Doctoral theses at NTNU, 2023:151

Hossein Darvishi

Machine-Learning Sensor Validation for Digital Twins

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical
Engineering
Department of Electronic Systems

NTNU
Norwegian University of
Science and Technology

Hossein Darvishi

Machine-Learning Sensor Validation for Digital Twins

Thesis for the Degree of Philosophiae Doctor

Trondheim, June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

ISBN 978-82-326-7007-9 (printed ver.)
ISBN 978-82-326-7006-2 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

IMT-report 2023:151

Doctoral theses at NTNU, 2023:151

Printed by NTNU Grafisk senter

Dedicated to
My Beloved Parents

Abstract

The rapid growth of digital twins (DTs), built upon Internet of Things (IoT) and Industrial IoT systems, demands a large variety of networked sensors' solutions. Indeed, networked sensors enable various sophisticated applications of DT by gathering/integrating sensor data, meanwhile, sensor failures can potentially undermine DT representativeness and cause serious consequences. In this thesis, we propose three generic sensor fault detection, isolation and accommodation (SFDIA) architectures capable of promptly detecting sensor failures, identifying faulty sensors, and replacing their faulty data with reliable estimations.

More specifically, the first modular architecture is built upon a series of neural network (NN) estimators and a classifier, which allows the selection of the most suitable models among diverse NN models with respect to the application. Estimators correspond to virtual sensors of all unreliable sensors (to reconstruct normal behavior and replace the isolated faulty sensor within the system), whereas the classifier is used for detection and isolation tasks.

This architecture is enhanced further to fully exploit the spatio-temporal correlation of sensor data and provide real-time detection, isolation and accommodation of multiple faulty sensors. A multi-dimensional classifier in the enhanced architecture is responsible for interpreting residual signals (from previous stages) to detect and identify faulty sensors, and provide feedback to a controller block. The controller is policing inputs-outputs of two banks of NNs which are providing estimations and predictions of all unreliable sensors within the system, thus supporting nearly-instantaneous SFDIA performance.

In the third proposed architecture, for the first time, we address the problem of SFDIA in large-size networked systems. Current available machine-learning solutions are either based on shallow networks unable to capture complex features from input graph data or on deep networks with overshooting complexity in the case of large number of sensors. To overcome these challenges, we propose a new framework for sensor validation based on a deep recurrent graph convolutional architecture (DRGCA) which jointly learns a graph structure and models spatio-temporal inter-dependencies. the proposed two-block DRGCA (i) constructs the virtual sensors in the first block to refurbish anomalous (i.e. faulty) behavior of unreliable sensors and to accommodate the isolated faulty sensors and (ii) performs the detection and isolation tasks in the second block by means of a classifier.

A detailed performance evaluation on different real-world datasets is conducted. Results prove the effectiveness of the proposed architectures in detection, isolation and accommodation of faults. Performance comparison shows their superiority over state-of-the-art machine-learning-based architectures.

Preface

The thesis is submitted to the Norwegian University of Science and Technology (NTNU) for the partial fulfillment of the requirements for the degree of Doctor of Philosophy.

The most of the doctoral work has been performed at the Department of Electronic Systems, NTNU, Trondheim, Norway. The work has been conducted under the supervision of Professor Pierkuigi Salvo Rossi at the Department of Electronic Systems, NTNU, and co-supervision of Professor Stefan Werner at the Department of Electronic Systems, NTNU, and Assistant Professor Domenico Ciuonzo at the Department of Electrical Engineering and Information Technologies, University of Naples "Federico II", from January 2020 to March 2023.

From June 2022 to January 2023, I was with the Signal Processing Laboratory (LTS4) in the Electrical Engineering Institute of the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, under the supervision of Professor Pascal Frossard during my visit.

The members of the assessment committee are: Professor Mikael Gidlund at the Department of Computer and Electrical Engineering (DET), Mid Sweden University; Associate Professor Sabita Maharjan at the Department of Informatics, University of Oslo; and Professor Kimmo Kansanen at the Department of Electronic Systems, NTNU.

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

Acknowledgements

When I look back, it certainly feels like it was a few months ago when I started my PhD at the Norwegian University of Science and Technology (NTNU) on the 10th of January 2020. Beyond doubt, PhD was the hardest, most rewarding, confidence-boosting, and uplifting challenge I have ever faced, but meeting kind and supportive people made it an enjoyable experience along the way. I am grateful to the Lord, who gave me the powers and abilities to complete this work.

I wish to especially thank my supervisor Pierluigi Salvo Rossi, who has been a great mentor, a sharp brain and invaluable support throughout the PhD. He acted as both researcher and friend in supporting my research and life. I would like to extend my thanks to my co-supervisor Domenico Ciuonzo for his knowledge, constant support, friendship, and scientific collaboration on most of my academic works included in this thesis.

I am thankful to Professor Pascal Frossard for hosting my research stay and the opportunity to be part of his professional and friendly group during a visit to the Signal Processing Laboratory (LTS4) in the Electrical Engineering Institute of the Swiss Federal Institute of Technology (EPFL), Switzerland. I also thank all the friendly people from the LTS4 and the other departments at EPFL, especially Anne, Abdellah, Ali (Gilani), Ali (Hariri), Ahmet, Caroline, William, Marcele, Yamin and Arun. I would also like to thank Professor Antonio Ortega at the University of Southern California, US, for useful discussions during the development of my PhD.

Working at NTNU was very cheerful and vibrant, due to the nice friends, colleagues and faculty staff. I would like to thank them all, specially Asaad, Mohammed Ayalew, Pouria, Naseh, Vinay, Reza (Mirzaeifard), Reza (Noormohammadi), Mohammad, Ashkan, Ehsan, Farrokh, Fazel, Roghayeh, Gianluca, Ali, Stefan, Naveen, Giampiero, Kimmo, Ragnar, Randi, Nina, Eric and Lars Arne.

Finally, I owe my deepest appreciation to my family who always believed in me and have provided unconditional support for my PhD work.

Contents

List of Tables

List of Figures

Abbreviations and Symbols

Abbreviations

IoT	Internet of Things
DT	Digital twin
SFDIA	Sensor fault detection, isolation and accommodation
ML	Machine learning
NN	Neural network
M-SFDIA	Modular SFDIA
CCS	Carbon capture and storage
MLP	Multi-layer perceptron
GCN	Graph convolutional network
MAE	Mean absolute error
RMSE	Root mean square error
MAPE	Mean absolute percentage error
ARX	Auto-regressive models with exogenous inputs
RF	Random forest
SVM	Support vector machines
WSN	Wireless sensor networks
AE	Autoencoder
FCC	Fully-connected cascade
RNN	Recurrent neural network
ReLU	Rectified linear unit
CNN	Convolutional neural network
LSTM	Long-short term memory
GRU	Gated recurrent unit
DRGCA	Deep recurrent graph convolutional architecture
AQ	Air quality

CO	Carbon monoxide
NMH	Non-metanic hydrocarbon
NO_x	Nitrogen oxides
NO_2	Nitrogen dioxide
O_3	Ozone
CO_2	Carbon dioxide
PMSM	Permanent magnet synchronous motor
CalTrans	California department of transportation
PeMS	Performance measurement system
Nadam	Nesterov-accelerated adaptive moment estimation
Adam	Adaptive moment estimation
Tanh	Hyperbolic tangent
MSE	Mean square error
ROC	Receiver operating characteristic
PDF	Probability density function
GNN	Graph neural network
GDN	Graph deviation network
AGCRN	Adaptive graph convolutional recurrent network
OM-SFDIA	Optimized M-SFDIA

Symbols

$\mathcal{G}$	Undirected graph
N	Total number of nodes/sensors
$\mathcal{V}$	Finite set of N nodes (i.e. sensors)
$\mathcal{E}$	Set of edges
$\boldsymbol{A}$	Adjacency matrix
$\boldsymbol{L}$	Graph Laplacian
$\boldsymbol{D}$	Diagonal degree matrix
$\boldsymbol{L}_{\mathcal{G}}$	Normalized graph Laplacian
$\{\boldsymbol{u}_i\}_{i=1}^{N}$	Set of orthonormal eigenvectors
$\{\lambda_i\}_{i=1}^{N}$	Eigenvalues
$\boldsymbol{U}$	Fourier basis
$\boldsymbol{\Lambda}$	Diagonal eigenvalues matrix
$\mathcal{F}$	Fourier graph transform
$\boldsymbol{x}$	Signal
$\boldsymbol{\theta}$	Graph parameter
$g_{\boldsymbol{\theta}}$	Graph filter
$\hat{g}_{\boldsymbol{\theta}}$	Spectral graph filter
$\mathcal{K}$	Chebyshev order
$\{T_k(x)\}_{k=0}^{\mathcal{K}}$	Chebyshev polynomials
$\widetilde{\boldsymbol{\Lambda}}$	Rescaled eigenvalues matrix
$\lambda_{\max}$	Largest eigenvalue
θ'_k	kth Chebyshev coefficient
$\widetilde{\boldsymbol{L}}$	Rescaled normalized Laplacian
$\widetilde{\boldsymbol{A}}$	Rescaled adjacency matrix
$\widetilde{\boldsymbol{D}}$	Rescaled degree matrix
$\boldsymbol{X}$	Graph signal
F	Number of graph filters
$\boldsymbol{\Theta}$	Learnable-filter parameter matrix (i.e. the GCN weight matrix)
$\mathrm{GCN}_{\boldsymbol{\Theta}}$	GCN function
$\boldsymbol{b}$	Additive constant vector
m	Starting time instant of the fault
C	Number of features per sensor
$\boldsymbol{x}_n[k]$	Normalized healthy readings at time step k
$\boldsymbol{x}_n^b[k]$	Normalized possibly bias-faulty readings at time step k
$\boldsymbol{x}_n^g[k]$	Normalized possibly noise-faulty readings at time step k
$\boldsymbol{x}_n^f[k]$	Normalized possibly freeze-faulty readings at time step k
$\boldsymbol{x}_n^d[k]$	Normalized possibly drift-faulty readings at time step k

M	Fault length parameter
K	Fault length parameter
$\boldsymbol{w}[k]$	Zero-mean additive Gaussian noise vector at time step k
σ_g^2	Noise variance
s_b	Auxiliary parameter
a_b	Auxiliary parameter
$\mathcal{S}_U$	Set of unreliable sensors
$\mathcal{S}_R$	Set of reliable sensors
N_U	Number of unreliable sensors
N_R	Number of reliable sensors
L_v	Size of sliding window for the estimators
L_c	Size of sliding window for the classifier
$y_s[n]$	Estimator output of sensor $s \in \mathcal{S}_U$ at time step n
$x_s[n]$	Sensor $s \in \mathcal{S}_U$ measurement
$e_s[n]$	Residual measurement of sensor $s \in \mathcal{S}_U$ at time step n
$\boldsymbol{d}[n]$	Decision vector at time step n
$d_i[n]$	Element of the decision vector, corresponding to ith sensor
γ	Decision threshold
$f_s(\cdot)$	Function model of the sth sensor estimator
$\boldsymbol{e}_U[n]$	Vector of the dissimilarity measurements of the unreliable set at time instant n
$\boldsymbol{x}_{U,s}[n]$	Vector of the unreliable set measurements except sensor s at time instant n
$\boldsymbol{x}_R[n]$	Vector of the reliable set measurements at time instant n
$g(\cdot)$	Function model of classifier for M-SFDIA architecture
H_v	Number of hidden layers for the estimator models
N_v	Number of hidden nodes per layer for the estimator models
H_c	Number of hidden layers for the classifier model
N_c	Number of hidden nodes per layer for the classifier model
H	General number of hidden layers
N_g	General number of hidden nodes per layer
L	General size of sliding window
P_d	Probability of detection
P_f	Probability of false alarm
$\boldsymbol{x}_{(s)}$	Vector of the unreliable set measurements except sensor s at time instant n
$\hat{x}_s[n]$	Estimation of the sensor s measurement at time step n
$\tilde{x}_s[n]$	Prediction of the sensor s measurement at time step n
L_e	Size of sliding window for the estimators
L_p	Size of sliding window for the predictors

$g_s(\cdot)$	Function model of the sth sensor predictor
$e_{E,s}[n]$	Difference of the sth sensor reading with their respective estimation at time step n
$e_{P,s}[n]$	Difference of the sth sensor reading with their respective prediction at time step n
$\boldsymbol{e}_U[n]$	Residual vector containing the residual signals of all N_U sensors at time step n
$\boldsymbol{h}(\cdot)$	Function model of the classifier
$\mathcal{I}_U$	Set of identified faulty sensors
$\phi_{E,s}$	Average residual signal for the sth estimator
$\phi_{P,s}$	Average residual signal for the sth predictor
τ	System tolerable level of deviation
υ	controller threshold
$e_{AE,s}[n]$	AE squared error for the sth unreliable sensor at time n
σ	AE decision threshold
F_R	Fault rate
$\boldsymbol{x}_n[k]$	measured parameters of nth sensor at time k
$\boldsymbol{X}[k]$	Matrix of all the N nodes recordings at time k
$\hat{\boldsymbol{X}}[k]$	Estimates of the present readings at time k
$\boldsymbol{f}_{\varsigma}(\cdot)$	Function model of the NN-based estimator
$\boldsymbol{\varsigma}$	Trainable parameters of function $\boldsymbol{f}_{\varsigma}(\cdot)$
$\boldsymbol{\Delta}[k]$	Absolute difference between sensors reading and their associated virtual reading at time k
$\boldsymbol{h}_{\boldsymbol{\vartheta}}(\cdot)$	Function model of the NN-based classifier
$\boldsymbol{\vartheta}$	Trainable parameters of function $\boldsymbol{h}_{\boldsymbol{\vartheta}}(\cdot)$
M_c	Size of sliding window for the classifier
M_e	Size of sliding window for the estimators
$\boldsymbol{E}_g$	Node embedding matrix
l	Embedding dimension
$\boldsymbol{W}_g$	Weight pool tensor
$\boldsymbol{B}$	Additive learnable bias matrix
$\boldsymbol{B}_g$	Bias pool matrix
$\boldsymbol{E}_a$	Randomly-initialized embedding matrix
$\boldsymbol{E}_a$	Trainable node embedding matrix of the AGCRN
$\boldsymbol{W}_z$	Trainable parameter of the AGCRN
$\boldsymbol{W}_r$	Trainable parameter of the AGCRN
$\boldsymbol{W}_{\hat{h}}$	Trainable parameter of the AGCRN
$\boldsymbol{B}_z$	Trainable parameter of the AGCRN
$\boldsymbol{B}_r$	Trainable parameter of the AGCRN
$\boldsymbol{B}_{\hat{h}}$	Trainable parameter of the AGCRN

$\boldsymbol{Z}[\cdot]$	Update gate
$\boldsymbol{R}[\cdot]$	Reset gate
$\hat{\boldsymbol{H}}[\cdot]$	Candidate activation matrix
$\hat{\boldsymbol{A}}$	Pseudo-Laplacian of graph $\mathcal{G}$
$\boldsymbol{H}[.]$	State matrix
$\boldsymbol{H}_i[.]$	State matrix of ith AGCRN layer
w	Number of samples in each batch
$\mathcal{L}_{\text{est}}(\cdot)$	MAE loss function of DRGCA estimator
$\boldsymbol{X}_j$	Fault-free readings of sensors for j sample in a batch
$\hat{\boldsymbol{X}}_j(\varsigma)$	Estimation of sensors readings for j sample in a batch
$\tanh(\cdot)$	Tanh activation function
$\sigma(\cdot)$	Sigmoid activation function
$\mathcal{L}_{\text{cl}}(\cdot)$	DRGCA classifier loss function
$\boldsymbol{\vartheta}_{\text{shared}}$	Shared trainable parameters of function $\boldsymbol{h}_{\boldsymbol{\vartheta}}(\cdot)$
$\boldsymbol{\vartheta}_n$	Trainable parameters of the nth learning task of function $\boldsymbol{h}_{\boldsymbol{\vartheta}}(\cdot)$
ρ_n	Preference level of the nth learning task
$\mathcal{L}_n^{\text{BCE}}(\cdot)$	Binary cross-entropy loss function of the nth learning task
y_n^j	$0/1$ representation of the true (i.e. labeled) fault status of nth sensor for j sample in a batch
d_n^j	Entry of classifier output of nth sensor for j sample in a batch
$\mathcal{O}(\cdot)$	Landau notation
$\mathcal{U}(a,b)$	Uniform probability density function with support $[a,b]$
$\mathcal{U}_d(a,b)$	Discrete-uniform probability density function with support $\{a, a+1, \dots, b\}$
$\mathcal{N}(b,c)$	Gaussian probability density function with mean b and variance c
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate Gaussian probability density function with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\mathcal{B}(p)$	Bernoulli distribution with parameter p

Chapter 1

Introduction

Industry 4.0 identifies the current fourth industrial revolution, whose aim is an increased level of automation through the effective combination of the Internet of Things (IoT), cyber-physical systems and cloud computing technologies [1]. Under the umbrella of Industry 4.0, digital twins (DTs) have garnered striking interest over the last few years through the process of industry digital transformation [2]. DTs are designed to create a digital model of (complicated) assets or processes. Fundamentally, a DT can be defined as a digital profile that mirrors a physical object or process, i.e. the physical twin, and provides a bidirectional interaction between the physical and digital parts. Leveraging DTs, operators can simulate complex systems behaviour, test/predict asset changes in specific scenarios, and remotely control/monitor/steer systems. DTs have been widely employed in various sectors such as industry [3], health care [4] and smart cities [5, 6], where their capabilities to visualize and treat with a perpetual stream of real-time sensor data is enabling new opportunities.

While DTs are highly dependent on data collected by sensors, the latter are unfortunately prone to errors. Faulty data may lead to system instability and eventually jeopardize system reliability with possible noxious outcomes ranging from loss or critical damage to the asset (viz. financial and time loss) and/or environmental hazardous impact to serious injury to people or death in the worst case. Failure sources can be classified into three types [7, 8, 9, 10]:

- *Hardware/software failure* — the sensor itself is inaccurate or faulty due to its bad quality, being out of lifetime, bad calibration, and/or software failure;
- *Typical harsh environment/condition* — the system operates under a setting in which sensor survival is difficult and its performance deteriorates rapidly;

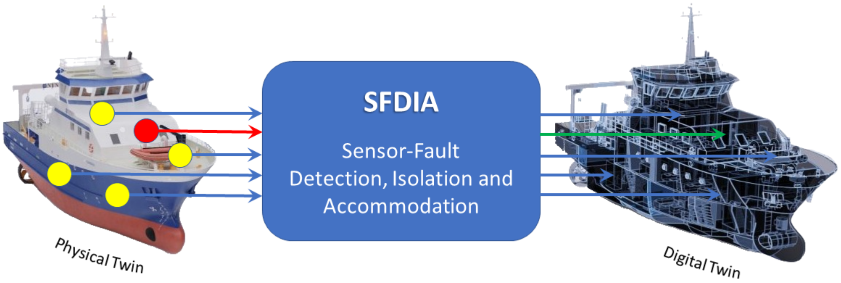


Figure 1.1: The SFDIA system.

- *Malicious cyber-attack* — an attempt is perpetrated to abuse or take advantage of the system functionality.

To ensure the successful rollout of a DT, it is crucial to continuously monitor and regulate received sensory data before feeding the DT with them.

From this perspective, in this thesis, we have tried to address sensors failure by the **Sensor Fault Detection, Isolation and Accommodation (SFDIA)** framework as a key response for deploying DTs while assuring reliable performance (see Fig. 1.1). SFDIA indeed consists of *three* parts:

- *Fault detection*, i.e. determining sensor fault(s) within the system's sensor network;
- *Fault isolation*, i.e. identifying specific faulty sensors and block their measurement feeding to DT;
- *Fault accommodation*, i.e. feeding DT with some other replaced trustworthy data.

We used deep neural networks (NNs) with novel architectures to improve the SFDIA performance in stationary scenarios.

There are several issues with the current research on the SFDIA problem. The conducted research in the literature mainly focuses on the development of SFDIA solutions for dependent DTs, where the framework is established for one specific application, while possible development of flexible frameworks can easily adapt to different scenarios/applications. Successful implementation of all three tasks of detection, isolation, and accommodation is one of the challenges in the SFDIA applications. Except for a few works, most research on the SFDIA problem is mostly conducted on fault detection [11, 12] and does not foresee all three tasks in their original formulation to insure real-time transfer of reliable data into DTs. Unlike previous, the explosion of the IoT in recent years and hundreds/thousands of sensors distributed all over the physical twin has raised the importance of scalable solutions for large-size networked systems to run possible simulations, what-if analysis, study specific scenarios and/or generate possible feed-backs/improvements on the physical twin. Equally important, the performance of current data-driven methods heavily depends on the dimensionality, and heterogeneity of the system. What's more, the computational complexity increases exponentially with the system network size and usually, this is paired with performance degradation and weak generalization.

1.1 Scope and Objective

In this study, we developed multiple robust and optimized SFDIA schemes to address the above issues of sensor failure in DT systems. In particular, this thesis is aiming at investigating signal processing and machine learning (ML) algorithms for *three tasks* of sensor fault detection, isolation and accommodation [13] in stationary situations to preserve the reliability and robustness of sensor-based systems. Accordingly, once a failure(s) has been detected/identified from the process loop it will be accommodated (replaced) with some other trusted data. We consider the three following objectives when designing SFDIA schemes to achieve reliable data transfer to DTs:

O1 Allowing flexible deployment of diversified ML techniques with a *modular* design; Enabling to adapt to various different applications and to better understand the commonalities and differences between ML modules, while abstracting from their specific technical details.

O2 Enabling integration of in-field and real-time raw data into DTs; Ensuring that the models process incoming data and generate decisions within narrow time windows.

O3 Strong generalization and scalability, i.e. capability to properly capture complex features within the data; The high dimensionality of datasets in big data applications leads to a more complex feature engineering.

1.2 Methodology

The thesis deals with state-of-the-art signal processing and ML algorithms (e.g. multi-layer perceptrons (MLPs) and graph NNs) and optimization tools to develop methods for SFDIA purposes possibly based on safety requirements and assess their performance on relevant use cases. We present motivations in line with the needs of the SFDIA literature, and the methods proposed in this thesis are compared to other state-of-the-art approaches. The developments, validation and analysis of the proposed methodology are conducted through numerical experiments using real-world and publicly available datasets, with applications in different scenarios, for the sake of a complete and reproducible assessment.

1.3 Thesis Contributions

This thesis proposes three ML frameworks for sensor validation to address objectives **O1**, **O2** and **O3**. In view of the previous discussion, some of the state-of-the-art methods are restricted to a given vertical domain (e.g. aircraft [14], vehicle [15] or HVAC system [16] monitoring), thus *lacking a general formulation*. Differently, proposed frameworks allow the development of general SFDIA schemes to be easily adapted to different application domains [3, 17]. Secondly, part of the literature evaluates corresponding proposals on *private* (e.g. [18, 19]) or *simulated* (e.g. [20, 16, 14]) measurement data, thus precluding reproducibility and convincing evaluation, respectively. The datasets and NNs in the proposed architectures are publicly-available, which helps reproducibility and further advances on the topic. In detail, the main *contributions* of this thesis are summarized as follows.

C1 Modularity and Real-Time Implementation: First, we presented modular SFDIA (M-SFDIA) scheme consist of a set of estimators (each associated to a sensor) providing residual signals as well as replacements (estimates) for faulty data. Therein a supervised classifier is trained to make detection & identification decisions upon the residual signals by leveraging their (possibly-nonlinear) relationships. Indeed, the proposed modular approach allows the implementation of diversified ML techniques for different modules and a more flexible deployment, also taking computational/hardware limitations into account, addressing both **O1** and **O2**. The contributions of Publications **P1**, **P2**, **P4** and **P6** can be summarized as follows.

- A novel machine-learning-based architecture for SFDIA is proposed. The proposed architecture jointly takes advantage of the temporal correlation of the measurements and of both reliable and unreliable sensors within the system to achieve a higher sensor validation performance.
- The performance of different NN-based virtual sensors and classifiers used within the M-SFDIA architecture are investigated and compared.

- The performance of the proposed approach (in terms of probabilities of detection, false alarm, correct classification, misclassification, etc.) is evaluated on different real-world datasets [21, 22, 23] corrupted with synthetically-generated sensor faults (bias and drifts) and compared with the state-of-the-art techniques [11, 24]. Synthetically-generated sensor faults have been considered to perform a systematic performance assessment of the proposed architecture. The focus of generated faults is on *weak faults*, which are very hard to detect and usually ignored in the literature [25, 26, 27, 28, 29].
- The designed M-SFDIA architecture for SFDIA in a Carbon capture and storage (CCS) system is discussed and evaluated.
- The impact of different hyperparameters, such as the number of layers and the number of nodes per layer, is assessed for the considered scenarios.

C2 Performance Improvement: The M-SFDIA architecture enhanced to fully explore (viz. learn) *spatial* and *temporal* dependence in sensory data and to directly address both **O1** and **O2** research objectives. Differently, the enhanced architecture relies on the novel use of a pair of regressors for each sensor performing estimation and prediction operations, along with a controlled feedback loop policing propagation of faults throughout the architecture. Hence, the joint adoption of regressors and the controlled fault propagation enables the proposed architecture to ultimately exploit spatio-temporal correlation within the system, thus supporting nearly-instantaneous fault detection and isolation performance. The contributions of Publications **P3** and **P5** can be summarized as follows.

- A *real-time* and *modular* data-driven SFDIA architecture is developed, fully exploring spatio-temporal correlation within the system. The proposed architecture consists of five building blocks (controller, estimators, predictors, residual calculator, classifier) arranged in *four layers*. Conversely, each predictor plays a complementary role (to the estimator) by using only previous data from the sensor under consideration to obtain an analogous virtual measurement.
- The proposed approach employs MLP NNs for both regression (estimation and prediction) and classification modules to capture and process analytical redundancy relations while keeping a *reasonable complexity* at the operational stage. In the latter case, a *multi-task* MLP NN (i.e. each sensor condition is seen as a binary classification task) is designed

for detecting and (if any) identifying multiple faulty sensors via a *single* NN.

- Moreover, classifier decisions, residual signals and virtual measurements are exploited by a *a specifically-designed controller* to make corrections on sensor models inputs and improve overall system performance both for detection and isolation tasks. Specifically, in a feedback loop, the controller is in charge of replacing corrupted input data and, consequently, avoiding propagation of faults throughout the architecture.

C3 **Scalability:** To address the dimensionality limitations (**O3**) of the state-of-the-art techniques over *large-scale IoT networks*, we propose a new framework for sensor validation based on a deep recurrent graph convolutional architecture (DRGCA) which jointly learns a graph structure of system and models spatio-temporal inter dependencies. More specifically, the proposed two-block architecture (i) constructs the virtual sensors in the first block to refurbish anomalous (i.e. faulty) behaviour of unreliable sensors and to accommodate the isolated faulty sensors and (ii) performs the detection and isolation tasks in the second block by means of a classifier. Accordingly, the contributions of the Publications **P7** can be summarized as follows.

- We present the use of an enhanced graph convolutional network (GCN), termed AGCRN, to model virtual sensors. The AGCRN captures close-grained spatio-temporal correlations in graph data based on the two modules and a recurrent design.
- To the best of our knowledge, this is the first attempt to propose the use of GCN-based design in the SFDIA framework. Our proposed DRGCA has capabilities for the detection, isolation and accommodation of unknown fault types without any pre-modifications.
- This is also the first attempt to address and successfully perform all three tasks of detection, isolation and accommodation of sensor faults within the SFDIA framework within the challenging scenario of large-scale IoT networks.
- The performance of the proposed approach in terms of mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE) and probabilities of detection, false alarm, and correct identification is evaluated on *publicly-available* datasets [30, 31].

1.3.1 List of Publications

The following works were conducted by the author of the dissertation in line with the research objectives presented in Section 1.1. These works are documented in papers **P1** to **P7** and comprise the contributions listed in Section 1.3. The list is composed of seven papers, of which six were published or accepted for publication, and one was submitted during the course of the Ph.D.

- **P1**: [32] H. Darvishi, D. Ciuonzo, E. R. Eide and P. Salvo Rossi, "A Data-Driven Architecture for Sensor Validation Based on Neural Networks," *2020 IEEE SENSORS*, 2020, pp. 1-4;
- **P2**: [33] H. Darvishi, D. Ciuonzo, E. R. Eide and P. Salvo Rossi, "Sensor-Fault Detection, Isolation and Accommodation for Digital Twins via Modular Data-Driven Architecture," in *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4827-4838, 2021;
- **P3**: [34] H. Darvishi, D. Ciuonzo and P. Salvo Rossi, "Real-Time Sensor Fault Detection, Isolation and Accommodation for Industrial Digital Twins," *2021 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 2021, pp. 1-6;
- **P4**: [35] H. Darvishi, D. Ciuonzo and P. Salvo Rossi, "Exploring a Modular Architecture for Sensor Validation in Digital Twins," *2022 IEEE Sensors*, 2022, pp. 1-4;
- **P5**: [36] H. Darvishi, D. Ciuonzo and P. Salvo Rossi, "A Machine-Learning Architecture for Sensor Fault Detection, Isolation and Accommodation in Digital Twins," in *IEEE Sensors Journal*, vol. 23, no. 3, pp. 2522-2538, 2023;
- **P6**: [37] A. Chawla, Y. Arellano, M. V. Johansson, H. Darvishi, K. Shaheen, M. Vitali, F. Finotti, P. Salvo Rossi, "IoT-based Monitoring in Carbon Capture and Storage Systems," in *IEEE Internet of Things Magazine*, vol. 5, no. 4, pp. 106-111, 2022;
- **P7**: [38] H. Darvishi, D. Ciuonzo and P. Salvo Rossi, "Deep Recurrent Graph Convolutional Architecture for Sensor Fault Detection, Isolation and Accommodation in Digital Twins," Submitted to *IEEE Sensors Journal*, 2023;

1.3.2 Papers Not Included in the Thesis

- **P8**: [39] M. Goodarzi, M. A. Sebt and H. Darvishi, "Target and Image Elevation Angles Separation Algorithm for Low-Angle Tracking with Monopulse Antenna," *2020 28th Iranian Conference on Electrical Engineering (ICEE)*, 2020, pp. 1-4;

- **P9**: [40] H. Darvishi, M. A. Sebt, D. Ciuonzo, and P. Salvo Rossi, “Tracking a Low-Angle Isolated Target via an Elevation-Angle Estimation Algorithm Based on Extended Kalman Filter with an Array Antenna,” *Remote Sensing*, vol. 13, no. 19, p. 3938, Oct. 2021;
- **P10**: [41] S. P. Talebi, H. Darvishi, S. Werner and P. Salvo Rossi, "Gradient-Descent Adaptive Filtering Using Gradient Adaptive Step-Size," *2022 IEEE 12th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2022, pp. 321-325;
- **P11**: [42] M. A. Sebt, M. Goodarzi and H. Darvishi, "Geometric Arithmetic Mean Method for Low Altitude Target Elevation Angle Tracking," in *IEEE Transactions on Aerospace and Electronic Systems*, 2023;

1.4 Thesis Outline

This thesis begins by providing the fundamentals of GCNs and the background necessary for the research developed presented here. Chapter 2 provides a literature review regarding the related work, the fundamentals of graph convolution and GCNs, and the description of the datasets and the framework for fault generation. Chapter 3 proposes three general SFDIA architectures and describes the different blocks of each SFDIA architecture for fault detection, isolation and accommodation in detail, and later highlights and compares the numerical performance for the datasets with different setups. Finally, Chapter 4 presents the conclusions and final remarks, and future works for the topics discussed in the thesis. In the second part of the thesis, the research articles which are the scientific contribution of the dissertation are presented.

