Real-Time Sensor Fault Detection, Isolation and Accommodation for Industrial Digital Twins

Hossein Darvishi
Department of Electronic Systems
Norwegian University of
Science and Technology
7491 Trondheim, Norway
Email: hossein.darvishi@ntnu.no

Domenico Ciuonzo
Department of Electrical Engineering
and Information Technologies (DIETI)
University of Naples “Federico II”
80125 Naples, Italy
Email: domenico.ciuonzo@unina.it

Pierluigi Salvo Rossi
Department of Electronic Systems
Norwegian University of
Science and Technology
7491 Trondheim, Norway
Email: salvorossi@ieee.org

***Abstract*—The development of Digital Twins (DTs) has bloomed significantly in last years and related use cases are now pervading several application domains. DTs are built upon Internet of Things (IoT) and Industrial IoT platforms and critically rely on the availability of reliable sensor data. To this aim, in this article, we propose a sensor fault detection, isolation and accommodation (SFDIA) architecture based on machine-learning methodologies. Specifically, our architecture exploits the available spatio-temporal correlation in the sensory data in order to detect, isolate and accommodate faulty data via a bank of estimators, a bank of predictors and one classifier, all implemented via multi-layer perceptrons (MLPs). Faulty data are detected and isolated using the classifier, while isolated sensors are accommodated using the estimators. Performance evaluation confirms the effectiveness of the proposed SFDIA architecture to detect, isolate and accommodate faulty data injected into a (real) wireless sensor network (WSN) dataset.**

***Index Terms*—Digital Twin (DT), Industry 4.0, fault tolerance, Internet of Things (IoT), neural networks, sensor validation.**

I. Introduction

The adoption of digital twins (DTs) built upon the Internet of Things (IoT) for industrial environments have grown significantly with the recent wave of digitalization. DTs are virtual representations of physical assets, which utilize the equipped sensors' data to elaborate and deliver real-time insights, predictions and improved decisions.

However, due to harsh environment [1], hardware limitation [2] and/or malicious attacks [3], [4], the data collected by sensors within the system can be faulty. The occurrence of sensor faults during normal system operation is inevitable and might lead to system-performance degradation and, in worst case when dealing with safety-critical systems, loss of lives. Therefore, sensor fault detection, isolation and accommodation (SFDIA) is an extremely important feature to implement in DTs in order to ensure system reliability and safety.

The current research trend mainly focuses on *analytical redundancy*, i.e. exploiting correlations within the system [5] to avoid the deployment of additional sensing hardware. A model-based SFDIA method was developed according to electrical dynamics equations for current sensors of grid side converters [6], where detection and isolation tasks were based on residual generations and linear state observer logic, while accommodation task was achieved by employing physical redundancy for the single-fault scenario. Analogously, a sensor-fault control strategy comprising of two sliding-mode observers and Luenberger observer was adopted for synchronous motor drives and resulted in high computational complexity [7]. Other model-based approaches have been developed with use of Kalman filters [8], [9], Bayesian methods [10] and observer-based methods [11]. Still, in general, it is seldom practical to develop an accurate model of a system due to the inherent complexity and variety of DTs' applications.

On the contrary, *data-driven approaches* (e.g. support vector machine [12], principal component analysis [13] and neural networks (NNs) [5]) are able to overcome this problem as they mostly rely on historical data. A multi-layer perceptron (MLP) NN, a class of feed-forward NNs, is employed by a modular SFDIA (M-SFDIA) method to diagnose faults in DTs [5], [14], while a fully-connected cascade NN is exploited in [15] to reduce the computational complexity. Also, alternative solutions are developed via hybrid approaches, e.g. using banks of NNs and adaptive linear networks, to reduce the computational complexity [16]. Finally, an unsupervised method uses an autoencoder (AE) NN as a classifier to detect faults and a denoising AE to clean the faulty data [4]. However, AE-based method is unable to perform the identification/isolation task within the SFDIA scheme.

