Fault Detection for Heterogeneous Multi-Agent Systems with Unknown Dynamics Using Distributed Autoencoder

Zeyuan Wang[1], Mohammed Chadli[1], Linlin Li[2], Steven X. Ding[3], Ketian Liang[3]

***Abstract*— This paper presents a novel approach addressing fault detection challenges for multi-agent systems through a machine-learning method using recurrent autoencoders. The main advantage lies in its ability to handle heterogeneous multi-agent systems with unknown dynamics. The approach features a distributed detection architecture based on a cluster representation that depends solely on the agents' relative outputs, integrating stable image representation and orthogonal projection. Unlike traditional observer-based methods, the fault detection framework employs distributed autoencoders for residual generation, offering a data-driven and model-free solution. The autoencoders are carefully designed for effective time-series data learning, incorporating gated recurrent units and neural networks. Simulation results validate the effectiveness of the proposed method, demonstrating excellent fault detection capabilities and highlighting the promising extension to more complex and generic systems.**

I. Introduction

With the advancement of technology, multi-agent systems (MASs) are widely applied across various fields. Through information sharing and coordinated control, MASs can accomplish difficult or impossible tasks for conventional single-agent systems[1], [2]. This has made the reliability and safety of MASs increasingly important, prompting extensive research into fault detection and diagnosis for MASs.

Over the past decades, observer-based and model-based fault detection has developed significantly [3], [4], [5]. Such techniques have also been applied for MASs with different system dynamics [6], [7], [8]. The core of observer-based methods lies in pre-designed observer's structure and designing the gain matrix to minimize the impact of disturbances or uncertainties on the residual signal while maximizing the sensitivity of fault signals, typically using H_-/H_∞ methods and linear matrix inequalities (LMIs) [9], [10], [7]. However, these methods require prior knowledge of the system model, and the associated LMI approach exhibits significant conservatism, especially when dealing with nonlinearity. Data-driven methods and multivariate statistic analysis techniques have emerged as alternatives [11]. However, they still encounter significant challenges when addressing nonlinear or complex heterogeneous MASs.

In recent years, artificial intelligence and machine learning technologies have experienced intensive development. Autoencoders (AEs), as a crucial unsupervised learning method, can learn existing patterns from data and are widely applied in data encoding, compression, and dimension reduction [12]. By integrating neural networks (NNs), AEs can be utilized for anomaly detection and fault detection in dynamic systems [13], [14], [15], [16], [17]. Furthermore, when combined with control theory and appropriately designed loss functions, the encoded latent variables become highly interpretable. Pioneer studies include [18], [19], which combined the stable image representation (SIR), the control theory, and the recurrent NNs for encoding highly explainable latent variables as reference signals of feedback control. This approach does not require prior knowledge of the system model and has been validated and applied in linear systems and affine nonlinear systems [18], [19]. However, to the best of the authors' knowledge, no study has examined AE-based methods for fault detection in MASs, nor addressed issues such as system topology, relative output information, and heterogeneity.

Motivated by the above studies, this paper proposes the first AE-based approach to address fault detection issues in MASs. The main contributions are as follows: 1) A distributed fault detection framework is proposed for heterogeneous MASs with unknown dynamics. 2) Based on the cluster transformation, gated recurrent unit (GRU)-based AEs are designed in a distributed architecture, followed by a fault agent isolation module. The designed AEs are highly explainable and guided by control theory and orthogonal projection techniques under the paradigm of stable image representation. 3) The loss function is carefully designed, and experiments demonstrate the impact of different regularization terms on the validation loss. Excellent fault detection ability are validated by performance metrics.

The paper is organized as follows. Section II gives preliminaries and the problem statement, including the cluster transformation, the SIR, and the orthogonal projection technique for fault detection. The detailed design of distributed AEs is presented in Section III. Simulation results are shown in Section IV, followed by the conclusion in Section V.

Notations: Let $\mathbb{R}, \mathbb{N}^+$ denote the set of real numbers and the set of positive integer numbers, respectively. I_n denotes an identity matrix of dimension n. De-

This work was supported by the China Scholarship Council under grant 202206020096 during the internship at the University of Duisburg-Essen, and by the National Natural Science Foundation of China under grant 62322303.

[1]Zeyuan Wang and Mohammed Chadli are with the University of Paris-Saclay Evry, IBISC Laboratory (EA 4526), 91020 Evry, France {zeyuan.wang, mohammed.chadli}@universite-paris-saclay.fr

[2]Linlin Li is with Key Laboratory of Knowledge Automation for Industrial Processes of Ministry of Education, School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China linlin.li@ustb.edu.cn

[3]Steven X. Ding and Ketian Liang are with the Institute for Automatic Control and Complex Systems, University of Duisburg-Essen, 47057 Duisburg, Germany {steven.ding, ketian.liang}@uni-due.de

note $\text{diag}(a_1, \ldots, a_N)$ an $N \times N$ diagonal matrix whose entries are $a_1, \ldots, a_N$. Denote a column vector $x = \text{col}\{x_1, \ldots, x_N\} = \left(x_1^T, \ldots, x_N^T\right)^T$. The conjugate of a rational transfer function matrix of a discrete-time system $G(z)$ is denoted by $G^{\sim}(z) = G^T(z^*)$. The communication topology of a MAS composed of N ($N \in \mathbb{N}^+$) agents is represented by an undirected connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of a vertex set $\mathcal{V}$ and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. An agent j is called a neighbor of agent i if they are connected and can have relative measurements of each other. Denote $\mathcal{N}_i = \{i_1, i_2, \ldots, i_{N_i}\}$ the set of neighbors of agent i with N_i neighbors. Denote $\tilde{\mathcal{N}}_i = \{i\} \cup \mathcal{N}_i$.