Chapter 2

Background

This chapter reviews the background of SFDIA systems, followed by parts from the theoretical framework. Specifically, relevant literature, technical concepts, sensor faults description, and working datasets are introduced. Specifically, a literature review and a brief introduction of different deep neural architectures which are employed in this thesis are presented in Section 2.1. We formalize the concepts of graph convolution, and graph neural network in Section 2.2.4. Later on, sections 2.3, 2.4 and 2.5 present the framework for fault generation and provide a description of sensors classification and the datasets, respectively.

2.1 Background on SFDIA systems

In the last years, the main advancements in sensor fault diagnosis technology have relied on the milestone concept of *redundancy* which embraces a wide spectrum of design solutions, e.g. redundancy can be accomplished by either *hardware* or *analytical* schemes. Within the class of *hardware-based approaches* (also referred to as physical-based approaches), multiple identical sensors (i.e. sensing the same physical parameter) along with a voting scheme (or more sophisticated techniques, see [43]) are employed to detect, isolate and accommodate sensor failures [44, 45, 46]. If the difference (namely, the residual signal) between the measured signal of a sensor and each other sensor in the set is considerably high, the aforementioned sensor is declared faulty and its data is replaced with those from the remaining (identical) sensors. For instance, the aforementioned assumptions apply to the case of homogeneous wireless sensor networks (WSNs), where neighboring nodes are assumed to measure roughly the same parameter [43]. Conventional physical-redundancy approaches however cannot handle cases with simultaneous failures of identical sensors, as they do not capitalize the statistical dependence of measurements originating from other sensor types [44, 45]. Moreover, in many applications, it is impractical to implement these approaches due to space and/or weight and/or cost constraints [45].

Accordingly, it is not surprising that methods adopting *analytical redundancy* have gained increasing attention within the research on SFDIA [47, 48, 28]. Unlike physical redundancy, the latter approaches exploit correlations and functional relationships within the system instead of introducing additional (redundant) hardware. Still, it is worth highlighting that the above two philosophies *are not* mutually exclusive and hybrid approaches can be pursued toward the sophisticated design of fault-tolerant DTs. Analytical redundancy can be usually implemented by either *model-based* or *data-driven* techniques.

Model-based SFDIA have been mostly investigated in the context of power systems [49], e.g. using electrical dynamics equations [47] or Luenberger observers [50]. Some other methods have focused on the detection and accommodation of proportional-type faults in nonlinear systems [51, 52]. Unfortunately, those methods (a) usually result in high complexity, (b) require an explicit, application-dependent, formulation of the analytical redundancy relationship among sensors and (c) are seldom able to handle multiple sensor faults simultaneously. On the contrary, *data-driven* approaches relying on historical data have recently received large interest, starting from simpler methods (e.g. auto-regressive models with exogenous inputs (ARX) [53]) to more complicated (non-linear) learning approaches (e.g. random forest (RF) [20], support vector machines (SVMs) [11, 12] and NNs [26, 32]). Indeed, *data-driven* techniques *do not require exact knowledge* of the mathematical model for sensor fault diagnosis.

Specifically, SVM-based classification was one of the relevant attempts to *detect* sensor faults in WSNs, in both batch [11] and online forms [12], which showed relatively small computational costs, but limited performance. Successive works [54, 15] have also employed the SVM approach to allow *both* detection and identification of faults: a binary classifier was trained from the residuals of each sensor. Specifically, in the former case [54], the residual signals were generated by comparing the true measurements with a single (global) observer designed by including fault models. Conversely, in the latter case [15], a residual was obtained from each (correlated) sensor pair via an ARX model, thus providing multiple classification outputs for a given sensor then aggregated at a higher level.

A second important class of approaches for SFDIA relies on the well-known Autoencoder (AE) NN [55, 48, 56, 16]. Indeed, the AE is an unsupervised learning technique capable of learning and extracting hidden representations from raw data and it is thus suited for fault detection. Hence, once trained, the AE can provide a reconstructed estimate of the sensors' measurements, thus allowing straightforward computation of residuals (i.e. the difference between inputs and outputs of the AE). Specifically, an AE-based (aided by exogenous inputs) sensor validation scheme for a heating, ventilation and air conditioning system was proposed in NNs [16]. Detection and identification are simply performed by comparing overall and per-sensor residuals to a given threshold. A similar AE-based SFDIA method is presented in [48] for an air quality controlling system, with identification scheme performed via a more involved sensor validity index. In both works [16, 48] accommodation is simply performed by using the AE output associated to the sensor(s) declared as faulty. Differently, a more sophisticated proposal uses an additional denoising AE (a supervised learning technique) to perform the accommodation task [56], namely to clean faulty data. Despite their simplicity, AE-based SFDIA approaches can suffer however from degraded performance under weak-faults, as the latter type of faults does not considerably impact correlations in data.

MLP NNs (including variants) have also been proved to perform satisfactorily for a number of relevant sensor fault diagnosis tasks [28, 14, 57], including heavy-duty diesel engines and aircrafts, based on a *sensor-centric viewpoint*. Indeed, in all the aforementioned works, *one MLP estimator per each sensor* is designed (solely on the basis of other sensors' measurements) and detection/identification is based on the evaluation of the residual vector. Accommodation is then performed by using the estimator(s) associated to the sensors declared as faulty. Specifically, the proposal in [14] adopts fully-connected cascade (FCC) NNs (i.e. MLPs allowing direct connections across different hidden layers) for the sensor estimator design, while [28] considers a hybrid structure with a linear NN and resource allocation network (a variant of well-known radial basis function NN) for the same task. More recently, a plain MLP estimator (exploiting the sole spatial correlation among sensors) has been proven to provide reliable detection with low false-alarm rate as well [57].

A different rationale is pursued in [26], where a *single* Deep belief network (a Bayesian type of NNs) has been trained (in a supervised fashion) to detect a faulty condition whereas sensor identification is naively carried out based on the maximum deviation from data mean-value. Along the same lines, a general approach is presented to detect and identify sensor faults using either a single Recurrent NN (RNN) or an MLP [18] for predicting next-step measurements and comparing them with actual ones. A disentanglement regularization term on the NN loss function is introduced to help the algorithm cope with propagation of faults to non-faulty sensors in the identification stage. Unfortunately, the accommodation stage is not taken into account in the above work. Interestingly, also a dynamic Bayesian network has succeeded in sensor fault detection and accommodation exploiting spatial and temporal correlations in the context of intelligent connected vehicles [19]. Still, its training difficulty (in terms of both parameter and structure learning) appears limiting in large-scale sensor systems.

2.2 Preliminaries

In the following, we focus on the different deep neural architectures which are employed for SFDIA, in the literature and this work. Before doing so, we need to introduce some notation.

Notation - $\boldsymbol{I}_N$ denotes the identity matrix of size N; $\mathbf{1}_{a\times b}$ denotes the matrix of all ones of size $[a \times b]$; $\mathbf{0}_N$ denotes the null vector of length N; $\{\cdot\}^T$ refers to the transpose operator, $[\cdot;\cdot]$ refers to concate operation, $|\cdot|$ indicates the absolute operation, $\odot$ denotes the entry-wise (Hadamard) product, $\otimes$ denotes the tensor product (whose meaning is specified each time is adopted), $*$ denotes the convolution operator, $\|\cdot\|_p$ denotes the p-norm, $\in$ is the set membership, and $\mathcal{O}(\cdot)$ denotes the Landau notation. $\mathcal{U}(a,b)$ (resp. $\mathcal{U}_d(a,b)$) denotes a uniform (resp. discrete-uniform) probability density function (PDF) with support $[a,b]$ (resp. $\{a, a+1, \dots, b\}$); $\mathcal{N}(b,c)$ (resp. $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$) denotes a Gaussian (resp. multivariate Gaussian) PDF with mean b and variance c (resp. with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$); $\mathcal{B}(p)$ denotes a Bernoulli distribution with parameter p.

2.2.1 Multi-Layer Perceptron (MLP)

It is a class of feedforward NNs that can model an arbitrary nonlinear mapping $f : \mathbb{R}^{i\times 1} \rightarrow \mathbb{R}^{j\times 1}$ between an input and an output vector. The NN is made of an arbitrary number of hidden layers, each consisting of an affine matrix operation on inputs, plus an entry-wise activation function. Considering a non-linear activation function e.g., Sigmoid function, rectified linear unit (ReLU) [58], the relationship between input and output data, will be a non-linear estimator. This makes the MLP a powerful tool for SFDIA problems which eases the implementation of sensor validation schemes for highly non-linear systems. For a given set of features (i.e. input) and a set of labels (i.e. output), the MLP can train a function approximator $f(\cdot)$ for either classification or regression tasks.

2.2.2 Convolutional Neural Network (CNN)

CNNs have been massively applied in many data-driven applications. A CNN is a specialized NN, inspired by visual mechanism of living organisms, designed for working with one-, two- and three-dimensional data. This is accomplished by chaining *convolutional* layers, each comprising a set of translation-invariant filters (kernels) with a limited extent (the “receptive field”). These layers are convolved with the input with the aim of extracting features of a certain input region. The aim of the convolutional layer is to extract high-level features from the input data that makes CNN able to capture the spatio-temporal correlations. In this work, one-dimensional CNNs have been used due to their appeal in (multivariate) time-series modeling.

2.2.3 Recurrent Neural Network (RNN)

This is a type of NN with attributes adapted to work for time series data or data that involves sequences. RNNs support processing of sequential data by allowing loopy connections, as opposed to feedforward NNs. RNNs have been utilized in many speech technology areas including fault diagnosis [59], speech recognition [60], and language modeling [61]. They are distinguished by their “memory” concept which helps them to store previous states to influence the current input for generating the output. It takes a sequence of n elements $(\boldsymbol{i}[0], \ldots, \boldsymbol{i}[n])$ as input, loops through these and outputs a value $\boldsymbol{o}[n]$. In each loop, the hidden layer acquires the "memory" from the previous loops. Still, vanilla RNNs are not able to model long-term dependencies affecting the output. Hence, two variants of vanilla RNNs are used in almost every application:

1. **Long-Short Term Memory (LSTM)**: first introduced by Hochreiter and Schmidhuber, 1997, it is still assiduously developed by researchers [62]. LSTM has the capability to cope with the RNN problem of modeling long-term dependencies [62].

2. **Gated Recurrent Unit (GRU)**: The GRU is a newer generation of RNNs presented by Cho *et al.* [63]. Despite the higher performance of LSTM over vanilla RNNs, excessive computational complexity for training an LSTM network is its main drawback [63]. The GRU is a simplified variant of LSTM using fewer training parameters: therefore it requires less memory and runs faster. Indeed, in each LSTM unit there are three gates associated to the cell state (forget, input, output) whereas in the GRU there are only two gates (reset and update).

2.2.4 Graph Convolutional Network (GCN)

In the following, the notion of graph convolution is recalled. Then, in Sec. 2.2.4.2, the actual implementation of graph convolutional layers is refreshed.

2.2.4.1 Convolution operation on graphs

The topological structure of a set of networked sensors can be described as an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V}$ is a finite set of N nodes (i.e. sensors), $\mathcal{E}$ is a set of edges that represents the connections between nodes, and $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix describing the connectivity of graph $\mathcal{G}$.

The *graph Laplacian* $\boldsymbol{L} \in \mathbb{R}^{N \times N}$ is a key operator in graph analysis [64], defined as $\boldsymbol{L} \triangleq (\boldsymbol{D} - \boldsymbol{A})$, namely the difference between the adjacency matrix $\boldsymbol{A}$ and the diagonal degree matrix $\boldsymbol{D}$ (where $d_{ii} = \sum_j a_{ij}$). The *normalized graph Laplacian* matrix $\boldsymbol{L}_{\mathcal{G}} = (\boldsymbol{I}_N - \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{D}^{-\frac{1}{2}})$ is a real symmetric positive semidefinite matrix with a complete set of orthonormal eigenvectors $\{\boldsymbol{u}_i\}_{i=1}^N \in \mathbb{R}^N$ (also known as Fourier functions) associated with real non-negative eigenvalues $\{\lambda_i\}_{i=1}^N$ representing the frequencies of the graph. Moreover, the graph normalized Laplacian spectrum [65] is contained in the span of $\{\lambda_i\}_{i=1}^N \in [0, 2]$. The graph normalized Laplacian is always diagonalizable by the Fourier basis $\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_N \end{bmatrix} \in \mathbb{R}^{N \times N}$ i.e. $\boldsymbol{L}_{\mathcal{G}} = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^T$, where $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, ..., \lambda_N) \in \mathbb{R}^{N \times N}$. The *Fourier graph transform* $\mathcal{F}$ of a signal $\boldsymbol{x} \in \mathbb{R}^N$ is defined by the Fourier basis $\mathcal{F}(\boldsymbol{x}) = \boldsymbol{U}^T \boldsymbol{x}$ and its inverse $\mathcal{F}^{-1}(\boldsymbol{x}) = \boldsymbol{U} \mathcal{F}(\boldsymbol{x})$.

Spectral convolution on the graph $\mathcal{G}$ is defined [66] as the signal $\boldsymbol{x}$ filtered by graph filter $g_{\boldsymbol{\theta}}$, i.e.

$$\begin{aligned} g_{\boldsymbol{\theta}} * \boldsymbol{x} &\triangleq \mathcal{F}^{-1}(\mathcal{F}(g_{\boldsymbol{\theta}}) \odot \mathcal{F}(\boldsymbol{x})) = \boldsymbol{U}(\boldsymbol{U}^T g_{\boldsymbol{\theta}} \odot \boldsymbol{U}^T \boldsymbol{x}) \\ &= \left[\boldsymbol{U} \hat{g}_{\boldsymbol{\theta}}(\boldsymbol{\Lambda}) \boldsymbol{U}^T\right] \boldsymbol{x} \,, \end{aligned} \tag{2.1}$$

where $\hat{g}_{\boldsymbol{\theta}}(\boldsymbol{\Lambda}) \triangleq \operatorname{diag}(\boldsymbol{U}^T g_{\boldsymbol{\theta}})$ is the spectral graph filter (in diagonal matrix form) parameterized by $\boldsymbol{\theta} \in \mathbb{R}^N$ in the Fourier domain to avoid the elementwise operation, namely

$$\hat{g}_{\boldsymbol{\theta}}(\boldsymbol{\Lambda}) = \begin{bmatrix} \hat{g}_{\theta_1}(\lambda_1) & 0 & \dots & 0 \\ 0 & \hat{g}_{\theta_2}(\lambda_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{g}_{\theta_N}(\lambda_N) \end{bmatrix} . \tag{2.2}$$

2.2.4.2 GCN layer

The defined filtering operation $g_{\boldsymbol{\theta}} * \boldsymbol{x}$ has a *quadratic* computational complexity $\mathcal{O}(N^2)$ due to the matrix multiplication with the Fourier basis $\boldsymbol{U}$ in Eq. (2.1). Also, eigendecomposition of $\boldsymbol{L}_{\mathcal{G}}$ is required (once) for carrying out spectral convolutions on $\mathcal{G}$. Then for large graphs (i.e. $N \gg 1$), both these operations can become *computationally expensive*. Equally important, there is *no guarantee of spatial localization* [67] of the graph filter $\hat{g}_{\boldsymbol{\theta}}(\boldsymbol{\Lambda})$ (i.e. a non-smooth filter). Spatial decay is an advantageous property to extract multi-scale patterns. The graph filter $\hat{g}_{\boldsymbol{\theta}}(\boldsymbol{\Lambda})$ can become a non-smooth spectral filter, while smoothness in the frequency domain corresponds to rapid spatial decay in the vertex domain.

Chebyshev polynomial approximation: To tackle the localization problem, $\hat{g}_{\boldsymbol{\theta}}(\boldsymbol{\Lambda})$ can be approximated by a truncated expansion up to order $\mathcal{K}$ of Chebyshev polynomials [68] $\{T_k(x)\}_{k=0}^{\mathcal{K}}$, namely $\hat{g}_{\boldsymbol{\theta}'}(\boldsymbol{\Lambda}) \approx \sum_{k=0}^{\mathcal{K}} \theta'_k \boldsymbol{T}_k(\widetilde{\boldsymbol{\Lambda}})$. In the latter approximation, $\widetilde{\boldsymbol{\Lambda}}$ denotes the rescaled matrix $\widetilde{\boldsymbol{\Lambda}} \triangleq (2\boldsymbol{\Lambda}/\lambda_{\max} - \boldsymbol{I}_N)$ (where $\lambda_{\max}$ denotes the largest eigenvalue of $\boldsymbol{L}_{\mathcal{G}}$), while θ'_k represents the kth Chebyshev coefficient ($\boldsymbol{\theta}' \in \mathbb{R}^{\mathcal{K}+1}$). The Chebyshev polynomials can be efficiently computed via the recurrence relation $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ with $T_0(x) = 1$ and $T_1(x) = x$. Although the graph filter is now K-localized with respect to the Kth-order polynomials of the Laplacian, the learning complexity is still not addressed because of the multiplication of the eigenvector matrix $\boldsymbol{U}$.

A solution to this problem is to directly learn the function of the normalized Laplacian $g_{\boldsymbol{\theta}'}(\boldsymbol{L}_{\mathcal{G}})$ [66]. Indeed, exploiting $\left(\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\right)^k = \boldsymbol{U}\boldsymbol{\Lambda}^k\boldsymbol{U}^T$, the equality $\boldsymbol{U}\boldsymbol{T}_k(\widetilde{\boldsymbol{\Lambda}})\boldsymbol{U}^T = \boldsymbol{T}_k(\widetilde{\boldsymbol{L}})$ holds, where $\widetilde{\boldsymbol{L}} \triangleq (2\boldsymbol{L}_{\mathcal{G}}/\lambda_{\max} - \boldsymbol{I}_N)$. Accordingly, graph convolution can be approximated as

$$g_{\boldsymbol{\theta}'} * \boldsymbol{x} \approx \sum_{k=0}^{\mathcal{K}} \theta'_k \left[\boldsymbol{U}\, \boldsymbol{T}_k(\widetilde{\boldsymbol{\Lambda}})\boldsymbol{U}^T\right] \boldsymbol{x} = \sum_{k=0}^{\mathcal{K}} \theta'_k \boldsymbol{T}_k(\widetilde{\boldsymbol{L}})\, \boldsymbol{x} \tag{2.3}$$

where $\boldsymbol{T}_k(\widetilde{\boldsymbol{L}}) \in \mathbb{R}^{N \times N}$ is the $\mathcal{K}$th order Chebyshev polynomial. The filtering operation is reduced to $\mathcal{O}(\mathcal{K}|\mathcal{E}|)$ operations.

Linear formulation of GCN: With first-order approximation (one-hop localization, i.e. $K = 1$) of Eq. (2.3) and further assuming[1] $\lambda_{\max} \approx 2$ and $\theta = \theta'_0 = -\theta'_1$, a layer-wise *linear* convolution operation can be defined to create a graph-based convolutional NN model, i.e.

$$g_\theta * \boldsymbol{x} = \theta\,(\boldsymbol{I}_N + \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}})\boldsymbol{x} \rightarrow \theta\,(\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}})\boldsymbol{x} \tag{2.4}$$

[1] These assumptions are made to constrain the number of trainable parameters (viz. reduce number of operations) and to address overfitting. Change in scale can be adapted by NN in the training phase.

with $\widetilde{\boldsymbol{A}} \triangleq \boldsymbol{A} + \boldsymbol{I}_N$ and $\widetilde{d}_{ii} \triangleq \sum_j \widetilde{a}_{ij}$. The last expression means that the matrix operation has been replaced with the so-called *re-normalization trick* [69]. Capitalizing the above result, the GCN layer for a graph signal $\boldsymbol{X} \in \mathbb{R}^{N \times C}$ with C features per node and F filters is formulated as:

$$\boldsymbol{Z} = \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{\Theta} \, , \tag{2.5}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{C \times F}$ is the learnable-filter parameter matrix (i.e. the GCN weight matrix), and $\boldsymbol{Z} \in \mathbb{R}^{N \times F}$ is the convolved signal matrix. Accordingly, the computational complexity of the GCN operation is $\mathcal{O}(FC|\mathcal{E}|)$ due to a sparse multiplication (with $\widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}}$). Hence, in general, the GCN layer can be expressed in its implicit form as:

$$\boldsymbol{Z} = \mathrm{GCN}_{\boldsymbol{\Theta}}(\boldsymbol{X}; \mathcal{G}) \, , \tag{2.6}$$

The aforementioned layer assumes the knowledge of the graph structure via the matrix $\widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}}$.

2.3 Sensor Fault

A fault in a system refers to a complete (or partial) malfunction and manifests over a permanent (or transient) time span. As shown in Fig. 2.1, the most common types of sensor faults in a sensor network are defined (a detailed discussion of sensor faults is found in [70, 71]). Depending on the characteristics of sensor data, faults can be classified as follows:

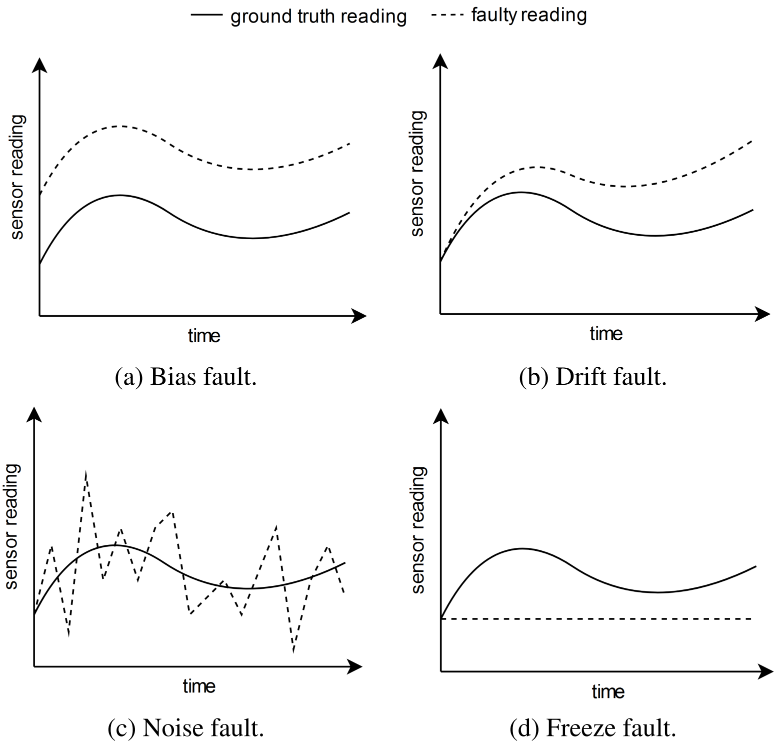
1. *Bias* fault: also known as offset fault, the deviation from nominal values is given by an additive constant bias;

2. *Drift* fault: sensor readings drift with a small slope from nominal values (drift faults are more subtle since they appear gradually over time and their effect is not very apparent);

3. *Noise* fault: an increased noise level in sensor readings (when noise power is much larger than usual, it is an indication of sensor malfunctioning).

4. *Freeze* fault: also known as stuck-at fault, the sensor readings stuck at a constant value (i.e. the variance of the readings becomes zero);

The impact of sensor faults would affect stability, reliability and accuracy of the system depending on the specific application. Hence, to fully utilize the expected properties of the DT, it is essential to continuously evaluate and amend sensor data.



(a) Bias fault.

(b) Drift fault.

(c) Noise fault.

(d) Freeze fault.

Figure 2.1: Types of sensor faults.

We consider transient synthetically-generated *bias*, *drift*, *noise*, and *freeze* fault types. Bias fault represents sudden faults, while drift fault well represents gradually-appearing faults. Finally, noise faults well represent sensors subject to external disturbances. It is worth highlighting that the practice of modeling simulated faults superimposed to real data is a common practice in the evaluation of SFDIA systems (e.g. [11, 18, 56]), as (i) real faulty measurements are sporadic and very hard to obtain and (ii) simulated faults also allow quantifying accommodation performance. This is also to highlight the *generality* of the proposed architectures in accommodating diversified faulty conditions. Details about the modeling of each fault type are provided.

2.3.1 Bias Fault

This fault type manifests as an additive constant vector $\boldsymbol{b} \in \mathbb{R}^C$ inserted to the normal operation of generic nth sensor for M consecutive samples, i.e.

$$\boldsymbol{x}_n^b[k] = \begin{cases} \boldsymbol{x}_n[k] + \boldsymbol{b} \ , & 0 \leq k - m < M \\ \boldsymbol{x}_n[k] \ , & \text{otherwise} \end{cases} \tag{2.7}$$

where m denotes the starting time instant of the fault, C is the number of features per sensor, $\boldsymbol{x}_n[k]$ and $\boldsymbol{x}_n^b[k]$ are the normalized healthy and possibly bias-faulty readings at time step k, respectively.

2.3.2 Drift Fault

For this type of fault, an additive term drifts gradually to the bias level vector $\boldsymbol{b}$ in M samples and then remains at the same value for K samples ($M > K$), namely:

$$\boldsymbol{x}_n^d[k] = \begin{cases} \boldsymbol{x}_n[k] + \frac{(k-m+1)}{M} \boldsymbol{b} \ , & 0 \leq k - m < M \\ \boldsymbol{x}_n[k] + \boldsymbol{b} \ , & M \leq k - m < M + K \\ \boldsymbol{x}_n[k] & \text{otherwise} \end{cases} \tag{2.8}$$

where $\boldsymbol{x}_n^d[k]$ is the possibly drift-faulty readings at time step k.

2.3.3 Noise Fault

This fault type is also considered to evaluate the performance of the proposed architecture in unseen fault scenarios. Specifically, a zero-mean additive Gaussian noise vector $\boldsymbol{w}[k] \sim \mathcal{N}(\boldsymbol{0}_C, \sigma_g^2 \boldsymbol{I}_C)$ is added for M consecutive samples, i.e.

$$\boldsymbol{x}_n^g[k] = \begin{cases} \boldsymbol{x}_n[k] + \boldsymbol{w}[k] \ , & 0 \leq k - m < M \\ \boldsymbol{x}_n[k] & \text{otherwise} \end{cases} \tag{2.9}$$

where $\boldsymbol{x}_n^g[k]$ is the possibly noise-faulty readings at time step k and σ_g^2 represents the noise variance.

2.3.4 Freeze Fault

For freeze-type faults, sensor output stuck at previous reading for M consecutive samples as follows

$$\boldsymbol{x}_n^f[k] = \begin{cases} \boldsymbol{x}_n[m-1] \ , & 0 \leq n - m < M \\ \boldsymbol{x}_n[k] \ , & \text{otherwise} \end{cases} \tag{2.10}$$

where $\boldsymbol{x}_n^f[k]$ is the possibly-faulty readings at time step k.

2.3.5 Random Fault Generation

The four types of faults considered in this work are synthetically generated [11, 18, 56] according to the corresponding models detailed above on the top of the *real measurement data*. Unless otherwise stated, the fault level b (note that $\boldsymbol{b}$ reduces to a scalar since $C = 1$) is generated as $b = (2s_b - 1) \cdot a_b$ where $s_b \sim \mathcal{B}(1/2)$ and $a_b \sim \mathcal{U}(0.2, 0.4)$ to represent weak faults. The variance for noise faults is similarly generated as $\sigma_g^2 \sim \mathcal{U}(0.2, 0.4)$. The fault lengths are specified via the parameters M and K (see Eqs. (2.7), (2.8), (2.9) and (2.10)), which are assumed uniformly distributed as $M, K \sim \mathcal{U}_d(3, 11)$ to represent *transient faults*[2]. The random distribution of faults helps both the estimator and classifier to better generalize during the training phase and prevents focusing on a specific fault level/length. To verify the robustness of the proposed architectures against simultaneous faults, *up to three concurrent faulty sensors* were considered for the (fault-)generation process.

2.4 Sensors Classification

In the proposed methods, it is assumed that the sensors are divided into *two* sets: (i) the set of unreliable sensors $\mathcal{S}_U$, containing sensors that are vulnerable to faults; and (ii) the set of reliable sensors $\mathcal{S}_R$, which, depending on the working system, include sensors whose flawless functionality can be guaranteed [33]. This (ideal) level of reliability could be associated to: a meta-sensor modeling a group of identical sensors (enjoying hardware redundancy), high-quality sensors, a proper design and safe working environment, a device being at the middle of life span [72], or context measurement information which is assumed to have significantly higher reliability than the considered networked sensor system. In a more general sense, any reliable source of data correlated with unreliable sensors could be included in the set of reliable sensors. In the following, without loss of generality, it is assumed $\mathcal{S}_U = \{1, \ldots, N_U\}$ and $\mathcal{S}_R = \{N_U + 1, \ldots, N\}$, where N_U and N denote the number of unreliable sensors and total number of sensors, respectively. Also, for compactness, N_R denotes the cardinality of the reliable set $\mathcal{S}_R$ (i.e. $N_R = N - N_U$).

