section considering its potential for time-varying systems which requires the window size as small as possible to ensure high temporal resolution in tracking changes. Each model's input time series is identical and generated randomly with a uniform distribution to ensure that input time series are rich in frequency. The testing set is generated separately following the steps described above so that the neural network is tested with ARX models consisting of all combinations of delay terms with random coefficients not shown in the training set.

The trend of the accuracy of model term identification is shown in Fig. 5(a). It has been observed that with the increase of the number of coefficient sets generated for each model structure from 50 to 300, the accuracy of all types of neural networks increased evidently and rapidly, and the accuracy converges after 500. As suggested by Table 1, the macro-averaged AUC generally matches the trend of accuracy, which indicates that the 500 coefficient sets generated for each model structure are roughly sufficient to achieve an accuracy of > 91% in the identification of model terms. After the convergence of accuracy, both accuracy and macro-averaged AUC of RNN-based neural networks are higher than those of TCN-based neural networks indicating that the RNN-based neural networks perform better in identifying the terms of simple ARX models when the number of coefficient sets in the training set is sufficient. It is also observed that TCN-based neural networks have the best performance while LSTM-based methods have the worst performance when the number of coefficient sets is small.

For coefficient estimation, the results of which are presented in Fig. 5(b), the RMSE of estimated coefficients of all RNN-based neural networks decreases sharply with the number of coefficient sets increasing from 50 to 500. The value of RMSE also becomes stable after 500. However, the RMSE of TCN-based neural net-

Table 1 Performance of simple ARX model identification with a data length of 300 and various numbers of coefficient sets

Data length	The number of coefficient sets	Backbone types	Macro-averaged AUC	Accuracy	Coefficients estimation RMSE
300	50	2-layer GRU	0.905 0	0.603 0	**0.064 3**
		Bi-GRU	0.921 0	0.691 0	0.069 6
		2-layer LSTM	0.825 0	0.256 0	0.097 2
		Bi-LSTM	0.656 0	0.019 0	0.156 6
		TCN	**0.941 0**	**0.731 0**	0.102 5
300	100	2-layer GRU	0.919 0	0.651 0	0.053 6
		Bi-GRU	0.948 0	**0.807 0**	**0.033 8**
		2-layer LSTM	0.829 0	0.267 0	0.096 8
		Bi-LSTM	0.776 0	0.127 0	0.112 9
		TCN	**0.955 0**	0.802 0	0.095 7
300	300	2-layer GRU	**0.989 0**	**0.956 0**	0.031 0
		Bi-GRU	0.988 0	0.950 0	0.031 1
		2-layer LSTM	0.988 0	0.951 0	**0.030 0**
		Bi-LSTM	0.984 0	0.930 0	0.064 7
		TCN	0.978 0	0.903 0	0.099 5
300	500	2-layer GRU	0.990 0	0.958 0	0.029 0
		Bi-GRU	0.991 0	0.963 0	0.025 9
		2-layer LSTM	**0.992 0**	**0.965 0**	**0.021 4**
		Bi-LSTM	0.989 0	0.953 0	0.028 6
		TCN	0.981 0	0.919 0	0.088 3
300	1 000	2-layer GRU	**0.991 0**	0.961 0	0.023 9
		Bi-GRU	**0.991 0**	**0.962 0**	0.025 9
		2-layer LSTM	**0.991 0**	0.961 0	**0.020 7**
		Bi-LSTM	0.989 0	0.954 0	0.027 7
		TCN	0.981 0	0.921 0	0.088 3