II. Preliminaries and Problem Statement

A. Distributed fault detection based on relative output

In many cooperative systems, the relative output is essential for various purposes, such as accurate positioning and low-cost sensing. This paper considers a heterogeneous undirected and connected MAS in which each agent is equipped solely with relative output sensors, as described by the following discrete-time dynamics.

$$\mathcal{G}_i: \quad \begin{cases} x_i(t+1) = A_i x_i(t) + B_i u_i(t) + \eta_i(t) \\ z_{i_j}(t) = y_i(t) - y_j(t) + \epsilon_{i_j}(t), \quad j \in \mathcal{N}_i \end{cases} \tag{1}$$

where $y_i(t) = C_i x_i(t) + D_i u_i(t)$, $x_i \in \mathbb{R}^{n_{x_i}}, u_i \in \mathbb{R}^{n_{u_i}}$, $y_i \in \mathbb{R}^{n_y}$, $z_{i_j} \in \mathbb{R}^{n_y}$. η_i and ϵ_{i_j} denote zero-mean and Gaussian noises. z_{i_j} is the relative output w.r.t. the neighbor agent $j, j \in \mathcal{N}_i$. A_i, B_i, C_i, D_i are unknown matrices of appropriate dimensions.

To effectively use the relative output information, we first introduce the cluster transformation. A cluster $\mathcal{C}_i$ centered in agent i is defined as a set consisting of agent i and its neighbors: $\mathcal{C}_i = \left\{\mathcal{G}_j | j \in \tilde{\mathcal{N}}_i\right\}$. The cluster vectors are defined as follows.

$$\begin{gathered} x_{c,i} = \text{col}\left\{x_i, x_{i_1}, \ldots, x_{i_{N_i}}\right\} \\ u_{c,i} = \text{col}\left\{u_i, u_{i_1}, \ldots, u_{i_{N_i}}\right\} \\ \eta_{c,i} = \text{col}\left\{\eta_i, \eta_{i_1}, \ldots, \eta_{i_{N_i}}\right\} \\ y_{c,i} = \text{col}\left\{z_{i_1}, \ldots, z_{i_{N_i}}\right\}, \epsilon_{c,i} = \text{col}\left\{\epsilon_{i_1}, \ldots, \epsilon_{i_{N_i}}\right\} \end{gathered} \tag{2}$$

and the dynamic of the cluster i is described as

$$\mathcal{C}_i: \begin{cases} x_{c,i}(t+1) = A_{c,i} x_{c,i}(t) + B_{c,i} u_{c,i}(t) + \eta_{c,i}(t) \\ y_{c,i}(t) = C_{c,i} x_{c,i}(t) + D_{c,i} u_{c,i}(t) + \epsilon_{c,i}(t) \end{cases} \tag{3}$$

where $A_{c,i} = \text{diag}\{A_i, A_{i_1}, \ldots, A_{i_{N_i}}\}$, $B_{c,i} = \text{diag}\{B_i, B_{i_1}, \ldots, B_{i_{N_i}}\}$, $C_{c,i} = W_i \times \text{diag}\{C_i, C_{i_1}, \ldots, C_{i_{N_i}}\}$, $D_{c,i} = W_i \times \text{diag}\{D_i, D_{i_1}, \ldots, D_{i_{N_i}}\}$, where W_i is an $N_i n_y$-by-$(N_i+1)n_y$ matrix defined as follows.

$$W_i = \begin{pmatrix} I_{n_y} & -I_{n_y} & 0_{n_y} & \cdots & 0_{n_y} \\ I_{n_y} & 0_{n_y} & -I_{n_y} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0_{n_y} \\ I_{n_y} & 0_{n_y} & \cdots & 0_{n_y} & -I_{n_y} \end{pmatrix} \tag{4}$$

The following assumption holds for the purpose of fault agent isolation:

Assumption 1: 1) $\forall (i,j) \in \{1, \ldots, N\}^2$, $\mathcal{C}_i \neq \mathcal{C}_j$

2) At any given moment, there is at most one faulty agent.

Remark 1: The reader can refer to Theorem 1 and Remark 4 that $\mathcal{C}_i \neq \mathcal{C}_j$ is a necessary condition for fault agent isolation. The second assumption aims at simplification without loss of generality. For multiple faulty agents, the reader can refer to [20] by means of the topology separation.

Based upon the above cluster transformation, the MAS can be regarded as a combination of N clusters: $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_N$. During the nominal process, i.e., noise-free and fault-free process, the input and output data $\text{col}\{u_{c,i}, y_{c,i}\}$ belongs to the image space $\mathcal{I}_{\mathcal{C}_i}$, which is defined by the stable image representation (SIR) [21] via the latent variable $v_{c,i}$:

$$\mathcal{I}_{\mathcal{C}_i} = \left\{ \begin{pmatrix} u_{c,i} \\ y_{c,i} \end{pmatrix} \middle| \begin{pmatrix} u_{c,i} \\ y_{c,i} \end{pmatrix} = \begin{pmatrix} M_{c,i} \\ N_{c,i} \end{pmatrix} v_{c,i}, v_{c,i} \in \mathcal{L}_2 \right\} \tag{5}$$

where $(M_{c,i}, N_{c,i})$ is the right coprime factorization (RCF) of $\mathcal{C}_i$ (3). The SIR is equivalent to a closed-loop system with a state-feedback controller:

