Exploring a Modular Architecture for Sensor Validation in Digital Twins

Hossein Darvishi*, Domenico Ciuonzo†, Pierluigi Salvo Rossi*,‡
*Dept. Electronic Systems, Norwegian University of Science and Technology, 7491 Trondheim, Norway
†Dept. Electrical Engineering & Information Technologies, University of Naples "Federico II," Naples 80125, Italy
‡Dept. Gas Technology, SINTEF Energy Research, 7034 Trondheim, Norway
Email: hossein.darvishi@ntnu.no; domenico.ciuonzo@unina.it; salvorossi@ieee.org

***Abstract*—Decision-support systems rely on data exchange between digital twins (DTs) and physical twins (PTs). Faulty sensors (e.g. due to hardware/software failures) deliver unreliable data and potentially generate critical damages. Prompt *sensor fault detection, isolation and accommodation* (SFDIA) plays a crucial role in DT design. In this respect, data-driven approaches to SFDIA have recently shown to be effective. This work focuses on a modular SFDIA (M-SFDIA) architecture and explores the impact of using different types of neural-network (NN) building blocks. Numerical results of different choices are shown with reference to a wireless sensor network publicly-available dataset demonstrating the validity of such architecture.**

***Index Terms*—Digital Twin, fault tolerance, neural networks, sensor validation.**

I. Introduction

Digital twins (DTs) are largely applied to objects [1], systems [2], processes and services [3]. A DT requires data about assets/processes to create a virtual representation of the paired physical twin (PT), usually collected and provided in real time by sensors. However, the data flow from PTs to DTs is not necessarily reliable [4]–[6]: malfunctioning sensors can harm the system leading to performance degradation or even safety-critical issues. The relevance of sensor validation (i.e. deployment of strategies for sensor fault detection, isolation and accommodation (SFDIA) is thus apparent.

Recent advances on SFDIA mostly relies on *analytical redundancy* [7], i.e. the use of virtual sensors using exploiting data dependencies for monitoring purposes. Model-based SFDIA approaches are effective when physical representations of the model/process parameters are available. Popular approaches build upon Kalman filters [8], [9], observers [10] and Bayesian [11] methods, however complex non-linear systems remain challenging to deal with. Data-driven SFDIA approaches have gained attention due to their ability to handle complex systems without the need for exact knowledge of the underlying model. Popular approaches build upon principal component analysis [12], support vector machine [13] and neural network (NN) based methods [14]–[17]. A modular SFDIA (M-SFDIA) scheme has been recently proposed in [18], [19] based on multi-layer perceptron (MLP) blocks connected in three layers. The M-SFDIA architecture exploits jointly temporal and spatial dependencies of the sensors measurements. Accordingly, we here explore the impact of different building blocks within the M-SFDIA architecture.

[0]This work was partially supported by the Research Council of Norway under the Project 311902 (SIGNIFY) within the framework IKTPLUSS.

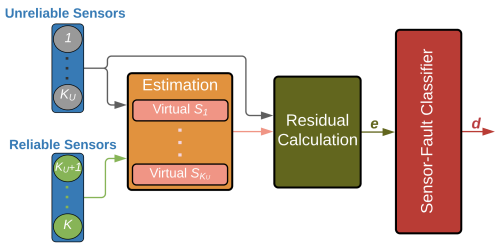


Fig. 1: Block diagram of the SFDIA system.

Specifically, the *contributions* of the paper are: (i) to investigate and compare the performance of different NN-based virtual sensors used within the M-SFDIA architecture; (ii) to compare the performance with a state-of-the-art SFDIA system based on autoencoders (AEs) [20]. For performance evaluation of the various structures, we considered a wireless sensor network (WSN) publicly-available dataset [21]. Also, synthetically-generated weak bias faults are superimposed to the real-world wireless sensor network (WSN) dataset.