In this article, the major motivation is to propose a machine-learning-based SFDIA architecture to exploit spatial and temporal correlations in the data collected from the sensors. To this end, two banks of MLP NNs are employed to perform estimation and prediction of sensors measurements in the system. Moreover, an MLP-based classifier is trained to classify faulty sensors based on dissimilarities between obtained predictions/estimates and actual sensors' readings. The proposed approach differs from our previous work (M-SFDIA [5]) in that we here introduce a bank of MLP NN predictors to better exploit the temporal correlations within each sensor in the proposed architecture. On contrary to AE-based architecture, the proposed architecture performs all three tasks (detection, isolation, and accommodation) by exploiting

This work was partially supported by the Research Council of Norway under the project SIGNIFY within the IKTPLUSS framework (project nr. 311902).

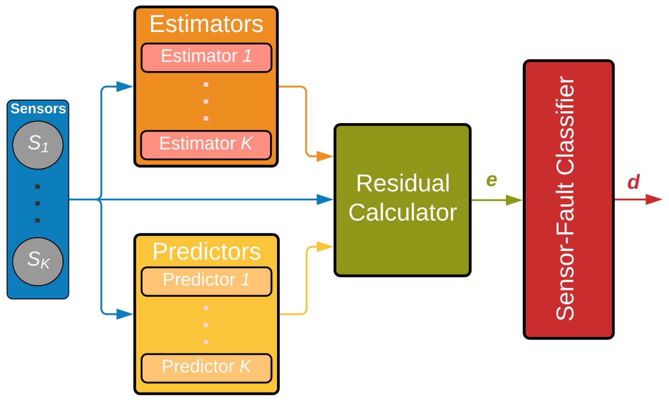


Figure 1: Block diagram of the proposed SFDIA architecture.

MLP modules. Moreover, the proposed architecture works on a real-time basis capable to detect faults online (promptly) as they occur. The considered SFDIA architecture is evaluated through synthetically-generated weak bias faults which were added to a real-world wireless sensor network (WSN) dataset with four sensors measuring temperature and humidity [17]. Numerical results illustrate the superiority of the proposed architecture compared to the M-SFDIA architecture.

The rest of this article is organized as follows. The proposed SFDIA architecture is explained in Sec. II. Sec. III presents the NNs' configuration and the dataset description. Simulation results and performance comparison are reported in Sec. IV. Finally, Sec. V ends the paper and provides direction for further study.

Notation - Lower-case bold letters indicate vectors, $\exp(\cdot)$ is the exponential function and $(\cdot)^T$ denotes transpose operator. $\mathcal{U}(a,b)$ (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.

II. SFDIA

The main idea of the proposed architecture is to exploit temporal and spatial correlations among sensors in the system. The block diagram of the proposed architecture is depicted in Fig. 1. The proposed method is based on *four* functional blocks: (a bank of) estimators (**B1**), (a bank of) predictors (**B2**), a residual calculator (**B3**) and a classifier (**B4**). Each block is described in what follows.

A. Estimators (B1) and Predictors (B2)

The two banks of estimators and predictors aim at modeling sensors within the system. According to Fig. 1, both these banks are equipped with K independent estimators and predictors, respectively. Herein, K denotes the number of faulty sensors. Each *estimator module* provides an estimation $\hat{x}_s[n]$ of the measurement of its corresponding sensor s at current time step n. Each sensor estimator receives $\boldsymbol{x}_{(s)}$ as input, i.e. the vector of all existing sensor readings except the one from the sensor under estimation s using a sliding window mechanism (from L_e previous time steps up to the current time step n).

Since each estimator module is not utilizing its corresponding sensor readings, *predictor modules* are there to play a complementary role. Specifically, each of the K predictors produces a prediction $\tilde{x}_s[n]$ of its corresponding sensor s at current time step n by receiving the readings x_s as input, i.e. readings from its corresponding (under prediction) sensor s using a sliding window mechanism (from L_p previous time steps up to time step $n-1$).

B. Residual Calculator (B3)

This block calculates the residual signals of estimation and prediction (namely $e_{e,s}[n]$ and $e_{p,s}[n]$) for each faulty sensor $s = 1, \ldots, K$ as the squared difference of sensors readings and their respective estimation and prediction values i.e.

$$e_{e,s}[n] = (x_s[n] - \hat{x}_s[n])^2, \tag{1}$$
$$e_{p,s}[n] = (x_s[n] - \tilde{x}_s[n])^2. \tag{2}$$

Residual signals capture the dissimilarity and incongruity between sensors readings vs. estimated and predicted values. Residual calculator plays a data pre-processing role for the classifier block by providing interpretable and parsed input.