$$\begin{cases} x_{c,i}(t+1) = (A_{c,i} + B_{c,i}K_{c,i})x_{c,i}(t) + B_{c,i}V_{c,i}v_{c,i}(t) \\ u_{c,i}(t) = K_{c,i}x_{c,i}(t) + V_{c,i}v_{c,i}(t) \\ y_{c,i}(t) = (C_{c,i} + D_{c,i}K_{c,i})x_{c,i}(t) + D_{c,i}V_{c,i}v_{c,i}(t) \end{cases} \tag{6}$$

where the feedback controller $K_{c,i} = \text{diag}\{K_i, K_{i_1}, \ldots, K_{i_{N_i}}\}$, the feedforward controller $V_{c,i} = \text{diag}\{V_i, V_{i_1}, \ldots, V_{i_{N_i}}\}$, and the latent variable $v_{c,i} = \text{col}\{v_i, v_{i_1}, \ldots, v_{i_{N_i}}\}$. The latent variable $v_j \in \mathbb{R}^{n_{u_j}}$ is a reference signal and can be designed to achieve different control objectives, such as consensus and formation control.

Remark 2: In the SIR (5), the "real" input data is $v_{c,i}$, which drives the system's evolution represented by the process data $(u_{c,i}, y_{c,i})$. In this case, the latent variable $v_{c,i}$ can also be regarded as a compression of $(u_{c,i}, y_{c,i})$, and the image space $\mathcal{I}_{\mathcal{C}_i}$ is a subspace spanned by the nominal process vector $\text{col}\{u_{c,i}, y_{c,i}\}$.

Instead of constructing observer-based residual signals, the projection-based approach can build more generalized residual generators [21]. It has been shown that an orthogonal projection into the image space $\mathcal{I}_{\mathcal{C}_i}$ can be obtained by the following operator $\mathcal{P}_{\mathcal{C}_i}: \mathcal{L}_2 \to \mathcal{L}_2$ [22]:

$$\mathcal{P}_{\mathcal{C}_i} = I_{\mathcal{C}_i,0} I_{\mathcal{C}_i,0}^{\sim} \tag{7}$$

where $I_{\mathcal{C}_i,0}$ is the normalized SIR of $\mathcal{I}_{\mathcal{C}_i}$ defined by the normalized RCF $(\mathcal{M}_{c,i}, \mathcal{N}_{c,i})$ of $\mathcal{C}_i$ satisfying Bézout's identity:

$$I_{\mathcal{C}_i,0}^{\sim}(z) I_{\mathcal{C}_i,0}(z) = \mathcal{M}_{c,i}^{\sim}(z)\mathcal{M}_{c,i}(z) + \mathcal{N}_{c,i}^{\sim}(z)\mathcal{N}_{c,i}(z) = I \tag{8}$$

More detailed description of $I_{\mathcal{C}_i,0}$ and $(\mathcal{M}_{c,i}, \mathcal{N}_{c,i})$ can be found in [18], [22]. Note that the orthogonal projection is a special case of an idempotent linear operator satisfying the following relation:

$$\mathcal{P}_{\mathcal{C}_i}^2 = \mathcal{P}_{\mathcal{C}_i} \tag{9}$$

By using $\mathcal{P}_{\mathcal{C}_i}$, the orthogonal projection of $\text{col}\{u_{c,i}, y_{c,i}\}$ into the image space $\mathcal{I}_{\mathcal{C}_i}$ is denoted by $\text{col}\{\hat{u}_{c,i}, \hat{y}_{c,i}\}$:

$$\begin{pmatrix} \hat{u}_{c,i} \\ \hat{y}_{c,i} \end{pmatrix} = \mathcal{P}_{\mathcal{C}_i} \begin{pmatrix} u_{c,i} \\ y_{c,i} \end{pmatrix} = I_{\mathcal{C}_i,0} I_{\mathcal{C}_i,0}^{\sim} \begin{pmatrix} u_{c,i} \\ y_{c,i} \end{pmatrix} \tag{10}$$

Based on the normalized SIR, the latent variable can be reformulated as $v_{c,i} = I^{\sim}_{\mathcal{C}_i,0}\begin{pmatrix} u_{c,i} \\ y_{c,i} \end{pmatrix}$ and $\begin{pmatrix} \hat{u}_{c,i} \\ \hat{y}_{c,i} \end{pmatrix} = I_{\mathcal{C}_i,0} v_{c,i}$. It has been shown that $v_{c,i}$ is a lossless compression of the nominal data [18].

In addition, the projected vector $\begin{pmatrix} \hat{u}_{c,i} \\ \hat{y}_{c,i} \end{pmatrix}$ can be interpreted in the following two situations:

- During the nominal process,

$$\begin{pmatrix} \hat{u}_{c,i} \\ \hat{y}_{c,i} \end{pmatrix} \in \mathcal{I}_{\mathcal{C}_i} \text{ and } \begin{pmatrix} \hat{u}_{c,i} \\ \hat{y}_{c,i} \end{pmatrix} = \begin{pmatrix} u_{c,i} \\ y_{c,i} \end{pmatrix} \tag{11}$$

- Under the presence of noises or/and faults, there exists a non-zero complement vector:

$$\begin{pmatrix} u^{\perp}_{c,i} \\ y^{\perp}_{c,i} \end{pmatrix} \neq 0 \text{ such that } \begin{pmatrix} u^{\perp}_{c,i} \\ y^{\perp}_{c,i} \end{pmatrix} = \mathcal{P}^{\perp}_{\mathcal{C}_i} \begin{pmatrix} u_{c,i} \\ y_{c,i} \end{pmatrix} \tag{12}$$

where $\mathcal{P}^{\perp}_{\mathcal{C}_i} = \mathcal{I} - \mathcal{P}_{\mathcal{C}_i}$. Then the cluster residual generator based on the orthogonal projection $\mathcal{P}_{\mathcal{C}_i}$ can be represented by the signal $r_{c,i}$:

$$r_{c,i} = \begin{pmatrix} u_{c,i} \\ y_{c,i} \end{pmatrix} - \begin{pmatrix} \hat{u}_{c,i} \\ \hat{y}_{c,i} \end{pmatrix} = (\mathcal{I} - \mathcal{P}_{\mathcal{C}_i}) \begin{pmatrix} u_{c,i} \\ y_{c,i} \end{pmatrix} \tag{13}$$

Remark 3: The traditional observer-based residual $r_{c,i} = y_{c,i} - \hat{y}_{c,i}$ is a special case of (13) by assigning $u_{c,i} = \hat{u}_{c,i}$. Therefore, (13) is more generalized and can be used to achieve better fault detectability.