The rest of the paper is organized as follows: the basic M-SFDIA architecture and related variations are presented in Sec. II; numerical results and performance discussion are found in Sec. III; Sec. IV provides some concluding remarks.[1]

II. M-SFDIA

We assume that K different sensors monitors the considered PT, the first K_U sensors being *unreliable* (i.e. vulnerable to faults) and the remaining ones *reliable* (i.e. their ideal functionality is guaranteed). Specifically, $x_k[n]$ denotes the measurement by the kth sensor at time n. Accordingly, $\boldsymbol{x}_{(-k)}[n]$, $k = 1, \ldots, K_U$, denotes the measurements at time n by all the sensors except the kth unreliable sensor. Finally, $x_k[n : n-L]$ (resp. $\boldsymbol{x}_{(-k)}[n : n-L]$) denotes the portion of time series (resp. multivariate time series) containing $L+1$ measurements up to time n.

[1]*Notation* - Lower-case bold letters indicate vectors; $\mathcal{U}(a,b)$ (resp. $\mathcal{U}_d(a,b)$) denotes a uniform (resp. discrete-uniform) probability density function (PDF) with support (a,b) (resp. $\{a, a+1, \ldots, b\}$), whereas $\mathcal{B}(p)$ denotes a Bernoulli PDF with activation probability p.

A. M-SFDIA Architecture

The system architecture is made of *three* layers (see Fig. 1): The *first layer* contains K_U independent virtual sensors, each being a NN-based estimator receiving measurements from all the sensors except the one under estimation and producing sensor-measurement estimates, namely

$$\hat{x}_k(n) \triangleq \mathcal{V}_k(\boldsymbol{x}_{(-k)}[n : n - L_v]) \tag{1}$$

The *second layer* computes the difference between estimates and actual measurements (i.e. *residual signals*), namely:

$$\boldsymbol{\Delta}[n] \triangleq \begin{bmatrix} (\hat{x}_1(n) - x_1(n)) & \cdots & (\hat{x}_{K_U}(n) - x_{K_U}(n)) \end{bmatrix}^T \tag{2}$$

The *last layer* is a NN-based classifier processing residual signals of all sensors pairs and providing a decision vector $\boldsymbol{d}[n]$ with elements $d_k[n] \in [0, 1],\ k = 1, \ldots, K_U$ denoting pseudo-probabilities of the sensors being faulty, namely

$$\boldsymbol{d}[n] \triangleq \mathcal{C}(\boldsymbol{\Delta}[n : n - L_c]) \tag{3}$$

L_v (resp. L_c) in Eq. (1) (resp. (3)) denotes the size of a sliding window selecting the inputs for the virtual sensors (resp. calssifier). A faulty sensor is detected and identified when the element(s) of the decision vector $\boldsymbol{d}[n]$ exceed(s) a predefined threshold (γ): $\max_{k=1}^{K_U} d_k[n] \gtrless \gamma$ is used for *detection*, while $\hat{k} = \arg\max_{k=1}^{K_U} d_k[n]$ is used for *identification*. Also, *accommodation* is performed by replacing the identified faulty sensor with the estimate from the corresponding virtual sensor.

B. NN-based Building Blocks

We considered different types of NN-based building blocks.
MLP: a class of feedforward NNs that can model arbitrary nonlinear mappings $f : \mathbb{R}^{i\times 1} \to \mathbb{R}^{j\times 1}$. The NN is made of an arbitrary number of hidden layers, each consisting of an affine matrix operation and an entry-wise nonlinear activation. The *baseline* M-SFDIA [19] uses MLP building blocks.
Convolutional NN (CNN): a specialized NN inspired by visual mechanism. A sequence of *convolutional* layers (each based on translation-invariant filters with limited extent) are responsible for feature extractions with increased level of abstraction. One-dimensional CNNs have shown to be appealing in (multivariate) time-series processing.
RNN: a class of NN suited for time series exploiting loopy connections for keeping memory of sequential information. Long-term dependencies in the data are usually captured when using two advanced types of RNNs: *long-short term* memory (LSTM) [22] and *gated recurrent unit* (GRU) [23].

III. Numerical Results and Discussion

A. WSN Dataset

The considered dataset was collected at the University of North Carolina [21] and is a collection of *two pairs* of temperature-humidity sensors placed outdoor and indoor. Only the four *fault-prone* temperature measurements (hence $K = K_U = 4$) during normal operation are used. The dataset is split into three subsets: 70%, 15% and 15% for training, validation, and testing, respectively, and *min-max scaling* is applied (with range extension learnt from the training set only).