[2] Under freeze fault, the fault length (M) is uniformly distributed between 100 and 400 consecutive samples due to smooth oscillating (WSN and PMSM) data-sets (which are presented in the following). Smaller fault lengths cause negligible faults on the working datasets.

2.5 Working Datasets

Three real-world datasets are applied to the proposed SFDIA system to evaluate the qualification of the system in different scenarios. There are several publicly-available datasets which applied to the proposed SFDIA systems to evaluate the qualification of the systems in different scenarios. Sensors in all datasets collect a single parameter (flow rate, pressure, etc.), i.e. $C = 1$. Before feeding the datasets to the proposed architectures, sensors readings in each dataset are normalized using min-max scaling on the training set to avoid polarization during the learning process. Finally, the entire rows containing missing values are ignored from the datasets. No other pre-processing has been considered, such as feature extraction, to help the learning procedure of our models. Although, for noisy datasets, smoothing techniques (e.g. moving average, Savitzky-Golay filter or quadratic regression) or low-pass filtering can be performed allowing the important patterns of data to stand out. Tab. 2.1 summarizes the statistics of each dataset. They will briefly be described in the following.

2.5.1 Air Quality (AQ) Dataset

The first dataset contains hourly-averaged measurements of an array of 5 metal oxide chemical sensors embedded in a gas multi-sensor device deployed on the field in an Italian city along with gas concentrations references from a certified analyzer [21]. The device was located in a polluted area, at road level of the city. AQ dataset was recorded during Mar. 2004-Feb. 2005.

Measurements contain carbon monoxide (CO), non-metanic hydrocarbons (NMH), nitrogen oxides (NO_x), nitrogen dioxide (NO_2) and ozone (O_3) gas concentrations, as well as measurements of temperature and humidity. For our analysis, the ground truth hourly-averaged concentrations provided by a co-located reference certified analyzer along with absolute humidity are ignored. Accordingly, in our numerical analysis, the five gas sensors are considered as the unreliable set ($N_U = 5$), whereas temperature and relative humidity are considered as the reliable set ($N_R = 2$).

2.5.2 WSN Dataset

The second dataset used in our evaluation has been collected at the University of North Carolina at Greensboro [22]. A labeled dataset collected from a single-hop and a multi-hop WSN using TelosB motes. The dataset consists of 4 sensors located indoor and outdoor measuring humidity and temperature. Measurements were collected during 6 hours at 5 seconds intervals. Anomalies indicated with label "1" in the original dataset were introduced to two sensors by using a water kettle which increased the temperature and humidity.

In what follows, only the multi-hop dataset with 4 temperature (T1 to T4) measurements is used as unreliable set ($N_U = 4$), and data with the indicated label "1" were ignored from this dataset. No reliable set is considered for this dataset ($N_R = 0$).

2.5.3 Permanent Magnet Synchronous Motor (PMSM) Dataset

The third dataset comprises several sensor data measurements from a permanent magnet synchronous motor collected by the LEA department at Paderborn University [23, 73]. Data-set measurements include ambient temperature, coolant temperature, voltage q and d components, current q and d components, motor speed, torque, rotor temperature, stator yoke temperature, stator tooth temperature, and stator winding temperature. Original measurements contain 52 sessions, with each session being $1 \sim 6\mathrm{h}$ long and sampled at intervals of 0.5 seconds.

We have considered a sample interval of 15 seconds[3] (by down-sampling) and ignored the ambient and rotor measurements. Summation of q and d components of voltage and current are treated as final voltage and current measurements. The reliable set consists of 3 stator temperatures[4] ($N_R = 3$), and other remaining measurements form the unreliable set ($N_U = 5$).

2.5.4 DeFACTO Dataset

DeFACTO is a highly instrumented experimental facility of SINTEF research company that includes a 139 meters long horizontal loop and a 90 meters deep vertical U-tube, enabling the study of both horizontal and vertical flow phenomena relevant to transport phenomenon for CCS [37]. The carbon dioxide (CO_2) loops operate at up to 160 bar, the vertical section has a tight heat transfer system that allows operation at temperatures between 5°C and 35°C. The experimental studies comprise steady-state liquid or gas flow and transient phenomena, including rapid depressurization and cavitation. We use the SINTEF dataset for experiment conducted at the DeFACTO facility on Dec. 13, 2021. Measurements were collected during 1337 minutes at 5 seconds intervals. We only used the temperature measurements from the 6 temperature sensors installed on the surface of the CO_2

[3] In the paper [36], the readings were sampled with 1.5 s-intervals and the first 55k readings were picked after sampling.

[4] In the paper [36], only the stator yoke temperature is assumed to belong to the reliable set $\mathcal{S}_R$ (thus $N_R = 1$).

pipeline DeFACTO.

Table 2.1: Datasets description. The reliable sensors in each dataset are highlighted in italic.

Dataset	**Samples**	N_U	N_R	**Attributes**
AQ	8991	5	2	**Multivariate**, **time-series**; CO, NMH, NO_x, NO_2 and O_3 gas concentrations, as well as measurements of *temperature* and *humidity*
WSN	4589	4	0	**Multivariate**, **time-series**; four temperature sensors: two indoor, two outdoor
PMSM	55000	5	3 (1 in **P5**)	**Multivariate**, **time-series**; coolant temperature, voltage and current (summation of q and d components), motor speed, torque and 3 *stator temperatures*
DeFACTO	16042	6	0	**Multivariate**, **time-series**; six temperature sensors: installed on the surface of the CO_2 pipeline DeFACTO
PeMSD8	17856	170	0	**Multivariate - Graph shape**, **time-series**; traffic flow readings, with 277 edges
Water Tank	26998	100	0	**Multivariate - Graph shape**, **time-series**; pressure readings, with 388 edges

2.5.5 PeMSD8 Dataset

This real-world dataset contains traffic data collected in San Bernardino, California, during Jul.-Aug. 2016 [30]. The traffic data consists of all detector-based point data captured by the California Department of Transportation (CalTrans) performance measurement system[5] (PeMS). The traffic flow is collected by $N = 170$ sensors on eight roads in San Bernardino with a time interval of 5 minutes.

2.5.6 Water Tank Dataset

This dataset[6] collects measurements from a network of $N = 100$ water tanks connected through pipelines [31]. Tanks pressure is measured using pressure sensors to indicate the level of water in the tanks. When a tank's water level goes below

[5] https://pems.dot.ca.gov/

[6] https://github.com/IndustrialNetwork/GraphDataset

a certain threshold, tank starts to refill until it is full. The flow rate between two connected tanks is a non-linear function of the pressure and distance between the tanks. We use the first three measurements' logs of the dataset which roughly contains 27k samples.

Chapter 3

Sensor Fault Detection, Isolation and Accommodation

In this chapter, the contributions of thesis for the SFDIA problem are described briefly. Section 3.1 presents a summary of the works Publications **P1**, **P2**, **P4** and **P6**, in which we proposed a modular ML-based framework for sensor validation, termed M-SFDIA. Section 3.2 describes the extended M-SFDIA architecture and provides the performance comparison of the proposed scheme with other methods via numerical results, which is taken from Paper **P3** and **P5**. In Section 3.3, the final proposed data-driven SFDIA architecture is presented and the functionalities of each block are illustrated and results from Paper **P7** are presented.

3.1 Modular-SFDIA (M-SFDIA) Architecture

In view of the previous discussion in Sec. 2.1, some proposals are restricted to a given vertical domain (e.g. aircraft [14], vehicle [15] or HVAC system [16] monitoring), thus *lacking a general formulation*. Secondly, part of the literature evaluates corresponding proposals on *private* (e.g. [18, 19]) or *simulated* (e.g. [20, 16, 14]) measurement data, thus precluding reproducibility and convincing evaluation, respectively. Thirdly, a number of the discussed works evaluate their proposals only on a *single fault type* (e.g. bias [18, 48] or drift [28]). Equally important, some architectures are only *limited to fault detection* [11, 12]. On the other hand, some recent proposals do not foresee all three tasks in their original formulation, e.g. the identification and accommodation tasks in [56] and [18], respectively. Still, even when all three tasks can be carried out, in some cases *only spatial correlation* [28, 57, 16] is used to accommodate faulty measurements. Fi-

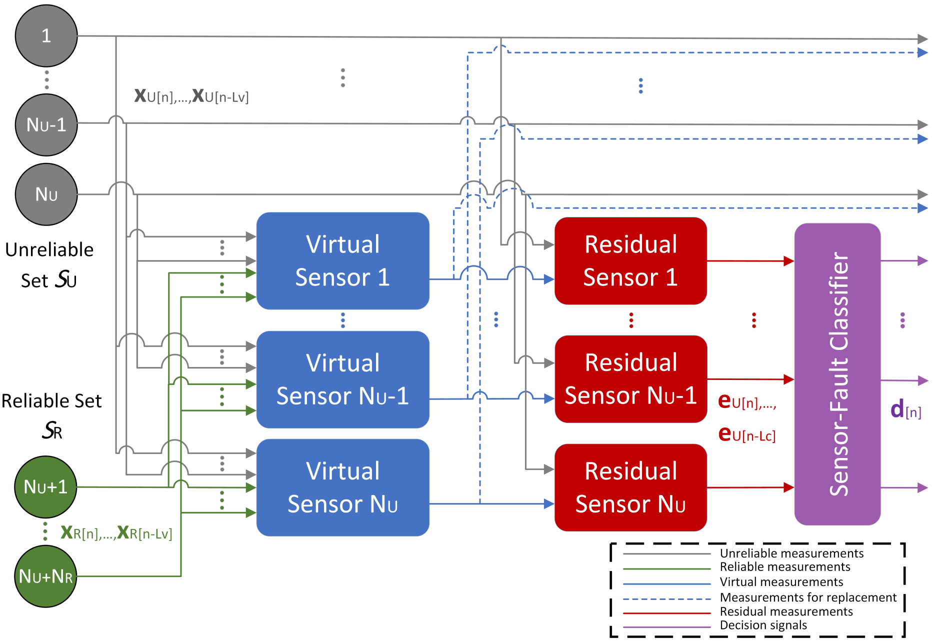


Figure 3.1: Block diagram of the M-SFDIA system.

nally, some approaches have a *limited modularity* [56, 48, 18].

Accordingly, the proposed framework allows the development of a general SFDIA scheme to be easily adapted to different application domains. The proposed architecture jointly takes advantage of the temporal correlation of the measurements and of both reliable and unreliable sensors within the system to achieve a higher sensor validation performance.

3.1.1 System Architecture for M-SFDIA Scheme

The block diagram of the proposed M-SFDIA scheme is shown in Fig. 3.1, where similar blocks and similar data are reported in the same color. The input to the system is the set of measurements from all sensors. The system is based on *three* stages: (i) the first stage is made of N_U virtual sensors (representing estimation of unreliable sensors); (ii) the second stage is made of N_U analogous residual-computation units; and (iii) the third stage is made of a (multi-task) classifier. The classifier at the third stage is performing detection and isolation, while accommodation is done by exploiting the estimators' output.

More specifically, at the *first stage*, the virtual sensor $s \in \mathcal{S}_U$ receives as input the measurements from all sensors excluding sensor s (i.e. $(\mathcal{S}_U \cup \mathcal{S}_R - \{s\})$ for time instant n and L_v previous time instants (i.e. a sliding window), and produces as output an estimate of the measurement of sensor $s \in \mathcal{S}_U$, whose nth sample is denoted $y_s[n]$.

Then, at the *second stage*, the residual-computation unit $s \in \mathcal{S}_U$ receives as input the measurement $x_s[n]$ of sensor $s \in \mathcal{S}_U$ and the corresponding estimate $y_s[n]$ from the virtual sensor $s \in \mathcal{S}_U$ and produces as output a measure of dissimilarity of the pair, whose nth sample is denoted $e_s[n]$. Residual measurements are reflecting inconsistencies between the normal and faulty sensor operating status of unreliable sensors.

At the *third stage*, the classifier receives as input the dissimilarity measures from all the sensor pairs in the unreliable set $\mathcal{S}_U$ for time instant n and L_c previous time instants, and produces as output a decision vector about if and which sensor has undergone a failure. According to Fig. 3.1, the nth (soft-) decision vector is denoted $\boldsymbol{d}[n] = (d_1[n], d_2[n], \ldots, d_{N_U}[n])^T$ where $d_i[n] \in [0, 1],\ i = 1, \ldots, N_U$ denotes the probability of the ith sensor (corresponds to a specified unreliable sensor) being faulty. Ideally, a vector $\boldsymbol{d}[n]$ with all elements set to 0 denotes the event that no sensor has been declared in failure, while the set of unreliable sensors $\mathcal{S}_U$ is mapped bijectively into the first N_U positive integers with an arbitrary labeling function. The final decision is made based on whether the maximum element of vector $\boldsymbol{d}[n]$ exceeds a given threshold γ.

It is implicitly assumed that in the case that sensor $s \in \mathcal{S}_U$ is declared in failure, its measurement $x_s[n]$ is replaced with the estimate $y_s[n]$ from the corresponding virtual sensor. It is apparent how the considered architecture implements all the tasks of an SFDIA system: i.e. decision vector $\boldsymbol{d}[n]$ with an over threshold element represents the *detection task*; after a fault is detected, the specific sensor index i corresponding to the maximum element $d_i[n]$ of the decision vector performs the *isolation task* and replacing $x_s[n]$ with $y_s[n]$ employs the *accommodation task*, with the sensor s identified through the inverse labeling function. In what follows, we detail each of the three aforementioned stages.

3.1.1.1 Virtual sensor

An MLP NN, with $(L_v + 1)(N_U + N_R - 1)$ inputs, 1 output, and H_v hidden layers, each with N_v hidden nodes, has been considered for the implementation of the generic virtual sensor, i.e.

$$y_s[n] = f_s^{(H_v,N_v)}(\boldsymbol{x}_{U,s}[n], \ldots, \boldsymbol{x}_{U,s}[n - L_v] \\ , \boldsymbol{x}_R[n], \ldots, \boldsymbol{x}_R[n - L_v]) \ , \tag{3.1}$$

where f_s represents the MLP-based function model of the sth sensor. Each MLP has been trained using the Nesterov-accelerated adaptive moment estimation (Nadam) optimization algorithm using real-world datasets [74, 75]. The Nadam algorithm takes advantage of properties of the adaptive moment estimation (Adam) algorithm and incorporates Nesterov Accelerated Gradients into Adam. Hyperbolic tangent (Tanh) and identity activation functions are employed in hidden layers and the output layer, respectively. The mean square error (MSE) loss function is used for loss calculation in training phase.

The MLP is a simple architecture with proved performance of estimating nonlinear behavior [76, 77]. Numerical results show the excellent performance of MLP architecture. However, in the case of further requirement of extrapolating the long-term impact of the temporal dimension for time series datasets, more complicated architectures (e.g. CNN, RNNs and GRU networks [78, 79, 35]) are expected to present more appropriate results for the implementation of each virtual sensor. The modular design of this proposal allows exploring different types of NN models to select the most suitable NN models according to the application [35]. The publication **P4** [35] focuses on exploring the optimized model selection for SFDIA of WSN dataset. Data description, data pre-processing (in order to make it suitable for model training) and data contamination procedure (via synthetically-generated faults) are described in the next section.

3.1.1.2 Residual computation

For dissimilarity measure, we simply considered the error between the estimated value and the actual value, i.e.

$$e_s[n] = y_s[n] - x_s[n]. \tag{3.2}$$

In fault-free condition, it is expected that the residual measurements $e_s[n]$ be equal to zero, but in practice, it always contains non-zero values due to noise and imperfect estimation of sensor output. Hence, the classifier is introduced to discriminate faulty measurements from non-faulty measurements via pattern analysis of residual signals.

Table 3.1: Computational complexity of the MLPs constituting the proposed SFDIA architecture.

Layers	MLP	Complexity
first hidden layer	virtual sensor	$\mathcal{O}(L_v N_U N_v + L_v N_R N_v)$
	classifier	$\mathcal{O}(L_c N_U N_c)$
other hidden layers	virtual sensor	$\mathcal{O}(N_v^2)$
	classifier	$\mathcal{O}(N_c^2)$
output layer	virtual sensor	$\mathcal{O}(N_v)$
	classifier	$\mathcal{O}(N_U N_c)$
in total	virtual sensor	$\mathcal{O}(L_v N_U N_v + L_v N_R N_v + H_v N_v^2)$
	classifier	$\mathcal{O}(L_c N_U N_c + H_c N_c^2)$

3.1.1.3 Classifier

An MLP NN, with N_U inputs, N_U discrete output, and H_c hidden layer with N_c hidden nodes, has been considered for the implementation of the classifier, i.e.

$$\boldsymbol{d}[n] = g^{(H_c,N_c)}(\boldsymbol{e}_U[n], \ldots, \boldsymbol{e}_U[n - L_c]). \tag{3.3}$$

where $\boldsymbol{e}_U[n]$ is a vector of the dissimilarity measurements of the unreliable set at time instant n. Since there is a certain level of correlation between temporal samples of residual signals, L_c previous time instants are also fed to the classifier to exploit the temporal correlation among measurements.

The binary cross-entropy loss function along with the same optimization algorithm (Nadam) and activation function (Tanh) for hidden layers as in the virtual sensors are employed in the classifier. Moreover, N_U sigmoid activation function is used at the output layer of the classifier. The fault-signal generation is described in the next section.

3.1.1.4 Computational complexity

The computational complexity of the proposed M-SFDIA structure is calculated hereunder in terms of the big-$\mathcal{O}$ notation for one input sample. The computational complexity for each layer of the virtual sensor and classifier is specified in Tab. 3.1.

It is worth noticing that in Tab. 3.1, the impact of Tanh and sigmoid operations for virtual sensors and the classifier has been neglected. Finally, with respect to the computational complexity of both MLPs and assuming equal number of hidden layers ($H_v = H_c = H$), nodes per hidden layer ($N_v = N_c = N_g$) and time delays ($L_v = L_c = L$), the computational complexity involved with the proposed architecture is approximately $\mathcal{O}(LN_U^2N_g + LN_RN_UN_g + HN_UN_g^2)$. Thus, the proposed architecture has *polynomial complexity*, and the complexity grows *quadratically* as a function of the number of nodes per layer (N_g) and number of unreliable sensors (N_U).

3.1.2 Numerical Results

In this section, performance of the proposed M-SFDIA architecture is examined and compared with recent research works by using the aforementioned real-world datasets. Each dataset is divided into three parts. On each dataset, we used 70% of data for training MLPs (training set), 15% for validating (validation set) and the last 15% block of data for testing purposes (test set). *Early stopping* method is used to avoid over-fitting during the training phase [80]. In this method, error on the validation set is monitored and if after 20 consecutive epochs validation set error did not improve, the training process is stopped.

In addition, to better understand the effect of fault strength on detection accuracy, strong fault signals with maximum level b uniformly distributed between 0.6 and 0.9 are considered for comparison with weak fault signals. A more detailed configuration for the proposed M-SFDIA scheme can be found in Publication **P2** [33] and **P4** [35].

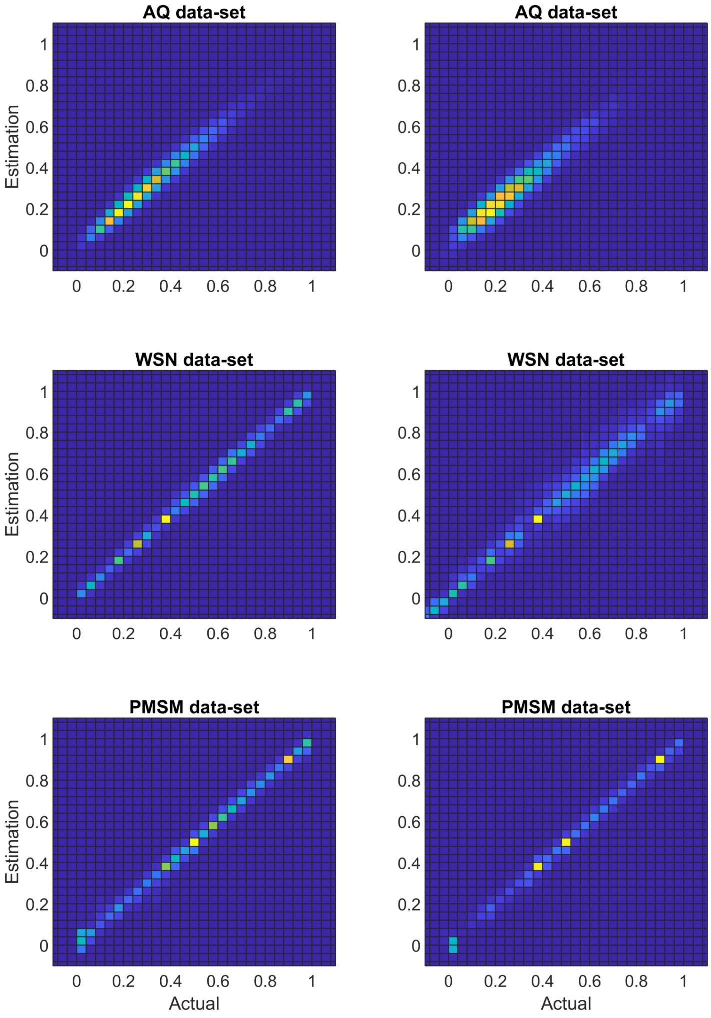
3.1.2.1 Virtual sensors performance

Performance of the configuration with $H_v = 1$ hidden layer, $N_v = 10$ nodes per hidden layer and $L_v = 10$ is considered acceptable (details are in publication **P2** [33]), thus in the following, we will refer to this specific configuration. The 2D-PDF plots of the estimated and actual values for virtual sensors in configuration 1×10 are shown in Fig. 3.2, both for the training and the test sets. It is worth noticing that the test set of the WSN dataset exceeds the defined normalization lower-bound which is the result of normalization on the training set.

3.1.2.2 Classifier fault detection and classification performance

Synthetically-generated faults have been added to the unreliable set of sensors to emulate faulty sensors. A classifier with $H_c = 2$ hidden layers, $N_c = 15$ nodes per hidden layer, and a memory of $L_c = 10$ has been trained.

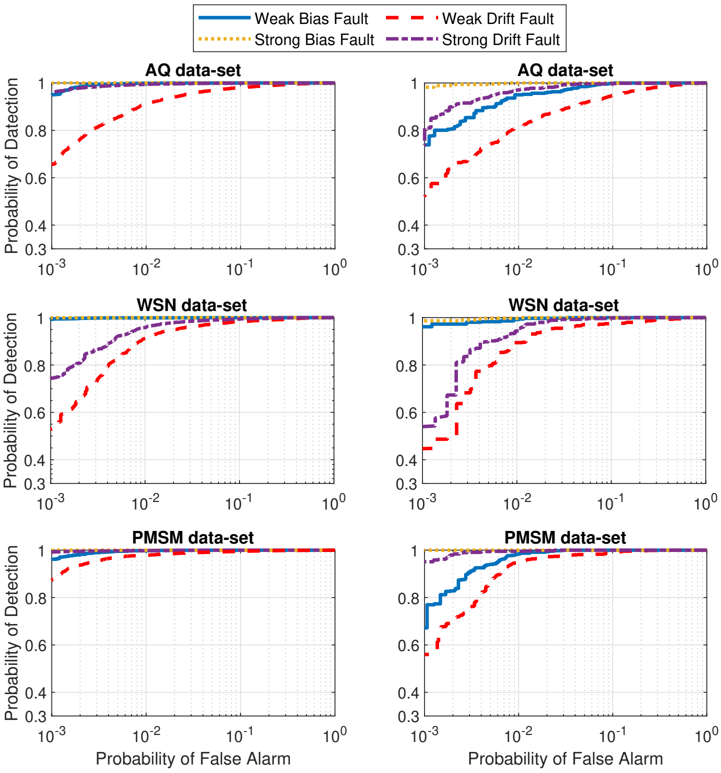
The probabilities of detection and false-alarm are two important metrics for evaluating the performance of a detector. Accordingly, in Fig. 3.3, fault detection performance is investigated in terms of both metrics by using the well-known receiver



(a) Training set **(b)** Test set

Figure 3.2: Averaged performance of the virtual sensors in configuration 1×10 and $L_v = 10$ in terms of 2D PDFs of the estimated and actual values.

operating characteristic (ROC) curves (i.e. by varying the threshold γ). Results highlight that, although the classifier is facing weak fault signals, it is still capable



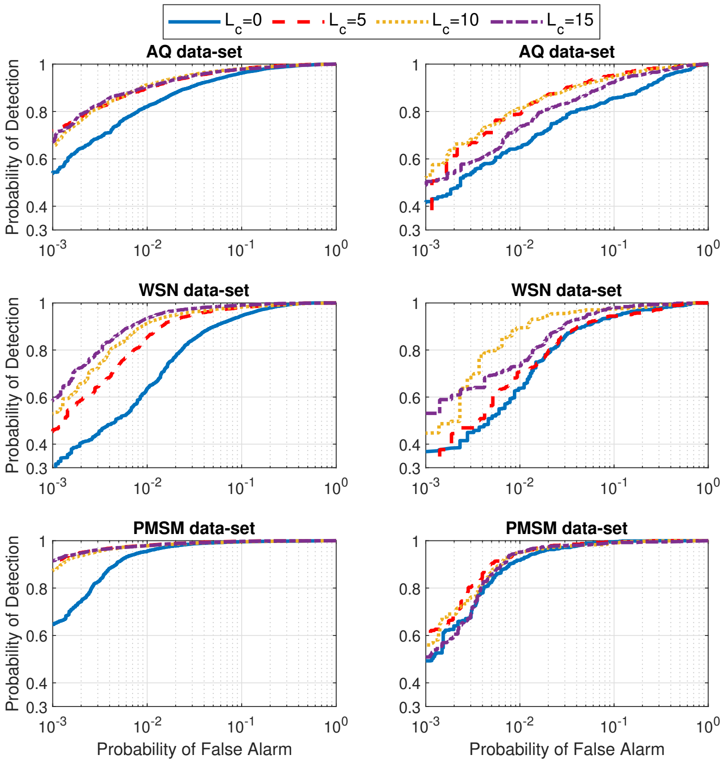
(a) Training set

(b) Test set

Figure 3.3: ROC curves of proposed SFDIA structure for all datasets under bias and drift faults.

to detect them with a very high probability for negligible false-alarm probability. Detection probability of bias faults is noticeably higher than drift faults over different false-alarm rates. This is originally due to the ramp-up phase of drift faults which takes the classifier more samples to detect faults. As illustrated in Fig. 3.3, WSN dataset has somewhat lower performance in comparison with the other two datasets (in case of drift faults). It is mainly because of very weak fault levels on this dataset according to its sensors' variation domains (see Tab. II in paper **P2** [33]). Conversely, detection performance of proposed architecture under strong faults is significantly higher than the detection performance under weak faults as shown in Fig. 3.3, which highlights the importance of detection and isolation of weak faults.

Figs. 3.4 and 3.5 demonstrate the effect of using time-delayed samples on the



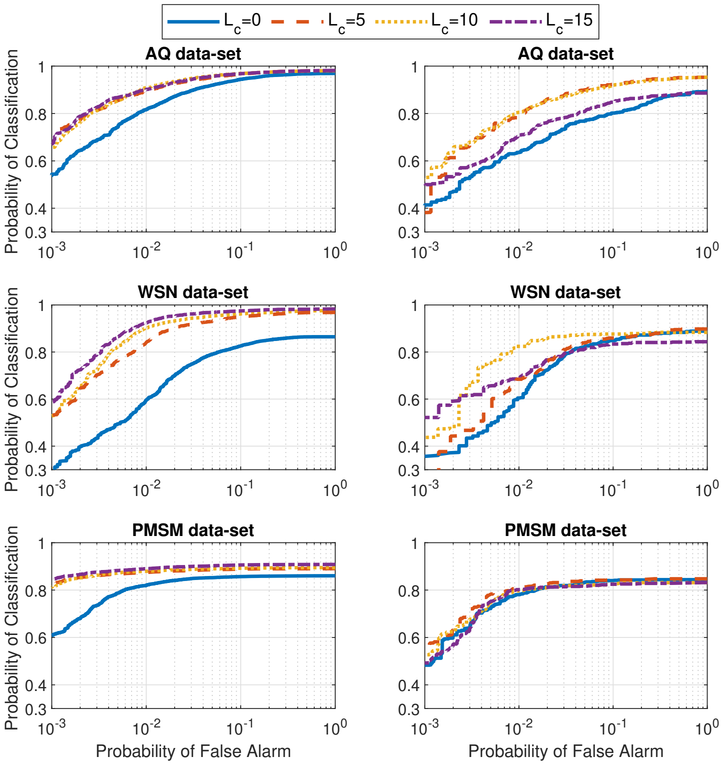
(a) Training set

(b) Test set

Figure 3.4: Detection performance of the classifier in configuration 2×15 for different number of previous time instants L_c in terms of ROC on each data-set.

classifier in the case of drift fault. There are certain improvements in detection performance and averaged classification (isolation) performance[1] when temporal correlation exists in sensor measurements. However, as it can be seen in both Fig. 3.4.(b) and 3.5.(b), the performance slightly reduces with increasing number of time delays ($L_c = 15$) due to the negligible temporal correlation between older samples and current sample in the measurements. Besides, in this scenario, increasing the window size should potentially lead to a performance improvement, however, a larger number of nodes in the hidden layers might be required to handle properly the increased number of input nodes. Differently, with a fixed network structure, increasing the window size might in practice saturate the learning cap-

[1] Averaged classification performance is the average of correct classification probability on all sensors in the dataset. Non-fault occurrence is considered a separate class.



