Explainable Intelligent Fault Diagnosis for Nonlinear Dynamic Systems: From Unsupervised to Supervised Learning

Hongtian Chen, *Member, IEEE*, Zhigang Liu, *Senior Member, IEEE*, Cesare Alippi, *Fellow, IEEE*, Biao Huang, *Fellow, IEEE*, and Derong Liu, *Fellow, IEEE*

***Abstract*—The increased complexity and intelligence of automation systems require the development of intelligent fault diagnosis (IFD) methodologies. By relying on the concept of a suspected space, this study develops explainable data-driven IFD approaches for nonlinear dynamic systems. More in detail, we parameterize nonlinear systems through a generalized kernel representation used for system modeling and the associated fault diagnosis. An important result obtained is a unified form of kernel representations, applicable to both unsupervised and supervised learning. More importantly, through a rigorous theoretical analysis we discover the existence of *a bridge* (i.e., a bijective mapping) between some supervised and unsupervised learning-based entities. Notably, the designed IFD approaches achieve the same performance by the use of this bridge. In order to have a better understanding of the results obtained, unsupervised and supervised neural networks are chosen as the learning tools to identify generalized kernel representations and design the IFD schemes; an invertible neural network is then employed to build the bridge between them. This study is a perspective article, whose contribution lies in proposing and detailing the fundamental concepts for explainable intelligent learning methods, contributing to system modeling and data-driven IFD designs for nonlinear dynamic systems.**

***Index Terms*—Intelligent fault diagnosis (IFD), nonlinear dynamic systems, the bridge, unsupervised learning, supervised learning, neural networks.**

I. Introduction

Over the past three decades, fault diagnosis has undergone tremendous development [1]–[7]. Due to the increasing demands of safe and economic operations, it is playing an essential role in system performance evaluation and maintenance [1]. Fault diagnosis has a broad spectrum of applications, for instance, ranging from chemical processes [8], electrical systems [9], intelligent transportation [10], aerospace engineering [11] to medical imaging [12].

This work was supported by the Natural Sciences and Engineering Research Council of Canada. (*Corresponding author: Zhigang Liu and Biao Huang.*)

Hongtian Chen and Biao Huang are with the Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada. Email: chtbaylor@163.com (H. Chen); biao.huang@ualberta.ca (B. Huang).

Zhigang Liu is with Institute of Rail Transit, Tongji University, Shanghai, 201804, China; School of Electrical Engineering, Southwest Jiaotong University, Chengdu, 611756, China. Email: liuzgcd@126.com.

Cesare Alippi is with the Faculty of Informatics, Università della Svizzera italiana, Lugano, 69000, Switzerland; the author is also with the Department of Electronics, Information and Bioengineering, Politecnico di Milano, Milan, 20133, Italy. Email: alippi@elet.polimi.it.

Derong Liu is with the School of Automation, Guangdong University of Technology, Guangzhou, 510006, China. Email: derong@gdut.edu.cn.

Fault diagnosis techniques have undergone a dramatic evolution with the development of control theory, computer science, big data, sensor technology, machine learning, etc. [13]. They are becoming more intelligent and diversified. A unified description of these developments is so-called intelligent fault diagnosis (IFD), which can be model-based [14], signal processing-based [15], and data-driven [16].

The success of deep learning has further promoted fault diagnosis through neural networks. These IFD approaches, as mentioned in [1], can accomplish tasks similar to what a human can do. Neural networks have the ability to extract helpful features so that distinguishing faulty conditions from normal conditions becomes simpler. The accompanying problem is an additional demand of sufficient labeled data [13].

Depending on the operating range, neural networks-based IFD approaches can be divided into static and dynamic methods. In [17], a recurrent neural network was used to detect and identify faults of railway track circuits. In order to diagnose incipient interturn faults, a probabilistic neural network was adopted in [18] with consideration of the network size. By using a deep convolutional neural network, [19] proposed a diagnosis method for the loose strands of the isoelectric line. Taking the topological structure of system data into account, a graph convolutional network was developed in [20] to diagnose machine faults. Most recently, [21] proposed two fault detection schemes, where the first design is based on the finite impulse response filter using a fully connected neural network and the second one constructs a recursive residual generator using a recurrent neural network.

Despite the aforementioned achievements, much of the inherent difficulty of the IFD designs arises from:

1) The presence of system nonlinearity and dynamics;
2) Fault diagnosis cannot be simply treated as a classification problem;
3) Lack of interpretability in many IFD approaches, since a good representation should possess explainable posterior knowledge.

These challenges motivate the studies in this perspective article with the following four-fold contributions.

- Data-based modeling and construction of residual generators via the composite operators, enabling the learning procedures to be described quantitatively.
- ***A bridge*** is built for construction of the generalized kernel representations, based on which unsupervised and

supervised learning-based IFD approaches can be transferrable between each other. An additional value of the bridge representation is for evaluating the performance of nonlinear IFD approaches.
- Unsupervised and supervised neural networks are employed in designing two specific IFD algorithms, whose purpose is to help us understand their fundamentals.
- In addition to the previous contribution, an invertible neural network is developed to build a bridge that allows unsupervised and supervised neural network-based IFD approach to transform between each other.

The remainder of this study is organized as follows. Section II introduces metrics, machine learning, and nonlinear dynamic systems. With the aid of composite operators, Section III is dedicated to constructing the bridge between unsupervised and supervised learning-based IFD approaches. Section IV details two specific IFD algorithms using unsupervised and supervised neural networks, followed by an invertible neural network-based bridge. Section V concludes this study and delineates research opportunities of IFD.

Notations: All notations in the article are standard. $\mathcal{M}$ denotes a metric, whose subscript signifies a specific form; $\mathcal{R}^{\kappa}$ represents the space of real κ-dimensional vectors; $\mathrm{Tr}(\cdot)$ is the trace operator; $|\cdot|$ refers to the absolute value; $\circ$ is the cascade connection of multiple operators; the superscript "**f**" signifies the faulty conditions; $\hat{\kappa}$ is the estimate of κ; $\Pr(\cdot)$ is the probability; $\begin{pmatrix} \boldsymbol{\psi}_1 & \boldsymbol{\psi}_2 \\ \boldsymbol{\psi}_3 & \boldsymbol{\psi}_4 \end{pmatrix}$ is a composite operator (a matrix if and only if $\boldsymbol{\psi}_i$ is a linear operator) by stacking the operator $\boldsymbol{\psi}_i$ in a reasonable manner, where $i = 1, \cdots, 4$.

II. Preliminaries and Problem Formulation

A metric, also called a measure, is essential for obtaining the objective function of the machine learning approaches. Therefore, four representative metrics are introduced in this section, followed by description of the nonlinear dynamic systems of interest.

A. Metrics

Metrics are the measures that can quantitatively assess, compare, and track performance [22]. In the following, several metrics are introduced for the purposes like constructing residual signals, defining test statistics, and quantifying the influences of the unknown faults.

Consider two variables $\boldsymbol{\psi}_1 \in \mathcal{R}^{k_{\psi_1}}$ and $\boldsymbol{\psi}_2 \in \mathcal{R}^{k_{\psi_2}}$. If a metric $\mathcal{M}$ is chosen as the Euclidean distance, one has

$$\mathcal{M}_{\mathbf{euc}} = \|\boldsymbol{\psi}_1 - \boldsymbol{\psi}_2\|_2 , \tag{1}$$

where $k_{\psi_1} = k_{\psi_2}$. By introducing a covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\psi}}$, the Mahalanobis distance can be defined by

$$\mathcal{M}^2_{\mathbf{mah}} = (\boldsymbol{\psi}_1 - \boldsymbol{\psi}_2)^T \boldsymbol{\Sigma}_{\boldsymbol{\psi}}^{-1} (\boldsymbol{\psi}_1 - \boldsymbol{\psi}_2), \tag{2}$$

in which $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$ are from the same distribution with covariance $\boldsymbol{\Sigma}_{\boldsymbol{\psi}}$. By setting $\boldsymbol{\Sigma}_{\boldsymbol{\psi}}^{-1} = \mathbf{I}$, (3) becomes

$$\mathcal{M}^2_{\mathbf{spe}} = (\boldsymbol{\psi}_1 - \boldsymbol{\psi}_2)^T (\boldsymbol{\psi}_1 - \boldsymbol{\psi}_2), \tag{3}$$

which is called squared prediction error (SPE). A metric can also be defined by the correlation such as [23]

$$\mathcal{M}_{\mathbf{cor}}(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2) = k_{\boldsymbol{\psi}_1} - \mathrm{Tr}\left(|\boldsymbol{\Sigma}_{\boldsymbol{\psi}_1}^{-1/2} \boldsymbol{\Sigma}_{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2} \boldsymbol{\Sigma}_{\boldsymbol{\psi}_2}^{-1/2}| \right), \tag{4a}$$

$$\mathcal{M}_{\mathbf{cor}}(\boldsymbol{\psi}_2, \boldsymbol{\psi}_1) = k_{\boldsymbol{\psi}_2} - \mathrm{Tr}\left(|\boldsymbol{\Sigma}_{\boldsymbol{\psi}_2}^{-1/2} \boldsymbol{\Sigma}_{\boldsymbol{\psi}_2, \boldsymbol{\psi}_1} \boldsymbol{\Sigma}_{\boldsymbol{\psi}_1}^{-1/2}| \right), \tag{4b}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2}$ is the covariance matrix between $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$, and k_{ψ_1} may not equal to k_{ψ_2}.

B. Unsupervised and Supervised Machine Learning

Depending upon linear or nonlinear mappings, machine learning is generally divided into linear and nonlinear approaches. An important step is to evaluate performance through the defined matrices (such as $\mathcal{M}_{\mathbf{euc}}$, $\mathcal{M}_{\mathbf{mah}}$, $\mathcal{M}_{\mathbf{spe}}$, and $\mathcal{M}_{\mathbf{cor}}$), and the subsequent step will be devoted to revealing the relationship between unsupervised and supervised machine learning approaches.

Define $\boldsymbol{\psi}_2 = \boldsymbol{\mathcal{S}} \boldsymbol{\psi}_3$, where $\boldsymbol{\mathcal{S}}$ represents the mapping of supervised learning methods. Considering the objective function to minimize (1), the following relation holds

$$\begin{aligned} \min \mathcal{M}_{\mathbf{euc}} &= \min \left\| [\mathbf{I} \quad -\boldsymbol{\mathcal{S}}] \begin{bmatrix} \boldsymbol{\psi}_1 \\ \boldsymbol{\psi}_3 \end{bmatrix} \right\|_2 \\ &= \min \left\| \underbrace{\boldsymbol{\mathcal{P}} \circ [\mathbf{I} \quad -\boldsymbol{\mathcal{S}}]}_{\text{unsupervised learning}} \begin{bmatrix} \boldsymbol{\psi}_1 \\ \boldsymbol{\psi}_3 \end{bmatrix} \right\|_2 , \end{aligned} \tag{5}$$

where $\boldsymbol{\mathcal{P}}$ is any (linear or nonlinear) operator that does not cause information loss or change the minimization of (5). In (1) and (5), $\boldsymbol{\psi}_1$ is an reference signal; $\boldsymbol{\psi}_2$ and $\boldsymbol{\psi}_3$ are the output and input of supervised learning models, respectively. By introducing $\boldsymbol{\mathcal{P}}$, $\boldsymbol{\mathcal{P}} \circ [\mathbf{I} \quad -\boldsymbol{\mathcal{S}}]$ is a composite operator whose input is $[\boldsymbol{\psi}_1^T \quad \boldsymbol{\psi}_3^T]^T$. Instead of optimizing $\boldsymbol{\mathcal{S}}$, we can treat $\boldsymbol{\mathcal{P}} \circ [\mathbf{I} \quad -\boldsymbol{\mathcal{S}}]$ as a whole item to learn. In this manner, the objective function given in (5) can be minimized through an unsupervised learning approach. Similarly, the objective functions defined by other metrics (including $\mathcal{M}_{\mathbf{mah}}$, $\mathcal{M}_{\mathbf{spe}}$, and $\mathcal{M}_{\mathbf{cor}}$) can also be achieved.

It is seen that, through simple mathematical derivations, an objective function can be achieved by using both the supervised and unsupervised approaches.

Remark 1. *The concept behind the aforementioned transformations is a bridge linking the unsupervised to supervised machine learning methods. This fact motivates the current study to focus on the development of a unified framework of the IFD and related parameter-identification approaches for nonlinear systems.* ∇

C. Nonlinear Dynamic Systems and Objectives

Consider a nonlinear dynamic system driven by

$$\begin{aligned} \mathbf{x}(k+1) &= \boldsymbol{\phi}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k)), \\ \mathbf{y}(k) &= \boldsymbol{\upsilon}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{v}(k), \end{aligned} \tag{6}$$

where k is the discrete time index; $\mathbf{x}(k) \in \mathcal{R}^{k_x}$, $\mathbf{u}(k) \in \mathcal{R}^{k_u}$, and $\mathbf{y}(k) \in \mathcal{R}^{k_y}$ are the state, input, and output vectors,

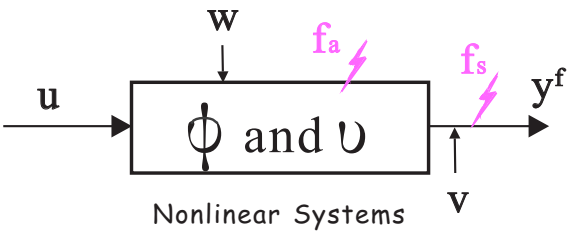


Fig. 1: Nonlinear dynamic systems with actuator and sensor faults.

respectively; $\mathbf{w}(k)$ and $\mathbf{v}(k)$ are random noise variables; $\boldsymbol{\phi}(\cdot)$ and $\boldsymbol{\upsilon}(\cdot)$ are continuous nonlinear mappings.

Both the actuator and sensor faults affect (6). Mathematically, the nonlinear system with faults becomes

$$\begin{aligned}\mathbf{x}^{\mathbf{f}}(k+1) &= \boldsymbol{\phi}(\mathbf{x}(k),\mathbf{u}(k),\mathbf{w}(k),\mathbf{f}_{\mathbf{a}}(k)),\\ \mathbf{y}^{\mathbf{f}}(k) &= \boldsymbol{\upsilon}(\mathbf{x}(k),\mathbf{u}(k),\mathbf{f}_{\mathbf{a}}(k)) + \mathbf{f}_{\mathbf{s}}(k) + \mathbf{v}(k),\end{aligned} \tag{7}$$

where $\mathbf{f_a}$ and $\mathbf{f_s}$ represent actuator and sensor faults, respectively. The schematic of (7) is given in Fig. 1. Now we introduce a lemma (given in [24]) that constitutes another foundation of this study.