Synthetically-generated *bias faults* are superimposed to the dataset[2]. A bias fault b with level $|b| \sim \mathcal{U}(0.2, 0.4)$ and $\text{sign}(b) \sim \mathcal{B}(0.5)$ is injected into the normalized dataset for $M \sim \mathcal{U}_d(2, 20)$ consecutive samples as

$$x'_{k,b}[n] = \begin{cases} x'_k[n] + b\,, & 0 \le n - m < M \\ x'_k[n]\,, & \text{otherwise} \end{cases} \tag{4}$$

where x'_k and $x'_{k,b}$ are the "normalized" and the "polluted" measurements of kth sensor, and m refers to the fault starting time.

B. Models

The reference MLP-based M-SFDIA discussed in [19] is compared with *seven* variants using the following building blocks: CNN with a single convolutional layer (size-3 kernel) and max-pooling layer (size-2 pad); GRU/LSTM with a single unit; GRU-CNN/LSTM-CNN combining the previous 2 types; GRU-RS/LSTM-RS stacking 2 units of the second type, following a return sequence (RS) mechanism.

In all networks, we consider 20 hidden nodes per hidden layer and the size of the input window is $L_v = L_c = 30$. Virtual sensors have a dense output layer with a single node and linear activation, while the classifier has a dense output layer with K_U nodes and sigmoidal activation. Mean square error (MSE) and binary cross-entropy are the loss functions used as optimization metric for the virtual sensors and the classifier, respectively. Virtual sensors were trained using healthy data, while the classifier was trained based on a loss capitalizing *multitask learning* using the polluted faulty data.[3] We use the superscripts $(\cdot)^{\text{vs}}$ and $(\cdot)^{\text{cl}}$ when NN building blocks refer to virtual sensors or classifier, respectively.

Additionally, results of our approach in terms of detection, identification and accommodation performance are compared with a state-of-the-art AE-based architecture in [20][4].

C. Performance Analysis and Comparison

Estimation Performance: Fig. 2 displays the statistics (median value, 95% confidence interval, and outliers) of the root mean squared error (RMSE) in the fault-free situation on the test set for each virtual sensors. $\texttt{MLP}^{\text{vs}}$ has the highest median over two out of four sensors (**S3** and **S4**), while $\texttt{GRU-RS}^{\text{vs}}$ and $\texttt{CNN}^{\text{vs}}$ outperform on average the other counterparts and provide the lowest RMSE value.
Detection and Isolation Performance: Fig. 3 shows the probabilities of *detection* and *classification* with respect to the probability of false alarm (set via γ) for different classifiers[5],

[2] A fault rate (ratio between the number of faulty and non-faulty samples) equal to 0.2 is considered. The proposed M-SFDIA approach can handle different types of faults, but those are not considered here for brevity.

[3] We leveraged the models provided by Keras Python API running on TensorFlow 2 to implement, train and test the models.

[4] We modified the decision logic of the AE architecture in order to enable the identification task which was not addressed in the original work.

[5] *Dashed* lines refer to the baseline M-SFDIA [19] and the AE architecture [20]. *Solid* curves refer to different classifiers using the same residual-signals (i.e. computed via $\texttt{GRU-RS}^{\text{vs}}$).

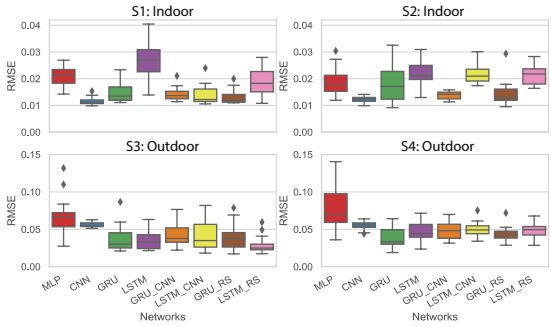
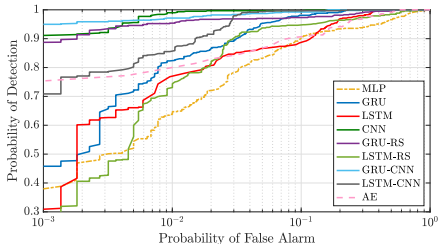
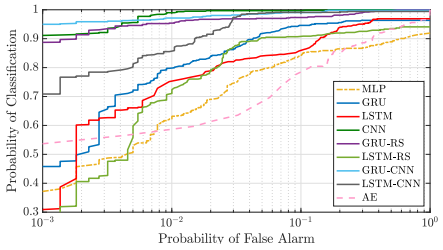


Fig. 2: Box-plot of estimation RMSE for each virtual sensor.