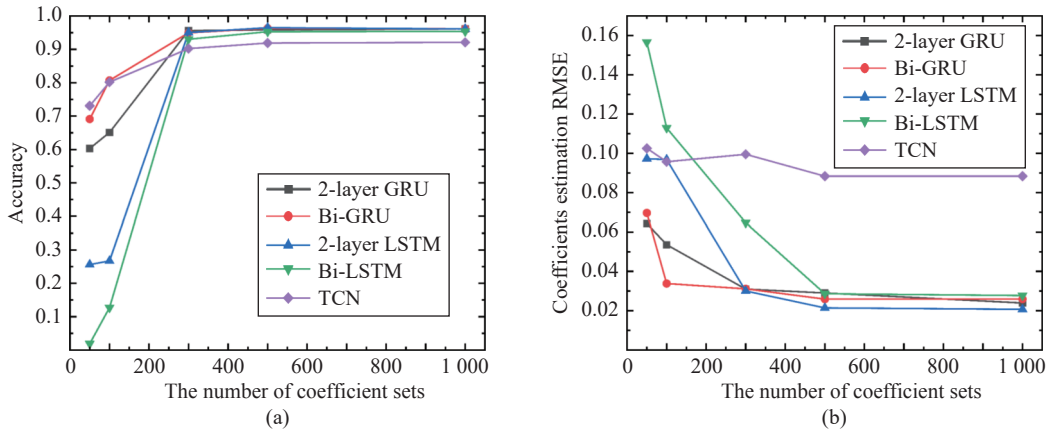


Fig. 5 The test results of different NN backbones with various numbers of coefficient sets. (a) Trend of the accuracy over the varying number of coefficient sets; (b) Trend of the coefficients estimation RMSE over the varying number of coefficient sets. (Colored figures are available in the online version at https://link.springer.com/journal/11633)

works does not decrease significantly as the number of coefficient sets increases. After the convergence, the RMSE of RNN-based neural networks is almost 4 times lower than that of TCN-based neural networks indicating that the RNN-based neural networks perform better in estimating the coefficient of simple ARX models with a sufficient number of coefficient sets in the training set.

3.1.2 Data length

Based on the above result, it was determined that 500 coefficient sets generated for each model structure are rel-

atively sufficient. Therefore, the second sensitivity test for simple ARX model identification was conducted with the number of coefficient sets of 500 fixed and the length of time series varied.

To generate the training data by simulation, the maximum time delays of both input and output delay terms were still set to 10. A total of 500 coefficient sets were generated randomly with a uniform distribution for each model structure variation, meanwhile, the stability condition requiring characteristic roots to be inside the unit cycle is imposed. Other parameters were set the same as those in the above section. The length of the simulation model input time series was set from 100 to 1 000, as shown in Table 2.

The trend of identification accuracy of delay terms is shown in Fig. 6(a). It has been observed that when the length of the time series increases from 100 to 300, the accuracy of all RNN-based neural networks increases evidently, while the accuracy converges after 300. On the other hand, although the accuracy of TCN-based neural networks is already high at the beginning, the accuracy of all RNN-based neural networks surpasses that of TCN-based neural networks when their accuracy converges. It can be observed from Table 2 that the trend of macro-averaged AUC also matches the trend of accuracy, which indicates that RNN-based models can generally outperform the TCN-based neural networks when the time series length is sufficient, and a time series length of 300 is roughly adequate to obtain promising delay term identification results.

Fig. 6(b) shows the result of coefficient estimation. The RMSE values of 2-layer GRU and 2-layer LSTM are already relatively small with a data length of 100, while the RMSE values of the rest neural networks converge after 300. After the convergence, the RMSE values of RNN-based neural networks are still about 3 times lower than that of TCN-based neural networks indicating that the RNN-based neural networks work better in estimating the coefficient of simple ARX models with sufficient time series length.