Lemma 1. *An operator $\mathcal{K}$ is called the generalized kernel representation for the nonlinear system (6) if for $\mathbf{w}=\mathbf{0}$ and $\mathbf{v}=\mathbf{0}$, the following relationship holds*

$$\mathcal{K}(z)\begin{bmatrix}\mathbf{u}(z)\\ \mathbf{y}(z)\end{bmatrix} = \mathbf{0}, \tag{8}$$

for an initial system state $\mathbf{x}(0)$ and a given $\mathbf{u}$.

In (8), z signifies the discrete variables in the z domain. Motivated by [16], [21], [23], and [25], this study focuses on IFD and its related parameter identification for nonlinear dynamic systems. The three objectives are casted as follows.

- To construct a unified IFD framework that can include both unsupervised and supervised learning-based approaches;
- To design two IFD approaches using the unsupervised and supervised neural networks, respectively;
- To quantify the fault influences in each situation, whose purpose is to mine the interpretability of IFD methods.

III. A Unified IFD Framework: From Supervised to Unsupervised Learning

By using the composite operators and suspected spaces (see [23]), this section will present data-driven implementation of residual generators using both unsupervised and supervised learning, with a focus on construction of the bridge between unsupervised and supervised learning-based approaches.

A. Data-based Modeling

Because both $\mathbf{x}(k)$ and the innovation from $\mathbf{x}(k-1)$ to $\mathbf{x}(k)$ are unknown, the extended form of (6) is usually adopted for system identification and fault diagnosis. For obtaining the extended form, several notations, referring to data and operators, are introduced:

$$\begin{aligned}&\boldsymbol{\phi}_{\wp_1\wp_2}\wp_2(k): \quad \wp_2(k)\in\mathcal{R}^{k_{\wp_2}} \to \wp_1(k+1)\in\mathcal{R}^{k_{\wp_1}},\\ &\boldsymbol{\upsilon}_{\wp_1\wp_2}\wp_2(k): \quad \wp_2(k)\in\mathcal{R}^{k_{\wp_2}} \to \wp_1(k)\in\mathcal{R}^{k_{\wp_1}},\\ &\boldsymbol{\phi}^{k}_{\wp_1\wp_1} = \boldsymbol{\phi}^{k-1}_{\wp_1\wp_1}\circ\boldsymbol{\phi}_{\wp_1\wp_1} = \boldsymbol{\phi}^{k-2}_{\wp_1\wp_1}\circ\boldsymbol{\phi}_{\wp_1\wp_1}\circ\boldsymbol{\phi}_{\wp_1\wp_1},\end{aligned} \tag{9}$$

and

$$\wp_s(k) = [\wp^T(k) \quad \cdots \quad \wp^T(k+s)]^T \in \mathcal{R}^{(s+1)k_\wp}. \tag{10}$$

In (9), $\wp_1$ and $\wp_2$ are vectors; $\boldsymbol{\phi}_{\wp_1\wp_2}$ and $\boldsymbol{\upsilon}_{\wp_1\wp_2}$ are linear or nonlinear operators, respectively; $\boldsymbol{\phi}^{k}_{\wp_1\wp_1}$ represents a high-order composite operator. In (10), s is the stack length. Furthermore, $\wp_1$, $\wp_2$, and $\wp$ can be replaced by any variable in (6) and (7); $\boldsymbol{\phi}_{\cdot,\cdot}$ and $\boldsymbol{\upsilon}_{\cdot,\cdot}$ can also be replaced by any operator.

Remark 2. *In order to parameterize nonlinear dynamic systems, the composite operators defined in (9), similar to Koopman operators [26], simplify the treatment processes for obtaining an equivalent system representation [27].* ∇

By using the notation given in (9), we rewrite (6) as

$$\mathbf{x}(k+1) = \underbrace{\begin{pmatrix}\boldsymbol{\phi}_{\mathbf{xx}} & \boldsymbol{\phi}_{\mathbf{xu}} & \boldsymbol{\phi}_{\mathbf{xw}}\end{pmatrix}}_{\boldsymbol{\phi}}\begin{bmatrix}\mathbf{x}(k)\\ \mathbf{u}(k)\\ \mathbf{w}(k)\end{bmatrix}, \tag{11a}$$

$$\mathbf{y}(k) = \underbrace{\begin{pmatrix}\boldsymbol{\upsilon}_{\mathbf{yx}} & \boldsymbol{\upsilon}_{\mathbf{yu}} & \boldsymbol{\upsilon}_{\mathbf{yv}}\end{pmatrix}}_{\boldsymbol{\upsilon}}\begin{bmatrix}\mathbf{x}(k)\\ \mathbf{u}(k)\\ \mathbf{v}(k)\end{bmatrix}, \tag{11b}$$

where $\boldsymbol{\upsilon}_{\mathbf{yv}} = \mathbf{I}$. Similar to the parity space in [2] and [11], the data-based system model can be described as

$$\mathbf{y}_s(k) = \begin{pmatrix}\boldsymbol{\Upsilon}_{\mathbf{x}} & \boldsymbol{\Upsilon}_{\mathbf{u}}\end{pmatrix}\begin{bmatrix}\mathbf{x}(k)\\ \mathbf{u}_s(k)\end{bmatrix} + \boldsymbol{\Upsilon}_{\mathbf{w}}\mathbf{w}_s(k) + \mathbf{v}_s(k), \tag{12}$$

in which the composite operators $\boldsymbol{\Upsilon}_{\mathbf{x}}$, $\boldsymbol{\Upsilon}_{\mathbf{u}}$, and $\boldsymbol{\Upsilon}_{\mathbf{w}}$ are

$$\boldsymbol{\Upsilon}_{\mathbf{x}} := \begin{pmatrix}\boldsymbol{\upsilon}_{\mathbf{yx}}\\ \vdots\\ \boldsymbol{\upsilon}_{\mathbf{yx}}\boldsymbol{\phi}^{s}_{\mathbf{xx}}\end{pmatrix} : \quad \mathcal{R}^{k_x} \to \mathcal{R}^{(s+1)k_y}, \tag{13}$$

$$\boldsymbol{\Upsilon}_{\mathbf{u}} := \begin{pmatrix}\boldsymbol{\upsilon}_{\mathbf{yu}} & \cdots & \mathbf{0}\\ \vdots & \ddots & \vdots\\ \boldsymbol{\upsilon}_{\mathbf{yx}}\boldsymbol{\phi}^{s-1}_{\mathbf{xx}}\boldsymbol{\phi}_{\mathbf{xu}} & \cdots & \boldsymbol{\upsilon}_{\mathbf{yu}}\end{pmatrix} : \mathcal{R}^{(s+1)k_u} \to \mathcal{R}^{(s+1)k_y}, \tag{14}$$

$$\boldsymbol{\Upsilon}_{\mathbf{w}} := \begin{pmatrix}\mathbf{0} & \cdots & \mathbf{0}\\ \vdots & \ddots & \vdots\\ \boldsymbol{\upsilon}_{\mathbf{yx}}\boldsymbol{\phi}^{s-1}_{\mathbf{xx}}\boldsymbol{\phi}_{\mathbf{xw}} & \cdots & \mathbf{0}\end{pmatrix} : \mathcal{R}^{(s+1)k_x} \to \mathcal{R}^{(s+1)k_y}. \tag{15}$$

In the data-based model (12), the state $\mathbf{x}(k)$ is generally unknown and needs to be estimated by using the past data. For example, the studies [21], [28], [29] suggest to estimate $\mathbf{y}$ through data in the past moving window, i.e.,

$$\mathbf{e}(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k), \hat{\mathbf{y}}(k) = \boldsymbol{\upsilon}(\hat{\mathbf{x}}(k), \mathbf{u}(k)), \tag{16a}$$

$$\hat{\mathbf{x}}(k+1) = \boldsymbol{\phi}(\hat{\mathbf{x}}(k), \mathbf{u}(k)) + \boldsymbol{\ell}(\mathbf{y}(k) - \hat{\mathbf{y}}(k)), \tag{16b}$$

in which $\boldsymbol{\ell} : \mathcal{R}^{k_y} \to \mathcal{R}^{k_x}$ signifies the (unknown but existing) projection from $\mathbf{e}$ to $\hat{\mathbf{x}}$. Therefore, one can obtain that

$$\begin{aligned} &\hat{\mathbf{x}}(k) = \boldsymbol{\phi}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}^{s_p+1}\hat{\mathbf{x}}(k-s_p-1) + [\boldsymbol{\mathcal{L}}_{\mathbf{p},\mathbf{u}} \quad \boldsymbol{\mathcal{L}}_{\mathbf{p},\mathbf{y}}]\mathbf{z}_{\mathbf{p}}(k), \\ &\boldsymbol{\phi}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}^{s_p+1}\hat{\mathbf{x}}(k-s_p-1) \approx \mathbf{0}, \\ &\Longrightarrow \quad \hat{\mathbf{x}}(k) \approx \underbrace{[\boldsymbol{\mathcal{L}}_{\mathbf{p},\mathbf{u}} \quad \boldsymbol{\mathcal{L}}_{\mathbf{p},\mathbf{y}}]}_{\boldsymbol{\mathcal{L}}_{\mathbf{p}}}\mathbf{z}_{\mathbf{p}}(k), \end{aligned} \tag{17}$$

where $\boldsymbol{\mathcal{L}}_{\mathbf{p},\mathbf{u}}$ and $\boldsymbol{\mathcal{L}}_{\mathbf{p},\mathbf{y}}$ are the composite operators similar to (14)-(15); the past stacked vector $\mathbf{z}_{\mathbf{p}}(k)$ is defined by

$$\begin{aligned} &\mathbf{z}_{\mathbf{p}}(k) = \begin{bmatrix} \mathbf{u}_{\mathbf{p}}(k) \\ \mathbf{y}_{\mathbf{p}}(k) \end{bmatrix}, \mathbf{u}_{\mathbf{p}}(k) = \mathbf{u}_{s_p}(k-s_p-1), \\ &\mathbf{y}_{\mathbf{p}}(k) = \mathbf{y}_{s_p}(k-s_p-1). \end{aligned} \tag{18}$$

Combining (12) with (17) yields an equivalent model:

$$\begin{aligned} \mathbf{y}_s(k) &= \boldsymbol{\Upsilon}_{\mathbf{x}}\boldsymbol{\mathcal{L}}_{\mathbf{p}}\mathbf{z}_{\mathbf{p}}(k) + \boldsymbol{\Upsilon}_{\mathbf{u}}\mathbf{u}_s(k) + \boldsymbol{\Upsilon}_{\mathbf{e}}\mathbf{e}_s(k) \\ &= (\boldsymbol{\Upsilon}_{\mathbf{x}}\boldsymbol{\mathcal{L}}_{\mathbf{p}} \quad \boldsymbol{\Upsilon}_{\mathbf{u}}) \begin{bmatrix} \mathbf{z}_{\mathbf{p}}(k) \\ \mathbf{u}_s(k) \end{bmatrix} + \boldsymbol{\Upsilon}_{\mathbf{e}}\mathbf{e}_s(k) \end{aligned} \tag{19}$$

where $\mathbf{e}_s$ contains the influences caused by $\mathbf{w}$ and $\mathbf{v}$, and $\boldsymbol{\Upsilon}_{\mathbf{e}}$ has the following form:

$$\boldsymbol{\Upsilon}_{\mathbf{e}} = \begin{pmatrix} \mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}}\boldsymbol{\phi}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}^{s-1}\boldsymbol{\ell}_{\hat{\mathbf{x}}\mathbf{e}} & \cdots & \mathbf{I} \end{pmatrix}. \tag{20}$$

Remark 3. *By the use of the composite operator, (19) details a generalized nonlinear predictor from $\mathbf{z}_{\mathbf{p}}$ and $\mathbf{u}_s$ to the current system output $\mathbf{y}_s$, while resembling a linear model. It makes system modeling possible and explainable, especially achieving a consensus among different IFD approaches using both unsupervised and supervised learning.* ∇

B. Data-driven Implementation of Residual Generators

On the basis of Lemma 1, the following corollary is obtained, whose embryonic development credits to [21].

Corollary 1. *An operator $\boldsymbol{\mathcal{K}}_{s_p+s}$ is called the data-driven generalized kernel representation for (6) if for $\mathbf{w} = \mathbf{0}$ and $\mathbf{v} = \mathbf{0}$, the following relationship holds*

$$\underbrace{(\boldsymbol{\mathcal{K}}_{\mathbf{z}}^{\mathbf{unsp}} \quad \boldsymbol{\mathcal{K}}_{\mathbf{u}}^{\mathbf{unsp}} \quad \boldsymbol{\mathcal{K}}_{\mathbf{y}}^{\mathbf{unsp}})}_{\boldsymbol{\mathcal{K}}_{s_p+s}^{\mathbf{unsp}}} \begin{bmatrix} \mathbf{z}_{\mathbf{p}} \\ \mathbf{u}_s \\ \mathbf{y}_s \end{bmatrix} = \mathbf{0}, \tag{21a}$$

$$or \ \underbrace{(\underline{\boldsymbol{\mathcal{K}}_{s_p+s}} \quad -\mathbf{I})}_{\boldsymbol{\mathcal{K}}_{s_p+s}^{\mathbf{sp}}} \begin{bmatrix} \mathbf{z}_{\mathbf{p}} \\ \mathbf{u}_s \\ \mathbf{y}_s \end{bmatrix} = \mathbf{0}, \tag{21b}$$

for an initial system state $\mathbf{x}(0)$ and an arbitrary $\mathbf{u}_s$, where the superscripts "$\mathbf{unsp}$" and "$\mathbf{sp}$" signify the unsupervised and supervised learning scenarios, respectively.