(a) Detection Performance



(b) Isolation Performance

Fig. 3: Detection and isolation performance of different classifier models by using ROC curves.

i.e. the receiver operating characteristic (ROC) curves, when synthetically-generated weak-bias faults are superimposed. The probability of detection (resp. classification) refers to the probability that the system correctly detects (resp. isolates) the faulty sensor(s). In the latter case, we consider the average probability of classification over all the unreliable sensors.

The baseline MLP-based M-SFDIA has the worst performance. Specifically, $\texttt{GRU-CNN}^{\text{cl}}$ and $\texttt{CNN}^{\text{cl}}$ models achieve the highest performance (in terms of detection and isolation): $\geq 95\%$ (resp. $\geq 90\%$) detection/isolation rate under false alarm rate of 10^{-2} (resp. of 10^{-3}). It is apparent that CNNs and RNNs are better in capturing more complex spatio-temporal dependencies in the data.

Accommodation Performance: In Fig. 4 the error between the accommodated samples with actual non-faulty sensor measurements as well as the difference between miss-detected faulty measurements with actual non-faulty sensor measurements is considered when false-alarm probability is 10^{-2}. Both $\texttt{CNN}^{\text{cl}}$ and $\texttt{GRU-CNN}^{\text{cl}}$ present the smallest accommodation error as: (i) they miss-detect less faults and (ii) they rely on better virtual sensors.

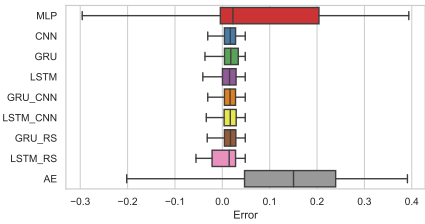


Fig. 4: Accommodation performance comparison in terms of averaged distribution of the error signals for the fixed false alarm probability of 10^{-2}.

TABLE I: Run-Time Per-Epoch (RTPE in seconds) and Number of Trainable Parameters (TP) of the baseline and different NN models. The RTPE is in the format avg. obtained over 3-folds.

Model	**Virtual Sensor/AE**		**Classifier/Denoising-AE**	
	RTPE	TP	RTPE	TP
MLP	0.0394	1901	0.0676	3004
CNN	0.0585	481	**0.1555**	**1464**
GRU	0.4021	1521	0.5516	1644
LSTM	0.3619	1941	0.5904	2084
GRU-CNN	0.2664	2741	**0.4257**	**2864**
LSTM-CNN	0.2511	3501	0.4318	3624
GRU-RS	**0.8166**	**4041**	1.1501	4164
LSTM-RS	0.7491	5221	1.2002	5364
AE	0.0618	41102	0.2797	41102

Complexity Assessment: Tab. I compares the computational complexity of the considered systems by showing the Run-Time Per-Epoch (RTPE) of each architecture paired with the corresponding number of Trainable Parameters (TP), which is related to the theoretical complexity of the training phase. The baseline $\texttt{MLP}^{\text{vs}}$ has the smallest RTPE, while the more complex (and better performing) $\texttt{GRU-RS}^{\text{vs}}$ model takes longer time to train. Also, it is worth noting that the baseline $\texttt{MLP}^{\text{cl}}$ and the AE, despite exhibiting the worst performance, have a larger number of TP than the best performing classifiers (reported in bold in Tab. I).

IV. Conclusions

In this paper, different types of NN models were exploited within a common M-SFDIA architecture. To validate the effectiveness of various configurations, we have injected synthetically-generated weak bias faults to a publicly-available WSN dataset. By using GRU-RS models as virtual estimators and GRU-CNN model for the classifier, we achieved detection and isolation probabilities of about 0.95 for false-alarm probability equal to 10^{-3}, which is $\approx 3\times$ better than the performance of the baseline configuration. The performance gain is due to better handling of the spatio-temporal dependencies in the data.

REFERENCES

[1] S. N. Bairampalli, F. Ustolin, D. Ciuonzo, and P. Salvo Rossi, "Digital moka: Small-scale condition monitoring in process engineering," *IEEE Sens. Lett.*, vol. 5, no. 3, pp. 1–4, 2021.