Compared to the effect of the increasing number of coefficient sets, the increase in time series length has less influence on both delay terms identification and coefficient estimation. A time series with a length of 300 and

Table 2 Performance of simple ARX model identification with the number of coefficient sets of 500 and various data lengths

Data length	The number of coefficient sets	Backbone types	Macro-averaged AUC	Accuracy	Coefficients estimation RMSE
100	500	2-layer GRU	0.955 0	0.842 0	0.018 2
		Bi-GRU	0.942 0	0.768 0	0.061 4
		2-layer LSTM	0.954 0	0.841 0	**0.017 6**
		Bi-LSTM	0.881 0	0.482 0	0.072 7
		TCN	**0.981 0**	**0.916 0**	0.106 5
200	500	2-layer GRU	0.954 0	0.839 0	0.023 2
		Bi-GRU	**0.988 0**	**0.949 0**	0.023 7
		2-layer LSTM	0.962 0	0.855 0	**0.020 5**
		Bi-LSTM	0.945 0	0.632 0	0.063 5
		TCN	0.985 0	0.935 0	0.092 4
300	500	2-layer GRU	0.990 0	0.958 0	0.029 0
		Bi-GRU	0.991 0	0.963 0	0.025 9
		2-layer LSTM	**0.992 0**	**0.965 0**	**0.021 4**
		Bi-LSTM	0.989 0	0.953 0	0.028 6
		TCN	0.981 0	0.919 0	0.088 3
500	500	2-layer GRU	**0.991 0**	**0.962 0**	0.023 5
		Bi-GRU	0.990 0	**0.962 0**	0.025 7
		2-layer LSTM	**0.991 0**	**0.962 0**	**0.020 2**
		Bi-LSTM	0.986 0	0.953 0	0.027 9
		TCN	0.983 0	0.933 0	0.077 0
1 000	500	2-layer GRU	0.991 0	0.963 0	0.021 7
		Bi-GRU	**0.993 0**	**0.970 0**	**0.020 0**
		2-layer LSTM	0.990 0	0.959 0	0.020 5
		Bi-LSTM	0.986 0	0.940 0	0.025 7
		TCN	0.985 0	0.933 0	0.064 6

500 coefficient sets is sufficient to generate an adequate-sized training set.

3.1.3 Additive white noise

Simple ARX models with different levels of additive white noise were tested to evaluate the robustness of the models. The maximum time delays of both input and output delay terms were also set to 10. Based on the results of the above tests, the training set size with 500 coefficient sets and a data length of 300 for each model were adopted. Additive white noise with a signal-to-noise ratio (SNR) ranging from 30 dB to 10 dB was generated and applied to the system output in both training data and test data to evaluate the robustness against the noise of each neural network, the result of which is presented in Table 3. To evaluate the effect of noise level on the identification of delay terms, we further adopted the area under the precision-recall curve (AUPR) to assess the performance. Unlike the AUC, which measures a model's ability to distinguish between positive and negative samples, the AUPR evaluates the model's ability to detect positive samples[30].

It has been observed that the proposed neural network-based framework generally works well to identify the ARX model with a certain level of additive white noise. As the SNR reduced, the accuracy of all neural network backbones declined. The accuracy of 2-layer LSTM, Bi-GRU and 2-layer LSTM is higher than those of Bi-LSTM and TCN when the SNR is 30 dB and 20 dB. At 10 dB, the accuracy of 2-layer GRU is significantly higher than other neural network backbones. The RMSE values of coefficient estimation and the macro-averaged AUC generally followed the same trend as the accuracy. The AUC did not change much when the SNR decreased to 10 dB, indicating that the NN-based approach did not