The obtained $\boldsymbol{\mathcal{K}}_{s_p+s}$, by using the unsupervised or supervised learning methods, can be directly applied in constructing a unified residual generator such that

$$\mathbf{r}_s(k) = \boldsymbol{\mathcal{K}}_{s_p+s} \begin{bmatrix} \mathbf{z}_{\mathbf{p}}(k) \\ \mathbf{u}_s(k) \\ \mathbf{y}_s(k) \end{bmatrix} \in \mathcal{R}^{(s+1)k_y}, \tag{22}$$

in fault-free conditions; $\boldsymbol{\mathcal{K}}_{s_p+s}$ can be replaced by $\boldsymbol{\mathcal{K}}_{s_p+s}^{\mathbf{unsp}}$ and $\boldsymbol{\mathcal{K}}_{s_p+s}^{\mathbf{sp}}$ provided in Corollary 1. When (6) is affected by faults, $\mathbf{r}_s(k)$ given in (22) becomes

$$\mathbf{r}_s^{\mathbf{f}}(k) = \mathbf{r}_s(k) + \mathbf{f}_{\mathbf{a},\mathbf{term}} + \mathbf{f}_{\mathbf{s},\mathbf{term}}, \tag{23}$$

where $\mathbf{f}_{\mathbf{a},\mathbf{term}}$ and $\mathbf{f}_{\mathbf{s},\mathbf{term}}$ are the actuator and sensor fault-related terms, respectively; their forms can be obtained via composite operators. Then the test statistics [1], such as T^2, can be defined on the residual signal $\mathbf{r}_s$, i.e.,

$$T^2(\mathbf{r}_s(k)) = \mathbf{r}_s^T(k)\boldsymbol{\Sigma}_{\mathbf{r}_s}^{-1}\mathbf{r}_s(k), \ \boldsymbol{\Sigma}_{\mathbf{r}_s}^{-1} = \mathbb{E}(\mathbf{r}_s(k)\mathbf{r}_s^T(k)). \tag{24}$$

Correspondingly, the threshold is determined by the chosen confidence level α:

$$J_{th} \leftarrow \Pr(T^2(\mathbf{r}_s(k)) > J_{th}) = \alpha. \tag{25}$$

The following theorem presents the essence of this study, sketching ***the bridge*** (i.e., *one-to-one mapping*) between unsupervised and supervised learning-based parameter identification, together with the IFD approaches, for nonlinear systems.

Theorem 1. *Consider the two residual generators defined by*

$$\mathbf{r}_s^{\mathbf{unsp}}(k) = \boldsymbol{\mathcal{K}}_{s_p+s}^{\mathbf{unsp}} \begin{bmatrix} \mathbf{z}_{\mathbf{p}}(k) \\ \mathbf{u}_s(k) \\ \mathbf{y}_s(k) \end{bmatrix}, \mathbf{r}_s^{\mathbf{sp}}(k) = \boldsymbol{\mathcal{K}}_{s_p+s}^{\mathbf{sp}} \begin{bmatrix} \mathbf{z}_{\mathbf{p}}(k) \\ \mathbf{u}_s(k) \\ \mathbf{y}_s(k) \end{bmatrix} \tag{26}$$

where $\boldsymbol{\mathcal{K}}_{s_p+s}^{\mathbf{unsp}}$ and $\boldsymbol{\mathcal{K}}_{s_p+s}^{\mathbf{sp}}$ are defined by

$$\boldsymbol{\mathcal{K}}_{s_p+s}^{\mathbf{unsp}} = [\mathbf{0} \quad \mathbf{0} \quad \mathbf{I}] - \boldsymbol{\mathcal{U}}_{\mathbf{y}}, \tag{27a}$$

$$\boldsymbol{\mathcal{K}}_{s_p+s}^{\mathbf{sp}} = (-\boldsymbol{\Upsilon}_{\mathbf{x}}\boldsymbol{\mathcal{L}}_{\mathbf{p}} \quad -\boldsymbol{\Upsilon}_{\mathbf{u}} \quad \mathbf{I}), \tag{27b}$$

in which $\mathcal{U}_{\mathbf{y}}$ is a segment of unsupervised learning $\mathcal{U}$:

$$\begin{bmatrix} \hat{\mathbf{z}}_{\mathbf{p}}(k) \\ \hat{\mathbf{u}}_s(k) \\ \hat{\mathbf{y}}_s(k) \end{bmatrix} = \underbrace{(\mathcal{U}_{\mathbf{z}} \ \mathcal{U}_{\mathbf{u}} \ \mathcal{U}_{\mathbf{y}})}_{\mathcal{U}} \begin{bmatrix} \mathbf{z}_{\mathbf{p}}(k) \\ \mathbf{u}_s(k) \\ \mathbf{y}_s(k) \end{bmatrix}. \tag{28}$$

There exists a nonlinear operator $\mathcal{P}_{\mathbf{sp/unsp}}$ such that

$$\mathbf{r}_s^{\mathbf{sp}}(k) = \mathcal{P}_{\mathbf{sp/unsp}} \mathbf{r}_s^{\mathbf{unsp}}(k), \tag{29}$$

and the inverse function of $\mathcal{P}_{\mathbf{sp/unsp}}$, denoted as $\mathcal{P}_{\mathbf{unsp/sp}} = \mathcal{P}_{\mathbf{sp/unsp}}^{-1}$, must exist such that

$$\mathbf{r}_s^{\mathbf{unsp}}(k) = \mathcal{P}_{\mathbf{unsp/sp}} \mathbf{r}_s^{\mathbf{sp}}(k). \tag{30}$$

Proof: The complete proof is given in Appendix A. ■

The following remark is made to set forth contributions of Theorem 1.

Remark 4. *With the help of the bridge, Theorem 1 provides us with an elegant way to evaluate the performance of both unsupervised and supervised learning-based IFD approaches in the sense of fault-detection power.* ▽

Based on the metric $\mathcal{M}_{\mathbf{euc}}^2$, the suspected space is used to design IFD approaches, whose purpose is to gain a more in-depth understanding of Theorem 1.

C. IFD Using Unsupervised Learning

In order to develop the unsupervised learning-based IFD approaches, a suspected space, denoted as $\mathcal{Y}^{\mathbf{unsp}}$, is defined according to $\mathcal{U}_{\mathbf{y}}$ in the following definition.

Definition 1. *Given any $\mathcal{U}$ in (28), $\underline{\mathbf{y}}_s^{\mathbf{unsp}}$ obtained by*

$$\underline{\mathbf{y}}_s^{\mathbf{unsp}} = \mathcal{U}_{\mathbf{y}} [\mathbf{z}_{\mathbf{p}}^T \ \ \mathbf{u}_s^T \ \ \mathbf{y}_s^T]^T, \tag{31}$$

spans a space $\mathcal{Y}^{\mathbf{unsp}}$ that is called the suspected space of (6).

Corresponding to $\mathcal{Y}^{\mathbf{unsp}}$, $\mathcal{Y}$ is called the measurement space, where $\mathbf{y}_s \in \mathcal{Y} \subset \mathcal{R}^{(s+1)k_y}$. Based on the metric defined in (1), the objective function of IFD approaches using unsupervised learning can be formulated as

$$\min \ \mathcal{M}_{\mathbf{euc}}(\mathbf{y}_s, \underline{\mathbf{y}}_s^{\mathbf{unsp}}), \tag{32a}$$

$$\Longrightarrow \mathcal{U}_{\mathbf{y}}^* := \arg\min_{\mathcal{U}_{\mathbf{y}}} \ \mathcal{M}_{\mathbf{euc}}(\mathbf{y}_s, \underline{\mathbf{y}}_s^{\mathbf{unsp}}), \tag{32b}$$

where the superscript "$*$" signifies the best solution. It is of interest to find that the residual space, denoted as $\mathcal{E}_s^{\mathbf{unsp}}$, can be obtained based on $\mathcal{U}_{\mathbf{y}}^*$:

$$\begin{aligned} \mathbf{e}_s &= \mathbf{y}_s - \underline{\mathbf{y}}_s^{\mathbf{unsp},*}, \mathbf{e}_s \in \mathcal{E}_s^{\mathbf{unsp}} \subset \mathcal{R}^{(s+1)k_y}, \\ \underline{\mathbf{y}}_s^{\mathbf{unsp},*} &= \mathcal{U}_{\mathbf{y}}^* [\mathbf{z}_{\mathbf{p}}^T \ \ \mathbf{u}_s^T \ \ \mathbf{y}_s^T]^T \in \mathcal{Y}^{\mathbf{unsp},*}, \end{aligned} \tag{33}$$

which satisfies

$$\mathcal{E}_s^{\mathbf{unsp}} \perp \mathcal{Y}^{\mathbf{unsp},*}, \ \ \mathcal{E}_s^{\mathbf{unsp}} \ (\approx) \perp \mathcal{Y}. \tag{34}$$

Theorem 2. *Considering a nonlinear system (6), its generalized kernel representation is defined by $\mathcal{K}_{s_p+s}^{\mathbf{unsp}} = [\mathbf{0} \ \ \mathbf{0} \ \ \mathbf{I}] - \mathcal{U}_{\mathbf{y}}^*$, where $\mathcal{U}_{\mathbf{y}}^*$ is obtained through (32b). For the faults $\mathbf{f_a}(k)$ and $\mathbf{f_s}(k)$ occurring from the k-th time instant, $\mathbf{r}_s^{\mathbf{unsp}}(k)$ given in (26) becomes*

$$\begin{aligned} \mathbf{r}_s^{\mathbf{unsp,f}}(k) &= \mathbf{e}_s(k) + \mathbf{f}_{\mathbf{a,term}}^{\mathbf{unsp}}(k) + \mathbf{f}_{\mathbf{s,term}}^{\mathbf{unsp}}(k), \\ \mathbf{f}_{\mathbf{a,term}}^{\mathbf{unsp}}(k) &= \underline{\boldsymbol{\Upsilon}}_{\mathbf{f_a}} \mathbf{f}_{\mathbf{a},s}(k), \ \mathbf{f}_{\mathbf{s,term}}^{\mathbf{unsp}}(k) = \underline{\boldsymbol{\Upsilon}}_{\mathbf{f_s}} \mathbf{f}_{\mathbf{s},s}, \end{aligned} \tag{35}$$

where $\mathbf{f}_{\mathbf{a,term}}^{\mathbf{unsp}}$ and $\mathbf{f}_{\mathbf{s,term}}^{\mathbf{unsp}}$ have the following forms:

$$\underline{\boldsymbol{\Upsilon}}_{\mathbf{f_a}} = \underline{\boldsymbol{\Upsilon}}_{\mathbf{u}}, \underline{\boldsymbol{\Upsilon}}_{\mathbf{f_s}} = \begin{pmatrix} \mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ -\boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}} \boldsymbol{\mathcal{A}}_{\boldsymbol{\ell}}^{s-1} \boldsymbol{\ell}_{\hat{\mathbf{x}}\mathbf{e}} & \cdots & \mathbf{I} \end{pmatrix}. \tag{36}$$

Then T^2 defined on $\mathbf{r}_s^{\mathbf{unsp,f}}(k)$ according to

$$T^2(\mathbf{r}_s^{\mathbf{unsp,f}}(k)) = \mathbf{r}_s^{\mathbf{unsp,f},T}(k) \boldsymbol{\Sigma}_{\mathbf{r}_s}^{-1} \mathbf{r}_s^{\mathbf{unsp,f}}(k) \tag{37}$$

has the optimal fault-detection power.

Proof: When (6) is fault-free, we have

$$\mathbf{r}_s^{\mathbf{unsp}}(k) = \mathbf{e}_s(k), \quad \boldsymbol{\Sigma}_{\mathbf{r}_s^{\mathbf{unsp}}} = \mathbb{E}(\mathbf{e}_s(k)\mathbf{e}_s^T(k)). \tag{38}$$

Since $\mathbf{f_a}$ and $\mathbf{f_s}$ respectively represent the actuator and sensor faults, it is reasonable to adopt

$$\boldsymbol{\phi}_{\mathbf{xf_a}} = \boldsymbol{\phi}_{\mathbf{xu}}, \ \boldsymbol{\upsilon}_{\mathbf{yf_a}} = \boldsymbol{\upsilon}_{\mathbf{yu}}, \ \text{ and } \ \boldsymbol{\upsilon}_{\mathbf{yf_s}} = \mathbf{I}. \tag{39}$$

When (6) is affected by $\mathbf{f_a}$ and $\mathbf{f_s}$, $\hat{\mathbf{x}}^{\mathbf{f}}(k)$ and $\mathbf{y}^{\mathbf{f}}(k)$ in (16a)-(16b) can be expressed by

$$\begin{aligned} \hat{\mathbf{x}}^{\mathbf{f}}(k+1) =& \boldsymbol{\phi}_{\hat{\mathbf{x}}\hat{\mathbf{x}}} \hat{\mathbf{x}}(k) + \boldsymbol{\phi}_{\hat{\mathbf{x}}\mathbf{u}} \mathbf{u}(k) + \boldsymbol{\phi}_{\hat{\mathbf{x}}\mathbf{u}} \mathbf{f_a}(k) \\ &+ \boldsymbol{\ell}(\mathbf{y}^{\mathbf{f}}(k) - \hat{\mathbf{y}}(k) - \mathbf{f_s}), \\ \mathbf{y}^{\mathbf{f}}(k) =& \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}} \hat{\mathbf{x}}(k) + \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\mathbf{u}} \mathbf{u}(k) + \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\mathbf{u}} \mathbf{f_a}(k) \\ &+ \boldsymbol{\upsilon}_{\mathbf{yf_s}} \mathbf{f_s}(k) + \mathbf{e}(k). \end{aligned} \tag{40}$$

Combining it with (A.2) yields (35) with $\underline{\boldsymbol{\Upsilon}}_{\mathbf{f_a}}$ and $\underline{\boldsymbol{\Upsilon}}_{\mathbf{f_s}}$ described by (36). Furthermore, taking expectation of (37) obtains

$$\begin{aligned} \mathbb{E}(T^2(\mathbf{r}_s^{\mathbf{unsp,f}}(k))) =& T^2(\mathbf{e}_s) + \mathbf{f}_{\mathbf{a,term}}^T \boldsymbol{\Sigma}_{\mathbf{r}_s^{\mathbf{unsp}}}^{-1} \mathbf{f}_{\mathbf{a,term}} \\ &+ \mathbf{f}_{\mathbf{f,term}}^T \boldsymbol{\Sigma}_{\mathbf{r}_s^{\mathbf{unsp}}}^{-1} \mathbf{f}_{\mathbf{f,term}}, \end{aligned} \tag{41}$$

because in general $\mathbf{e}_s(k)$, $\mathbf{f}_{\mathbf{a,term}}(k)$, and $\mathbf{f}_{\mathbf{f,term}}(k)$ are independent. As proved in [23], $\mathcal{U}_{\mathbf{y}}^*$ minimizes $\boldsymbol{\Sigma}_{\mathbf{r}_s^{\mathbf{unsp}}}$ via (32b); thus the last two terms in (41) are maximized to the greatest extent possible. It means $\mathcal{U}_{\mathbf{y}}^*$, a segment of $\mathcal{U}^*$, can achieve optimal performance in detecting $\mathbf{f_a}$ and $\mathbf{f_s}$.

The theorem is proved. ■

D. IFD Using Supervised Learning

In parallel to $\mathcal{Y}^{\mathbf{unsp}}$, another suspected space $\mathcal{Y}^{\mathbf{sp}}$ using supervised learning can be defined according to $\boldsymbol{\mathcal{S}}$ as follows.