Note that $r_{c,i} = 0$ holds if and only if each agent in $\mathcal{C}_i$ is both noise-free and fault-free, and $r_{c,i}$ is close to zero if $\mathcal{C}_i$ has noises but is fault-free. Based on this observation, the fault detection threshold $J_{th,i}$ for the cluster i can be set as follows.

$$\begin{cases} J_{th,i} = \sup\limits_{\binom{u_{c,i}}{y_{c,i}} \in \mathcal{Z}_{c,i}} J(r^{(t)}_{c,i}), \text{ s.t. FAR} \leq \gamma_i \\ J(r^{(t)}_{c,i}) = \dfrac{1}{L_w} \displaystyle\sum_{l=t-L_w+1}^{t} \|r_{c,i}(l)\|^2 \end{cases} \tag{14}$$

where γ_i is the pre-defined upper-bound of false alarm rate (FAR). $\mathcal{Z}_{c,i}$ is the set of noisy data col$\{u_{c,i}, y_{c,i}\}$ collected during the fault-free process. The fault decision is made based on the following detection logic.

$$\text{The cluster } i \text{ is } \begin{cases} \text{fault-free}, & \text{if } J(r^{(t)}_{c,i}) \leq J_{th,i} \\ \text{faulty}, & \text{if } J(r^{(t)}_{c,i}) > J_{th,i} \end{cases} \tag{15}$$

B. Problem Statement

The primary challenge of this study lies in the unknown dynamics of agents (1) and clusters (3). Without access to the system matrices, it is impossible to construct the orthogonal projection or residual generator analytically. Furthermore, the heterogeneity adds additional complexity. Therefore, this study aims to develop model-free residual generators and a fault detection strategy that remains robust to noise while sensitive to faults.

III. Fault Detection using Distributed Autoencoders

Figure 1 illustrates the proposed fault detection framework, consisting of two main components: the cluster fault detection module and the fault agent isolation module. The cluster fault detection module leverages cluster process data in (3). Each cluster employs a multi-layer gated recurrent unit (GRU)-based AE with fully connected NNs (FCNNs), as shown in Fig. 2. Trained on fault-free data, the AE can learn existing dynamic patterns, approximating the orthogonal projection (7) without requiring system models. The generated residual signal $r_{c,i}$ enables cluster fault detection by residual evaluation. Finally, the fault agent isolation module distinguishes the specific faulty agent using topology information.

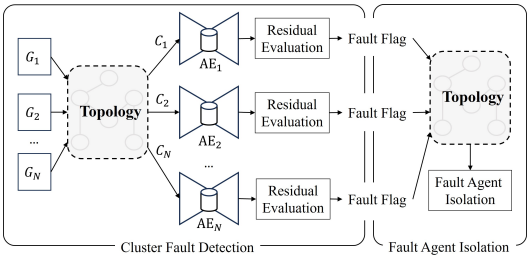

Fig. 1. Distributed fault detection and isolation framework

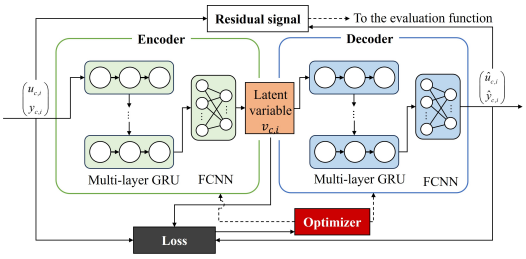

Fig. 2. The architecture of recurrent autoencoder

A. Cluster fault detection

The core of the cluster fault detection is the GRU-based AE shown in Fig. 2. The multi-layer GRU, alternatively called stacked-layer GRU, is shown in Fig. 3. Take the l-th layer for example: suppose the input sequence of length k is denoted by $\{X^l_{t-k+1}, \ldots, X^l_t\}$, and the output sequence is $\{X^{l+1}_{t-k+1}, \ldots, X^{l+1}_t\}$. $X^{l+1}_\tau, \tau \in \{t-k+1, \ldots, t\}$ is calculated by the following rule in (16) [23], [19].

$$\begin{aligned} r^l_\tau &= \sigma\left(W^l_{ir} X^l_\tau + b^l_{ir} + W^l_{hr} h^l_{(\tau'-1)} + b^l_{hr}\right) \\ z^l_\tau &= \sigma\left(W^l_{iz} X^l_\tau + b^l_{iz} + W^l_{hz} h^l_{(\tau'-1)} + b^l_{hz}\right) \\ n^l_\tau &= \tanh\left(W^l_{in} X^l_\tau + b^l_{in} + r^l_\tau \odot \left(W^l_{hn} h^l_{(\tau'-1)} + b^l_{hn}\right)\right) \\ h^l_{\tau'} &= \left(1 - z^l_\tau\right) \odot n^l_\tau + z_\tau \odot h^l_{(\tau'-1)} \\ X^{l+1}_\tau &= h^l_{\tau'},\ \tau' = \tau - t + k \end{aligned} \tag{16}$$