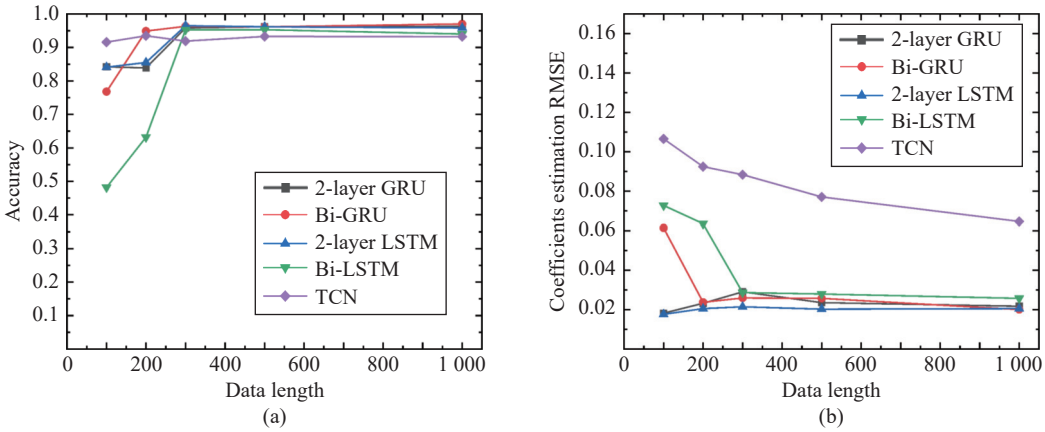


Fig. 6 The test results of different NN backbones with various data lengths. (a) Trend of the accuracy over the varying data length; (b) Trend of the RMSE of coefficients estimation over the varying data length. (Colored figures are available in the online version at https://link.springer.com/journal/11633)

Table 3 Performance of simple ARX model identification with additive white noise

SNR	Model	Macro-averaged AUC	AUPR	Accuracy	Coefficients estimation RMSE
30dB	2-layer GRU	0.947 0	0.972 0	0.802 0	0.060 8
	Bi-GRU	**0.949 0**	0.978 0	0.804 0	0.045 8
	2-layer LSTM	**0.949 0**	**0.981 0**	**0.807 0**	**0.041 2**
	Bi-LSTM	0.935 0	0.952 0	0.753 0	0.080 6
	TCN	0.927 0	0.931 0	0.719 0	0.091 7
20dB	2-layer GRU	0.921 0	0.942 0	0.721 0	0.052 9
	Bi-GRU	**0.929 0**	0.951 0	0.741 0	0.052 9
	2-layer LSTM	0.925 0	**0.967 0**	**0.742 0**	**0.048 0**
	Bi-LSTM	0.915 0	0.933 0	0.703 0	0.053 9
	TCN	0.904 0	0.875 0	0.667 0	0.097 5
10dB	2-layer GRU	**0.855 0**	**0.673 0**	**0.537 0**	**0.070 0**
	Bi-GRU	0.793 0	0.326 0	0.285 0	0.104 4
	2-layer LSTM	0.841 0	0.537 0	0.476 0	0.080 0
	Bi-LSTM	0.811 0	0.281 0	0.222 0	0.092 7
	TCN	0.843 0	0.528 0	0.473 0	0.104 4

misidentify many non-existing delay terms even with extremely low SNR. However, the AUPR decreased significantly when the SNR dropped to 10 dB, indicating that the proposed NN-based approach is more likely to be unable to detect existing delay terms at extremely low SNR. Generally, the proposed framework tends to detect a sparser model structure under high levels of noise. Overall, the 2-layer GRU performs better in both model terms identification and coefficients estimation when the SNR is low.

3.2 Complicated model identification

According to the result of the simple model identification, the increase of the time series length has less influence on both delay terms identification and coefficient estimation than the number of coefficient sets. Therefore, in this section, the identification of complicated models with multiple delay terms was tested with fixed data length and a varied number of coefficient sets. For the complicated testing model, the maximum time delays of both input and output delay terms were set to 3, as shown in (12).

To generate training data, the maximum time delays of both input and output delay terms were set to 3, resulting in a total of 49 model structure variations as described in (2). For the complicated ARX model simulation, the method to generate model input and coefficient sets is the same as the simple model simulation. The data length is fixed at 300. The number of coefficients changes from 500 to 1 500. The training set contains 60% of the data, while the validation set contains 20% of the data and the test set contains 20% of the data. The test results are shown in Table 4.

It has been observed that the proposed neural network-based framework generally works well to identify the complicated ARX model with multiple input-output delay terms. When the number of coefficient sets exceeded 500, both the accuracy and the macro-average AUC increased. In addition, the RMSE values of coefficient estimation continue to decrease. Therefore, as expected, complicated model identification requires a larger amount of training data than simple model identification. Compared with other types of neural network backbones, the 2-layer GRU generally performed better in both model term identification and coefficient estimation. However, compared with its performance of simple model identification, the accuracy and macro-average AUC are lower while the performance of coefficient estimation is better, indicating that the complicated model identification with multiple delay terms is more challenging.