(a) Training set **(b)** Test set

Figure 3.5: Averaged classification (isolation) performance of the classifier in configuration 2×15 for different number of previous time instants L_c in terms of ROC on each dataset.

ability.

3.1.2.3 Performance comparison

Table 3.2 compares the proposed architecture with two state-of-the-art techniques previously outlined in Sec. 2.1: (i) the SVM classifier [11] and (ii) the FCC NN [14] with 6 nodes. The SVM classifier has no control over the probability of false-alarm since it does not have any threshold mechanism. Hence, to provide a fair comparison, we tuned the threshold on the proposed architecture and on the FCC technique to achieve the same probability of false-alarm as the SVM classifier, and compared the probability of detection for all techniques in Tab. 3.2. Apparently, the detection performance of the proposed architecture outperforms the SVM technique for all fault types. The performance gap between these two tech-

Table 3.2: Detection accuracy of the proposed architecture compared to the SVM classifier and the FCC technique on the test set.

Dataset	Architecture	Metrics	Bias (%)		Drift (%)	
			Weak	Strong	Weak	Strong
AQ		P_f	2.82	0.01	2.32	0.17
	SVM	P_d	79.2	98.0	70.4	88.8
	FCC	P_d	98.5	-	85.2	87.9
	M-SFDIA	P_d	97.5	98.9	84.1	95.9
WSN		P_f	22.7	0.15	21.7	1.0
	SVM	P_d	95.9	98.5	88.2	90.3
	FCC	P_d	100	-	94.4	96.3
	M-SFDIA	P_d	100	98.9	98.2	94.2
PMSM		P_f	0.05	0.06	0.11	0.15
	SVM	P_d	34.9	92.3	31.8	77.7
	FCC	P_d	15.9	99.7	25.0	50.8
	M-SFDIA	P_d	58.1	99.8	56.0	96.2

niques in terms of detection accuracy becomes more evident under weak faults. More specifically, under weak drift fault for the PMSM dataset, the performance improvement in fault detection of the proposed architecture over the SVM technique is approximately 24.2%. The main reason lies in the fact that the SVM classifier takes raw-sensor data as input while the M-SFDIA architecture exploits the estimations of each sensor and feeds the residual data as input to the classifier which contains easy-to-interpret information about faults. The FCC technique exhibits similar detection performance as the proposed architecture over AQ and WSN datasets, while on the PMSM dataset, the proposed architecture turns out to be a better-performing SFDIA solution. In Tab. 3.2, the detection accuracy of the FCC technique with respect to the corresponding probability of false-alarm was not available for the WSN and AQ datasets under strong bias faults (as can be seen also in Fig. 3.6(a)). It is worth mentioning that the detection performance on the training set resembles those shown for the test set in Tab. 3.2.

As for the isolation task, the proposed architecture achieves significant gains over the FCC technique as observed in terms of classification performance shown in Fig. 3.6. More specifically, the proposed architecture takes advantage of MLP classifier while the FCC technique merely uses a sliding window mechanism. The relevance of the proposed architecture as an effective SFDIA scheme is apparent.

Finally, as for the accommodation task, Fig. 3.7 compares the accuracy of the virtual sensors which reveals better estimation capability of the MLPs from the

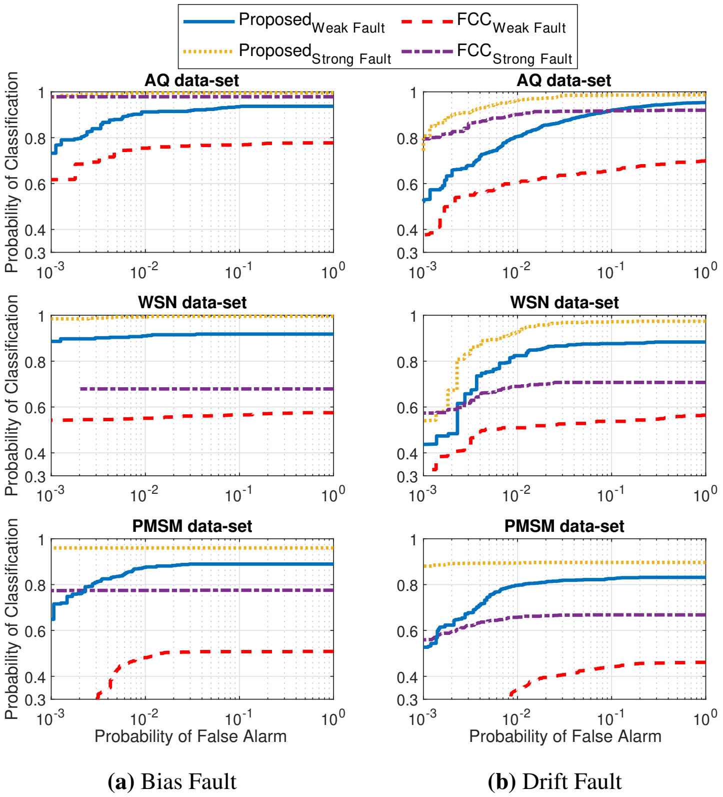
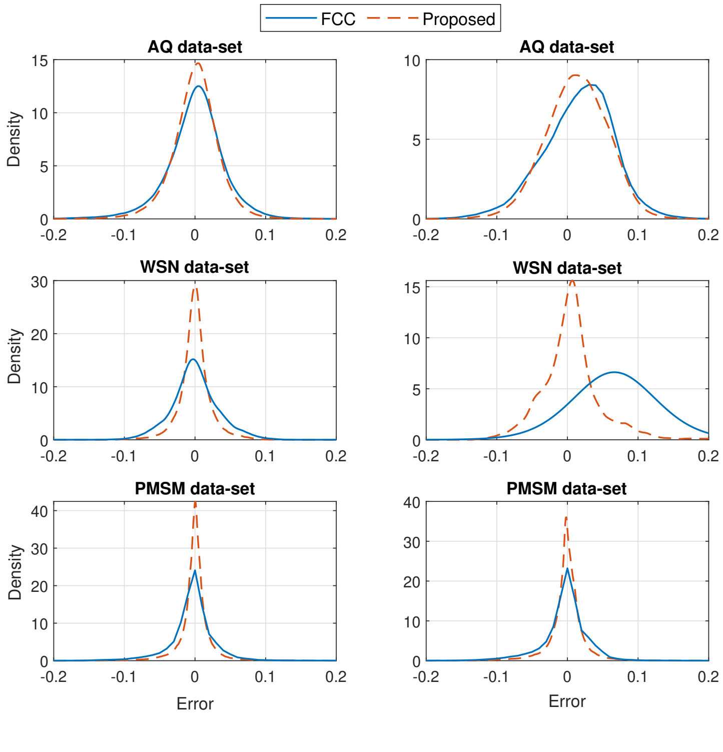


Figure 3.6: Averaged classification (isolation) performance comparison in terms of ROC for the test set on each dataset.

proposed architecture against the FCC NNs. The improvement is mainly due to the capability of the proposed technique to exploit temporal correlation. Finally, it is worth noticing that the isolation and accommodation performances of the SVM technique cannot be compared due to its incapability to classify and estimate faulty sensors.

3.1.2.4 Modular analysis

This section explores the impact of using different types of NN building blocks on M-SFDIA architecture to achieve the optimum configuration. Fig. 3.8 displays the statistics (median value, 95% confidence interval, and outliers) of the RMSE in the fault-free situation on the test set for each virtual sensor. `MLP`$^{\text{vs}}$ has the highest median over two out of four sensors (**S3** and **S4**), while `GRU-RS`$^{\text{vs}}$ and `CNN`$^{\text{vs}}$ outperform on average the other counterparts and provide the lowest RMSE value.



(a) Train Set

(b) Test Set

Figure 3.7: Accommodation performance comparison in terms of PDF of the error signals on each dataset.

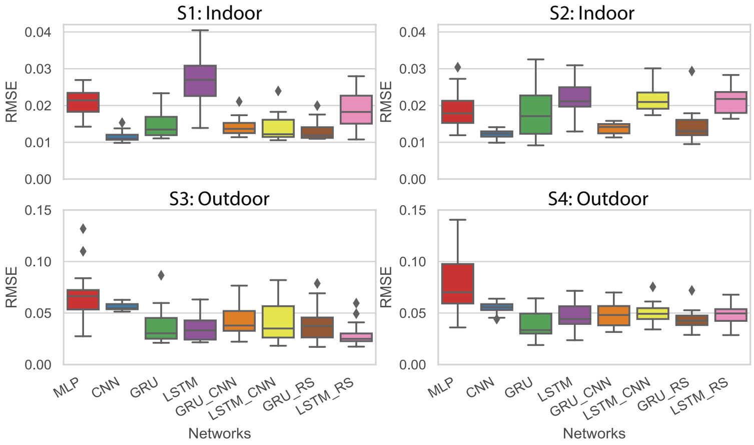
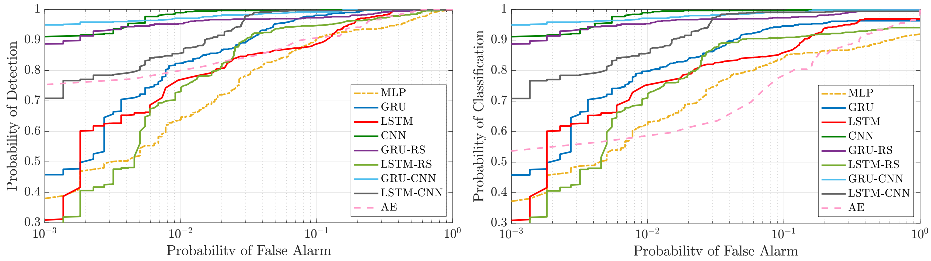


Figure 3.8: Box-plot of estimation RMSE for each virtual sensor on WSN dataset.



(a) Detection Performance.

(b) Isolation Performance.

Figure 3.9: Detection and isolation performance of different classifier models by using ROC curves on WSN dataset.

Fig. 3.9 shows the probabilities of detection and classification with respect to the probability of false-alarm (set via γ) for different classifiers[2], i.e. the 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 at capturing more complex spatio-temporal dependencies in the data.

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.

3.1.2.5 M-SFDIA performance on CCS systems

CCS is critical for climate-change policies and strategies targeting global warming within the Paris Agreement. Further, the proposed M-SFDIA framework has also been validated for the emerging CCS technology. A preliminary set of results are provided in Fig. 3.10, which demonstrates the efficacy of the proposed architecture in detecting the synthetically-generated bias faults. The results are performed using the real measurements from the temperature sensors installed on

[2] *Dashed* lines refer to the baseline M-SFDIA [33] and the state-of-the-art AE architecture [56]. *Solid* curves refer to different classifiers using the same residual-signals (i.e. computed via $\texttt{GRU-RS}^{\text{vs}}$).

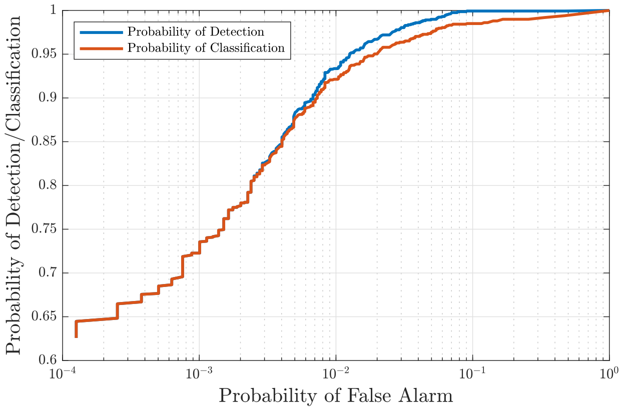


Figure 3.10: Detection and classification performance of the M-SFDIA scheme on the DeFACTO dataset.

the surface of the CO2 pipeline DeFACTO (described in publication **P6** [37]) with simulated faults. Figure 3.10 shows encouraging performance, however, these results are to be considered preliminary as more critical operational conditions need to be explored.

3.2 Real-Time and Modular SFDIA Architecture

Although the M-SFDIA has a promising modular architecture (from the estimators' design viewpoint), its main limitation is that it does not completely exploit the temporal correlations among sensors within the monitored system. Equally important, the M-SFDIA decision logic is designed to detect, isolate and accommodate only a single faulty sensor at a time.

In this section, the major motivation is to propose a machine-learning-based SFDIA architecture to exploit complete spatial and temporal correlations in the data collected from the sensors, which is in line with contributions **C2** (in Sec. 1.3). To this end, *a pair of regressors* are employed for each sensor to perform *estimation* and *prediction* operations of sensor measurements in the system. In the former case (M-SFDIA), each estimator is leveraging readings from other sensors only to obtain a virtual measurement. Conversely, each predictor plays a complementary role (to the estimator) by using only previous data from the sensor under consideration to obtain an analogous virtual measurement. Hence, their joint adoption enables the proposed architecture to ultimately exploit spatio-temporal correlation within the system, thus supporting nearly-instantaneous fault detection and isolation performance. Moreover, a controller in a feedback loop is preserving the

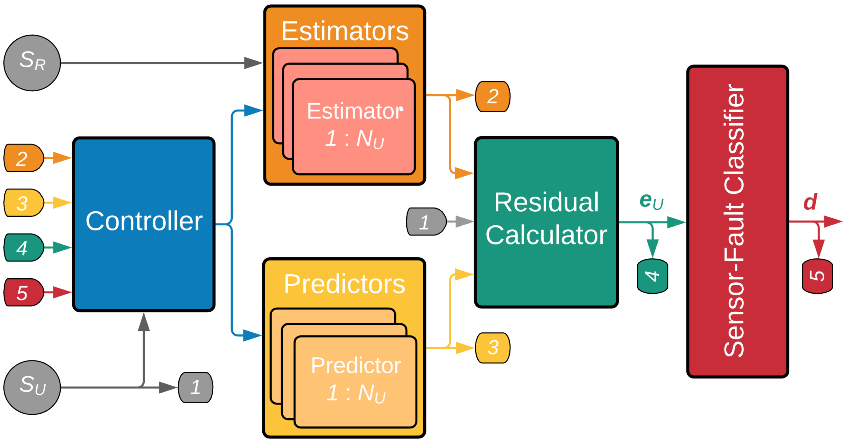


Figure 3.11: Block diagram of the second proposed SFDIA architecture.

performance of the proposed SFDIA architecture when faults occur.

The block diagram of the proposed SFDIA architecture is shown in Fig. 3.11. It consists of five building blocks (controller, estimators, predictors, residual calculator, classifier) arranged in *four layers*, whose function is explained as follows. The *first layer* contains two parallel blocks, the estimators block and the predictors block, each providing a virtual measurement for all the unreliable sensors in the system either regressed via other sensors' observations (i.e. the estimator) or based only on previous measurements of the same sensor under consideration (i.e. the predictor). The *second layer* is responsible for the computation of a discrepancy measure between the true and each calculated virtual measurement, usually in the form of a function of the residual signals. The *third layer* is fed with the aforementioned discrepancy measures and is able to perform a multidimensional classification to (a) detect a faulty condition and (b) identify the corresponding faulty sensors. Finally, the *fourth layer* controls the inputs of the blocks in the first layer in order to preserve estimators' and predictors' accuracy, by avoiding error propagation.

The present architecture improves over the proposed M-SFDIA in [33] where the main novelty is the introduction of the controller and the predictors. Despite the addition of these two modules, it is worth remarking that the proposed architecture retains the advantages of *modularity* and *real-time implementation*. Indeed, regarding the former property, the proposed approach allows the implementation of diversified ML techniques for different modules and a more flexible deployment, also taking computational/hardware limitations into account. Differently, regarding the latter property, each of the proposed modules can be flawlessly implemented in real-time since they are all based on a sliding window implementation. Finally, given the adoption of MLP-based solutions for the estimators/predictors (Sec. 3.2.1) and the classifier (Sec. 3.2.3), the proposed implementation also retains simplicity. The following subsections detail each of the four layers constituting the proposed approach.

3.2.1 First Layer: Estimation & Prediction

The first layer aims to model the unreliable sensors within the system and is based on *two subsystems*: (a) a bank of estimators and (b) a bank of predictors.

The bank of **estimators** is composed of N_U estimators (each associated to an unreliable sensor), each providing the estimation $\hat{x}_s[n]$ of the measurement (at current time step n) from its corresponding unreliable sensor $s \in \mathcal{S}_U$. Each estimator receives as input the vector $\boldsymbol{x}_{(s)}$ collecting all existing sensors readings (from current time step n back to L_e previous time steps using a sliding window mechanism) except the one from the sensor to be estimated $\{\mathcal{S}_U \cup \mathcal{S}_R - s\}$, i.e.

$$\hat{x}_s[n] = f_s^{(H_v,N_v)}(\boldsymbol{x}_{(s)}[n], \ldots, \boldsymbol{x}_{(s)}[n - L_e]) \ , \tag{3.4}$$

Previous time samples are fed into the estimators in order to exploit the *temporal correlation* among the input signals.

The bank of **predictors** operates a complementary approach. Each of the N_U predictors provides a prediction $\tilde{x}_s[n]$ of the measurement (at current time step n) from its corresponding unreliable sensor $s \in \mathcal{S}_U$. Each predictor receives as input the readings $x_s[\cdot]$ of the sensor to be predicted (from previous time step $n-1$ back to L_p previous time steps using a sliding-window mechanism), i.e.

$$\tilde{x}_s[n] = g_s^{(H_v,N_v)}\left(x_s[n-1], \ldots, x_s[n - L_p]\right) \ , \tag{3.5}$$

where $g_s^{(H_v,N_v)}(\cdot)$ denotes the function model of the MLP-based predictor for the sth sensor.

3.2.2 Second Layer: Residual Evaluation

The second layer computes the square of residual signals i.e. the difference of sensors reading with their respective estimation or prediction values, namely

$$e_{E,s}[n] = (x_s[n] - \hat{x}_s[n])^2, \tag{3.6}$$

$$e_{P,s}[n] = (x_s[n] - \tilde{x}_s[n])^2, \tag{3.7}$$

for each unreliable sensor $s \in \mathcal{S}_U$. Residual signals are used as input to the classifier in the third layer as they contain effective information for fault classification. It is worth noticing that the proposed SFDIA architecture enjoys modularity and generality: thus other discrepancy measures (other than that used in Eqs. (3.6) and (3.7)) may be adopted *without any substantial change* in the subsequent layers.

3.2.3 Third Layer: Classification

An MLP **classifier**, meant to work in real-time, is used for fault *detection* and the *identification* of the faulty sensors. Denoting $\boldsymbol{e}_U[n] = (e_{E,1}[n], \ldots, e_{E,N_U}[n], e_{P,1}[n], \ldots, e_{P,N_U}[n])^T$ the residual vector containing the residual signals of all N_U 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 $\boldsymbol{e}_U[n], \ldots, \boldsymbol{e}_U[n - L_c]$. Conversely, a decision vector $\boldsymbol{d}[n] = (d_1[n], d_2[n], \ldots, d_{N_U}[n])^T$ represents the output of the classifier and identifies which among the unreliable sensors are suspected to be in failure, i.e.

$$\boldsymbol{d}[n] = \boldsymbol{h}^{(H_c,N_c)}(\boldsymbol{e}_U[n], \ldots, \boldsymbol{e}_U[n - L_c]) \,, \tag{3.8}$$

where $\boldsymbol{h}^{(H_c,N_c)}(\cdot)$ denotes the function model of the MLP-based classifier, being H_c and N_c are the number of hidden layers and the number of neurons of the classifier, respectively.

Similar to M-SFDIA decision logic, $d_i[n] = 1$ (resp. $d_i[n] = 0$) represents the situation in which the system declares with maximum confidence the ith sensor to be faulty (resp. fault-free). As a consequence, a vector $\boldsymbol{d}[n] = \boldsymbol{0}_{N_U}$ indicates healthy operation of *all* the sensors within the system at time n.

Therefore, faulty sensors are identified via a threshold-based logic for each of the components of the decision vector. The considered threshold will be denoted γ in what follows. Different from M-SFDIA approach, herein faulty sensors are detected and identified/isolated when the entries of the decision vector $\boldsymbol{d}[n]$ exceed the threshold γ. Specifically, $\max_{s=1}^{N_U} d_s[n] \gtrless \gamma$ is used for detection. Accordingly, for the identification task, the set of identified faulty sensors (denoted with $\mathcal{I}_U$) is obtained as $\mathcal{I}_U \triangleq \{s \in \mathcal{S}_U \, : \, d_s[n] > \gamma\}$.

It is worth mentioning that, from overall SFDIA system perspective, the measurements from the sensors declared faulty are *replaced* (viz. *accommodated*) with their corresponding *estimates* in order to preserve system utility.

3.2.4 Fourth Layer: Control

The role of the control block is to preserve the performance of the proposed SFDIA method when faults occur. Referring to Fig. 3.11, this block operates at the beginning of each time step, and controls inputs-outputs of both estimators and predictors regarding the latest residual signals and the decision vector $\boldsymbol{d}[n-1]$.

The symbol $\phi_{E,s}$ (resp. $\phi_{P,s}$) denotes the average residual signal for the sth estimator (resp. predictor) computed with a moving average over a window of size L_r starting from time step $n-1$ while *excluding* the identified faulty time steps. The signal $\phi_{E,s}$ (resp. $\phi_{P,s}$) of the unreliable sensor s is used by the controller as a metric to define the estimation (resp. prediction) accuracy of the corresponding estimator (resp. predictor).

In the first step, after applying the proposed SFDIA scheme at time step $(n-1)$, the elements of the decision vector $\boldsymbol{d}[n-1]$ larger than a predefined threshold v identify faulty sensors for the controller. Then, the following process will be conducted at the beginning of each time step n. To keep the discussion simple, we will generically refer to sth sensor as the one identified as faulty.

As for the **predictor controlling** scheme, if the estimator's average residual signal $\phi_{E,s}$ is smaller than a certain value τ (i.e. the system tolerable level of deviation), the estimator output $\hat{x}_s[n-1]$ *replaces* the respective sensor input $x_s[n-1]$ to the corresponding predictor. In other words, the predictor in Eq. (3.5) will be then fed as:

$$\tilde{x}_s[n] = g_s^{(H_v,N_v)}(\underbrace{\hat{x}_s[n-1]}_{\text{replacement}}, \ldots, x_s[n-L_p]\,), \tag{3.9}$$

This logic is intended to use only those estimates whose quality is better than the faulty-data within the sth predictor.

As for the **estimator controlling** scheme, if the predictor's average residual signal $\phi_{P,s}$ smaller than both (i) the system tolerable level of deviation τ and (ii) $\phi_{E,s}$, the predictor output $\tilde{x}_s[n]$ is obtained and *replaces* the respective sensor input $x_s[n]$ (updates all estimators' input vectors except $\boldsymbol{x}_{(s)}[n]$) to the estimators. In other words, we have $\forall s_\star \in \mathcal{S}, s_\star \neq s$:

$$\hat{x}_{s_\star}[n] = f_{s_\star}^{(H_v,N_v)}(\underbrace{\tilde{\boldsymbol{x}}_{(s_\star)}[n]}_{\text{replacement}}, \ldots, \boldsymbol{x}_{(s_\star)}[n-L_e])\,, \tag{3.10}$$

where the vector $\tilde{\boldsymbol{x}}_{(s_\star)}[n]$ collects all existing sensors readings except for $s_\star$ and with sth reading being replaced by $\tilde{x}_s[n]$. Otherwise, if $\phi_{E,s}$ is smaller than the system tolerable level of deviation[3], the estimator output $\hat{x}_s[n]$ is obtained and replaces the respective sensor input $x_s[n]$ (updates all input vectors except $\boldsymbol{x}_{(s)}[n]$) to the estimators. Specifically, $\forall s_\star \in \mathcal{S}, s_\star \neq s$:

$$\hat{x}_{s_\star}[n] = f_{s_\star}^{(H_v,N_v)}(\underbrace{\hat{\boldsymbol{x}}_{(s_\star)}[n]}_{\text{replacement}} , \ldots , \boldsymbol{x}_{(s_\star)}[n - L_e]) \, , \tag{3.11}$$

where the vector $\hat{\boldsymbol{x}}_{(s_\star)}[n]$ collects all existing sensors readings except for $s_\star$ and with sth reading being replaced by $\hat{x}_s[n]$. This logic is intended to replace the input faulty-data with estimates/predictions whose accuracy are better than the input faulty-data (i.e. $x_{(s)}[n]$) to all estimators (except the corresponding sensor s estimator). We highlight that, in all three cases, *no architectural modification* (i.e. varying input size for the estimators and predictors) *is required* for the blocks of the proposed SFDIA method. Conversely, in the case of *no-fault detected*, this block merely slides the window forward in time to update both $\phi_{P,s}$ and $\phi_{E,s}$ by using the recent residual signals $\boldsymbol{e}_U[n-1]$.

It is worth remarking that substitution of faulty inputs with either estimated or predicted values maintains estimators' and predictors' accuracy (by avoiding error propagation) and results in better accommodation performance as well as increased detection rate.

The detailed configuration of NN models and training summary of the proposed architecture is given in publication **P5** [36].

3.2.5 Numerical Results

The effectiveness of the second proposed architecture for detection, isolation and accommodation of sensor faults has been assessed by means of a comprehensive analysis conducted on the three previously-described real-world datasets. Then, the working principle of the two relevant SFDIA baselines used for comparison is recalled (Sec. 3.2.5.1). Finally, the SFDIA performance is reported and discussed (Sec. 3.2.6).

3.2.5.1 Considered baselines

Results of the proposed approach in terms of detection, identification and accommodation performance are compared with *two* state-of-the-art architectures: (i) M-SFDIA [33] and (ii) AE [56].

Similar to the proposed method, our previous M-SFDIA proposal is able to detect

[3] In other words, the corresponding estimator is providing better accuracy than the corresponding predictor, i.e. $\phi_{E,s} < \phi_{P,s}$ and $\phi_{E,s} < \tau$.

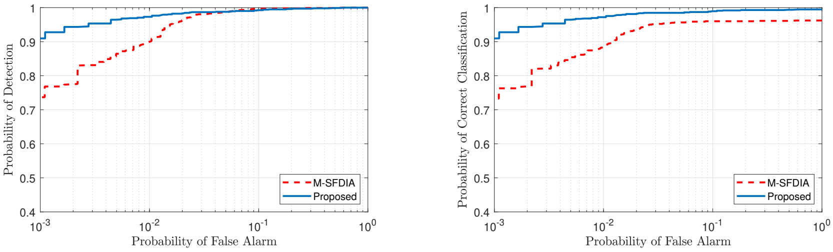
and isolate faulty sensors from patterns within the input residual signals. However, solely a bank of estimators is used to derive the residual signals, and to accommodate unreliable sensors in M-SFDIA method. Additionally, the controller block is absent in M-SFDIA. Furthermore, the original M-SFDIA's decision logic was designed to detect, isolate and accommodate only *up to one faulty sensor*. For this reason and for the sake of a fair comparison, the same decision logic as the proposed method was used (see Sec. 3.2.3) to enable the M-SFDIA method to detect, isolate and accommodate multiple sensors simultaneously.

Conversely, the AE-based architecture devised in [56] is based on a *two-stage* approach. Specifically, the first stage is represented by a (standard) AE to learn data correlations among sensors, and detect anomalies (viz. faults) by tracking the MSE between input and output of the AE. As for the accommodation task, a second stage based on a (supervised) denoising AE is then used to clean faulty data. It is worth noticing that the identification task for AE architecture was not addressed in the original work [56]. Indeed, in the aforementioned AE-based method, the overall MSE of input and output (reconstructed) vector of the first AE is compared to a predefined threshold for fault detection only. As opposed to the aforementioned decision logic, herein (for the sake of a fair comparison) the squared error between the corresponding input and output *for each entry* (viz. unreliable sensor) is traced. Then, this error is compared with a predefined threshold σ, enabling the AE method to both detect & identify the faulty sensors[4]. Specifically, similar to the proposed method, $\max_{s=1}^{N_U} e_{AE,s}[n] \gtrless \sigma$ is used for detection, where $e_{AE,s}[n]$ is the squared error for the sth unreliable sensor. Accordingly, for the identification task, the set of identified faulty sensors is obtained as $\mathcal{I}_U \triangleq \{s \in \mathcal{S}_U : e_{AE,s}[n] > \sigma\}$.

3.2.6 Performance Analysis and Comparison

First, in Fig. 3.12, *detection and classification* (i.e. detection plus isolation) performance of the proposed architecture in the absence of the controller block is being evaluated by means of the corresponding ROC curves. 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 $P_f = 10^{-2}$. Con-

[4] Numerical results (not shown for brevity) based on the original detection logic as [56], namely $\sum_{s=1}^{N_U} e_{AE,s}[n] \gtrless \sigma$ (and a matched identification logic, i.e. $\mathcal{I}_U \triangleq \{s \in \mathcal{S}_U : e_{AE,s}[n] > \sigma/N_U\}$) highlighted worse performance than the considered variant, due to the inability to cope with weak (and transient) faults.



(a) Detection Performance

(b) Classification Performance

Figure 3.12: Detection performance and averaged correct classification performance of proposed architecture in the absence of the controller by using ROC curves for WSN dataset.