where σ is the sigmoid function, $\odot$ is the Hadamard product. The superscript l means the l-th layer. $r^l_\tau, z^l_\tau, n^l_\tau$ are the reset, update, and new gates,

respectively. $\{W_{ir}^l, W_{hr}^l, W_{iz}^l, W_{hz}^l, W_{in}^l, W_{hn}^l\}$ and $\{b_{ir}^l, b_{hr}^l, b_{iz}^l, b_{hz}^l, b_{in}^l, b_{hn}^l\}$ are weights and bias to be optimized. $h_{\tau'}^l, \tau' \in \{1, \ldots, k\}$ is the hidden state of the τ'-th cell in the l-th layer GRU. h_0^l is the initial hidden state, which is set as 0 by default. The input X_τ^l can be different embeddings depending on its position of GRU:

1) First layer in the encoder: $X_\tau^{(l=1)} = \text{col}\{u_{c,i}(\tau), y_{c,i}(\tau)\}$.
2) First layer in the decoder: $X_\tau^{(l=1)} = v_{c,i}(\tau)$.
3) Middle layer in the encoder/decoder: $l \geq 2$, $X_\tau^{(l)}$ is the output of the last layer.

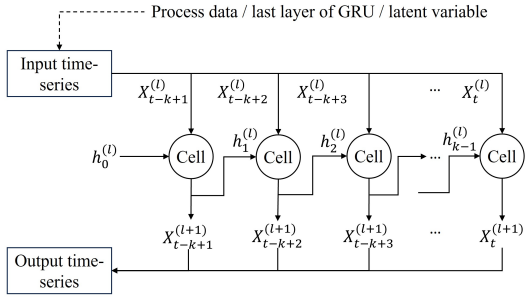

Fig. 3. The l-th layer of GRU

Note that the recurrent nature of GRU is for the analog of the cluster dynamic (3), which enables the GRU-based AE to handle time-series data better than static AEs. The FCNN is stacked after the multi-layer GRU for reshaping and more redundancy.

The entire AE of cluster i can be regarded as a function f_i described by:

$$\begin{cases} f_i\left(\begin{pmatrix} u_{c,i}^{(t)} \\ y_{c,i}^{(t)} \end{pmatrix}\right) = f_{de,i}\left(\theta_{de,i}, f_{en,i}\left(\theta_{en,i}, \begin{pmatrix} u_{c,i}^{(t)} \\ y_{c,i}^{(t)} \end{pmatrix}\right)\right) \\ v_{c,i}^{(t)} = f_{en,i}\left(\theta_{en,i}, \begin{pmatrix} u_{c,i}^{(t)} \\ y_{c,i}^{(t)} \end{pmatrix}\right) \approx I_{\widetilde{\mathcal{C}}_i,0} \begin{pmatrix} u_{c,i}^{(t)} \\ y_{c,i}^{(t)} \end{pmatrix} \\ \begin{pmatrix} \hat{u}_{c,i}^{(t)} \\ \hat{y}_{c,i}^{(t)} \end{pmatrix} = f_{de,i}\left(\theta_{de,i}, v_{c,i}^{(t)}\right) \approx I_{\mathcal{C}_i,0} v_{c,i}^{(t)} \end{cases} \tag{17}$$

where θ_{en} and θ_{de} are the parameters of encoder and decoder. $\begin{pmatrix} u_{c,i}^{(t)} \\ y_{c,i}^{(t)} \end{pmatrix}$ is a data sequence of length k recorded from $t-k+1$ to t. It follows from the projection method in (10) that the $f_{de,i}(\theta_{de,i}, \cdot)$ is for the purpose of an analog of $I_{\mathcal{C}_i,0}$ and the encoder $f_{en,i}(\theta_{en,i}, \cdot)$ is for its conjugate $I_{\widetilde{\mathcal{C}}_i,0}$. In other words, the objective of AE is to learn the SIR where the nominal data pattern exists.

We define the AE-based residual signal as follows.

$$r_{c,i}(t) = \begin{pmatrix} u_{c,i}(t) \\ y_{c,i}(t) \end{pmatrix} - \begin{pmatrix} \hat{u}_{c,i}(t) \\ \hat{y}_{c,i}(t) \end{pmatrix} \tag{18}$$

by extracting the latest data points of the input and the output sequences of AE. The evaluation function value is computed with a sliding window of length L_w based on (14).

The AE is trained during the fault-free process with noises by the generated dataset of (u_i, y_i). The loss function is designed for multiple purposes and is composed of three parts:

$$\mathcal{L}_i = \mathcal{L}_{i,1} + w_{i,2}\mathcal{L}_{i,2} + w_{i,3}\mathcal{L}_{i,3} \tag{19}$$

where each loss value is calculated batch-wise. Denote $\mathcal{B}$ a set of indexes in the batch. $w_{i,2}, w_{i,3}$ are the positive weighting factors for regularization. The losses are defined as follows.