3.3 Comparison with existing methods

We then evaluate the ARX model identification performance by comparing the one-step ahead prediction between the model derived from the proposed framework and conventional AIC and BIC methods. A simulation model is constructed to generate the data. The model is described as follows:

$$y(k) = -0.709 \times u(k-9) - 0.686 \times y(k-10) + e(k) \quad (13)$$

where the additive white noise with SNR of 20 dB is applied to the output. And 300 data points are generated through simulation.

According to the results in Section 3.1.3, the three best performing NN backbones under the SNR of 20 dB

Table 4 Performance of complicated model identification multiple input-output delay terms with the maximum time delay 3

Data length	The number of coefficient sets	Model	Macro-averaged AUC	Accuracy	Coefficients estimation RMSE
300	500	2-layer GRU	0.907 0	0.596 0	0.120 4
		Bi-GRU	**0.908 0**	**0.605 0**	0.114 9
		2-layer LSTM	0.905 0	0.593 0	**0.114 5**
		Bi-LSTM	0.902 0	0.589 0	0.117 5
		TCN	0.832 0	0.338 0	0.192 4
300	1 000	2-layer GRU	**0.947 0**	**0.747 0**	**0.081 9**
		Bi-GRU	0.926 0	0.669 0	0.101 0
		2-layer LSTM	0.939 0	0.711 0	0.086 6
		Bi-LSTM	0.920 0	0.634 0	0.113 1
		TCN	0.849 0	0.384 0	0.153 9
300	1 500	2-layer GRU	**0.953 0**	**0.765 0**	**0.079 4**
		Bi-GRU	0.917 0	0.628 0	0.097 0
		2-layer LSTM	0.941 0	0.712 0	0.084 9
		Bi-LSTM	0.913 0	0.622 0	0.107 2
		TCN	0.865 0	0.386 0	0.146 6

including the 2-layer GRU (GRU2L), bidirectional GRU (BiGRU) and 2-layer LSTM (LSTM2L) are employed in the experiment. After identification, all NN-based models have derived exact sparse model terms as the ground truth model in (13), while both AIC and BIC methods identified with redundant delay terms of output from $y(k-1)$ to $y(k-10)$. Fig.7 illustrates the one-step ahead prediction performance comparison of the identified ARX models derived from AIC, BIC and proposed neural network-based methods. The ground truth data are depicted in grey and the predictions from five different models derived from AIC, BIC, GRU2L, BiGRU and LSTM2L methods are depicted in blue, orange, yellow, purple and green, respectively. The goodness of fit for each model's prediction is measured by the normalized root mean square error (NRMSE). It has been observed that all models closely track the actual test data, with slight variations in NRMSE. The models derived from GRU2L and BiGRU show the highest prediction accuracy, outperforming those derived from AIC and BIC methods. This is likely due to the redundant delay terms identified by AIC and BIC causing overfitting, especially with noise. These results highlight the effectiveness of the proposed NN-based framework in identifying both structure and parameters in the presence of noise.

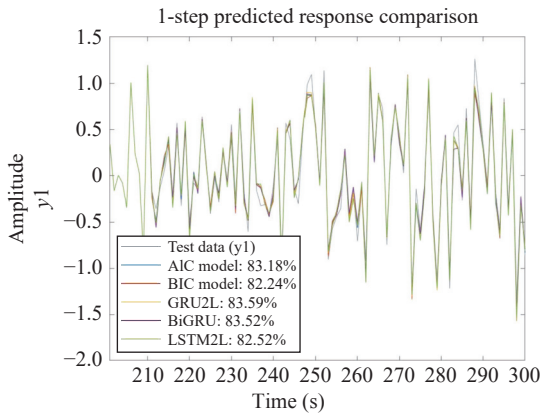


Fig. 7 The one-step ahead prediction performance comparison of the identified ARX models derived from AIC, BIC and proposed neural network-based method with simulation data. Each model's prediction is represented with a distinct colour. The legend provides the goodness of fit for each model's prediction, measured by the NRMSE. (Colored figures are available in the online version at https://link.springer.com/journal/11633)