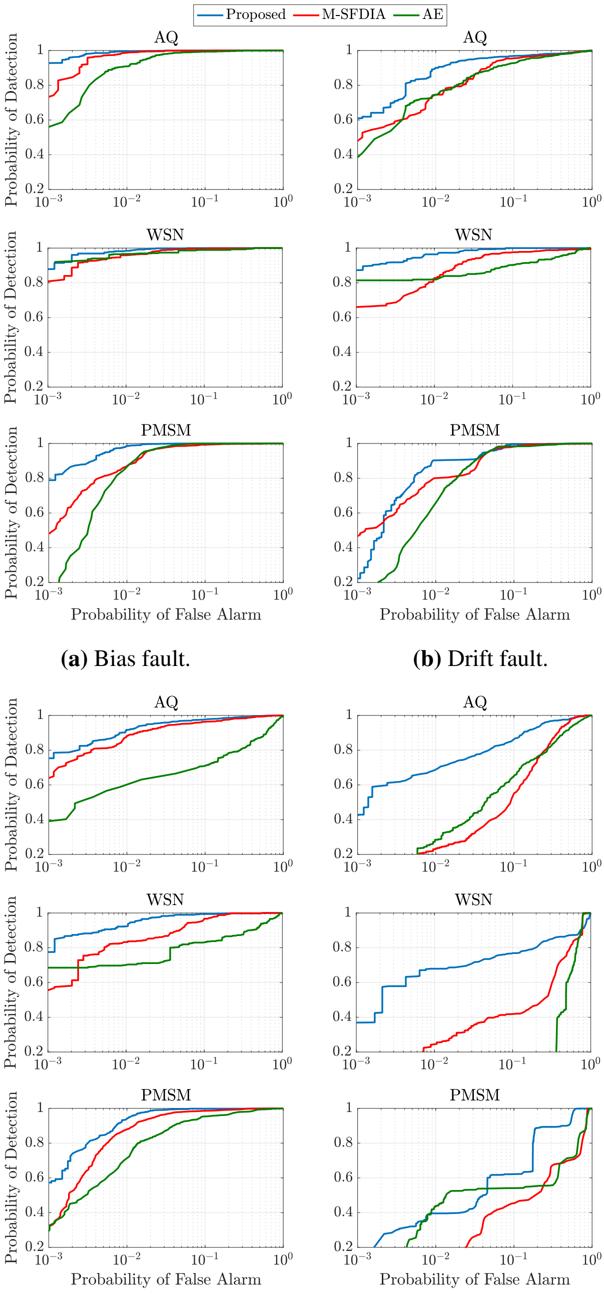
versely, regarding the classification task (under the same false-alarm constraint), the M-SFDIA (resp. the proposed) architecture achieves a probability of correct classification close to 0.90 (resp. 0.98). The above results highlight ideal identification performance for our approach, i.e. no additional errors caused by identifying the correct source of fault. More analysis for the proposed scheme in the absence of the controller block can be found in Publication **P3** [34].

Fig. 3.13 illustrates *fault detection performance* in terms of probability of detection vs. probability of false-alarm in the presence of the controller block. In this case, a fault rate[5] $F_R = 0.1$ is considered. Also, ROC performance is reported *separately* for each of the three datasets and for all four fault typologies considered. It is evident that the proposed architecture outperforms the two baselines for all four fault types. Specifically, the best detection rate is attained on AQ dataset when bias faults are present. Also, for all architectures, detection accuracy under bias faults appears to be generally higher than the other types of faults. Moreover, as can be seen, AE architecture fails to detect freeze faults on the WSN dataset. Indeed, drift and freeze faults are “trickier” to detect since they slowly appear in the system and have a less-appreciable effect on spatio-temporal correlations within the system.

Delving into real-time performance of SFDIA architectures, in Tab. 3.3 a *detection delay analysis*[6] for the fixed false-alarm rate of 10^{-2} is reported. Specifically, the *expected detection delay* is evaluated, defined as the average number of samples needed by an SFDIA architecture to detect a faulty sensor. The latter delay is indeed another important indicator of the SFDIA framework performance, which has a crucial effect on DTs functionality. In the experiments, the fault rate is set

[5] Fault rate refers to the ratio between the number of faulty and non-faulty samples.

[6] Every span of simultaneous faults is considered as a unified fault.



(a) Bias fault.

(b) Drift fault.

(c) Noise fault.

(d) Freeze fault.

Figure 3.13: Detection performance in terms of ROC curves for all architectures over different fault types.

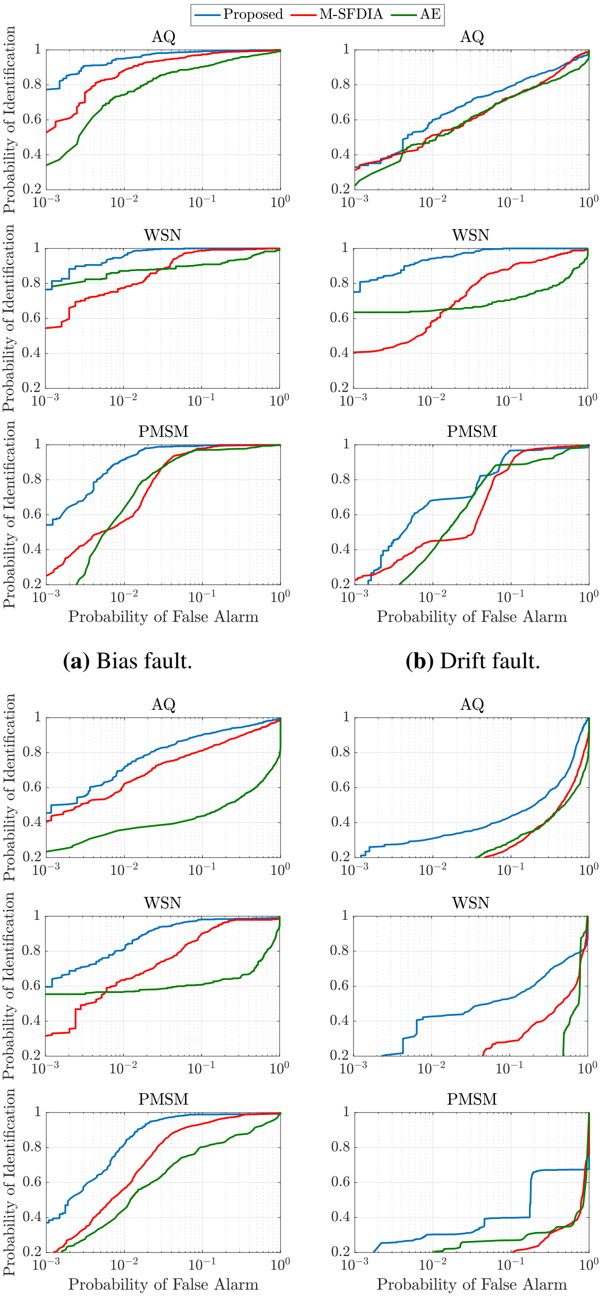
Table 3.3: Detection delay Analysis. Results refer to bias and drift faults and are in the format avg. (± std.) delayed samples obtained for a fault rate $F_R = 0.5$.

Dataset	Fault type	Proposed	M-SFDIA	AE
AQ	Bias	0.06 (± 0.30)	0.39 (± 1.11)	0.50 (± 1.33)
	Drift	1.77 (± 1.69)	2.33 (± 2.09)	4.04 (± 2.84)
WSN	Bias	0.28 (± 0.91)	0.84 (± 1.47)	0.31 (± 1.08)
	Drift	0.72 (± 1.12)	2.12 (± 2.14)	2.73 (± 2.12)
PMSM	Bias	0.10 (± 0.53)	1.24 (± 1.94)	3.61 (± 3.62)
	Drift	**0.67 (± 1.21)**	**3.40 (± 2.68)**	**8.30 (± 4.85)**

to $F_R = 0.5$ to generate a sufficient number of fault events allowing to obtain a reliable estimate of the aforementioned metric. Results highlight that the proposed architecture achieves the *lowest detection delay* in comparison to the state-of-the-art for all datasets and fault types considered. Specifically, the average detection delay for the proposed architecture is confined below 1 sample (except for the AQ dataset with drift fault-types), whereas the other two architectures *always require a longer span to detect fault(s) within the system.* The most evident performance difference is observed on the PMSM dataset for drift faults (boldface in Tab. 3.3). Indeed, in the latter case, the proposed architecture detects weak faults on average after 0.67 samples whilst MSFDIA and AE architectures take on average 3.40 and 8.30 samples to detect the same faults, respectively. The reported difference corresponds to a *faster detection for our proposal* of more than 5× and 12× than the MSFDIA and AE architectures, respectively. The reduced detection delay of the proposed architecture is mainly due to the *joint* exploitation of estimation and prediction blocks (cf. Sec. 3.2.1), as they provide complementary (residual) information for the classifier.

The corresponding sensor-averaged identification performance (under the same fault rate) is depicted in Fig. 3.14. Here in Fig. 3.14, the proposed architecture performs even better over other methods since it manages to reduce fault propagation within the architecture itself and avoid functionality degradation using the controlling block. Replacing faulty sensors with their estimates or predictions by the controller provides the classifier with easier interpretative residual signals.

The *accommodation performance* in terms of the RMSE is shown in Fig. 3.15, where fault rate $F_R = 0.1$ is considered. Herein the term *error* means the difference between sensor healthy values before adding the fault and the accommodated values provided by the SFDIA architecture (or the original values, in the case



(a) Bias fault.

(b) Drift fault.

(c) Noise fault.

(d) Freeze fault.

Figure 3.14: Averaged identification (isolation) performance in terms of ROC curves for all architectures over different fault types.

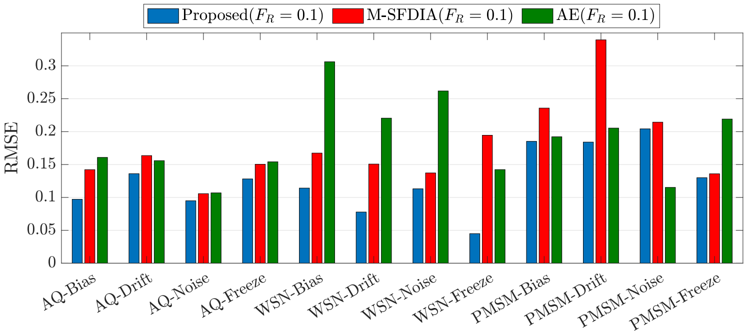
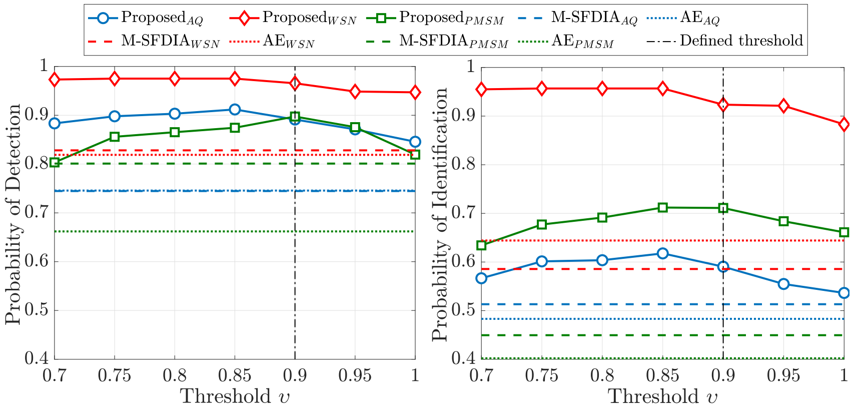


Figure 3.15: Comparison of accommodation performance in term of RMSE ($P_f = 10^{-2}$).



(a) Detection Sensitivity

(b) Identification Sensitivity

Figure 3.16: Impact of threshold (υ) on the detection and identification accuracy ($P_f = 10^{-2}$). Threshold $\upsilon = 1$ associated to a zero-effect of the controller (i.e. off-circuit controller).

of an undetected/unidentified fault). First of all, it is apparent that the proposed architecture outperforms the M-SFDIA architecture by presenting more accurate replacements for faulty data. The reason is that the proposed architecture relies on a combined estimator/predictor pair for each sensor and a controller block to continuously improve the accommodation performance by modifying their inputs based on the decision vector obtained from the classifier in a closed-loop fashion. Conversely, the M-SFDIA architecture does not take advantage of these excessive data. Finally, the proposed architecture outperforms AE-based SFDIA on all three available datasets (except for PMSM-Noise), with a higher improvement (viz. RMSE reduction) in the case of WSN dataset.

To deepen the investigation of the controller block, a *sensitivity analysis* was also performed, focusing on detection and identification performance of the proposed architecture, by varying the threshold υ during the test phase. More specifically, Fig. 3.16 shows the detection and identification performance of the proposed method with respect to the threshold υ. To better apprehend the impact of the threshold υ, the detection and identification performance of the state-of-the-art counterparts were reported as a lower bound. Results highlight quite smooth performance trends on the three datasets with respect to the threshold υ. Interestingly, the predefined threshold $\upsilon = 0.9$ based on the validation set is pretty near to the optimum value on the test set.

3.3 Deep Recurrent Graph Convolutional Architecture (DRGCA)

Although data-driven approaches have received large attention in the recent years since they do not require an explicit formulation of the relationships between sensors (as opposed to model-based approaches, e.g. [81, 82]), they suffer several disadvantages:

- the performance of basic ML methods heavily depends on the non-linearity, dimensionality, and heterogeneity of the system;
- shallow NNs suffer weak generalization, i.e. they are unable to properly capture complex features within the data;
- weak scalability, i.e. the computational complexity increases exponentially with the system network size and usually this is paired with performance degradation.

M-SFDIA deep networks proposed so far, besides difficulties of optimization, have limitations in dealing with *non-Euclidean spatial structures* (e.g. traffic systems) due to *modularity* design. In this section, we propose a data-driven-based DRGCA for SFDIA of *large-scale IoT networks*. Recently, Graph NNs (GNNs) have gained significant attention as a promising graph-based paradigm to perform fault detection. GNNs are capable to exploit effectively both temporal and spatial correlations among neighboring nodes (sensors) in large-size IoT systems, thus providing excellent accuracy in fault diagnosis. GCNs are feed-forward NNs with convolution operation generalized to graphs of arbitrary structure [83]. GCNs have been used successfully for drug synthesis, action recognition, image classification, link prediction, load prediction, and fault diagnosis [84, 85, 86, 87]. Although GNNs have

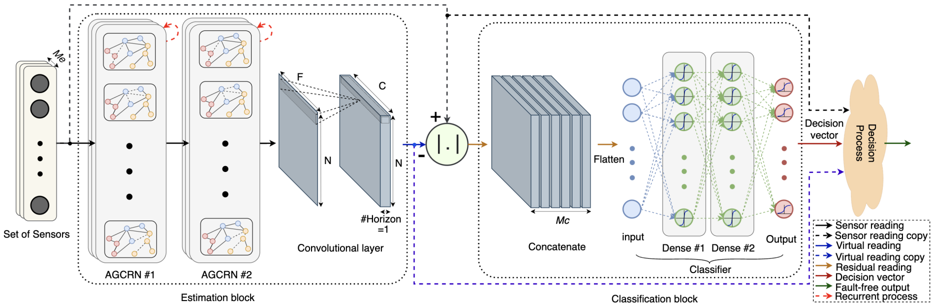


Figure 3.17: Block diagram of the conceptual SFDIA framework, which consists of two main blocks: estimation and classification.

been recently considered for anomaly detection [88], they are mostly unexplored within the SFDIA framework. Some recent works have explored GNN classifiers for fault detection and classification of power transformers [86], graph deviation networks (GDNs) for sensor anomaly detection [89], and adaptive graph convolutional recurrent network (AGCRN) for traffic forecasting [90].

In this work, we exploit the AGCRN in a *denoising configuration* (i.e. reconstruction of data from falsified input) as the building block for developing reliable virtual sensors. This configuration also assists the NN to better explore existing inter-dependencies among neighbouring sensors. Subsequently, the residuals (i.e. the difference between the readings from the real sensors and the virtual sensors) are concatenated and processed by a classifier in order to detect and identify the presence of faulty sensor(s). Furthermore, the virtual readings are employed to accommodate the isolated (viz. identified) faulty sensor(s). Accordingly, this research is aligned with research objective **O3** and the main contributions of this work are summarized under contributions **C3** (in Sec. 1.3).

3.3.1 Proposed DRGCA for SFDIA

We consider a large-size sensor system made of N correlated sensors, i.e. $N \gg 1$, each measuring C parameters. The measured parameters of nth sensor at time k are denoted with $\boldsymbol{x}_n[k] \in \mathbb{R}^C$, while the matrix $\boldsymbol{X}[k] = \{\boldsymbol{x}_1[k], \ldots, \boldsymbol{x}_N[k]\}^T \in \mathbb{R}^{N \times C}$ collects the recordings of all the N nodes at same time instant. While observing the stream of measurements $\ldots, \boldsymbol{X}[k-1], \boldsymbol{X}[k], \boldsymbol{X}[k+1], \ldots$, a subset of these sensors (corresponding to the rows of these matrices) may be subject to weak and transient faults. Employing SFDIA is necessary to make DTs reliable when operating under faulty conditions.

As shown in Fig. 3.17, the proposed SFDIA architecture is made of *two* NN-based blocks: (i) the estimation block and (ii) the classification block. Finally, the architecture is topped with (iii) a threshold-based decision & accommodation logic. The input to the *estimation block* of the proposed architecture is the set of readings from all the sensors within a sliding window. The estimation block models virtual sensors of *all* the sensors within the system. The *classification block* provides probabilistic predictions on the faulty condition of each sensor on the basis of the residual between each sensor and its corresponding virtual sensor. The *decision process* detects faulty conditions and identifies faulty sensors: once the faulty sensor(s) is (are) identified, the proposed architecture isolates the faulty sensors (i.e. position in failure status) and accommodates their measurements with the associated virtual readings to the DT throughout the process. High-level specification for each block is provided in what follows.

3.3.1.1 Estimation Block

This block aims to model the sensors (i.e. design the virtual sensors) within the system and is composed of a single multi-dimensional estimator providing the estimates $\hat{\boldsymbol{X}}[k] \in \mathbb{R}^{N\times C}$ of the present readings (at time k). The estimator receives as input a series of previous (time-correlated) sensors readings $\{\boldsymbol{X}[k-1], \ldots, \boldsymbol{X}[k-M_e]\}$ over a window of size M_e, a tunable hyperparameter of the proposed estimator. Equally important, it is designed to capitalize the spatial correlation among sensors via the graph $\mathcal{G}$, i.e.

$$\hat{\boldsymbol{X}}[k] = \boldsymbol{f}_{\varsigma}(\boldsymbol{X}[k-1], \ldots, \boldsymbol{X}[k-M_e]; \mathcal{G}(\varsigma)) \, , \tag{3.12}$$

where $\boldsymbol{f}_{\varsigma}(\cdot) : \mathbb{R}^{N\times C\times M_e} \rightarrow \mathbb{R}^{N\times C}$ denotes the function model of the NN-based estimator which models the current sensors readings and ς collects its trainable parameters. The notation $\mathcal{G}(\varsigma)$ in Eq. (3.12) underlines that we aim at learning *also* the graph structure of the system *during the training phase*.

3.3.1.2 Classification Block

As shown in Fig. 3.17, the classification block is made of an NN-based classifier meant to work in real-time to *detect* fault(s) and also *identify* the faulty sensor(s). To accomplish this task, the classifier relies on the concept of *residuals*, i.e. the absolute difference between sensors reading and their associated virtual reading, namely

$$\boldsymbol{\Delta}[k] = |\hat{\boldsymbol{X}}[k] - \boldsymbol{X}[k]| \, , \qquad \boldsymbol{\Delta}[k] \in \mathbb{R}^{N\times C} \, , \tag{3.13}$$

where the absolute operation $|\cdot|$ should be interpreted elementwise. We highlight that DRGCA relies on a *decoupled* design between estimation/classification blocks: thus other discrepancy measures other than Eq. (3.13) may be adopted without substantial changes in the subsequent layers.

Accordingly, the proposed classifier collects a concatenated sequence of residuals as inputs, namely $\{\boldsymbol{\Delta}[k], \boldsymbol{\Delta}[k-1], \ldots\}$, and based on the above inputs, a soft decision vector $\boldsymbol{d}[k] = \begin{bmatrix} d_1[k] & \cdots & d_N[k] \end{bmatrix}^T$ is provided as the output at time k. Therein, each element of the decision output $d_n[k] \in [0,1],\ n = 1, \ldots, N$, refers to a pseudo-probability of the nth sensor to be faulty at the corresponding time instant. More specifically, the considered classifier is modeled as follows:

$$\boldsymbol{d}[k] = \boldsymbol{h}_{\boldsymbol{\vartheta}}(\boldsymbol{\Delta}[k], \ldots, \boldsymbol{\Delta}[k - M_c + 1]) , \tag{3.14}$$

where $\boldsymbol{h}_{\boldsymbol{\vartheta}}(\cdot) : \mathbb{R}^{N \times C \times M_c} \to \mathbb{R}^N$ denotes the function model of the NN-based classifier and $\boldsymbol{\vartheta}$ collects the classifier trainable parameters. The model accepts residuals using a sliding window mechanism with a memory of size M_c, a tunable hyperparameter of the proposed approach.

3.3.1.3 Decision Process

The value of decision element $d_n[k]$ is assumed to represent *accurately* the architecture confidence in declaring the nth sensor to be faulty at time k, with $d_n[k] = 0$ (resp. $d_n[k] = 1$) being the utmost confidence on declaring the nth sensor non-faulty (resp. faulty). Consequently, *fault detection* is performed by checking if any entries of the decision vector $\boldsymbol{d}[k]$ exceed a given threshold γ, namely $\max_{n=1}^{N} d_n[k] > \gamma$. Consistently, *fault identification* is based on the set of indices $\mathcal{I} \triangleq \{n \in \{1, \ldots, N\} : d_n[k] > \gamma\}$. Finally, the declared faulty sensors after identification are *accommodated* (viz. isolated and replaced) by their associated virtual sensors in real-time to preserve the DT functionality. More specifically:

$$\boldsymbol{x}_s[k] \to \hat{\boldsymbol{x}}_s[k] \qquad \forall s \in \mathcal{I} \tag{3.15}$$

where $\hat{\boldsymbol{x}}_s[k]$ denotes the sth row of $\hat{\boldsymbol{X}}[k]$, i.e. the sth virtual sensor. We underline that the proposed SFDIA architecture runs in "open loop", i.e. accommodated measurements are not fed back into the estimation block. This is to grant decoupled design and avoid complex transients when a fault is detected/identified.

The following two subsections are devoted to the definition of the NNs (including their training phase) implementing the estimation ($\boldsymbol{f}_{\varsigma}(\cdot)$, Sec. 3.3.2) and the classification blocks ($\boldsymbol{h}_{\boldsymbol{\vartheta}}(\cdot)$, Sec. 3.3.3), respectively.

3.3.2 NN-Based Estimation

In the proposed DRGCA, the AGCRN layer is adopted as the stepping stone for the design of the estimation block (and thus model the whole set of virtual sensors) in the proposed SFDIA architecture [90]. Indeed, AGCRN addresses *three* strict *limitations* of GCNs, via the following *advancements*:

Node-specific patterns: GCN-based models are designed to effectively capture the shared spatial patterns (i.e. inter-dependencies) among sensors within the system. Indeed, having shared learnable-filter parameters $\boldsymbol{\Theta} \in \mathbb{R}^{C \times F}$ is quite useful to reduce the number of parameters while remaining on obtaining the prominent shared dependencies among sensors. Except for the shared patterns, the GCN fails to apprehend possible *diversified* node-specific patterns. On the contrary, assigning trainable parameters on each node level (i.e. a tensor $\boldsymbol{\Theta} \in \mathbb{R}^{N \times C \times F}$ with non-parametric dependence) would fit the bill, but unfortunately, drastically increases the network size. Hence, to reach a reasonable compromise, $\boldsymbol{\Theta}$ is factorized as $\boldsymbol{\Theta} = \boldsymbol{E}_g \otimes \boldsymbol{W}_g$, where: (i) $\boldsymbol{E}_g \in \mathbb{R}^{N \times l}$ is a node embedding matrix, where $l \ll N$ is the embedding dimension; (ii) $\boldsymbol{W}_g \in \mathbb{R}^{l \times C \times F}$ is a weight pool tensor.[7] Similarly, the additive learnable bias matrix $\boldsymbol{B} \in \mathbb{R}^{N \times F}$ is factorized as $\boldsymbol{B} = \boldsymbol{E}_g \boldsymbol{B}_g$, where $\boldsymbol{B}_g \in \mathbb{R}^{l \times F}$ denotes the bias pool matrix. Specifically, we have:

$$\boldsymbol{Z} = \left(\widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{X} \right) \otimes (\boldsymbol{E}_g \otimes \boldsymbol{W}_g) + \boldsymbol{E}_g \boldsymbol{B}_g \, . \tag{3.16}$$

where the entries of $\boldsymbol{Z}$ are obtained by interpreting the tensor product $\otimes$ as $\{\boldsymbol{Z}\}_{ik} = \sum_j \left(\widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{X} \right)_{ij} (\boldsymbol{E}_g \boldsymbol{W}_g)_{ijk}$, where $\left(\widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{X} \right) \in \mathbb{R}^{N \times C}$ and $(\boldsymbol{E}_g \boldsymbol{W}_g) \in \mathbb{R}^{N \times C \times F}$.

Learned adjacency matrix: the graph convolution operation (i.e. Eq. (2.5)) is completely dependent on the pre-defined adjacency matrix $\widetilde{\boldsymbol{A}}$ (as $\widetilde{\boldsymbol{D}}$ can be readily obtained from the former) to capture the spatial dependencies. The adjacency matrix is usually obtained by utilizing (intuitive) notions of similarity and/or distance functions [91, 92]. Unfortunately, the pre-defined graph generated in the aforementioned fashion is unable to contain domain knowledge of spatial dependencies and, equally important, is not related to the considered task. AGCRN learns the spatial dependencies by introducing an embedding matrix $\boldsymbol{E}_a \in \mathbb{R}^{N \times l_a}$. Then, multiplication of randomly-initialized embedding matrix $\boldsymbol{E}_a$ and its transpose $\boldsymbol{E}_a^T$ would infer the GCN about the spatial inter-dependencies between nodes, i.e.

$$\widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} = \mathrm{softmax}(\mathrm{ReLU}(\boldsymbol{E}_a \boldsymbol{E}_a^T)) \, , \tag{3.17}$$

where l_a is the node embedding dimension, $\mathrm{ReLU}(\cdot)$ function is used to force non-negative matrix and $\mathrm{softmax}(\cdot)$ function is utilized to normalize column-wise the final adaptive matrix.

Complex temporal correlations: AGCRN integrates the (node-specific and

[7] We underline that in such case $\boldsymbol{\Theta}$ is obtained as a matrix-tensor multiplication between $\boldsymbol{E}_g$ and $\boldsymbol{W}_g$ by contracting the second dimension of the matrix with the first dimension of the tensor according to Einstein notation, i.e. $\{\boldsymbol{\Theta}\}_{ik\ell} = \sum_j \{\boldsymbol{E}_g\}_{i,j} \{\boldsymbol{W}_g\}_{j,k,\ell}$.

graph-adaptive) GCN layer with the concept of GRU [90] to capture also the complex and long-term temporal correlations.

Capitalizing the aforementioned advancements, the AGCRN layer is then formally defined as

$$\begin{aligned}
\hat{\boldsymbol{A}} &= \mathrm{softmax}(\mathrm{ReLU}(\boldsymbol{E}\boldsymbol{E}^T)) \\
\boldsymbol{Z}[k] &= \sigma\left(\hat{\boldsymbol{A}}[\boldsymbol{X}[k];\boldsymbol{H}[k-1]] \otimes (\boldsymbol{E}\otimes\boldsymbol{W}_z) + \boldsymbol{E}\boldsymbol{B}_z\right) \\
\boldsymbol{R}[k] &= \sigma\left(\hat{\boldsymbol{A}}[\boldsymbol{X}[k];\boldsymbol{H}[k-1]] \otimes (\boldsymbol{E}\otimes\boldsymbol{W}_r) + \boldsymbol{E}\boldsymbol{B}_r\right) \\
\hat{\boldsymbol{H}}[k] &= \tanh\left(\hat{\boldsymbol{A}}[\boldsymbol{X}[k];\boldsymbol{R}[k]\odot\boldsymbol{H}[k-1]] \otimes \left(\boldsymbol{E}\otimes\boldsymbol{W}_{\hat{h}}\right) + \boldsymbol{E}\boldsymbol{B}_{\hat{h}}\right) \\
\boldsymbol{H}[k] &= \boldsymbol{Z}[k]\odot\boldsymbol{H}[k-1] + (\boldsymbol{1}_{N\times F} - \boldsymbol{Z}[k])\odot\hat{\boldsymbol{H}}[k]\,,
\end{aligned} \tag{3.18}$$

where $\boldsymbol{E}\in\mathbb{R}^{N\times l}$, $\boldsymbol{W}_z\in\mathbb{R}^{l\times(C+F)\times F}$, $\boldsymbol{W}_r\in\mathbb{R}^{l\times(C+F)\times F}$, $\boldsymbol{W}_{\hat{h}}\in\mathbb{R}^{l\times(C+F)\times F}$, $\boldsymbol{B}_z\in\mathbb{R}^{l\times F}$, $\boldsymbol{B}_r\in\mathbb{R}^{l\times F}$ and $\boldsymbol{B}_{\hat{h}}\in\mathbb{R}^{l\times F}$ are the trainable parameters of the AGCRN. In the GRU-inspired layer of Eq. (3.18), the matrices $\boldsymbol{Z}[\cdot]\in\mathbb{R}^{N\times F}$, $\boldsymbol{R}[\cdot]\in\mathbb{R}^{N\times F}$ and $\hat{\boldsymbol{H}}[\cdot]\in\mathbb{R}^{N\times F}$ represent the update gate, the reset gate, and candidate activation matrix, respectively. In the above equation the tensor products $\boldsymbol{E}\otimes\boldsymbol{W}_z$, $\boldsymbol{E}\otimes\boldsymbol{W}_r$ and $\boldsymbol{E}\otimes\boldsymbol{W}_{\hat{h}}$ have analogous meaning as $\boldsymbol{E}_g\otimes\boldsymbol{W}_g$ in Eq. (3.16). The same reasoning applies for the products $\hat{\boldsymbol{A}}[\boldsymbol{X}[k];\boldsymbol{H}[k-1]]\otimes(\boldsymbol{E}\otimes\boldsymbol{W}_z)$, $\hat{\boldsymbol{A}}[\boldsymbol{X}[k];\boldsymbol{H}[k-1]]\otimes(\boldsymbol{E}\otimes\boldsymbol{W}_r)$ and $\hat{\boldsymbol{A}}[\boldsymbol{X}[k];\boldsymbol{R}[k]\odot\boldsymbol{H}[k-1]]\otimes\left(\boldsymbol{E}\otimes\boldsymbol{W}_{\hat{h}}\right)$ when compared with $\left(\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{X}\right)\otimes(\boldsymbol{E}_g\otimes\boldsymbol{W}_g)$ in the same Eq. (3.16). Accordingly, the AGCRN output at time k is represented by the matrix $\boldsymbol{H}[k]\in\mathbb{R}^{N\times F}$. The matrix $\hat{\boldsymbol{A}}\in\mathbb{R}^{N\times N}$ is the (estimated) pseudo-Laplacian of graph $\mathcal{G}$, whereas $\sigma(\cdot)$ and $\tanh(\cdot)$ are the (entry-wise) sigmoid and hyperbolic tangent activation functions, respectively. In order to reduce the number of learning parameters, AGCRN unifies the embedding matrices associated to node-specific patterns and graph structure (i.e. $\boldsymbol{E}_g$ and $\boldsymbol{E}_a$) into a *single embedding matrix* $\boldsymbol{E}$.