1) $\mathcal{L}_{i,1}$ is the standard mean-square error of input and output to minimize the reconstruction error during the fault-free process:

$$\mathcal{L}_{i,1} = \frac{1}{|\mathcal{B}|} \sum_{p \in \mathcal{B}} \left\| \begin{pmatrix} u_{c,i}^{(p)} \\ y_{c,i}^{(p)} \end{pmatrix} - \begin{pmatrix} \hat{u}_{c,i}^{(p)} \\ \hat{y}_{c,i}^{(p)} \end{pmatrix} \right\|_2^2 \tag{20}$$

2) $\mathcal{L}_{i,2}$ is for satisfying the idempotence property of the orthogonal projection in (9):

$$\mathcal{L}_{i,2} = \frac{1}{|\mathcal{B}|} \sum_{p \in \mathcal{B}} \left\| \begin{pmatrix} \hat{u}_{c,i}^{(p)} \\ \hat{y}_{c,i}^{(p)} \end{pmatrix} - f_i\left(\begin{pmatrix} \hat{u}_{c,i}^{(p)} \\ \hat{y}_{c,i}^{(p)} \end{pmatrix}\right) \right\|_2^2 \tag{21}$$

3) $\mathcal{L}_{i,3}$ is designed for minimizing the construction error of the latent variable $v_{c,i}$:

$$\mathcal{L}_{i,3} = \frac{1}{|\mathcal{B}|} \sum_{p \in \mathcal{B}} \left\| v_{c,i}^{(p)} - v_{c,i,0}^{(p)} \right\|_2^2 \tag{22}$$

where $v_{c,i,0}^{(p)}$ is the known value during the data generation associated with $\begin{pmatrix} u_{c,i}^{(p)} \\ y_{c,i}^{(p)} \end{pmatrix}$.

For validation purposes, we use the standard mean-square error:

$$\mathcal{L}_{val} = \frac{1}{|\mathcal{B}|} \sum_{p \in \mathcal{B}} \left\| \begin{pmatrix} u_{c,i}^{(p)}(t_p) \\ y_{c,i}^{(p)}(t_p) \end{pmatrix} - \begin{pmatrix} \hat{u}_{c,i}^{(p)}(t_p) \\ \hat{y}_{c,i}^{(p)}(t_p) \end{pmatrix} \right\|_2^2 \tag{23}$$

where (t_p) means the latest data point at $t = t_p$ of the p-indexed sequence in batch $\mathcal{B}$.

B. Fault agent isolation

In order to isolate the fault agent, we first present the following Lemma.

Lemma 1: The presence of faults in agent i has no impact on the residual signal $r_{c,j}$ of the cluster $\mathcal{C}_j$, s.t. $\mathcal{C}_j \cap \mathcal{C}_i = \varnothing$.

Proof: Since $\mathcal{C}_j \cap \mathcal{C}_i = \varnothing$, faults in agent i only affect the dynamic of cluster $\mathcal{C}_i$ but have no impact on the model of $\mathcal{C}_j$. It is straightforward that from the definition of the image space in (5), the nominal data $\text{col}\{u_{c,j}, y_{c,j}\}$ still belongs to $\mathcal{I}_{\mathcal{C}_j}$, where the RCF $(M_{c,j}, N_{c,j})$ remaining unchanged, and thus $r_{c,j} = 0$ still holds. ■

The following theorem provides a criterion for isolating faulty agents.

Theorem 1 (Fault agent isolation): Under the Assumption 1, the agent i is faulty if and only if

1) all the cluster residual signals $r_{c,k}, k \in \tilde{\mathcal{N}}_i$ indicate the presence of fault,
2) and for any $\mathcal{C}_j$ s.t. $\mathcal{C}_j \cap \mathcal{C}_i = \varnothing$, $r_{c,j}$ is not affected by the fault.

Proof: "⇒": if agent i is faulty, all the clusters containing agent i are faulty, which are $\mathcal{C}_i, \mathcal{C}_{i_1}, \ldots \mathcal{C}_{i_{N_i}}$. Based

on Lemma 1, for the cluster j s.t. $\mathcal{C}_j \cap \mathcal{C}_i = \varnothing$, $r_{c,j}$ is not affected.

"$\Leftarrow$": Suppose that agent $\mathcal{G}_l$ is faulty. Then $\mathcal{G}_l \in \mathcal{C}_k$, $\forall k \in \tilde{\mathcal{N}}_i$. Note the fact that $\mathcal{G}_i \in \mathcal{C}_k$, $\forall k \in \tilde{\mathcal{N}}_i$. Based on the Assumption 1, only one agent is faulty, and thus $l = i$ follows. ■

Remark 4: The above proof demonstrates that the condition 1) $\forall (i,j) \in \{1,\dots,N\}^2$, $\mathcal{C}_i \neq \mathcal{C}_j$ in Assumption 1 is a sufficient and necessary condition for fault detection. To better illustrate the necessity, consider a counterexample of a three-agent system that is fully connected, i.e., $\mathcal{C}_1 = \mathcal{C}_2 = \mathcal{C}_3 = \{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3\}$. A fault in any of the three agents would affect all cluster residuals, making fault agent isolation impossible. Despite this, the assumption could be relaxed if some agents had absolute outputs. However, this scenario falls outside the scope of the current study and will be addressed in future research.

IV. Simulation Example

To validate the proposed approach, we present a simulation example of a four-agent system associated with the topology in Fig. 4. The agents' dynamic matrices are defined as follows:

$$A_i = \begin{pmatrix} a_{i1} & 0.01 \\ 0 & a_{i2} \end{pmatrix}, B_i = \begin{pmatrix} 0 \\ 0.01 \end{pmatrix}, C_i = \begin{pmatrix} 1 \\ 0 \end{pmatrix}^T, D_i = 0 \quad (24)$$

$$\epsilon_{i_j} \sim \mathcal{N}(0, \sigma_1^2),\ \eta_i \sim \begin{pmatrix} \mathcal{N}(0,\sigma_2^2) \\ \mathcal{N}(0,0) \end{pmatrix} \quad (25)$$

associated with a sampling time $T_s = 0.01s$. $a_{21} = a_{22} = 0.99$, $a_{31} = 0.985$, $a_{32} = 0.995$, $a_{41} = 0.98$, $a_{42} = 0.995$, $\sigma_1 = 0.01$, $\sigma_2 = 0.005$, $a_{11} = a_{12} = 0.98$. The training set is prepared with $100,000$ fault-free data points per agent illustrated in Fig. 5, and the ratio between the training set and validation set is $8:2$. Since each A_i is stable and to generate the dataset, we can set $u_i = v_i$, where the input is designed to have the output covering at least 85% operation range.