4 Discussions

Despite the proposed end-to-end ARX model identification framework effectively identifying both delay terms and parameters with limited data length and relatively low SNR, there are still some limitations. Due to the constraints of supervised learning, which is adopted in the pretraining in this study, the training data and corresponding labels must be generated from simulated models that are enumerated under a certain maximum time delay. This poses a significant challenge for enumerating all model combinations in highly nonlinear systems, especially under high degrees of nonlinearity. With the development of zero-shot learning[31], which is able to categorize unknown classes without training data covering all circumstances, future studies could potentially extend to nonlinear system identification using zero-shot learning.

On the other hand, since the proposed approach achieves promising results with limited data length, it provides the opportunity to identify time-varying systems based on a sliding window approach. As the system is regarded as time-invariant within each sliding window, the window size should be relatively small to ensure high temporal resolution in tracking changes[1]. Therefore, the ability of the identification method to effectively identify the model structure and parameters with limited data length is crucial for the sliding window approach. Future studies will fully explore its potential to address time-varying systems.

5 Conclusions

In this paper, a novel end-to-end ARX model identification framework was proposed to identify arbitrary linear stable SISO systems by order-wise neural networks trained with finite simulation data. To be more specific, the training data can be generated by simulating ARX models consisting of all combinations of input and output delay terms under the estimated maximum input and output time delays of the system to be identified. Then the proposed order-wise neural networks can be trained to identify arbitrary stable linear ARX models under the estimated maximum input and output time delays. For the identification of simple ARX models, order-wise neural networks with RNN backbones outperformed TCN-backbone when the training data is sufficient. While the TCN-based neural networks outperform the RNN-based neural networks with a small training set characterised by shorter data length. Furthermore, the RNN-based neural networks also perform better with additive white noise. For the identification of complicated ARX models, RNN-based neural networks also perform better in both model term identification and coefficient estimation. Compared with the simple ARX model identification, the identification of complicated ARX models is more challenging and requires a larger training set. Overall, the proposed end-to-end ARX model identification framework proves its capability to identify ARX model terms and coefficients simultaneously with finite simulation training data. This study opens up a new research opportunity for parametric system identification utilising the power of deep learning. A limitation of this work is that it is only able to identify the stable LTI system. The future study will be extended to nonlinear and time-varying systems.

Acknowledgements

Open access funding provided by Cranfield University, UK.

Declarations of conflict of interest

The authors declared that they have no conflicts of interest to this work.

Open Access

References

[1] S. A. Billings. *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio–temporal Domains*, Chichester, UK: Wiley, pp. 1–15, 2013. DOI: 10.1002/9781118535561.

[2] L. Faes, G. Nollo, K. H. Chon. Linear and nonlinear parametric model identification to assess Granger causality in short-term cardiovascular interactions. In *Proceedings of Computers in Cardiology*, Bologna, Italy, pp. 793–796, 2008. DOI: 10.1109/CIC.2008.4749161.

[3] J. Kon, Y. Yamashita, T. Tanaka, A. Tashiro, M. Daiguji. Practical application of model identification based on ARX models with transfer functions. *Control Engineering Practice*, vol. 21, no. 2, pp. 195–203, 2013. DOI: 10.1016/J.CONENGPRAC.2012.09.021.

[4] V. A. O. Alves, R. Juliani Correa De Godoy, C. Garcia. Searching the optimal order for high order models - SISO case. In *Proceedings of IEEE International Conference on Control Applications*, Dubrovnik, Croatia, pp. 843–848, 2012. DOI: 10.1109/CCA.2012.6402339.