Definition 2. *Given any $\boldsymbol{\mathcal{S}}$ in (A.8), $\underline{\mathbf{y}}_s^{\mathbf{sp}}$ obtained by*

$$\underline{\mathbf{y}}_s^{\mathbf{sp}} = \boldsymbol{\mathcal{S}} [\mathbf{z}_{\mathbf{p}}^T \ \ \mathbf{u}_s^T]^T, \tag{42}$$

spans a space $\mathcal{Y}^{\mathbf{sp}}$ that is also the suspected space of (6), where $\boldsymbol{\mathcal{S}}$ is defined by

$$\boldsymbol{\mathcal{S}} = (\boldsymbol{\mathcal{S}}_{\mathbf{z}} \ \boldsymbol{\mathcal{S}}_{\mathbf{u}}) := (\boldsymbol{\Upsilon}_{\mathbf{x}} \boldsymbol{\mathcal{L}}_{\boldsymbol{p}} \ \ \boldsymbol{\Upsilon}_{\mathbf{u}}). \tag{43}$$

Similar to (32a)-(32b), the optimal $\boldsymbol{\mathcal{S}}^*$ can be obtained via

$$\min\ \mathcal{M}_{\mathbf{euc}}(\mathbf{y}_s, \underline{\mathbf{y}}_s^{\mathbf{sp}}), \tag{44a}$$

$$\Longrightarrow \boldsymbol{\mathcal{S}}^* := \arg\min_{\boldsymbol{\mathcal{S}}}\ \mathcal{M}_{\mathbf{euc}}(\mathbf{y}_s, \underline{\mathbf{y}}_s^{\mathbf{sp}}), \tag{44b}$$

based on which $\boldsymbol{\mathcal{Y}}^{\mathbf{sp},*}$ can be obtained.

By the use of $\boldsymbol{\mathcal{S}}^*$, the following theorem is derived for IFD using supervised learning.

Theorem 3. *Considering a nonlinear system (6), its generalized kernel representation is defined by* $\boldsymbol{\mathcal{K}}_{s_p+s}^{\mathbf{sp}} = (\boldsymbol{\mathcal{S}}^* \quad -\mathbf{I})$, *where* $\boldsymbol{\mathcal{S}}^*$ *is obtained through (44b). For the faults* $\mathbf{f_a}(k)$ *and* $\mathbf{f_s}(k)$ *occurring from the* k*-th time instant,* $\mathbf{r}_s^{\mathbf{sp}}(k)$ *given in (26) becomes*

$$\begin{aligned} \mathbf{r}_s^{\mathbf{sp,f}}(k) &= \boldsymbol{\Upsilon}_{\mathbf{e}}\mathbf{e}_s(k) + \mathbf{f}_{\mathbf{a,term}}^{\mathbf{sp}}(k) + \mathbf{f}_{\mathbf{s,term}}^{\mathbf{sp}}(k), \\ \mathbf{f}_{\mathbf{a,term}}^{\mathbf{sp}}(k) &= \boldsymbol{\Upsilon}_{\mathbf{f_a}}\mathbf{f}_{\mathbf{a},s}(k),\ \ \mathbf{f}_{\mathbf{s,term}}^{\mathbf{sp}}(k) = \boldsymbol{\Upsilon}_{\mathbf{f_s}}\mathbf{f}_{\mathbf{s},s}, \end{aligned} \tag{45}$$

where $\mathbf{f}_{\mathbf{a,term}}^{\mathbf{sp}}$ *and* $\mathbf{f}_{\mathbf{s,term}}^{\mathbf{sp}}$ *have the following forms:*

$$\boldsymbol{\Upsilon}_{\mathbf{f_a}} = \boldsymbol{\Upsilon}_{\mathbf{u}}, \boldsymbol{\Upsilon}_{\mathbf{f_s}} = \mathbf{I}. \tag{46}$$

Then T^2 *defined on* $\mathbf{r}_s^{\mathbf{sp,f}}(k)$ *according to*

$$T^2(\mathbf{r}_s^{\mathbf{sp,f}}(k)) = \mathbf{r}_s^{\mathbf{sp,f},T}(k)\boldsymbol{\Sigma}_{\mathbf{r}_s}^{-1}\mathbf{r}_s^{\mathbf{sp,f}}(k) \tag{47}$$

has the optimal fault-detection power.

Proof: The proof is similar to Theorem 2. Owing to space constraints, the detail is omitted here.

In addition, another sketch of the proof for the theorem is presented as follows. By using the bridge $\boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}}$ given in Theorem 1, simple mathematical manipulations can yield

$$T^2(\mathbf{r}_s^{\mathbf{sp}}(k)) = T^2(\mathbf{r}_s^{\mathbf{unsp}}(k)), \tag{48}$$

and (46). Furthermore, it can also be verified

$$T^2(\mathbf{r}_s^{\mathbf{sp,f}}(k)) = T^2(\mathbf{r}_s^{\mathbf{unsp,f}}(k)), \tag{49}$$

which has the same (i.e., optimal) performance of fault detection as $T^2(\mathbf{r}_s^{\mathbf{unsp,f}}(k))$, and thus completes this proof. ■

Remark 5. *As presented in Theorems 2 and 3, performance evaluation of the two proposed IFD frameworks for nonlinear systems becomes possible. The main reason is that the difference between* $\mathbf{y}_s(k)$ *and its estimation in* $\boldsymbol{\mathcal{Y}}^{\mathbf{unsp},*}$ *and* $\boldsymbol{\mathcal{Y}}^{\mathbf{sp},*}$ *can be measured through quantitative metrics.* ▽

E. Notes on the Bridge

In what follows, several remarks (including fault features and geometric interpretation) and perspectives (i.e., a more general version) are made to set forth contributions and essences of the bridge provided in Theorem 1.

1) Fault features: Along with Theorems 2 and 3, the fault features, corresponding to $\boldsymbol{\mathcal{Y}}^{\mathbf{unsp},*}$ and $\boldsymbol{\mathcal{Y}}^{\mathbf{sp},*}$, are

$$\mathbf{r}_s^{\mathbf{unsp,f}} = \mathbf{e}_s + \mathbf{f}_{\mathbf{a,term}}^{\mathbf{unsp}} + \mathbf{f}_{\mathbf{s,term}}^{\mathbf{unsp}}, \tag{50a}$$

$$\mathbf{r}_s^{\mathbf{sp,f}} = \boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}}\mathbf{e}_s + \mathbf{f}_{\mathbf{a,term}}^{\mathbf{sp}} + \mathbf{f}_{\mathbf{s,term}}^{\mathbf{sp}}. \tag{50b}$$

It is interesting to verify

$$\begin{aligned} \boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}}\mathbf{f}_{\mathbf{a,term}}^{\mathbf{unsp}} &= \mathbf{f}_{\mathbf{a,term}}^{\mathbf{sp}}, \\ \boldsymbol{\mathcal{P}}_{\mathbf{unsp/sp}}\mathbf{f}_{\mathbf{a,term}}^{\mathbf{sp}} = \boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}}^{-1}\mathbf{f}_{\mathbf{a,term}}^{\mathbf{sp}} &= \mathbf{f}_{\mathbf{a,term}}^{\mathbf{unsp}}, \end{aligned} \tag{51}$$

for $\mathbf{f}_{\mathbf{a,term}}$, and

$$\begin{aligned} \boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}}\mathbf{f}_{\mathbf{s,term}}^{\mathbf{unsp}} &= \mathbf{f}_{\mathbf{s,term}}^{\mathbf{sp}}, \\ \boldsymbol{\mathcal{P}}_{\mathbf{unsp/sp}}\mathbf{f}_{\mathbf{s,term}}^{\mathbf{sp}} &= \mathbf{f}_{\mathbf{s,term}}^{\mathbf{unsp}}, \end{aligned} \tag{52}$$

for $\mathbf{f}_{\mathbf{s,term}}$. Combining (50a)-(50b) with (51) and (52) yields

$$\boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}}\mathbf{r}_s^{\mathbf{unsp,f}} = \mathbf{r}_s^{\mathbf{sp,f}} = \boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}} \circ \boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}}^{-1}\mathbf{r}_s^{\mathbf{sp,f}}, \tag{53}$$

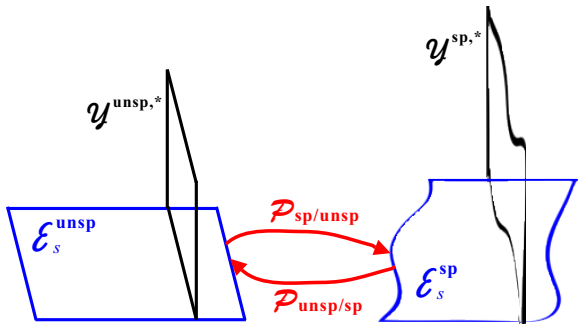
which indicates that, based on the bridge given in Theorem 1, the fault features can be transformed between each other. Also, (53) can be used as an auxiliary evidential statement to illustrate the validity of Theorems 2 and 3 because of

$$\begin{aligned} \boldsymbol{\Sigma}_{\mathbf{r}_s^{\mathbf{unsp}}} &= \mathbb{E}(\mathbf{e}_s(k)\mathbf{e}_s^T(k)), \\ \boldsymbol{\Sigma}_{\mathbf{r}_s^{\mathbf{sp}}} &= \boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}}\boldsymbol{\Sigma}_{\mathbf{r}_s^{\mathbf{unsp}}}\boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}}^T. \end{aligned} \tag{54}$$

such that

$$\begin{aligned} \mathbb{E}(T^2(\mathbf{r}_s^{\mathbf{unsp}}(k))) &= \mathbb{E}(T^2(\mathbf{r}_s^{\mathbf{sp}}(k))), \\ \mathbb{E}(T^2(\mathbf{r}_s^{\mathbf{unsp,f}}(k))) &= \mathbb{E}(T^2(\mathbf{r}_s^{\mathbf{sp,f}}(k))). \end{aligned} \tag{55}$$

2) Geometric interpretation: Fig. 2 provides the geometric descriptions of both unsupervised and supervised learning-aided parameter identification, where the bridge, $\boldsymbol{\mathcal{P}}_{\mathbf{unsp/sp}}$ and $\boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}}$, is highlighted by the two red curves. It is worth mentioning that $\boldsymbol{\mathcal{Y}}^{\mathbf{unsp},*}$ and $\boldsymbol{\mathcal{Y}}^{\mathbf{sp},*}$ reconstruct the dynamic behaviors of nonlinear systems by carrying the maximum variation information (i.e., the largest uncertainty). They are suitable for nonlinear parameter identification. At the same time, $\boldsymbol{\mathcal{E}}^{\mathbf{unsp},*}$ and $\boldsymbol{\mathcal{E}}^{\mathbf{sp},*}$ with the minimum uncertainty are the best choices for fault diagnosis [16].

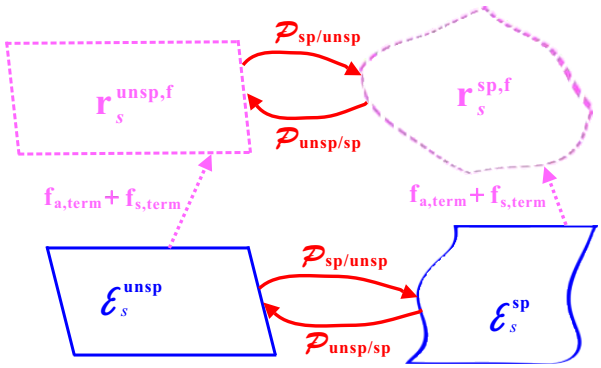


(a) unsupervised learning (b) supervised learning

Fig. 2: Geometric descriptions of the suspected and residual spaces in system identification.

Fig. 3 sketches the changes caused by $\mathbf{f_a}$ and $\mathbf{f_s}$ in residual spaces, together with the same bridge in both normal and faulty scenarios.

3) A generalized version: In Theorems 1 to 3, $\mathcal{M}_{\mathbf{euc}}$ is chosen as the metric to measure the difference not only between $\mathbf{y}_s$ and $\underline{\mathbf{y}}_s^{\mathbf{unsp}}$ when utilizing unsupervised learning to identify $\boldsymbol{\mathcal{K}}_{s_p+s}^{\mathbf{unsp}}$, but also between $\mathbf{y}_s$ and $\underline{\mathbf{y}}_s^{\mathbf{sp}}$ when utilizing supervised learning to estimate $\boldsymbol{\mathcal{K}}_{s_p+s}^{\mathbf{sp}}$. For the fault diagnosis purpose, the same performance can be obtained by using $\mathcal{M}_{\mathbf{cor}}$ defined in (4a)-(4b). As pointed in [23] and [25], an IFD design relying on the $\mathcal{M}_{\mathbf{cor}}$ metric is a generalized version



(a) unsupervised learning (b) supervised learning

Fig. 3: Geometric descriptions of the residual spaces in both normal and faulty scenarios.

that can provide multiple optimal solutions to both $\mathcal{K}_{s_p+s}^{\mathbf{unsp}}$ and $\mathcal{K}_{s_p+s}^{\mathbf{sp}}$.