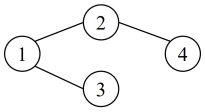

Fig. 4. Topology of a four-agent system

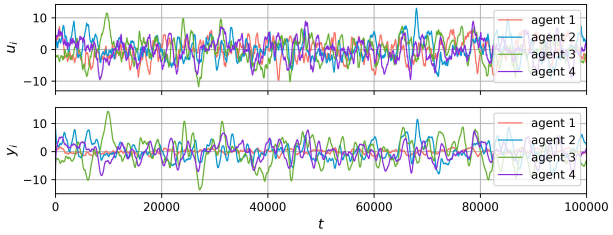

Fig. 5. Prepared dataset

Using parameter sweeping and tuning, we analyze the impact of different regularization terms on the validation loss in order to select an optimal combination. For example, Fig. 6 illustrates the validation loss with different regularization weights of cluster 1. It can be seen that a middle value of $w_{1,2}$ and a higher value of $w_{1,3}$ are favorable for validation loss. The final parameters are selected in the Table. I, and the corresponding training and validation loss per epoch is shown in Fig. 7.

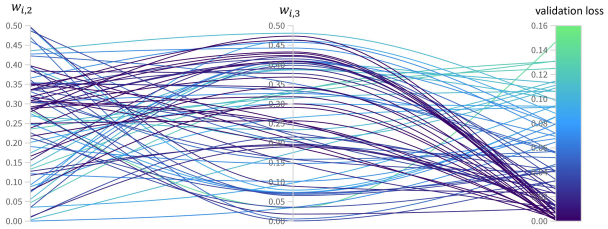

Fig. 6. Impact of the regularization weights on the validation loss, with $i = 1$

TABLE I
Simualtion Configuration

Parameters	Configuration
Encoder GRU hidden layers	4
Decoder GRU hidden layers	4
GRU hidden size	20
$w_{1,2}$, $w_{1,3}$	0.166, 0.343
$w_{2,2}$, $w_{2,3}$	0.318, 0.4296
$w_{3,2}$, $w_{3,3}$	0.336, 0.4631
$w_{4,2}$, $w_{4,3}$	0.217, 0.2907
Epochs	70
Learning rate	0.0005
Optimizer	Adam
Batch size	64
Input sequence length	10
Sliding window length L_w	30

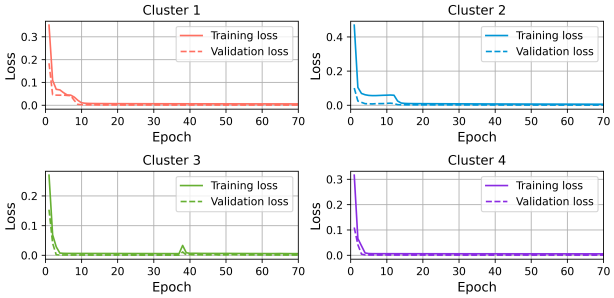

Fig. 7. Training and validation loss

To evaluate the fault detection capability of the proposed method, an additive process fault of value $(0, 0.08)^T$ is injected in agent 2 during $3000 \leq t \leq 5000$, and also injected in agent 3 during $8000 \leq t \leq 9000$. The detection threshold $J_{th,i}$ is set to maintain a low FAR as outlined in Table. II. The detection results are shown in Fig. 8, which validates the proposed methods. During $3000 \leq t \leq 5000$, clusters 1,2, and 4 are affected by the fault, allowing us to identify agent 2 as faulty based on Theorem 1. Similarly, the evaluation function values during $8000 \leq t \leq 9000$ in clusters 1 and 3 indicate a fault in agent 4. Table. II summarizes the performance metrics of the AEs for the four clusters, including FAR, missed detection rate (MDR), accuracy, and F1-score. It is evident that under small FARs, the proposed

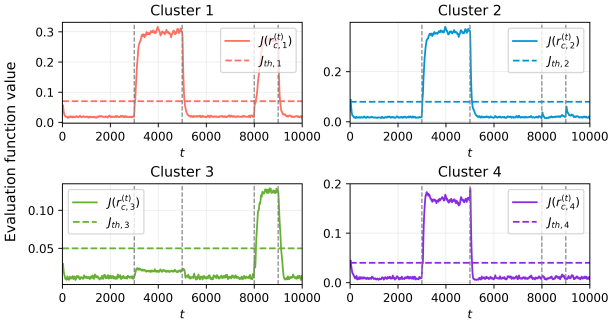

Fig. 8. Detection results

AEs achieve high accuracy and F1 scores, highlighting the proposed method as an excellent alternative for fault detection for heterogeneous MASs under the constraints of unknown dynamics.

TABLE II

FAULT DETECTION PERFORMANCE METRICS

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
$J_{th,i}$	0.0800	0.0900	0.0600	0.0500
FAR	0.0300	0.0495	0.0364	0.0196
MDR	0.0350	0.0090	0.0750	0.0155
Accuracy	0.9895	0.9982	0.9925	0.9969
F1-score	0.9821	0.9955	0.9610	0.9922

V. CONCLUSION

In this study, a data-driven fault detection scheme based on GRUs and AEs is successfully developed for heterogeneous MASs. Leveraging cluster transformation using relative output, the fault detection problem is formulated utilizing the SIR and the orthogonal projection technique. The challenge of unknown dynamics is addressed through the learning ability of GRU-based AEs, providing an effective alternative to the traditional observer-based methods and the SIR/projection-based methods. The loss function of AEs is designed with regularized terms, and an analysis of the weighting factor is performed to determine the optimal parameter selection. Simulation results validate the proposed method with favorable fault detection performance metrics.