[5] B. Lindoff, J. Holst. Convergence analysis of the RLS identification algorithm with exponential forgetting in stationary ARX-structures. *International Journal of Adaptive Control and Signal Processing*, vol. 13, no. 1, pp. 1–22, 1999. DOI: 10.1002/(SICI)1099-1115(199902)13:1<1::AID-ACS520>3.0.CO;2-V.

[6] Q. C. Nguyen, V. H. Vu, M. Thomas. A Kalman filter based ARX time series modeling for force identification on flexible manipulators. *Mechanical Systems and Signal Processing*, vol. 169, Article number 108743, 2022. DOI: 10.1016/j.ymssp.2021.108743.

[7] B. M. Sanandaji, T. L. Vincent, M. B. Wakin, R. Tóth, K. Poolla. Compressive system identification of LTI and LTV ARX models. In *Proceedings of 50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, USA, pp. 791–798, 2011. DOI: 10.1109/CDC.2011.6160935.

[8] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974. DOI: 10.1109/TAC.1974.1100705.

[9] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978. DOI: 10.1214/aos/1176344136.

[10] M. H. Marzaki, M. Tajjudin, R. Adnan, M. H. F. Rahiman, M. H. A. Jalil. Comparison of different model structure selection using R.2, MDL and AIC criterion. In *Proceedings of 4th Control and System Graduate Research Colloquium*, Shah Alam, Malaysia, pp. 80–85, 2013. DOI: 10.1109/ICSGRC.2013.6653280.

[11] H. A. Rahim, F. Ibrahim, M. N. Taib. Model order selection criterion for monitoring haemoglobin status in dengue patients using ARX model. In *Proceedings of International Conference on Information Technology and Applications in Biomedicine*, Shenzhen, China, pp. 452–456, 2008. DOI: 10.1109/ITAB.2008.4570537.

[12] L. L. Cheng, A. Cigada, Z. Q. Lang, E. Zappa, Y. P. Zhu. An output-only ARX model-based sensor fusion framework on structural dynamic measurements using distributed optical fiber sensors and fiber Bragg grating sensors. *Mechanical Systems and Signal Processing*, vol. 152, Article number 107439, 2021. DOI: 10.1016/J.YMSSP.2020.107439.

[13] B. Uniejewski, G. Marcjasz, R. Weron. Understanding intraday electricity markets: Variable selection and very short-term price forecasting using LASSO. *International Journal of Forecasting*, vol. 35, no. 4, pp. 1533–1547, 2019. DOI: 10.1016/J.IJFORECAST.2019.02.001.

[14] M. Klingspor, A. Hansson, J. Löfberg. Input selection in ARX model estimation using group lasso regularization. *IFAC-PapersOnLine*, vol. 51, no. 15, pp. 897–902, 2018. DOI: 10.1016/J.IFACOL.2018.09.080.

[15] N. Zimmermann, T. Büchi, J. Mayr, K. Wegener. Self-optimizing thermal error compensation models with adaptive inputs using Group-LASSO for ARX-models. *Journal of Manufacturing Systems*, vol. 64, pp. 615–625, 2022. DOI: 10.1016/J.JMSY.2022.04.015.

[16] J. L. Elman. Finding structure in time. *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990. DOI: 10.1207/s15516709cog1402_1.

[17] S. Hochreiter, J. Schmidhuber. Long short-term memory. *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/NECO.1997.9.8.1735.

[18] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of the 8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, Doha, Qatar, pp. 103–111, 2014. DOI: 10.3115/v1/w14-4012.

[19] S. J. Bai, J. Z. Kolter, V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, [Online], Available: http://arxiv.org/abs/1803.01271, 2018.

[20] L. Ljung, C. Andersson, K. Tiels, T. B. Schön. Deep learn-

ing and system identification. *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1175–1181, 2020. DOI: 10.1016/J.IFACOL.2020.12.1329.