To achieve accurate virtual sensors, the corresponding NN-based estimator $\boldsymbol{f}_\varsigma(\cdot)$ is thus composed as follows. We stack two AGCRN layers to be able to extract complex spatial and temporal correlations among sensors. The output (viz. input) of the first (resp. second) ACGRN layer is the evolution of the state matrix $\boldsymbol{H}_1[k-M_e],\ldots,\boldsymbol{H}_1[k]$. Conversely, from the output of the second AGCRN, we extract only the most recent form of the state matrix, i.e. $\boldsymbol{H}_2[k]$ (a usual practice when stacking multiple recurrent layers). A two-dimensional convolutional layer[8] is then applied at the output of the second AGCRN layer to directly project the

[8] One-dimensional convolutional layer and linear layers were observed to perform considerably worse than the two-dimensional convolutional layer.

representation from $\boldsymbol{H}_2[k] \in \mathbb{R}^{N\times F}$ (number of sensors by number of AGCRN features) to obtain the estimate $\hat{\boldsymbol{X}}[k] \in \mathbb{R}^{N\times C}$ (number of sensors by number of node parameters).

The MAE loss function is utilized to train the estimator, i.e.

$$\mathcal{L}_{\text{est}}\left(\varsigma\right) = \frac{1}{w}\sum_{j=0}^{w-1}\left\|\hat{\boldsymbol{X}}_j(\varsigma) - \boldsymbol{X}_j\right\|_1, \tag{3.19}$$

where w is the number of samples in each batch, $\boldsymbol{X}_j$ denotes the fault-free readings of sensors and $\hat{\boldsymbol{X}}_j(\varsigma)$ refers to the corresponding estimate. The estimation block is learned in the so-called *denoising configuration*: the NN is then trained to predict the fault-free $\boldsymbol{X}[k]$ even in the presence of faulty sensors. Such configuration helps the model learn the latent representation of data and make a robust recovery of the clean original data. Finally, the Adam optimization algorithm [93] is used to optimize the above loss.

3.3.3 NN-Based Classification

In the proposed DRGCA, a deep feed-forward (viz. MLP) classifier was selected to implement the mapping $\boldsymbol{h}_{\boldsymbol{\vartheta}}(\cdot)$. Specifically, the input tensor $\{\boldsymbol{\Delta}[k], \ldots, \boldsymbol{\Delta}[k - M_c + 1]\}$ is flattened into a single vector with NCM_c entries. The considered MLP is made of $H_c = 2$ hidden layers, each with $N_c = 2N$ neurons, where N denotes the number of sensors. Tanh activation function (i.e. $\tanh(\cdot)$) is applied to each neuron in both hidden layers. Finally, the MLP network is terminated with N neuron outputs with sigmoid activation function (i.e. $\sigma(\cdot)$) to provide a pseudo-probability output within $[0, 1]$. The N outputs are the entries of the soft decision vector $\boldsymbol{d}[k]$.

To train the classifier and make it able to implement both detection & identification tasks, a loss capitalizing *multitask learning* is employed. In the following, each learning task is associated with the classification of the operating condition for the corresponding sensor. Specifically, a *weighted sum* of the losses of the N binary (fault/no-fault) detection tasks associated with the unreliable sensors is minimized, i.e.

$$\mathcal{L}_{\text{cl}}\left(\boldsymbol{\vartheta}_{\text{shared}}, \{\boldsymbol{\vartheta}_n\}_{n=1}^{N}\right) \triangleq \sum_{n=1}^{N}\rho_n\,\mathcal{L}_n\left(\boldsymbol{\vartheta}_{\text{shared}}, \boldsymbol{\vartheta}_n\right) \tag{3.20}$$

where $\boldsymbol{\vartheta} = \{\boldsymbol{\vartheta}_{\text{shared}}, \boldsymbol{\vartheta}_1, \ldots \boldsymbol{\vartheta}_N\}$. In the above formula, the weight ρ_n indicates the preference level of the nth task (i.e. detection of a fault at nth sensor). It is worth noticing that the multitask objective function allows the proposed classifier to solve multiple learning tasks *at once* (i.e. via a single NN). Accordingly, in the above expression, $\boldsymbol{\vartheta}_{\text{shared}}$ represents the vector of *shared* parameters of the MLP common to all the N different tasks (i.e. those corresponding to the $H_c = 2$ hidden layers), whereas $\boldsymbol{\vartheta}_n$ indicates the vector of parameters which are *task-specific* for nth learning task (i.e. those corresponding to the nth output).

Uniform weighting is adopted ($\rho_n = 1/N$) and a binary cross-entropy (BCE) loss function for *all* the N binary tasks $\mathcal{L}_1(\cdot), \dots, \mathcal{L}_N(\cdot)$ is used, i.e.

$$\begin{aligned}\mathcal{L}_n^{\text{BCE}}(\boldsymbol{\vartheta}_{\text{shared}}, \boldsymbol{\vartheta}_n) = -\frac{1}{w}\sum_{j=0}^{w-1}\{y_n^j \ln d_n^j(\boldsymbol{\vartheta}_{\text{shared}}, \boldsymbol{\vartheta}_n) + \\ (1 - y_n^j)\ln\big(1 - d_n^j(\boldsymbol{\vartheta}_{\text{shared}}, \boldsymbol{\vartheta}_n)\big)\}\end{aligned} \tag{3.21}$$

where y_n^j is the $0/1$ representation of the true (i.e. labeled) fault status and d_n^j denotes the entry of classifier output of nth sensor. Finally, w is the number of samples in each batch.

We underline that the overall loss is minimized by leveraging Nadam optimization algorithm [74]. More details about the proposed architecture configurations, and the training process of the NN-based estimation and classification block are summarized in the paper **P7** [38].

3.3.4 Numerical Results

In this section, we present the results of comprehensive experiments to demonstrate the effectiveness of the proposed DRGCA. Also, the proposed architecture is compared with several state-of-the-art methods[9].

Baselines: We compare our SFDIA architecture against *five* state-of-the-art SFDIA and anomaly detection methods:

- **AE**-based architecture addressed in [56] which is detailed in Sec. 3.2.5.
- **FCC** NN is used in a modular architecture to model the virtual sensors [14]. FCC NN is chosen instead of the popular MLP NN due to efficiency in terms of number of neurons and input size required for the SFDIA problem. Fault detection and identification are performed by evaluating the residual between each sensor and the corresponding FCC NN estimate.

[9] We modified the configuration of some baselines for sake of fair comparison (see Tab. II in paper **P7**).

- Our previous **M-SFDIA** proposal with modified decision logic as decribed in Sec. 3.2.5.

- **Optimized M-SFDIA** (OM-SFDIA) is the optimized class of M-SFDIA architecture which selects the best configuration of NN modules for SFDIA among various variants (see **P4** [35]). OM-SFDIA is enhanced to handle more complex spatio-temporal patterns in the data by using GRU model as virtual estimators and a CNN model for the classifier.

- **GDN** is a novel attention-based approach, which detects anomalies from a learned graph of relationships between sensors [89]. The GDN method was merely designed for anomaly detection purposes. We used our decision logic (see Sec. 3.3.1.3) upon the GDN graph attention-based output to enable this method to detect and isolate multiple simultaneously faulty sensors.

We also compared the proposed architecture with an SVM-based classifier [11]. Surprisingly, the SVM method entirely failed in detecting the faults on both datasets, thus we do not report those performances in the following. Other parameters settings & implementation details are provided in **P7** [38].

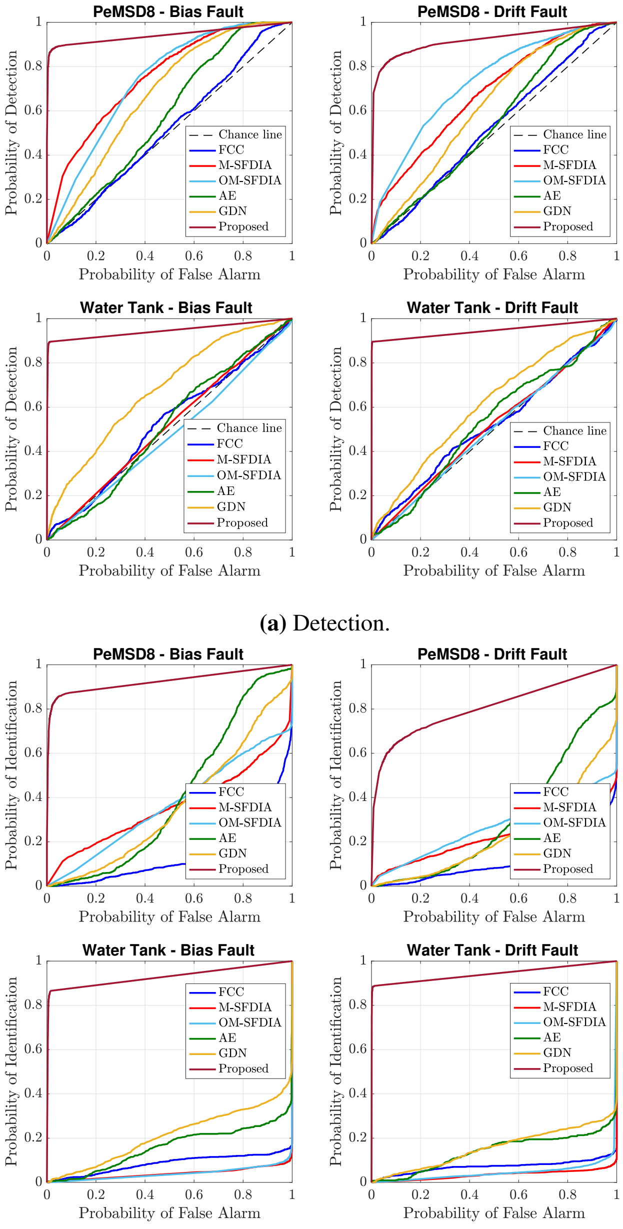
Table 3.4: Virtual sensors performance.

Datasets	**Models**	**Metrics**		
		MAE	RMSE	MAPE [%]
PeMSD8	AE	26.52	38.99	12.76
	FCC	20.64	33.81	14.41
	M-SFDIA	20.63	33.83	19.76
	OM-SFDIA	18.84	33.35	14.64
	GDN	24.24	36.57	14.68
	DRGCA	**13.28**	**21.47**	**8.85**
	Gain [%]	+29.51	+36.50	+30.64
Water Tank	AE	12.37	16.38	24.57
	FCC	15.50	20.00	30.77
	M-SFDIA	11.98	16.68	23.91
	OM-SFDIA	12.47	16.67	24.91
	GDN	6.63	9.12	14.35
	DRGCA	**0.50**	**0.76**	**1.00**
	Gain [%]	+92.25	+91.67	+93.03

Results & Analysis: *Virtual sensors performance* of our proposal vs. other baselines are reported in Tab. 3.4 for both the considered datasets. Overall, our proposed method clearly marks the best performance in all metrics as summarized in Tab. 3.4. Underlined values refer to the best-performing baselines on each metric. We can observe that both the graph-based methods, GDN method and our proposal, outperform other baselines on the Water Tank dataset, which indicates the effectiveness of GCN-based architecture in capturing spatial correlations. Nevertheless, the proposed architecture illustrates significant improvements (i.e. above 90% over GDN), thanks to its recurrent design. Better virtual sensors also imply higher *accommodation performance*, since these virtual measurements replace the real faulty sensors measurements upon classifier identification. There are no existing baselines that are as stable as our proposal. For instance, AE illustrates reasonably low MAPE on the PeMSD8 dataset, but failed on other metrics and the other dataset. On the contrary, our proposal shows *reliable estimations in all cases*.

Focusing our investigation toward *detection and identification* (viz. isolation) performance, in Fig. 3.18 we report the ROC curves. The probability of identification indicates the probability that SFDIA architecture correctly identifies the actual faulty sensor(s). The proposed DRGCA significantly outperforms all the baseline methods on both datasets, demonstrating its capability to detect and isolate sensor faults on graph data. The main reason behind this is that our proposal captures the faults and sensors' patterns by jointly utilizing spatio-temporal correlations due to its graph convolutional and recurrent design. The AE, M-SFDIA, OM-SFDIA, and FCC methods, regardless of their approved performance on small-size sensory networks [56, 14, 35, 33], show relatively poor detection performance because these methods have limited capability to discriminate faults from high-dimensional graphs. We notice that all the baseline methods almost failed to identify the correct faulty sensor(s), while the proposed architecture identifies faulty sensor(s) with bold confidence on both datasets. Moreover, our proposal obtains more considerable performance gains on detection and identification in the water tank dataset compared to which in the PeMSD8 dataset. We observe that the Water Tank dataset has a more spatial connection degree (# edges = 388 and # sensors = 100) than the PeMSD8 (# edges = 277 and # sensors = 170) dataset, which may lead to stronger spatial correlations.

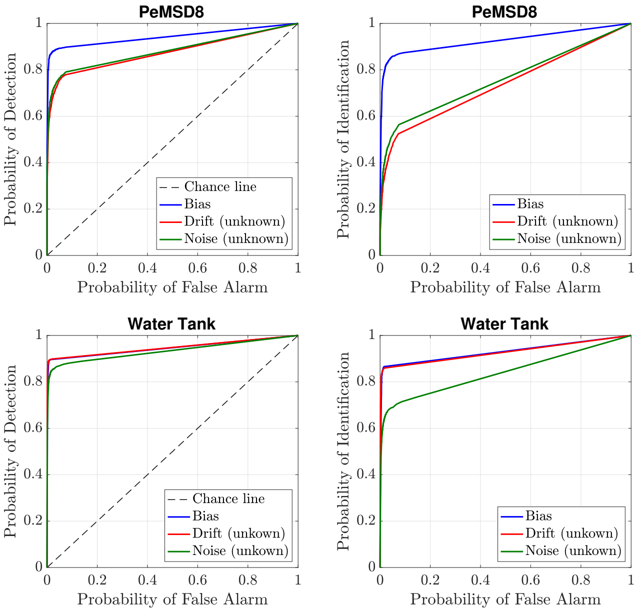
As a complementary analysis over unsupervised performance of the proposed architecture, in Fig. 3.19, we analyze the proposed architecture performance in situations that the architecture is not specifically trained for (i.e. without any supervision). In Fig. 3.19, we trained both NNs in our SFDIA architecture on bias fault type, while tested its (a) detection and (b) identification performance on *unseen* drift and noise fault types. Surprisingly, both detection and identification performances on unseen fault types are relatively close to the performances on trained



(a) Detection.

(b) Identification.

Figure 3.18: ROC curves on two graph-shape benchmark datasets. Sub-figures (a) and (b) show the detection performance, where the chance line is included, and the identification performance, respectively. Curves closer to the top-left corner indicate better performance (the closer the curve to the chance line, the less accurate the detection performance).



(a) Detection. (b) Identification.

Figure 3.19: Unsupervised (a) detection and (b) identification performance of the proposed architecture over unknown fault types (i.e. drift and noise fault) in terms of ROC curves. Both the estimator and classifier are trained over the bias fault type.

fault-type. This is basically because the proposed technique models the virtual sensors in the denoising configuration, this helps the estimator to focus on sensors' inter-dependencies and sensor-specific patterns rather than focusing on fault-type patterns. Moreover, perfect virtual sensors result in interpretable residual signals which further help the classifier to easier differ between faulty and non-faulty residual patterns.

Chapter 4

Conclusions and Future Works

The thesis aimed to tackle the SFDIA problem using data-driven approaches. The primary work was to present a general robust SFDIA architecture with the capability to adapt with different applications, while some state-of-the-art methods were restricted to a specified domain. Equally important, some architectures were only limited to the first fault detection task, or did not foresee all three tasks in their original formulation. Accordingly, we proposed M-SFDIA architecture which is capable of jointly taking advantage of the temporal correlation of the measurements and of both reliable and unreliable sensors within the system to achieve a higher sensor validation performance with respect to state-of-the-art methods. The classifier at the third stage of M-SFDIA architecture detects and isolates the faulty sensor from patterns within the input residual signals. The bank of estimators at the first stage allows to accommodate unreliable sensors by replacing the measurements from the identified faulty sensors. Estimators are also used at the second stage to derive the residual signals for the classifier.

In the scope of developing SFDIA schemes for real-world systems, we extended the M-SFDIA architecture with the major motivation to exploit complete spatial and temporal correlations in the data collected from the sensor to increase the overall accuracy. Complementary to the M-SFDIA architecture, we proposed a pair of regressors employed for each sensor to perform estimation and prediction operations of sensor measurements in the system. Moreover, a controller in a feedback loop is presented to preserve the performance of the proposed SFDIA architecture when faults occur. Our contribution represents a stepping stone towards the development of (modular) DTs based on sensor systems/networks in IoT contexts. The (four) designed layers of the extended design consist of estimation&prediction, residual, classification and controlling blocks. The controlling block is placed to track the classifier's decision output in order to boost overall system performance. This is accomplished by stopping fault propagation chain at the first layer by modifying estimators and predictors inputs with respect to the classifier's decision.

At the last part, we tackled the SFDIA problem of large-size networked IoT systems via a deep learning approach. This approach represents an opening gate toward transferring reliable data into DTs of large-size sensor systems/networks. We proposed a two-block architecture for SFDIA framework: in the first block, an estimator models virtual sensors and provides replacements for the identified faulty sensors within the system; in the second block, a classifier detects and identifies the faulty sensors. It is worth highlighting that the denoising design and the classification upon the residuals empower the proposed DRGCA to maintain its performance under unseen fault types.

We provided wide numerical analyses for comprehensive evaluation and comparison of the proposed architectures with other state-of-the-art methods. The proposed methods were trained and tested on different real-world and publicly-available datasets for the sake of a complete and reproducible assessment. For the sake of generalization, four types of faults were considered in this thesis: bias, drift, noise, and freeze. The proposed architectures yielded notably higher detection and isolation performance compared to the state-of-art counterparts under different fault types.

Future work can be focused on four aspects. First, including dynamic risk analysis in the design of IIoT systems in order to meet safety requirements when deploying DTs for safety-critical applications. A novel framework for risk management that improves real-time data collection, evaluates the performance of safety barriers and the overall impact on the system risk, and monitors and predicts risk changes and related decision support may assist safety and security standards.

Second, investigating non-stationary scenarios and the impact of diversity and redundancy in the graph. A non-stationary condition occurs when the deployment condition differs from the training condition. The change often results in a drop in performance. Dealing with non-stationarity is one of modern machine learning's greatest challenges.

Third, the investigation of reinforcement-learning algorithms for optimized controller design of the second proposal. The goal of reinforcement-learning is to acquire a policy function (a mapping from a state to an action) of a computer agent. Reinforcement-learning is able to find optimal policies using previous experiences without the need for previous information on the mathematical model, which makes this approach more applicable than other control-based systems.

Fourth, the application of explainable artificial-intelligence algorithms in interpreting (and improving) the proposed SFDIA approaches.

Bibliography

[1] M. Wollschlaeger, T. Sauter, and J. Jasperneite, “The future of industrial communication: Automation networks in the era of the Internet of Things and Industry 4.0,” *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[2] A. Rasheed, O. San, and T. Kvamsdal, “Digital twin: Values, challenges and enablers from a modeling perspective,” *IEEE Access*, vol. 8, pp. 21 980–22 012, 2020.

[3] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, “Digital twin in industry: State-of-the-art,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.

[4] J. Zhang, L. Li, G. Lin, D. Fang, Y. Tai, and J. Huang, “Cyber resilience in healthcare digital twin on lung cancer,” *IEEE Access*, vol. 8, pp. 201 900–201 913, 2020.

[5] A. Francisco, N. Mohammadi, and J. E. Taylor, “Smart city digital twin–enabled energy management: Toward real-time urban building energy benchmarking,” *Journal of Management in Engineering*, vol. 36, no. 2, p. 04019045, 2020.

[6] N. Mohammadi and J. E. Taylor, “Smart city digital twins,” in *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, pp. 1–5.

[7] 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.

[8] X. Luo, Y. Li, X. Wang, and X. Guan, “Interval observer-based detection and localization against false data injection attack in smart grids,” *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 657–671, 2021.

[9] Z. Zhang, A. Mehmood, L. Shu, Z. Huo, Y. Zhang, and M. Mukherjee, “A survey on fault diagnosis in wireless sensor networks,” *IEEE Access*, vol. 6, pp. 11 349–11 364, 2018.

[10] 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.

[11] 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.

[12] G. Su, D. Fang, S. Jian, and L. Fengmei, “Sensor fault detection with online sparse least squares support vector machine,” in *32nd Chinese Control Conference*, 2013, pp. 6220–6224.

[13] R. Isermann, *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2005.

[14] 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.

[15] K. Jeong, S. B. Choi, and H. Choi, “Sensor fault detection and isolation using a support vector machine for vehicle suspension systems,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3852–3863, 2020.

[16] M. Elnour, N. Meskin, and M. Al-Naemi, “Sensor data validation and fault diagnosis using auto-associative neural network for HVAC systems,” *Journal of Building Engineering*, vol. 27, p. 100935, 2020.

[17] H. Pham, J. Bourgeot, and M. E. H. Benbouzid, “Comparative investigations of sensor fault-tolerant control strategies performance for marine current turbine applications,” *IEEE Journal of Oceanic Engineering*, vol. 43, no. 4, pp. 1024–1036, 2018.

[18] 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 Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2020.

[19] H. Zhang, Q. Zhang, J. Liu, and H. Guo, "Fault detection and repairing for intelligent connected vehicles based on dynamic Bayesian network model," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2431–2440, 2018.

[20] P. Liu, Y. Zhang, H. Wu, and T. Fu, "Optimization of Edge-PLC-Based fault diagnosis with random forest in Industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9664–9674, 2020.

[21] S. De Vito, E. Massera, M. Piga, L. Martinotto, and G. Di Francia, "On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario," *Sensors and Actuators B: Chemical*, vol. 129, no. 2, pp. 750–757, 2008.

[22] 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.

[23] W. Kirchgässner, O. Wallscheid, and J. Böcker, "Empirical evaluation of exponentially weighted moving averages for simple linear thermal modeling of permanent magnet synchronous machines," in *IEEE 28th International Symposium on Industrial Electronics (ISIE)*, 2019, pp. 318–323.

[24] 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.

[25] K. Thiyagarajan, S. Kodagoda, L. Van Nguyen, and R. Ranasinghe, "Sensor failure detection and faulty data accommodation approach for instrumented wastewater infrastructures," *IEEE Access*, vol. 6, pp. 56 562–56 574, 2018.

[26] S. Mandal, B. Santhi, S. Sridhar, K. Vinolia, and P. Swaminathan, "Nuclear power plant thermocouple sensor-fault detection and classification using deep learning and generalized likelihood ratio test," *IEEE Transactions on Nuclear Science*, vol. 64, no. 6, pp. 1526–1534, 2017.

[27] S. Gutiérrez and H. Ponce, "An intelligent failure detection on a wireless sensor network for indoor climate conditions," *MDPI Sensors*, vol. 19, no. 4, p. 854, 2019.

[28] 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.

[29] L. Liu, G. Han, Y. He, and J. Jiang, "Fault-tolerant event region detection on trajectory pattern extraction for industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2072–2080, 2020.

[30] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *AAAI conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 922–929.

[31] H. Khorasgani, A. Hasanzadeh, A. Farahat, and C. Gupta, "Fault detection and isolation in industrial networks using graph convolutional neural networks," in *IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2019, pp. 1–7.

[32] 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.

[33] H. Darvishi, D. Ciuonzo, E. R. Eide, and P. Salvo Rossi, "Sensor-fault detection, isolation and accommodation for digital twins via modular data-driven architecture," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4827–4838, 2021.

[34] H. Darvishi, D. Ciuonzo, and P. Salvo Rossi, "Real-time sensor fault detection, isolation and accommodation for industrial digital twins," in *2021 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, vol. 1, 2021, pp. 1–6.

[35] H. Darvishi, D. Ciuonzo, and P. Salvo Rossi, "Exploring a modular architecture for sensor validation in digital twins," in *2022 IEEE Sensors*, 2022, pp. 1–4.

[36] H. Darvishi, D. Ciuonzo, and P. Salvo Rossi, "A machine-learning architecture for sensor fault detection, isolation and accommodation in digital twin," *IEEE Sensors Journal*, vol. 23, no. 3, pp. 2522–2538, 2023.

[37] A. Chawla, Y. Arellano, M. V. Johansson, H. Darvishi, K. Shaneen, M. Vitali, F. Finotti, and P. Salvo Rossi, "Iot-based monitoring in carbon capture and storage systems," *IEEE Internet of Things Magazine*, vol. 5, no. 4, pp. 106–111, 2022.

[38] H. Darvishi, D. Ciuonzo, and P. Salvo Rossi, "Deep recurrent graph convolutional architecture for sensor fault detection, isolation and accommodation in digital twins," *Submitted to IEEE Sensors Journal*, 2023.

[39] M. Goodarzi, M. A. Sebt, and H. Darvishi, "Target and image elevation angles separation algorithm for low-angle tracking with monopulse antenna," in *2020 28th Iranian Conference on Electrical Engineering (ICEE)*, 2020, pp. 1–4.

[40] H. Darvishi, M. A. Sebt, D. Ciuonzo, and P. Salvo Rossi, "Tracking a low-angle isolated target via an elevation-angle estimation algorithm based on extended kalman filter with an array antenna," *Remote Sensing*, vol. 13, no. 19, p. 3938, 2021.

[41] S. P. Talebi, H. Darvishi, S. Werner, and P. Salvo Rossi, "Gradient-descent adaptive filtering using gradient adaptive step-size," in *2022 IEEE 12th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2022, pp. 321–325.

[42] M. A. Sebt, M. Goodarzi, and H. Darvishi, "Geometric arithmetic mean method for low altitude target elevation angle tracking," *IEEE Transactions on Aerospace and Electronic Systems*, 2023.

[43] M. M. Gharamaleki and S. Babaie, "A new distributed fault detection method for wireless sensor networks," *IEEE Systems Journal*, vol. 14, no. 4, pp. 4883–4890, 2020.

[44] E. Dubrova, "Hardware redundancy," in *Fault-Tolerant Design*. Springer, 2013, pp. 5–86.

[45] A. A. Amin and K. Mahmood-Ul-Hasan, "Advanced fault tolerant air-fuel ratio control of internal combustion gas engine for sensor and actuator faults," *IEEE Access*, vol. 7, pp. 17 634–17 643, 2019.

[46] S. Yin, B. Xiao, S. X. Ding, and D. Zhou, "A review on recent development of spacecraft attitude fault tolerant control system," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 5, pp. 3311–3320, 2016.

[47] 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.

[48] J. Loy-Benitez, Q. Li, K. Nam, and C. Yoo, "Sustainable subway indoor air quality monitoring and fault-tolerant ventilation control using a sparse autoencoder-driven sensor self-validation," *Elsevier Sustainable Cities and Society*, vol. 52, p. 101847, 2020.

[49] M. Ruba, R. O. Nemes, S. M. Ciornei, and C. Martis, “Simple and robust current sensor fault detection and compensation method for 3-phase inverters,” *IEEE Access*, vol. 8, pp. 34 820–34 832, 2020.

[50] 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.

[51] C. Sun and Y. Lin, “Adaptive output feedback compensation for a class of nonlinear systems with actuator and sensor failures,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–10, 2021.

[52] J. Zhang, S. Li, and Z. Xiang, “Adaptive fuzzy output feedback event-triggered control for a class of switched nonlinear systems with sensor failures,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 5336–5346, 2020.

[53] C. Lo, J. P. Lynch, and M. Liu, “Distributed reference-free fault detection method for autonomous wireless sensor networks,” *IEEE Sensors Journal*, vol. 13, no. 5, pp. 2009–2019, 2013.

[54] J. Gao, J. Wang, P. Zhong, and H. Wang, “On threshold-free error detection for industrial wireless sensor networks,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2199–2209, 2018.

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

[56] 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.

[57] 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 International Journal of Aerospace Engineering*, 2018.

[58] A. F. Agarap, “Deep learning using rectified linear units (relu),” *arXiv preprint arXiv:1803.08375*, 2018.

[59] M. Alrifaey, W. H. Lim, and C. K. Ang, “A novel deep learning framework based RNN-SAE for fault detection of electrical gas generator,” *IEEE Access*, vol. 9, pp. 21 433–21 442, 2021.

[60] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional lstm and other neural network architectures,” *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[61] V. Mitra, H. Nam, C. Y. Espy-Wilson, E. Saltzman, and L. Goldstein, “Retrieving tract variables from acoustics: a comparison of different machine learning strategies,” *IEEE journal of selected topics in signal processing*, vol. 4, no. 6, pp. 1027–1045, 2010.

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

[63] 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.

[64] A. Ortega, *Introduction to Graph Signal Processing*. Cambridge University Press, 2022.

[65] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[66] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *30th International Conference on Neural Information Processing Systems (NIPS)*, 2016, p. 3844–3852.