In order to distinguish it from the solution based on $\mathcal{M}_{\mathbf{euc}}$, the notations in the generalized version are marked by "$\underline{=}$". Therefore, the objective functions of two generalized versions, respectively corresponding to unsupervised and supervised learning, can be formulated as follows.

$$\begin{aligned}&\min\ \mathcal{M}_{\mathbf{cor}}(\mathbf{y}_s,\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}),\\ =&\min\ \ k_{\mathbf{y}_s}-\text{Tr}\left(\left|\boldsymbol{\Sigma}_{\mathbf{y}_s}^{-1/2}\boldsymbol{\Sigma}_{\mathbf{y}_s,\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}}\boldsymbol{\Sigma}_{\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}}^{-1/2}\right|\right),\\ \Longleftrightarrow&\min\ \ k_{\mathbf{y}_s}-\text{Tr}\left(\boldsymbol{\Sigma}^{\mathbf{unsp},T}\boldsymbol{\Sigma}^{\mathbf{unsp}}\right),\end{aligned}\tag{56}$$

and

$$\begin{aligned}&\min\ \mathcal{M}_{\mathbf{cor}}(\mathbf{y}_s,\underline{\underline{\mathbf{y}}}_s^{\mathbf{sp}}),\\ =&\min\ \ k_{\mathbf{y}_s}-\text{Tr}\left(\left|\boldsymbol{\Sigma}_{\mathbf{y}_s}^{-1/2}\boldsymbol{\Sigma}_{\mathbf{y}_s,\underline{\underline{\mathbf{y}}}_s^{\mathbf{sp}}}\boldsymbol{\Sigma}_{\underline{\underline{\mathbf{y}}}_s^{\mathbf{sp}}}^{-1/2}\right|\right),\\ \Longleftrightarrow&\min\ \ k_{\mathbf{y}_s}-\text{Tr}\left(\boldsymbol{\Sigma}^{\mathbf{sp},T}\boldsymbol{\Sigma}^{\mathbf{sp}}\right),\end{aligned}\tag{57}$$

where $\boldsymbol{\Sigma}^{\mathbf{unsp}}$ and $\boldsymbol{\Sigma}^{\mathbf{unsp}}$ are obtained via the following singular value decompositions:

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{y}_s}^{-1/2}\boldsymbol{\Sigma}_{\mathbf{y}_s,\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}}\boldsymbol{\Sigma}_{\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}}^{-1/2}=&\underline{\underline{\boldsymbol{\Gamma}}}^{\mathbf{unsp}}\underline{\underline{\boldsymbol{\Sigma}}}^{\mathbf{unsp}}\underline{\underline{\boldsymbol{\Upsilon}}}^{\mathbf{unsp},T}\\ \boldsymbol{\Sigma}_{\mathbf{y}_s}^{-1/2}\boldsymbol{\Sigma}_{\mathbf{y}_s,\underline{\underline{\mathbf{y}}}_s^{\mathbf{sp}}}\boldsymbol{\Sigma}_{\underline{\underline{\mathbf{y}}}_s^{\mathbf{sp}}}^{-1/2}=&\underline{\underline{\boldsymbol{\Gamma}}}^{\mathbf{sp}}\underline{\underline{\boldsymbol{\Sigma}}}^{\mathbf{sp}}\underline{\underline{\boldsymbol{\Upsilon}}}^{\mathbf{sp},T}.\end{aligned}\tag{58}$$

Now we define two sets of linear mappings as follows.

$$\underline{\underline{\mathbf{G}}}_{\mathbf{y}_s}^{\mathbf{unsp}}=\underline{\underline{\boldsymbol{\Gamma}}}^{\mathbf{unsp},T}\boldsymbol{\Sigma}_{\mathbf{y}_s}^{-1/2},\underline{\underline{\mathbf{G}}}_{\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}}^{\mathbf{unsp}}=\underline{\underline{\boldsymbol{\Upsilon}}}^{\mathbf{unsp},T}\boldsymbol{\Sigma}_{\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}}^{-1/2},\tag{59a}$$

$$\underline{\underline{\mathbf{G}}}_{\mathbf{y}_s}^{\mathbf{unsp}}=\underline{\underline{\boldsymbol{\Gamma}}}^{\mathbf{unsp},T}\boldsymbol{\Sigma}_{\mathbf{y}_s}^{-1/2},\underline{\underline{\mathbf{G}}}_{\underline{\underline{\mathbf{y}}}_s^{\mathbf{sp}}}^{\mathbf{sp}}=\underline{\underline{\boldsymbol{\Upsilon}}}^{\mathbf{sp},T}\boldsymbol{\Sigma}_{\underline{\underline{\mathbf{y}}}_s^{\mathbf{sp}}}^{-1/2}.\tag{59b}$$

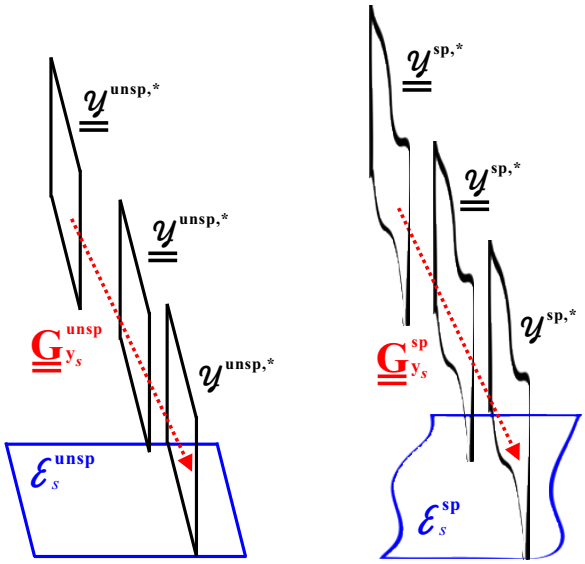
Then the two generalized residual generators can be constructed according to

$$\underline{\underline{\mathbf{r}}}_s^{\mathbf{unsp}}=\underline{\underline{\mathbf{G}}}_{\mathbf{y}_s}^{\mathbf{unsp}}\mathbf{y}_s-\underline{\underline{\boldsymbol{\Sigma}}}^{\mathbf{unsp}}\underline{\underline{\mathbf{G}}}_{\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}}^{\mathbf{unsp}}\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}},\tag{60a}$$

$$\underline{\underline{\mathbf{r}}}_s^{\mathbf{sp}}=\underline{\underline{\mathbf{G}}}_{\mathbf{y}_s}^{\mathbf{sp}}\mathbf{y}_s-\underline{\underline{\boldsymbol{\Sigma}}}^{\mathbf{sp}}\underline{\underline{\mathbf{G}}}_{\underline{\underline{\mathbf{y}}}_s^{\mathbf{sp}}}^{\mathbf{sp}}\underline{\underline{\mathbf{y}}}_s^{\mathbf{sp}}.\tag{60b}$$

It can be verified that

$$\boldsymbol{\Sigma}_{\underline{\underline{\mathbf{r}}}_s^{\mathbf{unsp}}}=\mathbf{I}-\underline{\underline{\boldsymbol{\Sigma}}}^{\mathbf{unsp}}\underline{\underline{\boldsymbol{\Sigma}}}^{\mathbf{unsp},T},\ \boldsymbol{\Sigma}_{\underline{\underline{\mathbf{r}}}_s^{\mathbf{sp}}}=\mathbf{I}-\underline{\underline{\boldsymbol{\Sigma}}}^{\mathbf{sp}}\underline{\underline{\boldsymbol{\Sigma}}}^{\mathbf{sp},T}\tag{61}$$



(a) unsupervised learning (b) supervised learning

Fig. 4: More general versions of suspected and residual spaces.

Furthermore, by substituting (59a) into (60a), one can obtain

$$\begin{aligned}&\underline{\underline{\mathbf{G}}}_{\mathbf{y}_s}^{\mathbf{unsp},-1}\underline{\underline{\mathbf{r}}}_s^{\mathbf{unsp}}\\ =&\mathbf{y}_s-\underline{\underline{\mathbf{G}}}_{\mathbf{y}_s}^{\mathbf{unsp},-1}\underline{\underline{\boldsymbol{\Sigma}}}^{\mathbf{unsp}}\underline{\underline{\mathbf{G}}}_{\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}}^{\mathbf{unsp}}\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}\\ =&\mathbf{y}_s-\boldsymbol{\Sigma}_{\mathbf{y}_s}^{1/2}(\underline{\underline{\boldsymbol{\Gamma}}}^{\mathbf{unsp}}\underline{\underline{\boldsymbol{\Sigma}}}^{\mathbf{unsp}}\underline{\underline{\boldsymbol{\Upsilon}}}^{\mathbf{unsp},T})\boldsymbol{\Sigma}_{\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}}^{-1/2}\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}\\ =&\mathbf{y}_s-\boldsymbol{\Sigma}_{\mathbf{y}_s,\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}}\boldsymbol{\Sigma}_{\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}}^{-1}\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}\end{aligned}\tag{62}$$

because $\underline{\underline{\mathbf{G}}}_{\mathbf{y}_s}^{\mathbf{unsp}}$ has full rank. Define

$$\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}=\underline{\underline{\boldsymbol{\mathcal{U}}}}_{\mathbf{y}}\begin{bmatrix}\mathbf{z}_{\mathbf{p}}\\ \mathbf{u}_s\\ \mathbf{y}_s\end{bmatrix}\tag{63}$$

where its optimal choices, $\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp},*}$ and $\underline{\underline{\boldsymbol{\mathcal{U}}}}_{\mathbf{y}}^*$, are obtained according to (56). It must be pointed out that $\underline{\underline{\boldsymbol{\mathcal{U}}}}_{\mathbf{y}}^*$ and its corresponding output $\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp},*}$ are not unique, because the objective functions are defined by $\mathcal{M}_{\mathbf{euc}}$. Then (62) can be further written as

$$\begin{aligned}&\underline{\underline{\mathbf{G}}}_{\mathbf{y}_s}^{\mathbf{unsp},-1}\underline{\underline{\mathbf{r}}}_s^{\mathbf{unsp}}\\ =&\mathbf{y}_s-\underbrace{\boldsymbol{\Sigma}_{\mathbf{y}_s,\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}}\boldsymbol{\Sigma}_{\underline{\underline{\mathbf{y}}}_s^{\mathbf{unsp}}}^{-1}\underline{\underline{\boldsymbol{\mathcal{U}}}}_{\mathbf{y}}^*}_{\mathcal{U}_{\mathbf{y}}^*}\begin{bmatrix}\mathbf{z}_{\mathbf{p}}\\ \mathbf{u}_s\\ \mathbf{y}_s\end{bmatrix}\\ =&\mathbf{e}_s=\mathbf{r}_s^{\mathbf{unsp}}\\ \Longleftrightarrow&\ \underline{\underline{\mathbf{r}}}_s^{\mathbf{unsp}}=\underline{\underline{\mathbf{G}}}_{\mathbf{y}_s}^{\mathbf{unsp}}\mathbf{r}_s^{\mathbf{unsp}}.\end{aligned}\tag{64}$$

Similarly, the following relationship holds for supervised learning-based IFD approaches:

$$\underline{\underline{\mathbf{r}}}_s^{\mathbf{sp}}=\underline{\underline{\mathbf{G}}}_{\mathbf{y}_s}^{\mathbf{sp}}\mathbf{r}_s^{\mathbf{sp}}.\tag{65}$$

Eventually, the following two relationships hold:

$$\begin{aligned} &T^2(\underline{\underline{\mathbf{r}}}_s^{\mathbf{unsp}}(k)) = \underline{\underline{\mathbf{r}}}_s^{\mathbf{unsp},T}(k)\underline{\underline{\mathbf{\Sigma}}}_{\underline{\underline{\mathbf{r}}}_s^{\mathbf{unsp}}}^{-1}\underline{\underline{\mathbf{r}}}_s^{\mathbf{unsp}}(k) \\ =&(\underline{\underline{\mathbf{G}}}_{\mathbf{y}_s}^{\mathbf{unsp}}\mathbf{r}_s^{\mathbf{unsp}})^T\underline{\underline{\mathbf{\Sigma}}}_{\underline{\underline{\mathbf{r}}}_s^{\mathbf{unsp}}}^{-1}\underline{\underline{\mathbf{G}}}_{\mathbf{y}_s}^{\mathbf{unsp}}\mathbf{r}_s^{\mathbf{unsp}} = T^2(\mathbf{r}_s^{\mathbf{unsp}}(k)) \end{aligned} \tag{66}$$

and

$$T^2(\underline{\underline{\mathbf{r}}}_s^{\mathbf{sp}}(k)) = T^2(\mathbf{r}_s^{\mathbf{sp}}(k)). \tag{67}$$

Combining (67) with (48) obtains

$$\begin{aligned} T^2(\underline{\underline{\mathbf{r}}}_s^{\mathbf{sp}}(k)) = T^2(\underline{\underline{\mathbf{r}}}_s^{\mathbf{unsp}}(k)) = \\ T^2(\mathbf{r}_s^{\mathbf{sp}}(k)) = T^2(\mathbf{r}_s^{\mathbf{unsp}}(k)), \end{aligned} \tag{68}$$

for normal operations, and

$$\begin{aligned} T^2(\underline{\underline{\mathbf{r}}}_s^{\mathbf{sp},\mathbf{f}}(k)) = T^2(\underline{\underline{\mathbf{r}}}_s^{\mathbf{unsp},\mathbf{f}}(k)) = \\ T^2(\mathbf{r}_s^{\mathbf{sp},\mathbf{f}}(k)) = T^2(\mathbf{r}_s^{\mathbf{unsp},\mathbf{f}}(k)), \end{aligned} \tag{69}$$

for faulty operations.

Following Theorems 2 and 3, T^2 test statistic defined on the two generalized residual generators also have the optimal fault-detection power. The readers can refer to [23] for the more rigorous analysis of the generalized version.

Remark 6. *We can find that the generalized versions of residual generators also deliver the least-square estimation of* $\mathbf{y}_s$*, because both* $\underline{\underline{\mathbf{G}}}_{\mathbf{y}_s}^{\mathbf{unsp}}$ *and* $\underline{\underline{\mathbf{G}}}_{\mathbf{y}_s}^{\mathbf{sp}}$ *play a role as normalization by limiting their covariance matrices to (61).* ∇

In order to have an insightful observation, Fig. 4 depicts multiple solutions to the suspected spaces, where Figs. 4(a) and (b) respectively correspond to unsupervised and supervised learning strategies.

IV. IFD Designs and Implementations: From Unsupervised to Supervised Neural Networks

A study of the existing approaches reveals that both the unsupervised and supervised machine learning techniques can be employed to enhance the IFD flexibility against system nonlinearity. For example, neural networks have received an increasing attention for the designs of IFD. Directly motivated by Theorems 1 to 3, the fully connected neural networks are used in this section for the fault diagnosis purpose.

A. Unit-time Delay Operators

A recurrent neural network is capable of describing the dynamic behaviors of nonlinear systems, thanks to the unit-time delays attached in neurons [30]. As the system order increases, more delay units are necessary to fit higher order dynamics, resulting in heavy computations together with problems associated with vanishing and exploding gradients [31]. To avoid such a situation and improve computation efficiency, the unit-time delay operator, denoted as z^{-1}, is used in the data pre-processing stage.

As shown in Fig. 5(a), (s_p+s+1) unit-time delay operators are defined on both $\mathbf{u}(k+s)$ and $\mathbf{y}(k+s)$ to obtain the inputs of an unsupervised neural network, i.e.,

$$\begin{aligned} &\mathbf{u}(k+s-1) = z^{-1}\mathbf{u}(k+s), \quad \cdots, \\ &\mathbf{u}(k-s_p-1) = z^{-(s_p+s+1)}\mathbf{u}(k+s), \\ &\mathbf{y}(k+s-1) = z^{-1}\mathbf{y}(k+s), \quad \cdots, \\ &\mathbf{y}(k-s_p-1) = z^{-(s_p+s+1)}\mathbf{y}(k+s). \end{aligned} \tag{70}$$

Similarly, (s_p+s+1) and (s_p+1) unit-time delay operators are defined on $\mathbf{u}(k+s)$ and $\mathbf{y}(k)$, respectively. The inputs of a supervised neural network given in Fig. 5(b) are $[\mathbf{u}_\mathbf{p}^T \ \mathbf{u}_s^T]$ and

$$\begin{aligned} &\mathbf{y}(k-1) = z^{-1}\mathbf{y}(k), \quad \cdots, \\ &\mathbf{y}(k-s_p-1) = z^{-(s_p+1)}\mathbf{y}(k). \end{aligned} \tag{71}$$

In addition, the unit-time delays are also used to obtain the reference outputs of both the unsupervised and supervised neural networks.