C. Sensor-Fault Classifier (B4)

In the *classifier block*, a single MLP classifier is exerted to detect and identify faulty sensors in a real-time manner. Denoting $\boldsymbol{e}[n] = (e_{e,1}[n], \ldots, e_{e,K}[n], e_{p,1}[n], \ldots, e_{p,K}[n])^T$ the residual vector containing the residual signals of all K sensors at time step n, the input of the classifier is the collection of residual vectors from L_c previous time steps up to current time step n, namely $\boldsymbol{e}[n], \ldots, \boldsymbol{e}[n - L_c]$. A (soft-)decision vector $\boldsymbol{d}[n] = (d_1[n], d_2[n], \ldots, d_K[n])^T$ represents the classifier output, where $d_i[n] \in [0,1]$, $i = 1, \ldots, K$ indicates the pseudo-probability (viz. confidence) for the i-th sensor being faulty. Specifically, a decision element $\{d_i(n) = 0\}$ indicates the highest confidence on sensor i being fault-free, whereas a decision element $\{d_i(n) = 1\}$ corresponds to the highest confidence on the faulty behaviour for the considered sensor. As a consequence, a vector $\boldsymbol{d}[n]$ with all elements set to 0 indicates healthy operation of all sensors within the system. Consequently, herein a faulty sensor is detected and identified/isolated when the entries of the decision vector $\boldsymbol{d}[n]$ exceed a predefined threshold γ. Specifically, $\max_{i=1}^{K} d_i[n] \gtrless \gamma$ is used for *detection*, whereas (upon detection) $\hat{k} = \arg\max_{i=1}^{K} d_i[n]$ is used for *identification*.

Ultimately, isolated faulty sensors are accommodated with the corresponding estimates from the *estimators block* to preserve system performance. Although the proposed SFDIA architecture can diagnose simultaneous faults of multiple sensor (by a slight modification of the identification logic), this issue is left to future work.

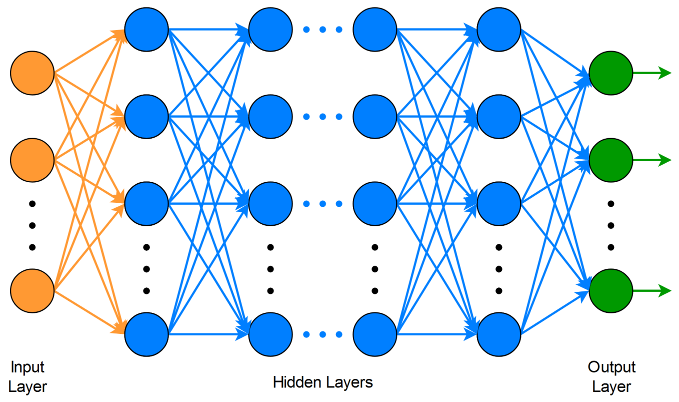


Figure 2: The generic structure of MLP.

III. MLP NNs and Dataset Setup

A. MLP NNs Setup

MLPs are feed-forward NNs capable to learn a function $\boldsymbol{h}(\cdot) : \mathbb{R}^l \rightarrow \mathbb{R}^m$ by means of a set of known labeled training samples, where l is the input dimension and m is the output dimension, which are broadly used for regression and classification tasks [18]. As shown in Fig. 2, MLPs are made of an input layer, one or more hidden layers and one output layer. Each neuron at the generic hidden/output layer executes a biased weighted sum of its inputs and processes the obtained value with an activation function to produce the output value. MLP NNs with appropriate number of hidden layers and number of neurons per hidden layer can model functions of arbitrary complexity with sufficient accuracy. In the following, we provide details about each MLP NN configuration employed in the proposed architecture.