[21] C. F. Fung, S. A. Billings, H. Zhang. Generalised transfer functions of neural networks. *Mechanical Systems and Signal Processing*, vol. 11, no. 6, pp. 843–868, 1997. DOI: 10.1006/mssp.1997.0112.

[22] T. A. Tutunji. Parametric system identification using neural networks. *Applied Soft Computing*, vol. 47, pp. 251–261, 2016. DOI: 10.1016/J.ASOC.2016.05.012.

[23] M. Nauta, D. Bucur, C. Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 312–340, 2019. DOI: 10.3390/MAKE1010019.

[24] A. Tank, I. Covert, N. Foti, A. Shojaie, E. B. Fox. Neural granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 8, pp. 4267–4279, 2022. DOI: 10.1109/TPAMI.2021.3065601.

[25] P. Khanarsa, A. Luangsodsa, K. Sinapiromsaran. Self-identification ResNet-ARIMA forecasting model. *WSEAS Transactions on Systems and Control*, vol. 15, no. 21, pp. 196–211, 2020. DOI: 10.37394/23203.2020.15.21.

[26] W. H. Tang, A. Röllin. Model identification for ARMA time series through convolutional neural networks. *Decision Support Systems*, vol. 146, Article number 113544, 2021. DOI: 10.1016/j.dss.2021.113544.

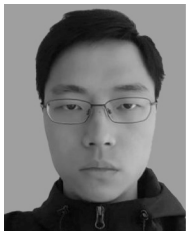
[27] Y. Zhang, Q. Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 12, pp. 5586–5609, 2022. DOI: 10.1109/TKDE.2021.3070203.

[28] S. Ruder. An overview of multi-task learning in deep neural networks, [Online], Available: https://doi.org/10.48550/arXiv.1706.05098, 2017.

[29] J. Q. Ma, Z. Zhao, J. L. Chen, A. Li, L. C. Hong, E. H. Chi. SNR: Sub-network routing for flexible parameter sharing in multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, USA, pp. 216–223, 2019. DOI: 10.1609/AAAI.V33I01.3301216.

[30] H. R. Sofaer, J. A. Hoeting, C. S. Jarnevich. The area under the precision-recall curve as a performance metric for rare binary events. *Methods in Ecology and Evolution*, vol. 10, no. 4, pp. 565–577, 2019. DOI: 10.1111/2041-210X.13140.

[31] F. Pourpanah, M. Abdar, Y. X. Luo, X. L. Zhou, R. Wang, C. P. Lim, X. Z. Wang, Q. M. J. Wu. A review of generalized zero-shot learning methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4051–4070, 2023. DOI: 10.1109/TPAMI.2022.3191696.



Aoxiang Dong received the M. Sc. degree in automation and control from the Newcastle University, UK in 2019. He is currently a Ph. D. degree candidate in system identification and data science at Cranfield University, UK.

His research interests include time series analysis, nonlinear Granger causality analysis, and explainable artificial intelligence.

E-mail: A.Dong@cranfield.ac.uk
ORCID iD: 0000-0002-8695-8812

Andrew Starr received the Ph. D. degree in condition-based maintenance for robotic production plant from the University of Manchester, UK in 1993. He is currently a professor of maintenance systems with the Cranfield University, UK.

His research interests include machine and structural damage detection, diagnostics and prognostics, autonomous inspection, "big data" for system monitoring and control, and applications in railway infrastructure and vehicles.

E-mail: a.starr@cranfield.ac.uk
ORCID iD: 0000-0001-9046-560X

Yifan Zhao received the Ph. D. degree in automatic control and system engineering from the University of Sheffield, UK in 2007. He is currently a professor of data science with the Cranfield University, UK. He is a senior member of IEEE and a fellow of Higher Education Academy.

His research interests include machine learning, computer vision, signal processing, nondestructive testing, active thermography, and nonlinear system identification.

E-mail: yifan.zhao@cranfield.ac.uk (Corresponding author)
ORCID iD: 0000-0003-2383-5724