B. Unsupervised IFD Approaches

Similar to (32a), the loss function L of the unsupervised neural network can be defined by

$$L^{\mathbf{unsp}} = \left\| \begin{bmatrix} \mathbf{z}_\mathbf{p} \\ \mathbf{u}_s \\ \mathbf{y}_s \end{bmatrix} - \underbrace{\boldsymbol{\mathcal{H}}^{\mathbf{unsp}}(\boldsymbol{\Theta}_\mathbf{z} \ \boldsymbol{\Theta}_\mathbf{u} \ \boldsymbol{\Theta}_\mathbf{y})}_{\boldsymbol{\mathcal{H}}^{\mathbf{unsp}}(\boldsymbol{\Theta})} \begin{bmatrix} \mathbf{z}_\mathbf{p} \\ \mathbf{u}_s \\ \mathbf{y}_s \end{bmatrix} \right\|_2^2, \tag{72}$$

where $\boldsymbol{\mathcal{H}}^{\mathbf{unsp}}(\boldsymbol{\Theta})$ is the architecture of the neural network with the hyper parameter $\boldsymbol{\Theta}$, and the subscripts of $\boldsymbol{\Theta}$ signify their corresponding outputs. Therefore, the optimal $\boldsymbol{\Theta}$ can be obtained by minimizing $L^{\mathbf{unsp}}$, i.e.,

$$\boldsymbol{\Theta}^* = (\boldsymbol{\Theta}_\mathbf{z}^* \ \ \boldsymbol{\Theta}_\mathbf{u}^* \ \ \boldsymbol{\Theta}_\mathbf{y}^*) = \arg\min_{\boldsymbol{\Theta}} \ L^{\mathbf{unsp}}, \tag{73}$$

based on which the residual signal using $\boldsymbol{\mathcal{H}}^{\mathbf{unsp}}(\boldsymbol{\Theta}^*)$ is

$$\mathbf{r}_s^{\mathbf{unsp}}(k) = \mathbf{y}_s - \boldsymbol{\mathcal{H}}^{\mathbf{unsp}}(\boldsymbol{\Theta}_\mathbf{y}^*) \circ \begin{bmatrix} \mathbf{z}_\mathbf{p} \\ \mathbf{u}_s \\ \mathbf{y}_s \end{bmatrix} = \text{(27a)}. \tag{74}$$

In Fig. 5 (a), $\mathbf{r}_s^{\mathbf{unsp}}(k)$ is highlighted in red and "$\times$" refers to the reconstruction errors that may be neglected.

Combining with Fig. 5 (a), the implementation procedures of an unsupervised neural network-based IFD approach are summarized in Algorithms 1 and 2.

C. Supervised IFD Approaches

Considering a supervised neural network $\boldsymbol{\mathcal{H}}^{\mathbf{sp}}(\boldsymbol{\Theta})$, its loss function is formulated as follows

$$L^{\mathbf{sp}} = \left\| \mathbf{y}_s - \boldsymbol{\mathcal{H}}^{\mathbf{sp}}(\boldsymbol{\Theta}) \begin{bmatrix} \mathbf{z}_\mathbf{p} \\ \mathbf{u}_s \end{bmatrix} \right\|_2^2, \tag{76}$$

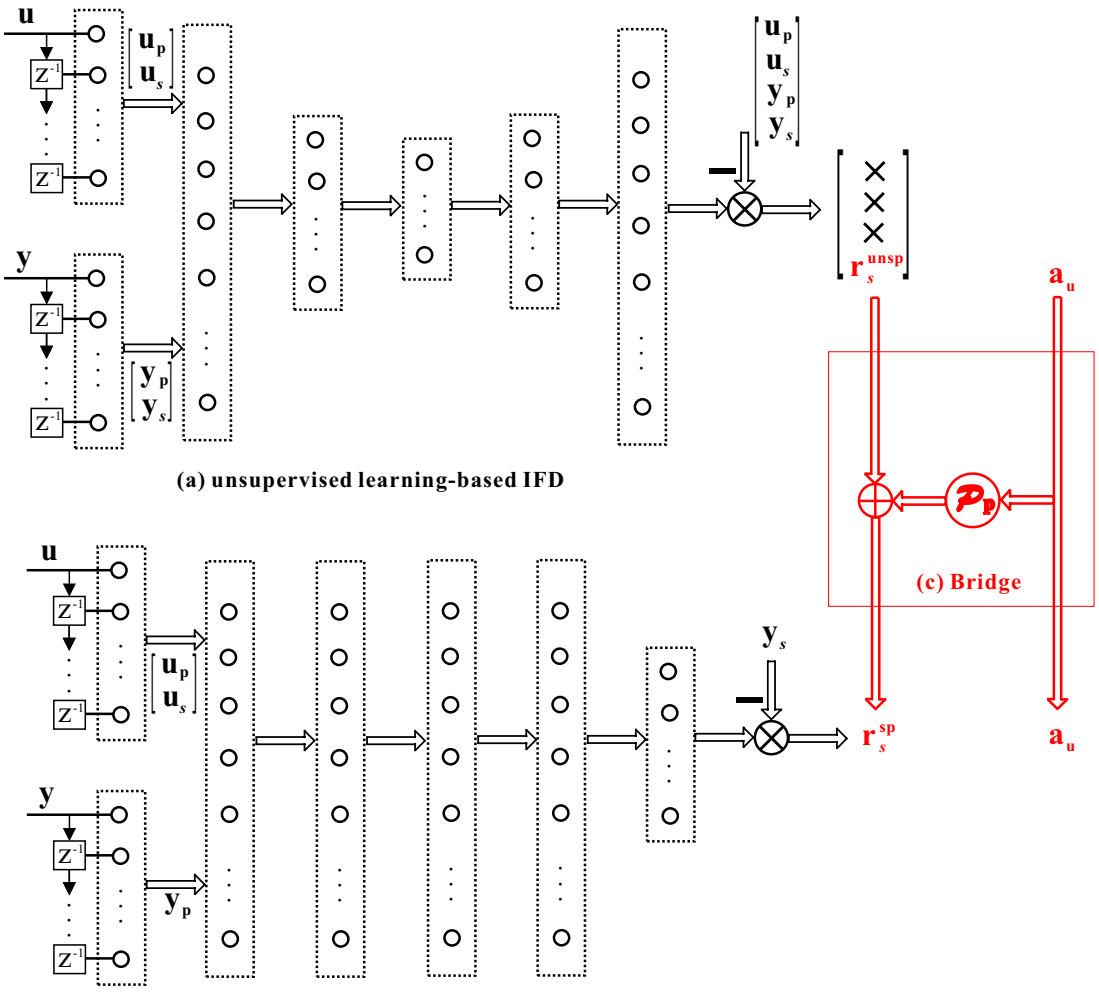


Fig. 5: Schematic of IFD frameworks with multiple unit-time delays, where "$\mathbf{a_u}$" represents an auxiliary variable, (a) is designed based an unsupervised neural network, (b) is implemented based a supervised neural network, and (c) highlighted in red is the bridge using an invertible neural network.

Algorithm 1 Off-line Learning: Unsupervised neural network-based IFD approaches

1: Collect system measurements $\mathbf{u}$ and $\mathbf{y}$;
2: Pre-process $\mathbf{u}$ and $\mathbf{y}$ by using the delay units according to (70) to obtain $\mathbf{z_p}$, $\mathbf{u}_s$ and $\mathbf{y}_s$;
3: Construct an unsupervised neural network whose architecture is $\mathcal{H}^{\mathbf{unsp}}(\Theta)$ and loss function is defined in (72);
4: Update Θ according to (73) and obtain $\Theta^*_{\mathbf{y}}$;
5: Generate the residual signal $\mathbf{r}_s^{\mathbf{unsp}}$ via (74);
6: Determine the threshold J_{th} for T^2 based on (25);
7: (If necessary) identify the fault features based on system knowledge or using faulty data.

when $\mathcal{M}^2_{\mathbf{euc}}(\mathbf{y}_s, \underline{\mathbf{y}}_s^{\mathbf{sp}})$ is adopted. Then the optimal hyper parameter can be obtained according to

$$\mathbf{\Theta}^* = \arg\min_{\mathbf{\Theta}} \; L^{\mathbf{sp}}, \tag{77}$$

Algorithm 2 Online Fault Diagnosis: Unsupervised neural network-based IFD approaches

1: Read real-time data and employ unit-time delay operators;
2: Calculate the online residual signal according to (74);
3: Compute the test statistic via (24);
4: Make an FD decision according to

$$\begin{aligned} T^2(\mathbf{r}_s^{\mathbf{unsp}}(\text{online})) - J_{th} < 0 \implies & \text{ Fault-free;} \\ \text{Otherwise} \implies & \text{ Faulty;} \end{aligned} \tag{75}$$

5: (If necessary) diagnose the fault by using $\mathbf{r}_s^{\mathbf{unsp}}(\text{online})$.

which allows for constructing the residual signal as follows.

$$\mathbf{r}_s^{\mathbf{sp}}(k) = \mathbf{y}_s - \boldsymbol{\mathcal{H}}^{\mathbf{sp}}(\mathbf{\Theta}^*) \begin{bmatrix} \mathbf{z_p} \\ \mathbf{u}_s \end{bmatrix} = (27\text{b}). \tag{78}$$

According to the analysis above and the schematic presented in Fig. 5(b), the complete designs and implementation proce-

dures are given in Algorithms 3 and 4.

Algorithm 3 Off-line Learning: Supervised neural network-based IFD approaches

1: Collect system measurements $\mathbf{u}$ and $\mathbf{y}$;
2: Pre-process $\mathbf{u}$ and $\mathbf{y}$ by using the delay units according to (70) and (71) to obtain $\mathbf{z_p}$ and $\mathbf{u}_s$;
3: Construct a supervised neural network whose architecture is $\mathcal{H}^{\mathbf{sp}}(\Theta)$ and loss function is defined in (76);
4: Update Θ via (77) and obtain Θ^*;
5: Generate the residual signal $\mathbf{r}_s^{\mathbf{sp}}$ via (78);
6: Determine the threshold J_{th} for T^2 based on (25);
7: (If necessary) identify the fault features based on system knowledge or using faulty data.

Algorithm 4 Online Fault Diagnosis: Supervised neural network-based IFD approaches

1: Read real-time data and employ unit-time delay operators;
2: Calculate the online residual signal according to (78);
3: Compute the test statistic via (24);
4: Make an FD decision according to

$$\begin{aligned} T^2(\mathbf{r}_s^{\mathbf{sp}}(\text{online})) - J_{th} < 0 \implies & \text{ Fault-free;} \\ \text{Otherwise} \implies & \text{ Faulty;} \end{aligned} \tag{79}$$

5: (If necessary) diagnose the fault by using $\mathbf{r}_s^{\mathbf{sp}}(\text{online})$.

D. An Invertible Neural Network-aided Bridge

In order to build the bridge between unsupervised and supervised neural networks-based IFD approaches, we introduce an auxiliary variable $\mathbf{a_u}$. The mapping generated by $\boldsymbol{\mathcal{H}}(\boldsymbol{\Theta}_{\mathbf{ext}})$ in the red square of Fig. 5(c) is

$$\begin{bmatrix} \mathbf{r}_s^{\mathbf{sp}} \\ \mathbf{a_u} \end{bmatrix} = \underbrace{\begin{pmatrix} \mathbf{I} & \boldsymbol{\mathcal{H}}(\boldsymbol{\Theta}_{\mathbf{p}}) \\ \mathbf{0} & \mathbf{I} \end{pmatrix}}_{\boldsymbol{\mathcal{H}}(\boldsymbol{\Theta}_{\mathbf{ext}})} \begin{bmatrix} \mathbf{r}_s^{\mathbf{unsp}} \\ \mathbf{a_u} \end{bmatrix} \tag{80}$$

where the loss function of $\boldsymbol{\mathcal{H}}(\boldsymbol{\Theta}_{\mathbf{ext}})$ is chosen as

$$L^{\mathbf{ext}} = \left\| \begin{bmatrix} \mathbf{r}_s^{\mathbf{sp}} \\ \mathbf{a_u} \end{bmatrix} - \boldsymbol{\mathcal{H}}(\boldsymbol{\Theta}_{\mathbf{ext}}) \begin{bmatrix} \mathbf{r}_s^{\mathbf{unsp}} \\ \mathbf{a_u} \end{bmatrix} \right\|_2^2. \tag{81}$$

Without loss of generality, we can choose $\mathbf{a_u} = \mathbf{r}_s^{\mathbf{unsp}}$ to prove the existence of the bridge. Then minimizing $L^{\mathbf{ext}}$ obtains

$$\boldsymbol{\Theta}_{\mathbf{ext}}^* = \underset{\boldsymbol{\Theta}_{\mathbf{ext}}}{\arg\min}\ L^{\mathbf{ext}} \iff \boldsymbol{\mathcal{H}}(\boldsymbol{\Theta}_{\mathbf{p}}^*) = \boldsymbol{\mathcal{P}}_{\mathbf{p}} \tag{82}$$

as shown in Fig. 5(c). It indicates that

$$\mathbf{r}_s^{\mathbf{sp}} = (\boldsymbol{\mathcal{P}}_{\mathbf{p}} + \mathbf{I})\mathbf{r}_s^{\mathbf{unsp}}, \mathbf{r}_s^{\mathbf{unsp}} = \mathbf{r}_s^{\mathbf{unsp}}, \tag{83}$$

and

$$\begin{aligned} \mathbf{r}_s^{\mathbf{unsp}} &= \mathbf{r}_s^{\mathbf{sp}} - \boldsymbol{\mathcal{P}}_{\mathbf{p}}\mathbf{r}_s^{\mathbf{unsp}} = \mathbf{r}_s^{\mathbf{sp}} - \boldsymbol{\mathcal{H}}(\boldsymbol{\Theta}_{\mathbf{p}}^*)\mathbf{r}_s^{\mathbf{unsp}} \\ &= \mathbf{r}_s^{\mathbf{sp}} - \boldsymbol{\mathcal{H}}(\boldsymbol{\Theta}_{\mathbf{p}}^*)\mathbf{a_u}, \\ \mathbf{a_u} &= \mathbf{a_u}. \end{aligned} \tag{84}$$

It is interesting to see that, even though the inverse operation is not used, we can obtain the bridge by

$$\begin{bmatrix} \mathbf{r}_s^{\mathbf{sp}} \\ \mathbf{a_u} \end{bmatrix} = \begin{pmatrix} \mathbf{I} & \boldsymbol{\mathcal{P}}_{\mathbf{p}} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{bmatrix} \mathbf{r}_s^{\mathbf{unsp}} \\ \mathbf{a_u} \end{bmatrix}, \tag{85}$$

and

$$\begin{bmatrix} \mathbf{r}_s^{\mathbf{unsp}} \\ \mathbf{a_u} \end{bmatrix} = \begin{pmatrix} \mathbf{I} & -\boldsymbol{\mathcal{P}}_{\mathbf{p}} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{bmatrix} \mathbf{r}_s^{\mathbf{sp}} \\ \mathbf{a_u} \end{bmatrix}. \tag{86}$$

In addition, the Jacobian matrix $\mathbf{J}$ of $\boldsymbol{\mathcal{H}}(\boldsymbol{\Theta}_{\mathbf{ext}})$ is

$$\mathbf{J} = \begin{pmatrix} \mathbf{I} & \boldsymbol{\mathcal{P}}_{\mathbf{p}} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}, \tag{87}$$

whose determinant is always positive, i.e.,

$$|\mathbf{J}| = \left| \begin{pmatrix} \mathbf{I} & \boldsymbol{\mathcal{P}}_{\mathbf{p}} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \right| = 1. \tag{88}$$

It means that $\boldsymbol{\mathcal{H}}(\boldsymbol{\Theta}_{\mathbf{ext}})$ and its component $\boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}}$ are globally invertible.