1) Estimators and Predictors: The considered MLP-based *estimators* are made of $(L_e+1)(K-1)$ input nodes, one single hidden layer with N_v hidden neurons, and one single output node. MLP-based *predictors* are made of L_p input nodes and a similar structure as the MLP-based estimators. Moreover, the hyperbolic tangent (Tanh) function is used as the activation function $f(\cdot)$ of the hidden layers, i.e.

$$f(z) = [\exp(z) - \exp(-z)] \;/\; [\exp(z) + \exp(-z)]\,, \quad (3)$$

where z is the biased weighed sum of inputs to a neuron. Differently, a linear activation function is used for the output layer in *all* MLP-based estimators and predictors. Training is done using the Nesterov-accelerated adaptive moment estimation (Nadam) [19] optimization algorithm with the mean square error (MSE) loss function.

2) Classifier: The MLP-based *classifier* is made of $2K(L_c+1)$ input nodes, two hidden layers with N_c hidden neurons per hidden layer, and K output nodes. The Tanh activation function is used for the hidden layers, and a logistic (sigmoid) activation function $g(\cdot)$ is used for each neuron of the output layer, i.e.

$$g(z) = 1 \,/\, [1 + \exp(-z)] \;, \quad (4)$$

Table I: Architecture Parameters

Parameter	**Estimator**	**Predictor**	**Classifier**
No. of input nodes	33	10	88
No. of hidden layers	1	1	2
No. of nodes per hidden layer	10	10	15
No. of output nodes	1	1	4
Hidden layers activation	Tanh	Tanh	Tanh
Output activation	Linear	Linear	Sigmoid
Optimization algorithm	Nadam	Nadam	Nadam
Loss function	MSE	MSE	BCE

The Nadam optimization algorithm is employed for training the classifier based on a loss capitalizing *multitask learning*. Indeed, given the multitask nature of the employed architectures, the loss function to be minimized depends on the specific parameters of the K binary fault-classification tasks. Accordingly, we aim to minimize a *weighted sum* of the losses of the K classification tasks considered, namely:

$$\mathcal{L}\,(\cdot) \triangleq \sum_{k=1}^{K} \lambda_k\, \mathcal{L}_k(\cdot) \quad (5)$$

with the usual binary cross-entropy (BCE) loss function used for *all* the K binary tasks $\mathcal{L}_1(\cdot), \ldots, \mathcal{L}_K(\cdot)$. Since our classifier is in charge of solving multiple learning tasks at once, the weight λ_k represents the preference level of the k^{th} task in the multitask objective function to be optimized. For simplicity, in this work, we use (simply) uniform weighting, i.e. $\lambda_k = 1/K$ for $k = 1, \ldots, K$.

B. WSN Dataset

The proposed method is evaluated using a real-world publicly-available WSN dataset generated at the University of North Carolina [17]. More specifically, we considered four sensors ($K = 4$): *two* indoor and *two* outdoor sensor nodes. Each sensor is twofold and measures both humidity and temperature for the time duration of six hours. Also, the original dataset includes some anomalies which we have discarded in our study in order to superimpose synthetically-generated faults and perform a statistical analysis. Only temperature measurements are considered in this study.

IV. Numerical Results

In this section, numerical performance on the WSN dataset of the proposed SFDIA architecture are presented and compared with those of the M-SFDIA proposed in our previous work [5]. Our analysis is carried out by dividing the dataset into a training set accounting for 85% of the samples and a test set made of the remaining 15%. Also, 15% of the training set is held out for validation purposes to avoid over-fitting of the MLP NNs. Samples of each sensor in the dataset are normalized to the range $[0, 1]$ using *min-max scaling*, i.e.

$$x_s' = (x_s - x_{\min}) \,/\, (x_{\max} - x_{\min}), \quad (6)$$

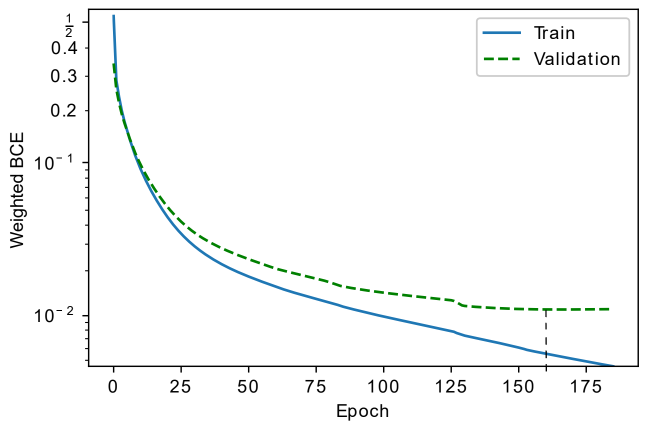


Figure 3: Weighted BCE loss of the classifier (cf. Eq. (5)) for training and validation sets during the training phase.