It is worth noting that although this work focuses on linear MASs, it is promising that the AE-based fault detection scheme can be extended to generic nonlinear MASs.

REFERENCES

[1] Z. Wang and M. Chadli, "Observer-based distributed dynamic event-triggered control of multi-agent systems with adjustable interevent time," *Asian Journal of Control*, vol. 26, no. 6, pp. 2783–2795, 2024.

[2] Z. Wang and M. Chadli, "Distributed Observer-Based Dynamic Event-Triggered Control of Multi-Agent Systems with Adjustable Inter-Event Time," in *Proceedings of the 62nd IEEE Conference on Decision and Control (CDC)*. Singapore, Singapore: IEEE, Dec. 2023, pp. 2391–2396.

[3] S. Xu, W. Huang, D. Huang, H. Chen, Y. Chai, M. Ma, and W. X. Zheng, "A Reduced-Order Observer-Based Method for Simultaneous Diagnosis of Open-Switch and Current Sensor Faults of a Grid-Tied NPC Inverter," *IEEE Transactions on Power Electronics*, vol. 38, no. 7, pp. 9019–9032, July 2023.

[4] Z. Wang and M. Chadli, "Distributed Joint Fault Estimation for Multi-Agent Systems via Dynamic Event-Triggered Communication," *IEEE Control Systems Letters*, vol. 8, pp. 868–873, May 2024.

[5] S. X. Ding, *Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms and Tools*, ser. Advances in Industrial Control. London: Springer, 2013.

[6] M. Chadli, M. Davoodi, and N. Meskin, "Distributed state estimation, fault detection and isolation filter design for heterogeneous multi-agent linear parameter-varying systems," *IET Control Theory & Applications*, vol. 11, no. 2, pp. 254–262, Oct. 2016.

[7] D. Liang, Z. He, R. Li, and Y. Yang, "Distributed fault detection for uncertain Lipschitz nonlinear multi-agent systems in finite frequency domain," *International Journal of Robust and Nonlinear Control*, vol. 32, no. 13, pp. 7594–7610, 2022.

[8] L. Zhang, X. Zhang, X. Zhao, and N. Zhao, "Membership-Function-Dependent Reachable Set Synthesis of Takagi-Sugeno Fuzzy Singular Multi-Agent Systems Subject to Deception Attacks," *IEEE Transactions on Automation Science and Engineering*, pp. 1–10, 2024.

[9] M. Chadli, A. Abdo, and S. X. Ding, "H-/H-infinity fault detection filter design for discrete-time Takagi–Sugeno fuzzy system," *Automatica*, vol. 49, no. 7, pp. 1996–2005, July 2013.

[10] M. Davoodi, N. Meskin, and K. Khorasani, "Simultaneous fault detection and consensus control design for a network of multi-agent systems," *Automatica*, vol. 66, pp. 185–194, Apr. 2016.

[11] L. Li, S. X. Ding, J. Qiu, and Y. Yang, "Real-Time Fault Detection Approach for Nonlinear Systems and its Asynchronous T–S Fuzzy Observer-Based Implementation," *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 283–294, Feb. 2017.

[12] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, July 2006.

[13] J. Qian, Z. Song, Y. Yao, Z. Zhu, and X. Zhang, "A review on autoencoder based representation learning for fault detection and diagnosis in industrial processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 231, p. 104711, Dec. 2022.

[14] G. Jiang, P. Xie, H. He, and J. Yan, "Wind Turbine Fault Detection Using a Denoising Autoencoder With Temporal Information," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 89–100, Feb. 2018.

[15] S. Wang, Y. Ju, P. Xie, and C. Cheng, "Fault detection using generalized autoencoder with neighborhood restriction for electrical drive systems of high-speed trains," *Control Engineering Practice*, vol. 143, p. 105804, Feb. 2024.

[16] J. Zhu, M. Jiang, and Z. Liu, "Fault Detection and Diagnosis in Industrial Processes with Variational Autoencoder: A Comprehensive Study," *Sensors*, vol. 22, no. 1, p. 227, Jan. 2022.

[17] R. Dhakal, C. Bosma, P. Chaudhary, and L. N. Kandel, "UAV Fault and Anomaly Detection Using Autoencoders," in *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*, Oct. 2023, pp. 1–8.

[18] L. Li, S. X. Ding, K. Liang, Z. Chen, and T. Xue, "Control theoretically explainable application of autoencoder methods to fault detection in nonlinear dynamic systems," *arXiv*, pp. 1–21, May 2023.

[19] S. Wu, H. Luo, Y. Jiang, J. Zhang, J. Tian, and S. Yin, "SIR-Aided Secure Transmission and Attack Detection for Security Management of Nonlinear Cyber-Physical System Using GRU Autoencoder," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 4, pp. 5529–5538, Apr. 2024.

[20] Y. Bai and J. Wang, "Fault detection and isolation using relative information for multi-agent systems," *ISA Transactions*, vol. 116, pp. 182–190, Oct. 2021.

[21] S. X. Ding, *Advanced methods for fault diagnosis and fault-tolerant control*. Berlin, Heidelberg: Springer, 2021.

[22] J. W. Hoffmann, "Normalized coprime factorizations in continuous and discrete time—a joint state-space approach," *IMA Journal of Mathematical Control and Information*, vol. 13, no. 4, pp. 359–384, Dec. 1996.

[23] K. Cho, B. v. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," Sept. 2014.