[67] M. Henaff, J. Bruna, and Y. LeCun, “Deep convolutional networks on graph-structured data,” *arXiv preprint arXiv:1506.05163*, 2015.

[68] D. K. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[69] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *4th International Conference on Learning Representations (ICLR)*, 2016, p. 1–14.

[70] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava, “Sensor network data fault types,” *ACM Trans. Sen. Netw.*, vol. 5, no. 3, Jun. 2009.

[71] T. Muhammed and R. A. Shaikh, "An analysis of fault detection strategies in wireless sensor networks," *Elsevier Journal of Network and Computer Applications*, vol. 78, pp. 267 – 287, 2017.

[72] P. Tchakoua, R. Wamkeue, M. Ouhrouche, F. Slaoui-Hasnaoui, T. A. Tameghe, and G. Ekemb, "Wind Turbine Condition Monitoring: State-of-the-Art Review, New Trends, and Future Challenges," *MDPI Energies*, vol. 7, no. 4, pp. 1–36, April 2014.

[73] W. Kirchgässner, O. Wallscheid, and J. Böcker, "Deep residual convolutional and recurrent neural networks for temperature estimation in permanent magnet synchronous motors," in *IEEE International Electric Machines & Drives Conference (IEMDC)*, 2019, pp. 1439–1446.

[74] T. Dozat, "Incorporating Nesterov momentum into Adam," in *4th International Conference on Learning Representations (ICLR)*, 2016.

[75] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[76] S. Gururajan, M. L. Fravolini, H. Chao, M. Rhudy, and M. R. Napolitano, "Performance evaluation of neural network based approaches for airspeed sensor failure accommodation on a small UAV," in *21st IEEE Mediterranean Conference on Control and Automation (MED)*, 2013, pp. 603–608.

[77] M. R. Napolitano, Y. An, and B. A. Seanor, "A fault tolerant flight control system for sensor and actuator failures using neural networks," *Elsevier Aircraft Design*, vol. 3, no. 2, pp. 103–128, 2000.

[78] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 162–169, 2017.

[79] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

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

[81] B. Pourbabaee, N. Meskin, and K. Khorasani, "Sensor fault detection, isolation, and identification using multiple-model-based hybrid kalman filter for gas turbine engines," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1184–1200, 2016.

[82] I. Samy, I. Postlethwaite, and D.-W. Gu, "Detection and accommodation of sensor faults in uavs- a comparison of nn and ekf based approaches," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 4365–4372.

[83] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," in *2nd International Conference on Learning Representations (ICLR)*, 2014.

[84] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5966–5978, 2021.

[85] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with multi-stream adaptive graph convolutional networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 9532–9545, 2020.

[86] W. Liao, D. Yang, Y. Wang, and X. Ren, "Fault diagnosis of power transformers using graph convolutional network," *CSEE Journal of Power and Energy Systems*, vol. 7, no. 2, pp. 241–249, 2021.

[87] Z. Yan, J. Ge, Y. Wu, L. Li, and T. Li, "Automatic virtual network embedding: A deep reinforcement learning approach with graph convolutional networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1040–1057, 2020.

[88] Y. Wu, H.-N. Dai, and H. Tang, "Graph neural networks for anomaly detection in industrial Internet of Things," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[89] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4027–4035.

[90] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *33rd International Conference on Neural Information Processing Systems (NIPS)*, 2020.

[91] L. Bai, L. Yao, S. Kanhere, X. Wang, Q. Sheng *et al.*, "STG2Seq: spatial-temporal graph to sequence model for multi-step passenger demand forecasting," in *28th International Joint Conference on Artificial Intelligence (IJCAI)*, 2019, p. 1981–1987.

[92] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” in *6th International Conference on Learning Representations (ICLR)*, 2018.

[93] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference for Learning Representations (ICLR)*, 2015.

Paper 1

A Data-Driven Architecture for Sensor Validation Based on Neural Networks

H. Darvishi, D. Ciuonzo, E. R. Eide and P. S. Rossi

2020 IEEE SENSORS

A Data-Driven Architecture for Sensor Validation Based on Neural Networks

Hossein Darvishi*, Domenico Ciuonzo†, Eivind Rosón Eide‡, Pierluigi Salvo Rossi*
*Department of Electronic Systems, Norwegian University of Science and Technology, 7491 Trondheim, Norway
†University of Naples "Federico II," Naples 80125, Italy
‡Kongsberg Digital AS, Norway
Email: hossein.darvishi@ntnu.no; domenico.ciuonzo@unina.it; eivind.roson.eide@kdi.kongsberg.com; salvorossi@ieee.org

***Abstract*—In this paper, we propose a novel sensor validation architecture, which performs sensor fault detection, isolation and accommodation (SFDIA). More specifically, a machine-learning based architecture is presented to detect faults in sensors measurements within the system, identify the faulty ones and replace them with estimated values. In our proposed architecture, sensor estimators based on neural networks are constructed for each sensor node in order to accommodate faulty measurements along with a classifier to determine the failure detection and isolation. Finally, numerical results are presented to confirm the effectiveness of the proposed architecture on a publicly-available air quality (AQ) chemical multi-sensor data-set.**

***Index Terms*—Fault tolerance, neural networks, sensors.**

I. Introduction

With the new wave of digitalization, digital twins are at the core of the development process within Industry 4.0. Accordingly, sensors constitute the driving force for the accomplishment of this concept [1]. However, sensors are prone to failure and faulty data may negatively affect functionalities of the monitored system. Accordingly, SFDIA is a crucial practice since it can hinder faulty sensors from leading systems to catastrophic consequences. In this context, numerous approaches have been developed in the literature related to the use of analytical redundancy techniques for sensor fault detection and isolation. Such techniques can be mainly categorized into two groups: *model-based* methods and *data-driven* (or more generally model-free) methods.

The most widely used model-based methods comprise (multiple-model) Kalman filter [2], [3] and observer-based [4] approaches. Despite their appeal, model-based methods require an accurate mathematical model of the system, whose constitutive parameters are difficult to apply in the presence of nonlinearities. On the other hand, data-driven methods for SFDIA schemes have attracted significant attention by the scientific community due their ease of implementation and capabilities to capture nonlinear behavior by learning from historical data [5]–[9]. Data-driven methods include neural networks (NNs) and other machine-learning approaches [6], [8]–[10], hidden Markov models [11], fuzzy logic [12] and principal component analysis [13], whose successful application has been demonstrated to manifold systems. These comprise diesel-engines, gas-turbines, wireless sensor networks and air-crafts.

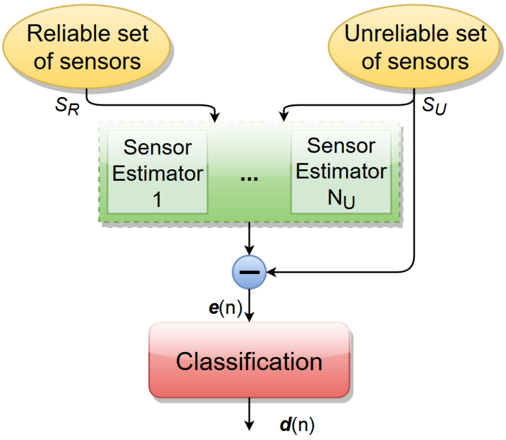


Fig. 1: The proposed system architecture for SFDIA.

In this work, we propose a machine-learning-based framework for sensor validation with different applications. The proposed architecture takes advantage of reliable and unreliable sensors' measurements as well as their temporal correlation. Synthetically-generated weak bias faults were added to a data-set of a chemical multi-sensor device to evaluate the presented SFDIA architecture. The benefits of the proposed approach are the flexibility in terms of the application domain, the capability to promptly deal with weak faults and (not explored here) with simultaneous faults of multiple sensors.

The outline of this manuscript is the following. Sec. II describes the proposed machine-learning based SFDIA architecture. The description of the considered multi-sensor data-set used in this work and numerical results are provided in Sec. III. Some final remarks are given in Sec. IV.

Notation – Lower-case bold letters and bold numbers denote vectors and $(\cdot)^T$, denotes transpose operator.

II. Proposed SFDIA Architecture

In this section, we briefly describe the three-layer system architecture (illustrated in Fig. 1). More specifically, we con-

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

sider a system monitored via $(N_R + N_U)$ different sensors. Sensors measurements constitute the input of the proposed SFDIA system, where measurements are divided into two sets: N_R reliable sensors (set S_R), which represent supportive data, and N_U unreliable sensors (set S_U), which are prone to failure. Still, we underline that the present architecture does not necessarily require the presence of reliable sensors.

A. Estimation Layer

According to Fig. 1, input sensors data enter the first layer with N_U independent sensor estimators, namely *virtual sensors*. Each virtual sensor receives all sensors' data except for the sensor under estimation from time instant n to $n - m$ (i.e. using a sliding window of length $m + 1$) as input and estimates the measurement of the sensor under estimation at time n as output. Outputs of the estimators are exerted to replace the isolated faulty data by the SFDIA system at the last layer. A classic multilayer perceptron (MLP) [7] architecture is considered for each virtual-sensor implementation.

B. Error Computation

The estimated measurement from each virtual sensor is then subtracted from the respective unreliable sensor measurement in the second layer to obtain N_U error signals, collected within $\boldsymbol{e}(n)$. Error signals measure the dissimilarity between the normal and faulty status of unreliable sensors, wherein the case of perfect estimation and no faulty sensors $\boldsymbol{e}(n) = \boldsymbol{0}$.

C. Classification Layer

The last stage of the proposed architecture consists of a classifier which aims at (i) detecting and (ii) identifying faulty measurements from the set of unreliable sensors S_U. In detail, the classification stage accepts the error vectors inputs at time instants n to $n - k$, namely $\boldsymbol{e}(n), \dots, \boldsymbol{e}(n - k)$ (i.e. a sliding window of length $k + 1$). The error vectors are used by the classifier as a metric for fault detection and isolation. Accordingly, the decision vector output is in the format $\boldsymbol{d}(n) = [d_0(n), d_1(n), \dots, d_{N_U}(n)]^T$ (with $d(n) \in [0, 1]$). Therein, $\{d_0(n) = 1\}$ denotes the event that no sensor failure is present, while other decision elements $\{d_i(n) = 1\}$ with $i = 1, \dots, N_U$ indicate failure on the ith unreliable sensor.

More specifically, the classifier is made of a two-layer MLP with a softmax output activation function and $N_U + 1$ output nodes. The classifier softmax output gives a decision vector representing the probability distributions of the vector of potential outcomes. Thus, decision element with the highest probability represents the occurred event

$$i_m = \underset{i \in 0, \dots, N_U}{\operatorname{argmax}} \; d_i(n) \,,$$

where i_m points to the largest element of the decision vector (i.e. it represents the event with the highest probability of occurrence). Finally, if an unreliable sensor is declared in failure, its measurements are replaced with the estimated values from the corresponding virtual sensor.

Briefly, $i_m = 0$ vs. $i_m \neq 0$ represents the *detection task*, being equivalent to "no fault detected" $\{d_0(n) = 1\}$ vs. "fault detected" $\{d_0(n) = 0\}$. In the case $i_m \neq 0$, the specific values of i_m performs the *isolation task* and replacing faulty sensor measurements with corresponding virtual sensor estimates employs the *accommodation task*.

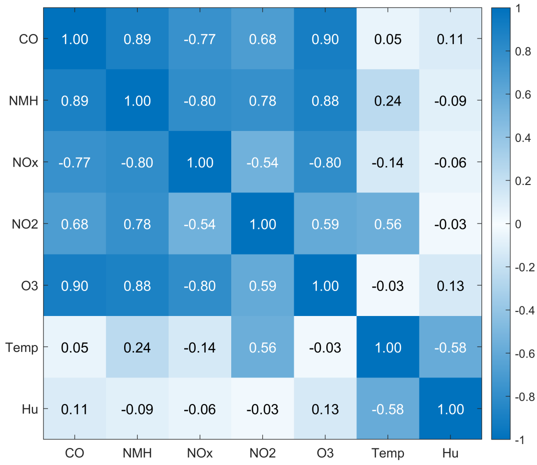


Fig. 2: correlation matrix of sensor pairs for AQ data-set.

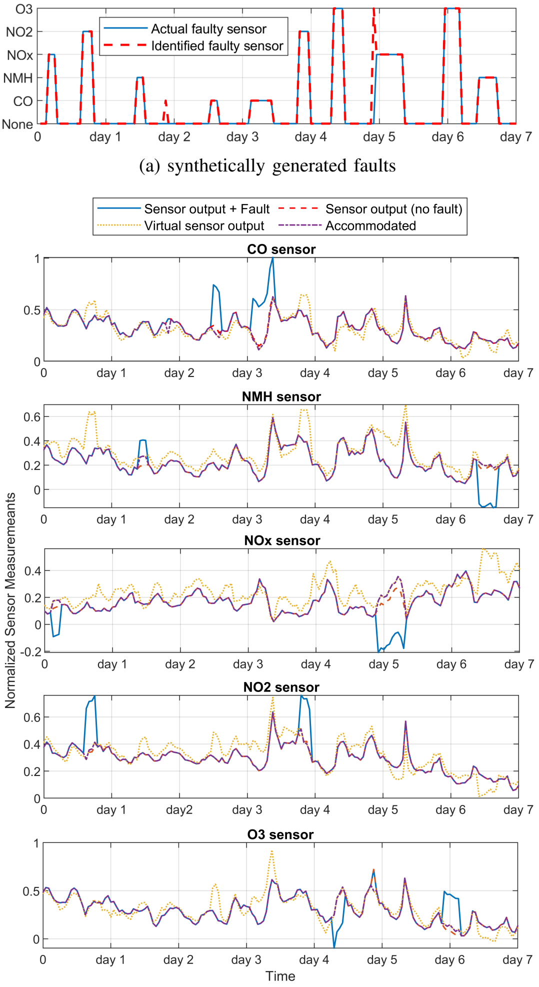
III. Data-Set Description and Numerical Results

The proposed architecture is applied to an air quality (AQ) data-set with 5 metal oxide chemical sensors embedded in an AQ chemical multi-sensor device installed on the field in an Italian city [14]. Hourly averaged measurements of the multi-sensor device consisting of carbon monoxide (CO), Non-Metanic Hydrocarbons (NMH), Nitrogen Oxides (NOx), Nitrogen Dioxide (NO2) and ozone (O3) gas concentrations are considered as unreliable set. Moreover, measurements of temperature (Temp) and humidity (Hu) in the AQ data-set are used within reliable set in this study. Accordingly, we have:

$$\mathcal{S}_U = \{\mathrm{CO}, \mathrm{NMH}, \mathrm{NOx}, \mathrm{NO2}, \mathrm{O3}\} \quad (N_U = 5)$$
$$\mathcal{S}_R = \{\mathrm{Temp}, \mathrm{Hu}\} \quad (N_R = 2)$$

Measurements of both sets are normalized via min-max normalization within the range $[0, 1]$. In addition, we dropped missed data before processing the data-set.

Our experimental analysis is carried out by dividing the data-set into a training set accounting for 85% of the first part of data-set and a test set accounting for the remaining 15%. The holdout validation method is used to prevent overfitting to some extent. Synthetically-generated bias faults are added to the AQ data set to verify the proposed architecture performance. To represent weak faults, we considered positive and negative additive bias faults. The bias absolute level ranges within $[20, 40]\%$ of each sensor measurements' variation domain on the train set. Five MLP virtual sensors (estimators) with one single hidden layer (made of 10 neurons) are trained to provide estimation of the $N_U = 5$ unreliable



(a) synthetically generated faults

(b) sensors' outputs

Fig. 3: Output of different stages of the proposed SFDIA architecture for bias faults over one week of the test set.

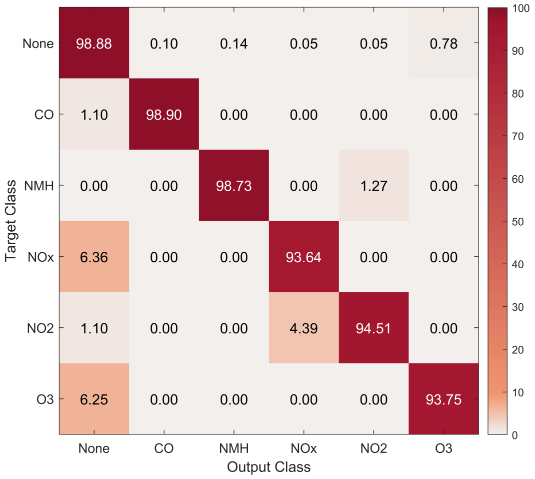


Fig. 4: Normalized confusion matrix for all classes during the test period. Numbers are in percent.

metal oxide chemical sensors. Differently, two hidden layers with 15 neurons per layer are considered for the classifier. Also, the size of the sliding window is assumed to span 10 samples for both the estimators and the classifier (i.e. $m = 10$ and $k = 10$).

Fig. 2 shows the correlation coefficient between different sensor pairs. Indeed, a higher correlation between sensor pairs would lead to more accurate estimators (viz. virtual sensors) in the first layer. As a result, this would imply a higher-precision classifier, since error signals represent difference in actual and virtual sensors' measurements. Results highlight significant dependencies among different pairs, which indicates the feasibility of our data-driven SFDIA.

The output of several parts of SFDIA architecture for one week of test set is shown in Fig. 3. More specifically, Fig. 3(a) monitors the faults on different sensors where the proposed architecture successfully detects and identifies all faults without delay in the system (dashed line) with only two false declaration samples (false positive) in the first and fifth days. As can be seen in Fig. 3(b), after fault identification, system accommodates isolated faulty data with its estimation to ensure the fault-free performance of the system.

Finally, the (normalized) confusion matrix on the test set is presented in Fig. 4. The confusion matrix shows excellent accuracy of the proposed architecture, i.e. classification rate about 96.5%. All classes show high precision over 90%, with the lowest precision exhibited on O3 and NOx sensors with values 93.75% and 93.64%, respectively.

IV. Conclusions and Future Directions

This manuscript presented a machine-learning based architecture for SFDIA scheme in real-time operation. MLP-based virtual sensors provided appropriate estimates of unreliable sensors to replace corresponding corrupted measurements in presence of faults, while an MLP-based classifier was responsible for detection and isolation of faults. The proposed architecture is validated by real-world data from AQ monitoring sensors, and results illustrate the prompt detection, isolation and accommodation of sensors' failures with less than 2.6% of faults on average remained undetected on the test set. Future directions will include the use of deep networks for the modules of the proposed SFDIA and type-of-fault classification.

REFERENCES

[1] S. Yin, S. X. Ding, and D. Zhou, "Diagnosis and prognosis for complicated industrial systems—Part I," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2501–2505, 2016.

[2] I. Samy, I. Postlethwaite, and D.-W. Gu, "Detection and accommodation of sensor faults in UAVs-a comparison of NN and EKF based approaches," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 4365–4372.

[3] B. Pourbabaee, N. Meskin, and K. Khorasani, "Sensor fault detection, isolation, and identification using multiple-model-based hybrid kalman filter for gas turbine engines," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1184–1200, 2016.

[4] S. Amin, X. Litrico, S. Sastry, and A. Bayen, "Cyber security of water SCADA Systems—Part I: analysis and experimentation of stealthy deception attacks," *IEEE Transactions on Control Systems Technology*, vol. 21, pp. 1963–1970, 09 2013.

[5] M. R. Napolitano, Y. An, and B. A. Seanor, "A fault tolerant flight control system for sensor and actuator failures using neural networks," *Elsevier Aircraft Design*, vol. 3, no. 2, pp. 103–128, 2000.

[6] 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.

[7] S. Gururajan, M. L. Fravolini, H. Chao, M. Rhudy, and M. R. Napolitano, "Performance evaluation of neural network based approaches for airspeed sensor failure accommodation on a small UAV," in *21st IEEE Mediterranean Conference on Control and Automation*, 2013, pp. 603–608.

[8] 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.

[9] S. Gutiérrez and H. Ponce, "An intelligent failure detection on a wireless sensor network for indoor climate conditions," *MDPI Sensors*, vol. 19, no. 4, p. 854, 2019.

[10] S. Mandal, B. Santhi, S. Sridhar, K. Vinolia, and P. Swaminathan, "Nuclear power plant thermocouple sensor-fault detection and classification using deep learning and generalized likelihood ratio test," *IEEE Transactions on Nuclear Science*, vol. 64, no. 6, pp. 1526–1534, 2017.

[11] C. Alippi, S. Ntalampiras, and M. Roveri, "Model-free fault detection and isolation in large-scale cyber-physical systems," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 1, pp. 61–71, 2017.

[12] S. Ogaji, L. Marinai, S. Sampath, R. Singh, and S. Prober, "Gas-turbine fault diagnostics: a fuzzy-logic approach," *Applied Energy*, vol. 82, no. 1, pp. 81 – 89, 2005.

[13] M. Z. Sheriff, M. Mansouri, M. N. Karim, H. Nounou, and M. Nounou, "Fault detection using multiscale PCA-based moving window GLRT," *Journal of Process Control*, vol. 54, pp. 47 – 64, 2017.

[14] S. De Vito, E. Massera, M. Piga, L. Martinotto, and G. Di Francia, "On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario," *Sensors and Actuators B: Chemical*, vol. 129, no. 2, pp. 750–757, 2008.

Paper 2

Sensor-Fault Detection, Isolation and Accommodation for Digital Twins via Modular Data-Driven Architecture

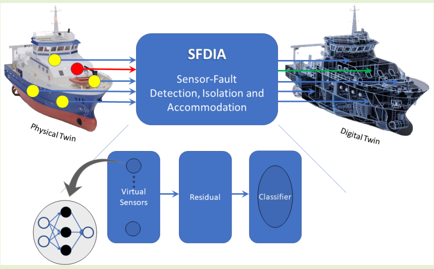
H. Darvishi, D. Ciuonzo, E. R. Eide and P. S. Rossi

IEEE Sensors Journal

Sensor-Fault Detection, Isolation and Accommodation for Digital Twins via Modular Data-Driven Architecture

Hossein Darvishi, *Student Member, IEEE*, Domenico Ciuonzo, *Senior Member, IEEE*, Eivind Rosón Eide and Pierluigi Salvo Rossi, *Senior Member, IEEE*

***Abstract*—Sensor technologies empower Industry 4.0 by enabling integration of in-field and real-time raw data into digital twins. However, sensors might be unreliable due to inherent issues and/or environmental conditions. This paper aims at detecting anomalies in measurements from sensors, identifying the faulty ones and accommodating them with appropriate estimated data, thus paving the way to reliable digital twins. More specifically, we propose a general machine-learning-based architecture for sensor validation built upon a series of neural-network estimators and a classifier. Estimators correspond to virtual sensors of all unreliable sensors (to reconstruct normal behaviour and replace the isolated faulty sensor within the system), whereas the classifier is used for detection and isolation tasks. A comprehensive statistical analysis on three different real-world data-sets is conducted and the performance of the proposed architecture validated under hard and soft synthetically-generated faults.**



***Index Terms*—Digital Twin, Fault Tolerance, Industry 4.0, Internet of Things, Machine Learning, Sensor Validation.**

I. Introduction

INDUSTRY 4.0 identifies the current fourth industrial revolution, whose aim is an increased level of automation through the effective combination of the Internet of Things (IoT), cyber-physical systems and cloud computing technologies [2]. Within this concept, sensors play a crucial role by measuring different physical parameters, thus enabling monitoring, controlling and decision-support capabilities [3]. While systems are highly dependent on data collected by sensors, the latter are unfortunately prone to errors. These errors can occur because of several reasons such as a harsh working environment, low battery level, limited life span (aging), improper calibration and hardware failures [4]. Corrupted data from sensors with failures may negatively affect both simple and more advanced functionalities of the system and result in overall system performance degradation and increased risk level. This would lead to consequences ranging from financial losses to serious safety issues (including life losses).

Reliable sensor measurements are vital for effective control and action-taking chain, and early reaction to faulty scenarios plays a critical role in risk management strategies while increasing safety and reliability. More specifically, a properly-working system should be able to perform: (i) *detection* (promptly detecting a fault condition within the system); (ii) *isolation* (identifying the faulty sensor) and (iii) *accommodation* (replacing the faulty data with some other trusted data). Accordingly, in this paper we propose a machine-learning-based framework for sensor validation. This framework allows developing a general sensor-fault detection, isolation, and accommodation (SFDIA) scheme to be easily adapted to different application domains, e.g. renewables in maritime scenarios [5]. In detail, the *contributions* of this paper are:

1) A novel machine-learning-based architecture for SFDIA is proposed. The proposed architecture jointly takes advantage of the temporal correlation of the measurements and of both reliable and unreliable sensors within the system to achieve a higher sensor validation performance.
2) The focus of generated faults is on *weak faults*, which are very hard to detect and usually ignored in the literature [6]–[10].
3) The performance of the proposed approach (in terms of probabilities of detection, false alarm, correct

This work was partially supported by the Research Council of Norway under the project SIGNIFY within the IKTPLUSS framework (project nr. 311902). Part of this work was presented at the conference IEEE Sensors, Rotterdam, The Netherlands, October 2020 (see [1])

H. Darvishi and P. Salvo Rossi are with the Department of Electronic Systems, Norwegian University of Science and Technology, 7491 Trondheim, Norway (e-mail: hossein.darvishi@ntnu.no; salvorossi@ieee.org).

D. Ciuonzo is with the University of Naples “Federico II”, 80125 Naples, Italy (e-mail: domenico.ciuonzo@unina.it).

E. R. Eide is with Kongsberg Digital AS, Norway (e-mail: eivind.roson.eide@kdi.kongsberg.com).

classification, misclassification, etc.) is evaluated on *three* different real-world data-sets [11]–[13] corrupted with synthetically-generated sensor faults (bias and drifts) and compared with two state-of-the-art techniques [14], [15]. The data-sets considered are *publicly-available*: this fosters reproducibility and further advances on the topic. Synthetically-generated sensor faults have been considered to perform a systematic performance assessment of the proposed architecture.

4) The impact of different hyperparameters, such as the number of layers and the number of nodes per layer, is assessed for the considered scenarios.

The rest of this paper is organized as follows. Sec. II provides a literature review regarding the related work. In Sec. III we introduce the proposed general SFDIA architecture and describe the different blocks for fault detection, isolation and accommodation. Then, in Sec. IV, we present the data description, contamination and pre-processing related to three independent data-sets with different applications. Accordingly, Sec. V highlights and compares the numerical performance for all the data-sets with different setups.

Finally, in Sec. VI we provide some concluding remarks and highlight future directions of research.

Notation - Lower-case bold letters denote vectors, $(\cdot)^T$ is the transpose operator, and $\mathcal{O}(\cdot)$ indicates the Landau notation.

II. Related Works

First practices for sensor validation were based on *hardware redundancy* [16]. These approaches used multiple sensors to measure the same parameter at the same point as well as a voting scheme to compensate sensors faults [16], [17]. However, hardware redundancy is unable to handle system noise and has some other serious drawbacks in terms of cost, weight, power consumption and size. Even more importantly, it is sensitive to simultaneous failure of all redundant sensors subject to the same harsh environmental conditions. Due to these reasons, alternative approaches based on *analytical redundancy* have gained more attention. Analytical-redundancy approaches attempt to develop reliable virtual sensors based on system model(s). More specifically, measurements collected by real sensors are compared with the values from the virtual ones to detect presence of faults and provide reliable measurements for replacement [9], [15], [18]. Various model-based and model-free (viz. data-driven) algorithms such as Kalman filter (KF) [19], [20], hidden Markov model [21], artificial neural networks (NN) [7], [22], and support vector machine (SVM) [14] have focused on detection and isolation tasks with application on aircraft sensor technologies, cyber-physical systems and wireless sensor networks (WSNs).

Early KF-based algorithms for detection and isolation were developed with an inherited drawback of being unable to deal with non-linearities [19]. Extended KF and multiple hybrid KFs were shown to overcome this issue through linearization around the state estimate and piece-wise linear models, respectively [20], [23]. Nevertheless, such solutions were heavily dependent on domain knowledge about the system which is not necessarily available.

As for data-driven approaches, multi-layer perceptron (MLP) architectures were considered for reducing probabilities of false alarm and miss detection through time-variant thresholds-based tests [22]. A method based on the SVM classifier was also proposed to detect faults through abnormal behaviors in the last three data measurements [14]. However, this method makes decision using redirected data to the server which results in delayed fault detection. Since the SVM classifier was only able to classify the faulty data, a deep belief network [7] coupled with a maximum squared error method for fault detection and isolation purposes was investigated. To address large data requirement of data-driven approaches, fault detection and isolation filters were derived in the state-space representation form by estimating system impulse response coefficients in the frequency domain via fast Fourier transform of input/output signals [24].

In the context of industrial WSNs, a threshold-free error detection (TED) method was developed [25]. TED relies on both temporal and spatial correlation between sensor readings. Recently, a method named TPE-FTED [10] based on an adjustable step window was proposed for online learning the changes of sensor readings in a dynamic environment. TPE-FTED deals with fault detection and isolation problem as a trajectory pattern extraction problem extracted from different sensing states. Then, TPE-FTED starts pattern matching as well as spatial-temporal constraint violation checking to detect the faulty sensor.

In summary, model-based algorithms require good knowledge of system model/ parameters and are difficult to implement in presence of nonlinearities. Conversely, data-driven algorithms may represent a valid alternative to analytical model-based algorithms: ease of implementation and capabilities to capture non-linear behavior by learning from historical data have increased attention toward data-driven algorithms for SFDIA schemes [8], [9], [15], [26]–[28].

An SFDIA scheme based on MLPs by consociating one main NN and a set of decentralized NNs has been proposed to create a system for detecting failures of gyro sensors of an aircraft [26]. Previous-time measurements of sensors under estimation were also used as the input of MLP NNs. A minimal radial basis function (MRAN) NN presented in [27] was able to reduce NN complexity by ignoring hidden neurons with less effect on the NN output. This algorithm was relatively slow in detecting faults after the occurrence of the faults. The performance of MLP and Extended MRAN NNs on sensor failure accommodation scheme were evaluated and compared through a study for failure on air data system [28]. This study showed similar performance of both NNs as online estimators, with slightly better performance of MLP NN in the training phase. SFDIA scheme presented in [15] employed a fully connected cascade (FCC) NN with only one neuron per layer connected to all previous layers. The proposed FCC NN was able to perform efficiently with a limited number of neurons and reduced computational complexity in comparison to MLP NN.