where $x_{\max}$ and $x_{\min}$ are the minimum and maximum readings in the training set for a given sensor s and x'_s represents the normalized reading of the sensor s.

Synthetic *bias* faults are generated and added to the WSN dataset in order to validate the proposed architecture performance. In order to avoid NNs from learning specific bias levels and/or duration of the generated faults, the modulus (resp. the sign) of the bias level has been generated as $|b| \sim \mathcal{U}(0.2, 0.4)$ (resp. $\text{sign}(b) \sim \mathcal{B}(0.5)$). Finally, the bias duration has been generated as $M \sim \mathcal{U}_d(3, 7)$. Then, a bias b is injected to the normal operation data-sets for M consecutive samples as

$$x'_{s,b}[n] = \begin{cases} x'_s[n] + b \,, & 0 \leq n - m < M \\ x'_s[n] \,, & \text{otherwise} \end{cases} \tag{7}$$

where $x'_{s,b}[n]$ is reading of sensor s with possible bias faults and m denotes the starting time instant of the fault. The performance analysis of the proposed architecture on different fault typologies (e.g. *drift* faults) is left to future work.

We considered $N_v = 10$ nodes per hidden layer and sliding windows with size $L_e = L_p = 10$ for all the estimators and the predictors within the architecture, while the classifier was implemented with $N_c = 15$ nodes per hidden layer and a sliding window with size $L_c = 10$. The parameters of the proposed architecture are summarized in Tab. I. For a fair comparison, the same parameters have been chosen for the M-SFDIA architecture [5].

Fig. 3 shows the trend of the weighted BCE loss for both training and validation sets during the training phase of the MLP-based classifier. Apparently, the validation loss settles after $\approx$ 160 epochs (as highlighted in the plot), while the training loss keeps decreasing for successive epochs. *Early-stopping* mechanism [20] was used to stop the training phase at this point and avoid over-fitting. Trends for the MSE loss on training/validation sets during the training phase of the MLP-based estimators and predictors resemble those shown for the classifier and are omitted for brevity.

First, in Fig. 4 the temporal behavior of the fault detection process is visualized over a portion of the test set. The proposed architecture provides better detection performance compared to the M-SFDIA due to a complete exploitation of the spatio-temporal correlation within the sensor data. Indeed, it is apparent how the M-SFDIA architecture exhibits missed detection of several faults for a given probability of false alarm, while the proposed architecture performs much better.

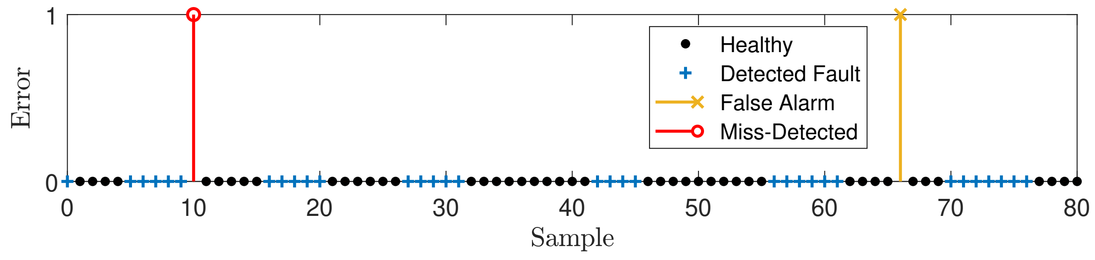
Fig. 5 illustrates detection and classification (i.e. detection plus isolation) performance by means of the corresponding ROC curves.[1] More specifically, the results show a clear performance improvement achieved by the proposed architecture w.r.t. the M-SFDIA architecture for *both* (i) *detection* and (ii) *classification* tasks. Regarding the former, the probability of detection for the M-SFDIA (resp. proposed) architecture approaches a value of ≈ 0.93 (≈ 0.98). The above results are obtained by setting the false-alarm probability to $P_f = 10^{-2}$. Conversely, regarding the classification task (under the same false-alarm constraint), the M-SFDIA (resp. the proposed) architecture achieves a probability of correct classification close to 0.90 (resp. 0.98). The above results highlight ideal identification performance for our approach, i.e. no additional errors caused by identifying the correct source of fault.