Also, (85), (86), and (88) can be used as an auxiliary evidence to illustrate the validity of Theorem 1.

Remark 7. *The solutions to the bridge depicted in Fig. 5(c) are not unique. For example, a "reversible residual network" given in [32] can also be directly employed to obtain an one-to-one mapping between* $\mathbf{r}_s^{\mathbf{unsp}}$ *and* $\mathbf{r}_s^{\mathbf{sp}}$. ∇

V. Conclusions

Thanks to the advanced machine learning techniques, there appears to have room for further development of nonlinear IFD. System data can provide us with information to reveal physical principles, making more informative inferences and decisions feasible. As the interpretability of these advanced methods is fundamental, the interplay between physical principles (such as a mathematical system description whose parameters may be unknown) and data becomes more critical.

This perspective article has developed three theorems related to IFD and related parameter identification for nonlinear dynamic systems. In order to obtain a unified form, Theorem 1 builds a bridge between unsupervised and supervised learning-based residual generators. Theorems 2 and 3, respectively corresponding to unsupervised and supervised learning, develop the IFD structures and illustrate their optimal performance for fault detection. With the aid of three different kinds of neural networks, Section IV details the specific designs and implementations of Theorems 1 to 3. This work lays a foundation for the further development of explainable IFD.

The perspective article ends with our opinions, expected challenges, and future research opportunities as follows.

- Not all IFD approaches are satisfactory from the viewpoint of fault-diagnosis performance. Our work provides researchers with some instructive guidance on designing more effective IFD algorithms, including both unsupervised and supervised learning-based schemes.
- As shown in Fig. 5 and Theorem 2, not all results obtained through unsupervised learning are useful for fault diagnosis. The conclusion is also true and easier to explain for the case of linear dynamic systems. For example, the purpose of singular value decomposition [33] and QR [5] used in linear approaches is not for dimension reduction, although they can do so. Keep in mind that the ultimate goal of unsupervised learning adopted in IFD approaches is always to minimize the reconstruction error of system outputs for both the linear and nonlinear systems.
- The initial step to apply the results obtained through this work must be to show the existence of the bridge $\mathcal{P}_{\mathbf{sp/unsp}}$ and $\mathcal{P}_{\mathbf{unsp/sp}}$ given in Theorem 1. Then a series of invertible machine learning tools, such as invertible neural networks in [34], can be employed to build this bridge. The process should not be reversed.
- An unmentioned condition in this study for nonlinear IFD approaches is that (6) is output reconstructible [35]. In fact, the condition is weak and easy to satisfy for our proposed IFD frameworks, because a multi-layer neural network can approximate a continuous function (such as $\phi(\cdot)$ and $\upsilon(\cdot)$ in Fig. 1) arbitrarily accurate [36].
- The data-based model (12) is obtained from (6) with the assumption of additivity given in (11a)-(11b). In practical applications, many classes of nonlinear systems satisfy (11a)-(11b) such as bounded systems, Hammerstein systems, and affine systems.
- It is a fact that nonlinear IFD approaches and the associated thresholds show dependence on $\mathbf{u}$ and $\mathbf{x}(0)$ [13]. In order to obtain a reasonable data set for training, a uniformly distributed $\mathbf{u}$ over different operation points is recommended so that a persistent excitation of the global system nonlinearities can be achieved.
- In addition to the bridge developed in this work, other aspects of system identification and IFD approaches deserve more investigations, such as (a) How many neurons should be used when utilizing a neural network? (b) How to determine the nonlinearity degree to approximate an unknown dynamic system without the problem of overfitting or underfitting? (c) What is the minimum size of training data for obtaining a desired performance? and so on.

APPENDIX A
THE PROOF OF THEOREM 1

Given an arbitrary unsupervised learning approach whose notation is $\mathcal{U}$, it generates the nonlinear mapping (28) with the objective function such as defined via $\mathcal{M}^2_{\mathbf{euc}}$:

$$\begin{aligned}&\min \left\| \begin{bmatrix} \mathbf{z_p} \\ \mathbf{u}_s \\ \mathbf{y}_s \end{bmatrix} - \begin{bmatrix} \hat{\mathbf{z}}_\mathbf{p} \\ \hat{\mathbf{u}}_s \\ \hat{\mathbf{y}}_s \end{bmatrix} \right\|_2^2 \\ \Longrightarrow &\min \ \|\mathbf{y}_s - \hat{\mathbf{y}}_s\|_2^2 = \min \ \|\mathbf{e}_s\|_2^2 . \end{aligned} \tag{A.1}$$

In (28), the input of $\mathcal{U}_\mathbf{z}$, $\mathcal{U}_\mathbf{u}$, and $\mathcal{U}_\mathbf{y}$ is $[\mathbf{z}_\mathbf{p}^T \ \mathbf{u}_s^T \ \mathbf{y}_s^T]^T$, and the subscripts correspond to the output variables.

Three steps are necessary to complete the proof.

Step 1): The generation of innovation error $\mathbf{e}_s$ using unsupervised learning. Based on (16a) and (16b), one can obtain

$$\begin{aligned} \mathbf{y}(k) =& \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}}\hat{\mathbf{x}}(k) + \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\mathbf{u}}\mathbf{u}(k) + \mathbf{e}(k) \\ \mathbf{y}(k+1) =& \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}}\hat{\mathbf{x}}(k+1) + \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\mathbf{u}}\mathbf{u}(k+1) + \mathbf{e}(k+1) \\ =& \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\mathbf{u}}\mathbf{u}(k+1) + \mathbf{e}(k+1) + \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}}\boldsymbol{\ell}\mathbf{y}(k) + \\ & \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}} \underbrace{(\boldsymbol{\phi}_{\hat{\mathbf{x}}\hat{\mathbf{x}}} - \boldsymbol{\ell}\boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}})}_{\boldsymbol{\mathcal{A}}_\ell} \hat{\mathbf{x}}(k) + \\ & \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}} \underbrace{(\boldsymbol{\phi}_{\hat{\mathbf{x}}\mathbf{u}} - \boldsymbol{\ell}\boldsymbol{\upsilon}_{\hat{\mathbf{y}}\mathbf{u}})}_{\boldsymbol{\mathcal{B}}_\ell} \mathbf{u}(k) \\ & \cdots \end{aligned} \tag{A.2}$$

Then, $\mathbf{e}_s(k)$ can be described by

$$\mathbf{e}_s(k) = \mathbf{y}_s(k) - \underline{\boldsymbol{\Upsilon}}_\mathbf{y}\mathbf{y}_s(k) - \underline{\boldsymbol{\Upsilon}}_\mathbf{x}\hat{\mathbf{x}}(k) - \underline{\boldsymbol{\Upsilon}}_\mathbf{u}\mathbf{u}_s(k) \tag{A.3}$$

where $\underline{\boldsymbol{\Upsilon}}_\mathbf{y}$, $\underline{\boldsymbol{\Upsilon}}_\mathbf{x}$, and $\underline{\boldsymbol{\Upsilon}}_\mathbf{u}$ are the nonlinear composite operators:

$$\underline{\boldsymbol{\Upsilon}}_\mathbf{y} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}}\boldsymbol{\ell} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}}\boldsymbol{\mathcal{A}}_\ell^{s-1}\boldsymbol{\ell} & \cdots & \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}}\boldsymbol{\ell} & \mathbf{0} \end{pmatrix}, \tag{A.4}$$

$$\underline{\boldsymbol{\Upsilon}}_\mathbf{u} = \begin{pmatrix} \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\mathbf{u}} & \mathbf{0} & \cdots & \mathbf{0} \\ \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}}\boldsymbol{\mathcal{B}}_\ell & \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\mathbf{u}} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}}\boldsymbol{\mathcal{A}}_\ell^{s-1}\boldsymbol{\mathcal{B}}_\ell & \cdots & \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}}\boldsymbol{\mathcal{B}}_\ell & \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\mathbf{u}} \end{pmatrix}, \tag{A.5}$$

$$\underline{\boldsymbol{\Upsilon}}_\mathbf{x} = \begin{pmatrix} \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}} \\ \vdots \\ \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}}\boldsymbol{\mathcal{A}}_\ell^{s} \end{pmatrix}. \tag{A.6}$$

Through some mathematical manipulations, (19) can be rewritten as

$$\mathbf{y}_s(k) = \boldsymbol{\Upsilon}_\mathbf{x}\hat{\mathbf{x}}(k) + \boldsymbol{\Upsilon}_\mathbf{u}\mathbf{u}_s(k) + \boldsymbol{\Upsilon}_\mathbf{e}\mathbf{e}_s(k) \tag{A.7}$$

which can be achieved by using supervised learning approaches, as shown in (19), i.e.,

$$\hat{\mathbf{y}}_s(k) = \underbrace{(\boldsymbol{\Upsilon}_\mathbf{x}\boldsymbol{\mathcal{L}}_p \ \ \boldsymbol{\Upsilon}_\mathbf{u})}_{\boldsymbol{\mathcal{S}}:=(\boldsymbol{\mathcal{S}}_\mathbf{z} \ \boldsymbol{\mathcal{S}}_\mathbf{u})} \begin{bmatrix} \mathbf{z}_\mathbf{p}(k) \\ \mathbf{u}_s(k) \end{bmatrix} \tag{A.8}$$

where $\boldsymbol{\mathcal{S}}$ is the nonlinear composite operator generated by supervised learning.

Step 2): A unified description of the transformation between $\mathbf{r}_s^{\mathbf{unsp}}$ and $\mathbf{r}_s^{\mathbf{sp}}$. Based on (A.3) and (A.8), two kinds of residual generators can be described by

$$\mathbf{r}_s^{\mathbf{unsp}} = \mathbf{y}_s - \boldsymbol{\mathcal{U}}_{\mathbf{y}} \begin{bmatrix} \mathbf{z_p} \\ \mathbf{u}_s \\ \mathbf{y}_s \end{bmatrix} = \mathbf{e}_s, \tag{A.9a}$$

$$\mathbf{r}_s^{\mathbf{sp}} = \mathbf{y}_s - (\boldsymbol{\mathcal{S}}_{\mathbf{z}} \;\; \boldsymbol{\mathcal{S}}_{\mathbf{u}}) \begin{bmatrix} \mathbf{z_p} \\ \mathbf{u}_s \end{bmatrix} = \boldsymbol{\Upsilon}_{\mathbf{e}} \mathbf{e}_s \tag{A.9b}$$

Step 3): The existence of $\boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}}^{-1}$. In fact, $\mathbf{r}_s^{\mathbf{unsp}}$ in (A.9a) and $\mathbf{r}_s^{\mathbf{sp}}$ in (A.9b) have shown

$$\mathbf{r}_s^{\mathbf{sp}} = \boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}} \mathbf{r}_s^{\mathbf{unsp}}, \; \boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}} := \boldsymbol{\Upsilon}_{\mathbf{e}}. \tag{A.10}$$

For the sake of simplicity, we rewrite (A.10) as

$$\mathbf{r}_s^{\mathbf{sp}} = \mathbf{r}_s^{\mathbf{unsp}} + \boldsymbol{\mathcal{P}}_{\mathbf{p}} \mathbf{r}_s^{\mathbf{unsp}} \tag{A.11}$$

where $\boldsymbol{\mathcal{P}}_{\mathbf{p}}$ is the component of $\boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}}$, i.e.,

$$\boldsymbol{\mathcal{P}}_{\mathbf{p}} = \begin{pmatrix} \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\upsilon}_{\hat{\mathbf{y}}\hat{\mathbf{x}}} \boldsymbol{\phi}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}^{s-1} \boldsymbol{\ell}_{\hat{\mathbf{x}}\mathbf{e}} & \cdots & \mathbf{0} \end{pmatrix}. \tag{A.12}$$

Consider a nonlinear operator $\boldsymbol{\mathcal{P}}_{\mathbf{unsp/sp}}$ and the Lipschitz constant of $\boldsymbol{\mathcal{P}}_{\mathbf{p}}$ is $\mathbf{Lip} < 1$. By defining a sequence $\{\mathbf{e}_{s,j}\} \in \mathbf{e}_s$, one can obtain the following innovation:

$$\begin{aligned} \boldsymbol{\mathcal{P}}_{\mathbf{unsp/sp}} :\; & \mathbf{e}_{s,j+1} = \boldsymbol{\Upsilon}_{\mathbf{e}} \mathbf{e}_s - \boldsymbol{\mathcal{P}}_{\mathbf{p}} \mathbf{e}_{s,j} \\ \Longrightarrow & \lim_{j \to \infty} \mathbf{e}_{s,j} = \boldsymbol{\mathcal{P}}_{\mathbf{unsp/sp}} \circ \boldsymbol{\Upsilon}_{\mathbf{e}} \mathbf{e}_s. \end{aligned} \tag{A.13}$$

Given a positive integer n, we have

$$\begin{aligned} &\|\mathbf{e}_{s,j+n} - \mathbf{e}_{s,j}\|_2 \leq \|\mathbf{e}_{s,j+n} - \mathbf{e}_{s,j+n-1}\|_2 + \cdots \\ &\quad \|\mathbf{e}_{s,j+1} - \mathbf{e}_{s,j}\|_2 \\ &= \|\boldsymbol{\mathcal{P}}_{\mathbf{p}}(\mathbf{e}_{s,j+n-1} - \mathbf{e}_{s,j+n-2})\|_2 + \cdots + \\ &\quad \|\boldsymbol{\mathcal{P}}_{\mathbf{p}}(\mathbf{e}_{s,j} - \mathbf{e}_{s,j-1})\|_2 \\ &\leq (\mathbf{Lip}^{j+n-1} + \cdots + \mathbf{Lip}^{j}) \|\mathbf{e}_{s,1} - \mathbf{e}_{s,0}\|_2 \\ &= \frac{1 - \mathbf{Lip}^{n-1}}{1 - \mathbf{Lip}} \mathbf{Lip}^{j} \|\mathbf{e}_{s,1} - \mathbf{e}_{s,0}\|_2 \\ &\leq \frac{\mathbf{Lip}^{j}}{1 - \mathbf{Lip}} \|\mathbf{e}_{s,1} - \mathbf{e}_{s,0}\|_2. \end{aligned} \tag{A.14}$$

It indicates that given an arbitrary small ε, there always exists a n such that

$$\|\mathbf{e}_{s,j+n} - \mathbf{e}_{s,j}\|_2 < \varepsilon \text{ or } \|\mathbf{e}_{s,j+n} - \mathbf{e}_{s,j}\|_2^2 < \varepsilon, \tag{A.15}$$

when $\mathbf{Lip} < 1$, which guarantees the existence of $\boldsymbol{\mathcal{P}}_{\mathbf{sp/unsp}}^{-1} = \boldsymbol{\mathcal{P}}_{\mathbf{unsp/sp}}$. It is worth mentioning that $\{\mathbf{e}_{s,j}\}$ is a Cauchy sequence [37] and (A.15) holds for $\mathbf{Lip} < 1$ because of the Banach fixed-point theorem [38]. Furthermore, $\mathbf{Lip} < 1$ is a weak condition/requirement to satisfy when choosing $\boldsymbol{\mathcal{P}}_{\mathbf{p}}$.