[2] L. Zhao, G. Han, Z. Li, and L. Shu, "Intelligent digital twin-based software-defined vehicular networks," *IEEE Netw.*, vol. 34, no. 5, pp. 178–184, 2020.

[3] S. Aheleroff, X. Xu, R. Y. Zhong, and Y. Lu, "Digital twin as a service (DTaaS) in Industry 4.0: an architecture reference model," *Advanced Engineering Informatics*, vol. 47, p. 101225, 2021.

[4] Z. Yang, N. Meratnia, and P. Havinga, "An online outlier detection technique for wireless sensor networks using unsupervised quarter-sphere support vector machine," in *IEEE ISSNIP'08*.

[5] A. Mahapatro and P. M. Khilar, "Fault diagnosis in wireless sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 2000–2026, 2013.

[6] N. R. Prasad and M. Alam, "Security framework for wireless sensor networks," *Wirel. Pers. Commun.*, vol. 37, no. 3-4, pp. 455–469, 2006.

[7] S. Gururajan, M. L. Fravolini, M. Rhudy, A. Moschitta, and M. Napolitano, "Evaluation of sensor failure detection, identification and accommodation (SFDIA) performance following common-mode failures of Pitot tubes," SAE Technical Paper, Tech. Rep., 09 2014.

[8] R. Saravanakumar, M. Manimozhi, D. Kothari, and M. Tejenosh, "Simulation of sensor fault diagnosis for wind turbine generators DFIG and PMSM using Kalman filter," *Energy procedia*, vol. 54, pp. 494–505, 2014.

[9] S. Huang, K. K. Tan, and T. H. Lee, "Fault diagnosis and fault-tolerant control in linear drives using the Kalman filter," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4285–4292, 2012.

[10] Y. Wang, N. Masoud, and A. Khojandi, "Real-time sensor anomaly detection and recovery in connected automated vehicle sensors," *IEEE Trans. Intell. Transp. Syst.*, 2020.

[11] N. Mehranbod, M. Soroush, and C. Panjapornpon, "A method of sensor fault detection and identification," *Journal of Process Control*, vol. 15, no. 3, pp. 321–339, 2005.

[12] Y. Tharrault, G. Mourot, and J. Ragot, "Fault detection and isolation with robust principal component analysis," in *IEEE MED'08*.

[13] S. Zidi, T. Moulahi, and B. Alaya, "Fault detection in wireless sensor networks through SVM classifier," *IEEE Sensors J.*, vol. 18, no. 1, pp. 340–347, 2018.

[14] S. Hussain, M. Mokhtar, and J. M. Howe, "Sensor failure detection, identification, and accommodation using fully connected cascade neural network," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1683–1692, 2015.

[15] F. Balzano, M. L. Fravolini, M. R. Napolitano, S. d'Urso, M. Crispoltoni, and G. del Core, "Air data sensor fault detection with an augmented floating limiter," *Hindawi Int. Journal of Aerospace Eng.*, 2018.

[16] H. Zhao, "Neural component analysis for fault detection," *Chemometrics and Intelligent Laboratory Systems*, vol. 176, 12 2017.

[17] D. Haldimann, M. Guerriero, Y. Maret, N. Bonavita, G. Ciarlo, and M. Sabbadin, "A scalable algorithm for identifying multiple-sensor faults using disentangled RNNs," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–14, 2020.

[18] H. Darvishi, D. Ciuonzo, E. R. Eide, and P. Salvo Rossi, "A data-driven architecture for sensor validation based on neural networks," in *IEEE Sensors'20*, pp. 1–4.

[19] ——, "Sensor-fault detection, isolation and accommodation for digital twins via modular data-driven architecture," *IEEE Sensors J.*, vol. 21, no. 4, pp. 4827–4838, February 2021.

[20] M. M. N. Aboelwafa, K. G. Seddik, M. H. Eldefrawy, Y. Gadallah, and M. Gidlund, "A machine-learning-based technique for false data injection attacks detection in industrial IoT," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8462–8471, 2020.

[21] S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie, and M. Palaniswami, "Labelled data collection for anomaly detection in wireless sensor networks," in *IEEE ISSNIP'10*.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[23] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.