Fig. 6 focuses on a snapshots for visual comparison of the accommodated output of both architectures for probability of false alarm of $P_f = 10^{-2}$ together with healthy and faulty measurements. Again, the proposed method successfully accommodates more faulty data, and presents greater accommodation performance. As a wrap-up, the PDFs of the error signals (i.e. the difference between the accommodated signals and the healthy signals) over the test set are shown in Fig. 7, for $P_f = 10^{-1}$ (top) and $P_f = 10^{-2}$ (bottom). Though both architectures use the *same estimation outputs* obtained by the MLP-based estimators to accommodate the detected faulty measurements, the proposed architecture provides better final accommodation performance. This is generally due to the higher detection and correct classification rates, which reflect into a larger number of faulty measurements replaced with corresponding reliable estimates.

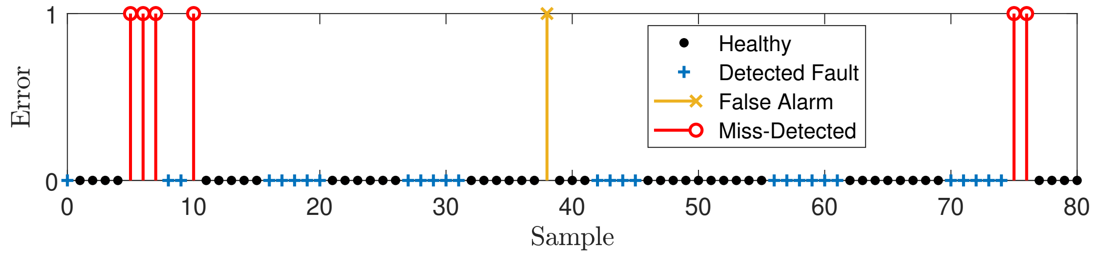
V. Conclusions and Future Directions

In this paper, we introduced SFDIA architecture based on machine-learning methods to empower design of DTs. We compared the performance of our novel architecture with a state-of-the-art M-SFDIA architecture using a real-world publicly-available WSN dataset. Unlike the M-SFDIA architecture, the proposed architecture utilizes the entire spatio-temporal correlations by introducing a block of MLP-based

[1]Receiver operating characteristics (ROC) curves show the trade-off between the *probability of detection* (resp. *probability of correct classification*) and the *probability of false alarm* by varying γ, when assessing detection (resp. identification) performance. In detail, the *probability of detection* refers to the proportion of faulty samples that are correctly detected (i.e. true-positive rate), while the *probability of false alarm* refers to the proportion of healthy samples that are incorrectly identified as faulty (i.e. false-positive rate). Finally, the *probability of correct classification* considers a correct event if the detected fault is associated to the actual sensor undergoing failure.