Hence, the proof of Theorem 1 is completed.

REFERENCES

[1] H. Chen, B. Jiang, S. X. Ding, and B. Huang, "Data-driven fault diagnosis for traction systems in high-speed trains: A survey, challenges, and perspectives," *IEEE Trans. Intell. Transp. Syst.*, 2021, doi.10.1109/TITS.2020.3029946.

[2] S. X. Ding, *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*, Berlin, Germany: Springer-Verlag, 2008.

[3] D. Zhou, Y. Zhao, Z. Wang, X. He, and M. Gao, "Review on diagnosis techniques for intermittent faults in dynamic systems," *IEEE Trans. Ind. Electron.*, vol. 67, no. 3, pp. 2337–2347, Mar. 2020.

[4] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Ann. Rev. Control*, vol. 36, no. 2, pp. 220–234, Dec. 2012.

[5] B. Huang and R. Kadali, *Dynamic Modelling, Predictive Control and Performance Monitoring: A Data-Driven Subspace Approach*, London, U.K.: Springer-Verlag, 2008.

[6] R. Isermann, "Model-based fault-detection and diagnosis–status and applications," *Ann. Rev. Control*, vol. 29, no. 1, pp. 71–85, Jan. 2005.

[7] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis: Part III: Process history based methods," *Comput. Chem. Eng.*, vol. 27, no. 3, pp. 327–346, Mar. 2003.

[8] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6418–6428, Nov. 2014.

[9] C. Dai, Z. Liu, K. Hu, and K. Huang, "Fault diagnosis approach of traction transformers in high-speed railway combining kernel principal component analysis with random forest," *IET Elect. Syst. Transp.*, vol. 6, no. 3, pp. 202–206, Sep. 2016.

[10] H. Chen, B. Jiang, N. Lu, and W. Chen, *Data-driven Detection and Diagnosis of Faults in Traction Systems of High-speed Trains*, Cham, Switzerland: Springer Nature, 2020.

[11] R. J. Patton and J. Chen, "Review of parity space approaches to fault diagnosis for aerospace systems," *J. Guid. Control Dyn.*, vol. 17, no. 2, pp. 278–285, Mar. 1994.

[12] Z. Zhao, W. Jiang, and W. Gao, "Health evaluation and fault diagnosis of medical imaging equipment based on neural network algorithm," *Comput. Intell. Neurosci.*, 2021, doi.org/10.1155/2021/6092461.

[13] S. X. Ding, *Advanced Methods for Fault Diagnosis and Fault-tolerant Control*, Berlin, Germany: Springer Nature, 2020.

[14] J. Chen and R. J. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*, 1998, Norwell, MA, USA: Kluwer.

[15] H. Wang, Z. Liu, A. Núñez, and R. Dollevoet, "Entropy-based local irregularity detection for high-speed railway catenaries with frequent inspections," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 10, pp. 3536–3547, Oct. 2019.

[16] S. X. Ding, *Data-driven Design of Fault Diagnosis and Fault-tolerant Control Systems*, London, U.K.: Springer-Verlag, 2014.

[17] T. de Bruin, K. Verbert, and R. Babuška, "Railway track circuit fault diagnosis using recurrent neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 523–533, Mar. 2017.

[18] J. Seshadrinath, B. Singh, and B. K. Panigrahi, "Incipient interturn fault diagnosis in induction machines using an analytic wavelet-based optimized Bayesian inference," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 990–1001, May. 2014.

[19] Z. Liu, L. Wang, C. Li, and Z. Han, "A high-precision loose strands diagnosis approach for isoelectric line in high-speed railway," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 1067–1077, Mar. 2018.

[20] T. Li, Z. Zhao, C. Sun, R. Yan, and X. Chen, "Multi-receptive field graph convolutional networks for machine fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 68, no. 12, pp. 12 739–12 749, Dec. 2021.

[21] H. Chen, Z. Chai, O. Dogru, B. Jiang, and B. Huang, "Data-driven designs of fault detection systems via neural networks-aided learning," *IEEE Trans. Neural Netw. Learn. Syst.*, 2021, doi.10.1109/TNNLS.2021.3071292.

[22] A. Bellet, A. Habrard, and M. Sebban, "Metric learning," *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 9, no. 1, pp. 1–151, Jan. 2015.

[23] H. Chen, Z. Chen, Z. Chai, B. Jiang, and B. Huang, "A single-side neural network-aided canonical correlation analysis with applications to fault diagnosis," *IEEE Trans. Cybern.*, 2021, doi.10.1109/TCYB.2021.3060766.

[24] Y. Yang, S. X. Ding, and L. Li, "Parameterization of nonlinear observer-based fault detection systems," *IEEE Trans. Autom. Control*, vol. 61, no. 11, pp. 3687–3692, Nov. 2016.

[25] H. Chen, L. Li, C. Shang, and B. Huang, "Fault detection for nonlinear dynamic systems with consideration of modeling errors: A data-driven approach," *IEEE Trans. Cybern.*, 2021, to be published.

[26] I. Mezić, "Koopman operator, geometry, and learning," *arXiv*, Oct. 2020, arXiv:2010.05377.

[27] A. L. Bruce, V. M. Zeidan, and D. S. Bernstein, "What is the Koopman operator? A simplified treatment for discrete-time systems," in *Amer. Contr. Conf.*, Jul. 2019, pp. 1912–1917.

[28] Y. Maki and K. A. Loparo, "A neural-network approach to fault detection and diagnosis in industrial processes," *IEEE Trans. Control Syst. Technol.*, vol. 5, no. 6, pp. 529–541, Nov. 1997.

[29] I. Rivals and L. Personnaz, "Black-box modeling with state-space neural networks," *Neural Adapt. Control Technol.*, vol. 15, pp. 237–264, Apr. 1996.

[30] S. Haykin, *Neural Networks and Learning Machines*, Hamilton, Canada: Pearson Education, 2010.

[31] H. Chen, Z. Chai, B. Jiang, and B. Huang, "Data-driven fault detection for dynamic systems with performance degradation: A unified transfer learning framework," *IEEE Trans. Instrum. Meas.*, 2021, doi.10.1109/TIM.2020.3033943.

[32] A. N. Gomez, M. Ren, R. Urtasun, and R. B. Grosse, "The reversible residual network: Backpropagation without storing activations," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Dec. 2017, pp. 2211–2221.

[33] J. Wang and S. J. Qin, "A new subspace identification approach based on principal component analysis," *J. Process Control*, vol. 12, no. 8, pp. 841–855, Dec. 2002.

[34] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother, and U. Köhe, "Analyzing inverse problems with invertible neural networks," *arXiv preprint*, Feb. 2019, 1808.04730v3.

[35] E. Sontag and Y. Wang, "Lyapunov characterizations of input to output stability," *SIAM. J. Control Optim.*, vol. 39, no. 1, pp. 226–249, Jan. 2000.

[36] K. I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Netw.*, vol. 2, no. 3, pp. 183–192, Mar. 1989.

[37] S. Lang, *Algebraic Number Theory*, New York, USA: Springer Science & Business Media, 2013.

[38] K. Ciesielski, "On Stefan Banach and some of his results," *Banach J. Math. Anal.*, vol. 1, no. 1, pp. 1–10, Apr. 2007.

Zhigang Liu (Senior Member, IEEE) received the Ph.D. degree in Power system and its Automation from Southwest Jiaotong University, China in 2003. He is currently a Full Professor of the School of Electrical Engineering, Southwest Jiaotong University, Chengdu. He is also a Guest Professor of Tongji University, Shanghai.

He has authored three books and published more than 200 peer-reviewed journal and conference articles. His research interests include the electrical relationship of EMUs and traction, detection, and assessment of pantograph-catenary in high-speed railway. Dr. Liu is an Associate Editor-in-Chief of IEEE Transactions on Instrumentation and Measurement, Associate Editor of IEEE Transactions on Neural Networks and Learning Systems, and IEEE Transactions on Vehicular Technology. He received the IEEE TIM's Outstanding Associate Editors for 2019 and 2020, and the Outstanding Reviewer of IEEE Transactions on Instrumentation and Measurement in 2018.

Hongtian Chen (Member, IEEE) received the B.S. and M.S. degrees in School of Electrical and Automation Engineering from Nanjing Normal University, China, in 2012 and 2015, respectively; and he received the Ph.D. degree in College of Automation Engineering from Nanjing University of Aeronautics and Astronautics, China, in 2019.

He had ever been a Visiting Scholar at the Institute for Automatic Control and Complex Systems, University of Duisburg-Essen, Germany, in 2018. Now he is a Post-Doctoral Fellow with the Department of Chemical and Materials Engineering, University of Alberta, Canada. His research interests include process monitoring and fault diagnosis, data mining and analytics, machine learning, and quantum computation; and their applications in high-speed trains, new energy systems, and industrial processes.

Dr. Chen was a recipient of the Grand Prize of Innovation Award of Ministry of Industry and Information Technology of the People's Republic of China in 2019, the Excellent Ph.D. Thesis Award of Jiangsu Province in 2020, and the Excellent Doctoral Dissertation Award from Chinese Association of Automation (CAA) in 2020.

Cesare Alippi (Fellow, IEEE) received the master's degree (cum laude) in electronic engineering and the Ph.D. degree from the Politecnico di Milano, Italy, in 1990 and 1995, respectively. He has been a Visiting Researcher with University College London, London, U.K.; the Massachusetts Institute of Technology, Cambridge, MA, USA; ESPCI, Paris, France; CASIA; and A*STAR, Singapore.

He is currently a Full Professor of information processing systems with the Politecnico di Milano and a Full Professor of cyber-physical and embedded systems with the Università della Svizzera italiana, Lugano, Switzerland. He holds five patents. He has published in 2014 a monograph on Intelligence for Embedded Systems (Springer) and (co)authored more than 200 papers in international journals and conference proceedings. His current research activity addresses adaptation and learning in nonstationary environments and intelligence for embedded systems.

Dr. Alippi is a member and the chair of other IEEE committees. He received the IEEE Instrumentation and Measurement Society Young Engineer Award in 2004, the IBM Faculty Award in 2013, the 2016 IEEE TNNLS Outstanding Paper Award, and the 2016 International Neural Networks Society (INNS) Gabor Award. Among the others, he was the General Chair of the International Joint Conference on Neural Networks (IJCNN) in 2012 and the Program Chair and Co-Chair in 2014 and 2011, respectively. He was the General Chair of the IEEE Symposium Series on Computational Intelligence in 2014. He was an Associate Editor of the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He is a Distinguished Lecturer of the IEEE Computational Intelligence Society (CIS), a member of the Board of Governors of INNS, the Vice-President of IEEE CIS, an Associate Editor of the IEEE Computational Intelligence Magazine.

Biao Huang (Fellow, IEEE) received the B.Sc. and M.Sc. degrees in automatic control from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 1983 and 1986, respectively, and the Ph.D. degree in process control from the University of Alberta, Edmonton, AB, Canada, in 1997.

In 1997, he joined the University of Alberta as an Assistant Professor with the Department of Chemical and Materials Engineering and is currently a Professor. He is the Natural Sciences and Engineering Research Council of Canada's Industrial Research Chair in Control of Oil Sands Processes and was also the Alberta Innovates Technology Futures Industry Chair in Process Control. He has applied his expertise extensively in industrial practice, particularly in the oil sands industry. His research interests include process control, system identification, control performance assessment, Bayesian methods, and state estimation.

Dr. Huang is a Fellow of the Canadian Academy of Engineering and the Chemical Institute of Canada. He was a recipient of Germany's Alexander von Humboldt Research Fellowship, the Canadian Chemical Engineer Society's Syncrude Canada Innovation and D. G. Fisher Awards, APEGA Summit Research Excellence Award, University of Alberta McCalla and Killam Professorship Awards, Petro-Canada Young Innovator Award, and a Best Paper Award from the Journal of Process Control.

Derong Liu (Fellow, IEEE) received the PhD degree in electrical engineering from the University of Notre Dame, USA, in 1994.

He became a Full Professor of Electrical and Computer Engineering and of Computer Science at the University of Illinois at Chicago in 2006. He was selected for the "100 Talents Program" by the Chinese Academy of Sciences in 2008, and he served as the Associate Director of The State Key Laboratory of Management and Control for Complex Systems at the Institute of Automation, from 2010 to 2016. He has published 13 books.

He received the International Neural Network Society's Gabor Award in 2018 and the IEEE CIS Neural Network Pioneer Award in 2022. He has been named a highly cited researcher consecutively for five years from 2017 to 2021 by Clarivate. He was the Editor-in-Chief of IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS from 2010 to 2015. He is the Editor-in-Chief of Artificial Intelligence Review (Springer). He is a Fellow of the IEEE, a Fellow of the International Neural Network Society, a Fellow of the International Association of Pattern Recognition, and a Member of Academia Europaea (The Academy of Europe).