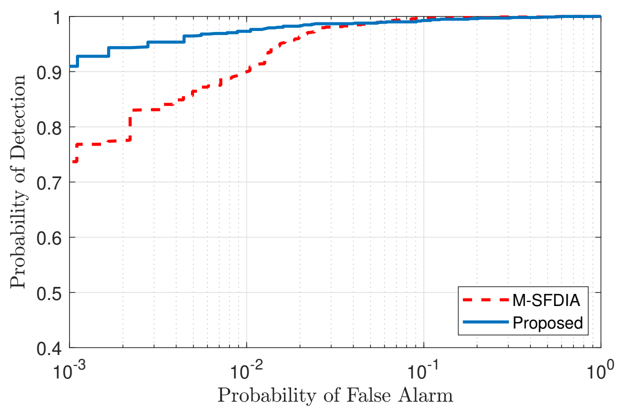


(a) Proposed

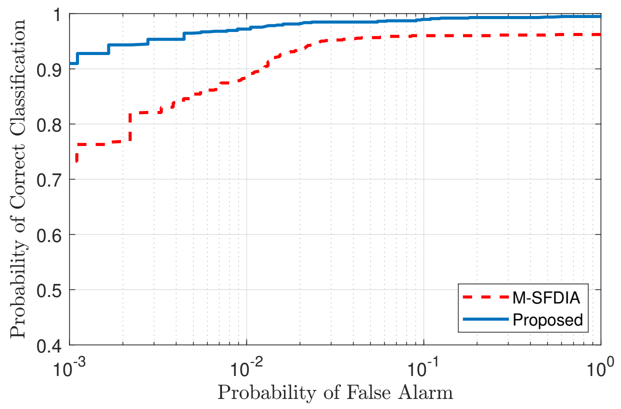


(b) M-SFDIA

Figure 4: A snapshot of the test set for false alarm rate of $P_f = 10^{-2}$.



(a) Detection Performance



(b) Classification Performance

Figure 5: Detection performance and averaged correct classification performance of both architectures by using ROC curves.

predictors. Estimators in both architectures exploit other sensors data to estimate corresponding sensor reading, while predictors in the proposed architecture play a complementary role by using previous data of the corresponding sensors and exploit better the available temporal correlation. Numerical results showed that the proposed architecture achieves better performance in term of probability of detection and probability of correct classification for fixed probability of false alarm. Moreover, the proposed architecture yields inferior accommodation error than the M-SFDIA architecture. In future works, we plan to exploit the classifier decisions to avoid fault propagation into the proposed SFDIA architecture. Future directions of research will include: (a) design of DTs which are robust to communication channel uncertainties, (b) the usage of explainable artificial intelligence techniques to interpret (and improve) the proposed SFDIA approach and (c) the capitalization of multimodal techniques for improved estimators' design.

References

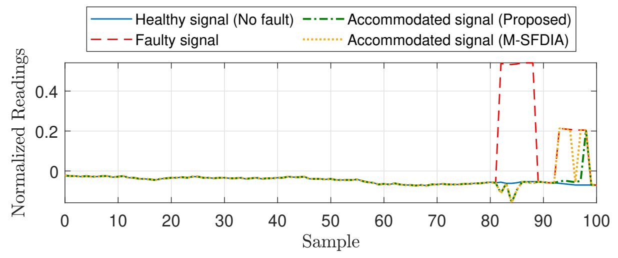
[1] Z. Yang, N. Meratnia, and P. Havinga, "An online outlier detection technique for wireless sensor networks using unsupervised quarter-sphere support vector machine," in *IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2008, pp. 151–156.

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

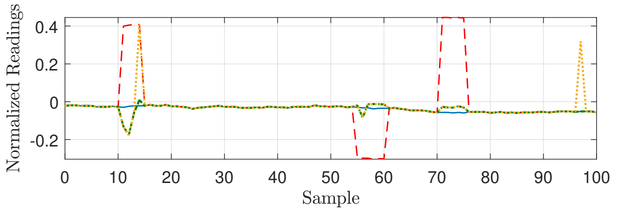
[3] N. R. Prasad and M. Alam, "Security framework for wireless sensor networks," *Wireless Personal Communications*, vol. 37, no. 3-4, pp. 455–469, 2006.

[4] 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 of Things Journal*, vol. 7, no. 9, pp. 8462–8471, 2020.

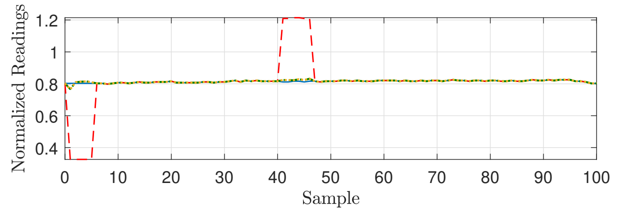
[5] H. Darvishi, D. Ciuonzo, E. R. Eide, and P. Salvo Rossi, "Sensor-fault detection, isolation and accommodation for digital twins via modular



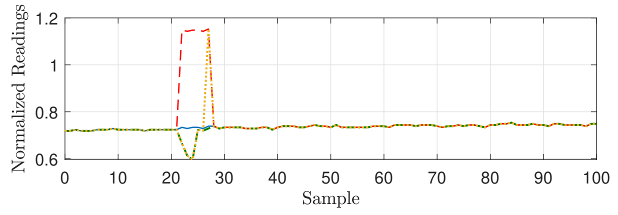
(a) Sensor 1



(b) Sensor 2

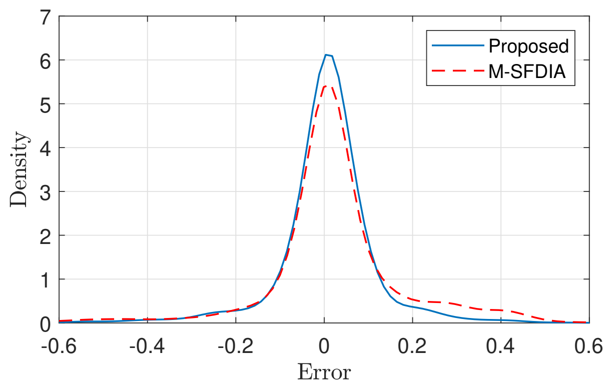


(c) Sensor 3

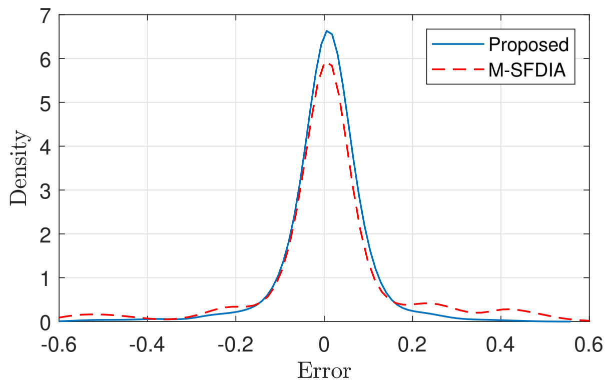


(d) Sensor 4

Figure 6: Visualization of accommodation performance vs. time.



(a) $P_f = 10^{-1}$



(b) $P_f = 10^{-2}$

Figure 7: Accommodation performance comparison in terms of PDF of the error signals for two different probabilities of false alarm.

data-driven architecture," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4827–4838, February 2021.

[6] P. M. Papadopoulos, L. Hadjidemetriou, E. Kyriakides, and M. M. Polycarpou, "Robust fault detection, isolation, and accommodation of current sensors in grid side converters," *IEEE Transactions on Industry Applications*, vol. 53, no. 3, pp. 2852–2861, 2017.

[7] S. K. Kommuri, S. B. Lee, and K. C. Veluvolu, "Robust sensors-fault-tolerance with sliding mode estimation and control for PMSM drives," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 17–28, 2018.

[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 Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4285–4292, 2012.

[10] 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.

[11] Y. Wang, N. Masoud, and A. Khojandi, "Real-time sensor anomaly detection and recovery in connected automated vehicle sensors," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

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

[13] W. Yu, T. Dillon, F. Mostafa, W. Rahayu, and Y. Liu, "A global manufacturing big data ecosystem for fault detection in predictive maintenance," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 183–192, 2020.

[14] 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 Conference*, 2020, pp. 1–4.

[15] S. Hussain, M. Mokhtar, and J. M. Howe, "Sensor failure detection, identification, and accommodation using fully connected cascade neural network," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1683–1692, 2015.

[16] G. Campa, M. Thiagarajan, M. Krishnamurty, M. R. Napolitano, and M. Gautam, "A neural network based sensor validation scheme for heavy-duty diesel engines," *ASME Journal of dynamic systems, measurement, and control*, vol. 130, no. 2, 2008.

[17] S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie, and M. Palaniswami, "Labelled data collection for anomaly detection in wireless sensor networks," in *6th IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2010, pp. 269–274.

[18] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, no. 5-6, pp. 183–197, 1991.

[19] T. Dozat, "Incorporating Nesterov momentum into Adam," in *International Conference on Learning Representations (ICLR)*, 2016.

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.