A NN-based sensor validation scheme for heavy-duty diesel engines was proposed using two banks of NN approximators to generate a residual signal for isolating faults and to produce an approximation of faulty sensor measurements [9]. A hybrid structure constructed of adaptive linear (ADALINE) NN for linear dominant operating conditions as well as MRAN NN for non-linear dominant operating conditions were considered to decrease complexity and computational load. However, the proposed scheme is still slow in detecting faults and requires a high number of neurons to approximate sensor output. In [8], the SFDIA approach based on artificial hydrocarbon networks (AHN) over WSN was presented. AHN is exploited to predict the temperature and detect the faulty sensor using in-field sensors and comparing it with information from a web service.

A distributed spike fault detection method was presented for linear time-invariant systems based on online learned pair-wise relationships of sensors using auto-regressive with exogenous input time-series model [29]. Another method utilized a seasonal auto-regressive integrated moving average models for forecasting surface temperature variation of concrete sewer pipes [6]. Predicted values were used as a reference measure for fault detection and replacement for faulty data. However, the presence of faults and anomalies reduces the forecasting performance of this method as it relies on previous measurements of the faulty sensor.

III. System Architecture for SFDIA Scheme

In the proposed SFDIA scheme, sensors are split into two groups: the unreliable set $\mathcal{S}_{\mathrm{U}}$ with N_{U} sensors that are prone to failures, and the reliable set $\mathcal{S}_{\mathrm{R}}$ with N_{R} reliable sensors. Indeed, in some applications some sensors could be more reliable because of sensor quality, hardware redundancy, proper design and working environment, being at middle of life time [30], or some other forms of protection in higher architectural layers. The proposed SFDIA algorithm can also handle the case of $\mathcal{S}_{\mathrm{R}}$ being the empty set ($N_{\mathrm{R}} = 0$). The objective is to detect, identify and accommodate failure of faulty sensors among the unreliable set whenever they happen. In the following, $x_{\mathrm{s}}[n]$ denotes the measurement from the generic sth sensor at time instant n. Without loss of generality, we number sensors 1 to N_{U} those belonging to the unreliable set, and $N_{\mathrm{U}}+1$ to $N_{\mathrm{U}}+N_{\mathrm{R}}$ those belonging to the reliable set, then we denote $\boldsymbol{x}_{\mathrm{U,s}}[n]$ and $\boldsymbol{x}_{\mathrm{R}}[n]$ the vectors collecting the measurements from the unreliable sensors with sth sensor excluded and from the reliable sensors, respectively, at time instant n.

The block diagram of the proposed SFDIA scheme is shown in Fig. 1, where similar blocks and similar data are reported in the same color. The input to the system is the set of measurements from all sensors. The system is based on *three* stages: (i) the first stage is made of N_{U} virtual sensors (representing estimation of unreliable sensors); (ii) the second stage is made of N_{U} analogous residual-computation units; and (iii) the third stage is made of a (multi-task) classifier. The classifier at the third stage is performing detection and isolation, while accommodation is done by exploiting the estimators' output.

More specifically, at the *first stage*, the virtual sensor $s \in \mathcal{S}_{\mathrm{U}}$ receives as input the measurements from all sensors excluding sensor s (i.e. $(\mathcal{S}_U \cup \mathcal{S}_R - \{s\})$ for time instant n and L_{v} previous time instants (i.e. a sliding window), and produces as output an estimate of the measurement of sensor $s \in \mathcal{S}_{\mathrm{U}}$, whose nth sample is denoted $y_{\mathrm{s}}[n]$.

Then, at the *second stage*, the residual-computation unit $s \in \mathcal{S}_{\mathrm{U}}$ receives as input the measurement $x_{\mathrm{s}}[n]$ of sensor $s \in \mathcal{S}_{\mathrm{U}}$ and the corresponding estimate $y_{\mathrm{s}}[n]$ from the virtual sensor $s \in \mathcal{S}_U$ and produces as output a measure of dissimilarity of the pair, whose nth sample is denoted $e_{\mathrm{s}}[n]$. Residual measurements are reflecting inconsistencies between the normal and faulty sensor operating status of unreliable sensors.

At the *third stage*, the classifier receives as input the dissimilarity measures from all the sensor pairs in the unreliable set $\mathcal{S}_{\mathrm{U}}$ for time instant n and L_{c} previous time instants, and produces as output a decision vector about if and which sensor has undergone a failure. According to Fig. 1, the nth (soft-) decision vector is denoted $\boldsymbol{d}[n] = (d_1[n], d_2[n], \ldots, d_{N_{\mathrm{U}}}[n])^T$ where $d_i[n] \in [0,1],\ i = 1, \ldots, N_{\mathrm{U}}$ denotes the probability of the ith sensor (corresponds to a specified unreliable sensor) being faulty. Ideally, a vector $\boldsymbol{d}[n]$ with all elements set to 0 denotes the event that no sensor has been declared in failure, while the set of unreliable sensors $\mathcal{S}_{\mathrm{U}}$ is mapped bijectively into the first N_{U} positive integers with an arbitrary labeling function. The final decision is made based on whether the maximum element of vector $\boldsymbol{d}[n]$ exceeds a given threshold γ. Nonetheless, the proposed SFDIA architecture (cf. Fig. 1), can detect, isolate and accommodate more than one sensor simultaneously. In this case, SFDIA scheme would present better performance for large scale systems. However this issue falls beyond the scope of this paper and will be explored in future works.

It is implicitly assumed that in the case that sensor $s \in \mathcal{S}_{\mathrm{U}}$ is declared in failure, its measurement $x_{\mathrm{s}}[n]$ is replaced with the estimate $y_{\mathrm{s}}[n]$ from the corresponding virtual sensor. It is apparent how the considered architecture implements all the tasks of a SFDIA system: i.e. decision vector $\boldsymbol{d}[n]$ with an over threshold element represents the *detection task*; after a fault is detected, the specific sensor index i corresponding to the maximum element $d_i[n]$ of the decision vector performs the *isolation task* and replacing $x_{\mathrm{s}}[n]$ with $y_{\mathrm{s}}[n]$ employs the *accommodation task*, with the sensor s identified through the inverse labeling function. In what follows, we detail each of three aforementioned stages.

1) Virtual Sensor:

An MLP NN, with $(L_{\mathrm{v}}+1)(N_{\mathrm{U}}+N_{\mathrm{R}}-1)$ inputs, 1 output, and H_{v} hidden layers, each with N_{v} hidden nodes, has been considered for the implementation of the generic virtual sensor, i.e.

$$y_{\mathrm{s}}[n] = f_{\mathrm{s}}^{(H_{\mathrm{v}},N_{\mathrm{v}})}(\boldsymbol{x}_{\mathrm{U,s}}[n], \ldots, \boldsymbol{x}_{\mathrm{U,s}}[n-L_{\mathrm{v}}] \\ , \boldsymbol{x}_{\mathrm{R}}[n], \ldots, \boldsymbol{x}_{\mathrm{R}}[n-L_{\mathrm{v}}])\ , \quad (1)$$

where f_{s} represents the MLP-based function model of the sth sensor. Each MLP has been trained using the Nesterov-accelerated adaptive moment estimation (Nadam) optimization algorithm using real-world data-sets [31], [32]. The Nadam algorithm takes advantage of properties of adaptive moment estimation (Adam) algorithm and incorporates Nesterov Accelerated Gradients to Adam. Hyperbolic tangent (tanh) and

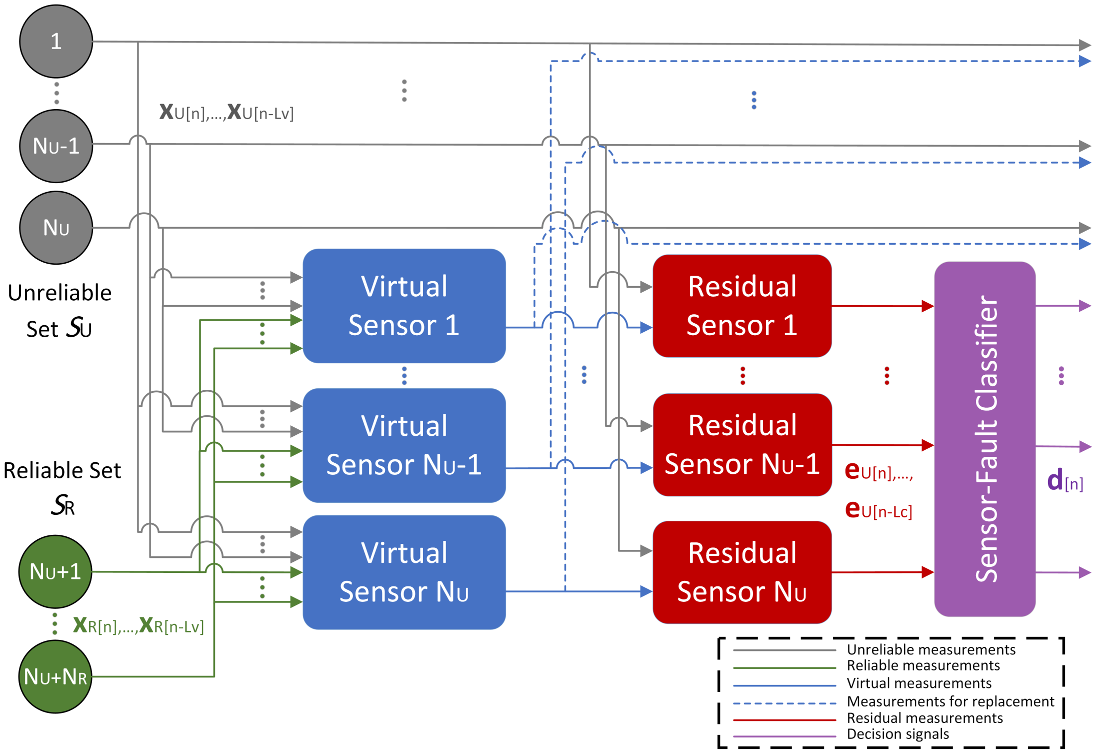


Fig. 1: Block diagram of the SFDIA system.

identity activation functions are employed in hidden layers and the output layer, respectively. Mean square error (MSE) loss function is used for loss calculation in training phase.

The MLP is a simple architecture with proved performance of estimating nonlinear behavior [26], [28]. Numerical results show the excellent performance of MLP architecture. However, in the case of further requirement of extrapolating long-term impact of the temporal dimension for time series data-sets, more complicated architectures (e.g. convolutional neural networks, recurrent neural networks (RNNs) and long short term memory networks (LSTMs) [33], [34]) are expected to present more appropriate results for the implementation of each virtual sensor. Data description, data pre-processing (in order to make it suitable for model training) and data contamination procedure (via synthetically-generated faults) are described in the next section.

2) Residual Computation: For dissimilarity measure, we simply considered the error between the estimated value and the actual value, i.e.

$$e_{\mathrm{s}}[n] = y_{\mathrm{s}}[n] - x_{\mathrm{s}}[n]. \tag{2}$$

In fault-free condition, it is expected that the residual measurements $e_{\mathrm{s}}[n]$ be equal to zero, but in practice, it always contains non-zero value due to noise and imperfect estimation of sensor output. Hence, the classifier is introduced to discriminate faulty measurements from non-faulty measurements via pattern analysis of residual signals.

3) Classifier: An MLP NN, with N_{U} inputs, N_{U} discrete output, and H_{c} hidden layer with N_{c} hidden nodes, has been considered for the implementation of the classifier, i.e.

$$\boldsymbol{d}[n] = g^{(H_{\mathrm{c}},N_{\mathrm{c}})}(\boldsymbol{e}_{\mathrm{U}}[n], \ldots, \boldsymbol{e}_{\mathrm{U}}[n-L_{\mathrm{c}}]). \tag{3}$$

where $\boldsymbol{e}_{\mathrm{U}}[n]$ is a vector of the dissimilarity measurements of the unreliable set at time instant n. Since there is a certain level of correlation between temporal samples of residual signals, L_{c} previous time instants are also fed to the classifier to exploit the temporal correlation among measurements.

The binary cross-entropy loss function along with the same optimization algorithm (Nadam) and activation function (tanh) for hidden layers as in the virtual sensors are employed in the classifier. Moreover, N_{U} sigmoid activation function is used at the output layer of the classifier. The fault-signal generation is described in the next section.

Computational Complexity: The computational complexity of the proposed SFDIA structure is calculated hereunder in terms of the big-$\mathcal{O}$ notation for one input sample. The computational complexity for each layer of the virtual sensor and classifier is specified in Tab. I.

It is worth noticing that in Tab. I, the impact of tanh and sigmoid operations for virtual sensors and the classifier has been neglected. Finally, with respect to the computational complexity of both MLPs and assuming equal number of

TABLE I: Computational complexity of the MLPs constituting the proposed SFDIA architecture.

Layers	MLP	Complexity
first hidden layer	virtual sensor	$\mathcal{O}(L_{\rm v}N_{\rm U}N_{\rm v} + L_{\rm v}N_{\rm R}N_{\rm v})$
	classifier	$\mathcal{O}(L_{\rm c}N_{\rm U}N_{\rm c})$
other hidden layers	virtual sensor	$\mathcal{O}(N_{\rm v}^2)$
	classifier	$\mathcal{O}(N_{\rm c}^2)$
output layer	virtual sensor	$\mathcal{O}(N_{\rm v})$
	classifier	$\mathcal{O}(N_{\rm U}N_{\rm c})$
in total	virtual sensor	$\mathcal{O}(L_{\rm v}N_{\rm U}N_{\rm v} + L_{\rm v}N_{\rm R}N_{\rm v} + H_{\rm v}N_{\rm v}^2)$
	classifier	$\mathcal{O}(L_{\rm c}N_{\rm U}N_{\rm c} + H_{\rm c}N_{\rm c}^2)$

hidden layers ($H_{\rm v} = H_{\rm c} = H$), nodes per hidden layer ($N_{\rm v} = N_{\rm c} = N$) and time delays ($L_{\rm v} = L_{\rm c} = L$), the computational complexity involved with the proposed architecture is approximately $\mathcal{O}(LN_{\rm U}^2N + LN_{\rm R}N_{\rm U}N + HN_{\rm U}N^2)$. Thus, the proposed architecture has *polynomial complexity*, and the complexity grows *quadratically* as a function of the number of nodes per layer (N) and number of unreliable sensors ($N_{\rm U}$).

IV. Data Description, Pre-Processing, and Contamination

A. Data Description

Three real-world data-sets are applied to the proposed SFDIA system to evaluate the qualification of the system in different scenarios.

1) Air Quality (AQ) Data-Set: The first data-set contains hourly-averaged measurements of an array of 5 metal oxide chemical sensors embedded in a gas multi-sensor device deployed on the field in an Italian city along with gas concentrations references from a certified analyzer [11]. The device was located in a polluted area, at road level of the city. AQ data-set was recorded during Mar. 2004-Feb. 2005.

Measurements contain carbon monoxide (CO), non-metanic hydrocarbons (NMH), nitrogen oxides (NO_x), nitrogen dioxide (NO_2) and ozone (O_3) gas concentrations, as well as measurements of temperature and humidity. For our analysis, the ground truth hourly-averaged concentrations provided by a co-located reference certified analyzer along with absolute humidity are ignored. Accordingly, in our numerical analysis, the five gas sensors are considered as the unreliable set ($N_{\rm U} = 5$), whereas temperature and relative humidity are considered as the reliable set ($N_{\rm R} = 2$).

2) Wireless Sensor Network (WSN) Data-Set: The second data-set used in our evaluation has been collected at the University of North Carolina at Greensboro [12]. A labeled data-set collected from a single-hop and a multi-hop WSN using TelosB motes. The data-set consists of 4 sensors located indoor and outdoor measuring humidity and temperature. Measurements were collected during 6 hours at 5 seconds interval. Anomalies indicated with label ”1” in the original data-set were introduced to two sensors by using a water kettle which increased the temperature and humidity.

In what follows, only the multi-hop data-set with 4 temperature (T1 to T4) measurements is used as unreliable set ($N_{\rm U} = 4$), and data with the indicated label ”1” were ignored from this data-set. No reliable set is considered for this data-set ($N_{\rm R} = 0$).

3) Permanent Magnet Synchronous Motor (PMSM) Data-Set: The third data-set comprises several sensor data measurements from a permanent magnet synchronous motor collected by the LEA department at Paderborn University [13], [35]. Data-set measurements include ambient temperature, coolant temperature (CT), voltage q and d components, current q and d components, motor speed (MS), torque (TRQ), rotor temperature, stator yoke temperature, stator tooth temperature, and stator winding temperature. Original measurements contain 52 sessions, with each session being $1 \sim 6\text{h}$ long and sampled at intervals of 0.5 seconds.

We have considered a sample interval of 15 seconds (by down-sampling) and ignored the ambient and rotor measurements. Summation of q and d components of voltage and current are treated as final voltage (V) and current (C) measurements. The reliable set consists of 3 stator temperatures ($N_{\rm R} = 3$), and other remaining measurements form the unreliable set ($N_{\rm U} = 5$).

B. Pre-processing

As commonly done in machine-learning applications, in order to avoid polarization in the training due to different ranges of different variables, measurements of each sensor have been normalized such to span the range $[0, 1]$ via min-max scaling

$$x'_{\rm s}[n] = \frac{x_{\rm s}[n] - x_{\min}}{x_{\max} - x_{\min}}\ , \tag{4}$$

where $x'_{\rm s}[n]$ represents the normalized measurements of the sth sensor, whereas $x_{\max}$ and $x_{\min}$ are the minimum and maximum of the training set for given sensor measurements. It is worth mentioning, in the normalization process, $x_{\max}$ and $x_{\min}$ are derived based on the training set of each data-set to present the real-world condition. Besides normalization, entire rows with missed data in data-sets are omitted. No other pre-processing has been considered, such as feature extraction, to help the learning procedure of the virtual sensors. Although, for noisy data-sets, smoothing techniques (e.g. moving average, Savitzky-Golay filter or quadratic regression) or low-pass filtering can be performed allowing the important patterns of data to stand out.

In proposed architecture, instead of using all sensor except the one under estimation as input of each virtual sensor, only the most correlated sensors could be considered as input. This would help containing complexity, specially for large-scale systems, while ensuring acceptable performance. Correlation matrix of all sensors could be obtained from the training set. However, this issue is beyond the scope of this paper and will not be here investigated. Architectures with different number of hidden layers has been compared in order to verify if a deep architecture can overcome the need for feature extraction for the specific problem.

C. Data Contamination

In order to build data-sets including sensor failures for training the SFDIA classifier and testing its performance, synthetic fault signals have been generated and injected to all three data-sets. Failure of a sensor could manifest in several ways [36]–[38]. The most common fault models are bias, drift, freezing and random fault. In this paper, without loss of generality, we considered *bias* and *drift* faults to represent hard and soft failures, respectively. The mathematical model for each of them is described in what follows.

1) Bias fault: In this type of failure (also known as step fault), a constant bias b for M consecutive samples was added to the sensor measurements, namely

$$x_{\mathrm{s}}[n] = \begin{cases} a_{\mathrm{s}}[n] + \nu_{\mathrm{s}}[n] + b \ , & 0 \leq n - m < M \\ a_{\mathrm{s}}[n] + \nu_{\mathrm{s}}[n] \ , & \text{else} \end{cases} \tag{5}$$

where $a_{\mathrm{s}}[n]$ is the ideal (without fault) measurement of the sth sensor and m is the starting time instant of the fault, while $\nu_{\mathrm{s}}[n]$ denotes the measurement noise. Sensor measurements in all three data-sets are (naturally) including measurement noise (i.e. they provide $a_{\mathrm{s}}[n] + \nu_{\mathrm{s}}[n]$).

2) Drift fault: This additive fault happens in $M + N$ consecutive samples when sensor output drifts up to the bias level b with M time instants

$$x_{\mathrm{s}}[n] = \begin{cases} a_{\mathrm{s}}[n] + \nu_{\mathrm{s}}[n] + \frac{b(n-m+1)}{M}, & 0 \leq n - m < M \\ a_{\mathrm{s}}[n] + \nu_{\mathrm{s}}[n] + b, & M \leq n - m < M + N \\ a_{\mathrm{s}}[n] + \nu_{\mathrm{s}}[n], & \text{else} \end{cases} \tag{6}$$

where N is the number of consecutive samples that the drift fault remains at the saturated bias level b. Also, we considered $M > N$ to stress the effect of the drift.

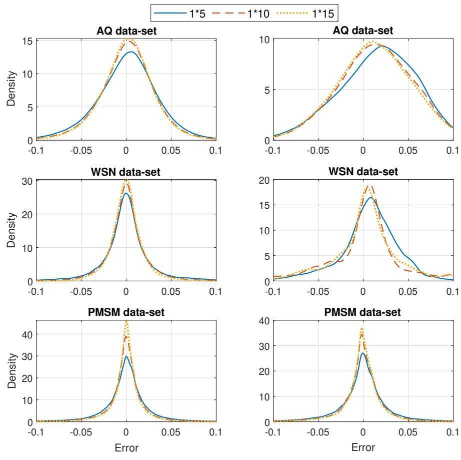
V. Numerical Results

In this section, performance of the proposed SFDIA architecture is examined and compared with recent research works by using the aforementioned real-world data-sets. Each data-set is divided into three parts. On each data-set, we used 70% of data for training MLPs (training set), 15% for validating (validation set) and last 15% block of data for testing purposes (test set). *Early stopping* method is used to avoid over-fitting during the training phase [39]. In this method, error on the validation set is monitored and if after 20 consecutive epochs validation set error did not improve, the training process is stopped.

We denote *variation domain* the size of the range spanned by a sensor with reference to the training set. Maximum level b of generated faults is assumed uniformly distributed between 0.2 and 0.4 (i.e. accounting for 20 to 40 percent of the corresponding variation domain) to represent weak fault signals. Positive and negative faults are generated randomly. Uniform distribution of maximum level b assures that the classifier will not learn on a specific level. Table II reports the variation domain for each sensor. The variation domain, which is always less or equal to the true range of each sensor (e.g. on WSN data-set in Tab. II, maximum variation domain is 3.72°C while usually temperature sensors range are around 150°C or even higher), is used as criterion since the true ranges were unknown. In addition, to better understand the effect of fault strength on detection accuracy, strong fault signals with maximum level b uniformly distributed between 0.6 and 0.9 are considered for comparison with weak fault signals.

TABLE II: Variation domain for each sensor.

Data-set	**Sensors Variation Domain**				
AQ	CO	NMH	NO$_x$	NO$_2$	O$_3$
	1392	1830	2360	2118	2270
WSN	T1 (°C)	T2 (°C)	T3 (°C)	T4 (°C)	-
	3.57	3.72	2.23	1.99	-
PMSM	CT	V	C	MS	TRQ
	3.50	6.00	7.24	3.25	6.33



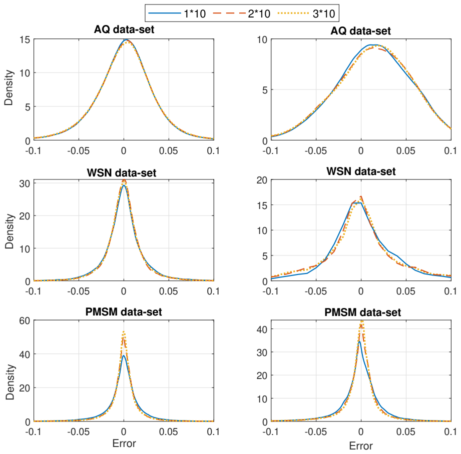
(a) Training set (b) Test set

Fig. 2: Averaged performance of the virtual sensors for different number of nodes N_{v} in terms of PDF of the error signals on each data-set. Different configurations are denoted with $H_{\mathrm{v}} \times N_{\mathrm{v}}$.

A. Virtual Sensors Performance

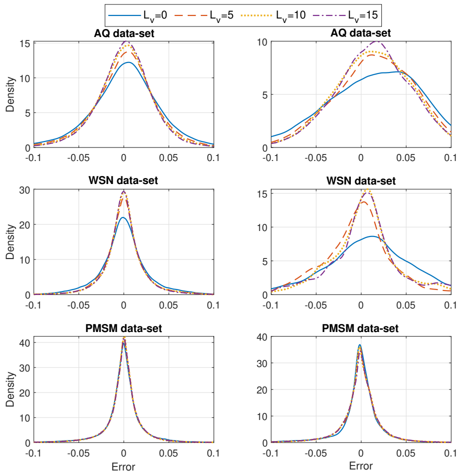
Virtual sensors with $N_{\mathrm{v}} \in \{5, 10, 15\}$ nodes per hidden layer and $H_{\mathrm{v}} \in \{1, 2, 3\}$ hidden layers have been trained and compared. In detail, virtual sensors' overall performance on both training and test sets are shown in Figs. 2 and 3 in terms of PDF of all sensors error signals ($e_{\mathrm{U}}[n]$) in each data-set.

The improvement of the performance with increasing the number of nodes (N_{v}) and hidden layers (H_{v}) is apparent, but variable for different data-sets. Fig. 2 seems to suggest the improvements with respect to the number of nodes per layer saturate approximately with N_{v}, while, as it can be seen in Fig. 3, adding more layers has only a relevant effect on the largest data-set (PMSM data-set). It must be said that



(a) Training set (b) Test set

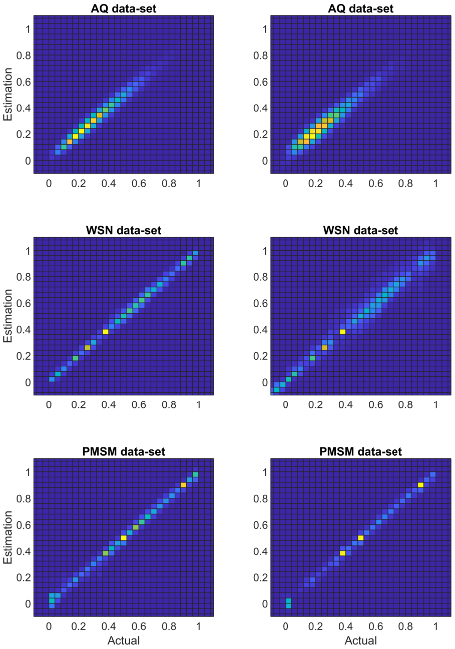
Fig. 3: Averaged performance of the virtual sensors for different number of hidden layers H_v in terms of PDF of the error signals on each data-set. Different configurations are denoted with $H_\text{v} \times N_\text{v}$.



(a) Training set (b) Test set

Fig. 4: Averaged performance of the virtual sensors in configuration 1×10 for different number of previous time instants L_v in terms of PDF of the error signals on each data-set.

deeper network structures require larger data-sets to update their weights and biases, thus the saturation effect might be due to the limited amount of available data. Fig. 4 illustrates the impact of input window size L_v on the virtual sensors performance. By employing delayed samples, the virtual sensors can exploit the temporal correlation between data samples to enhance estimation performance. However, the PMSM data-set has a very limited temporal correlation.



(a) Training set (b) Test set

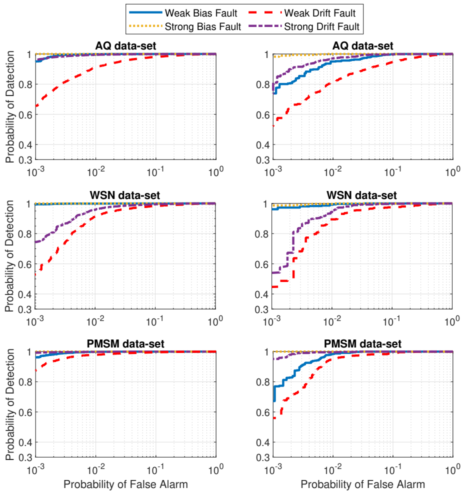
Fig. 5: Averaged performance of the virtual sensors in configuration 1×10 and $L_\text{v} = 10$ in terms of 2D PDFs of the estimated and actual values.

Performance of the configuration with $H_\text{v} = 1$ hidden layer, $N_\text{v} = 10$ nodes per hidden layer and $L_\text{v} = 10$ is considered acceptable, thus in the following, we will refer to this specific configuration. The 2D-PDF plots of the estimated and actual values for virtual sensors in configuration 1×10 are shown in Fig. 5, both for the training and the test sets. It is worth noticing that the test set of the WSN data-set exceeds the defined normalization lower-bound which is the result of normalization on the training set.

B. Classifier Fault Detection and Classification Performance

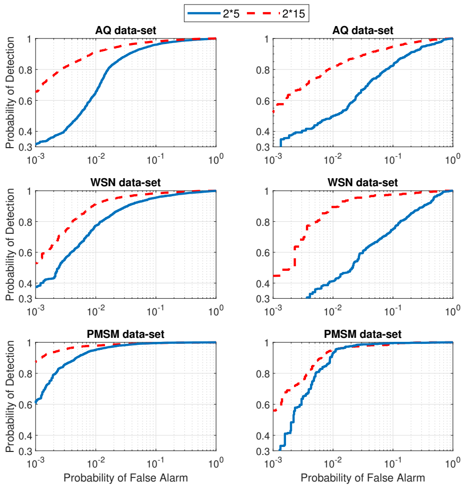
Synthetically-generated faults have been added to unreliable set of sensors to emulate faulty sensors. Different configurations for the classifier are compared in the following. Table III lists the number of parameters (weights and biases) to be trained during training phase in the classifier and each virtual sensor for different configurations.

A classifier with $H_\text{c} = 2$ hidden layers, $N_\text{c} = 15$ nodes per hidden layer and a memory of $L_\text{c} = 10$ has been trained. In this configuration, according to Tab. III, a total number of 725



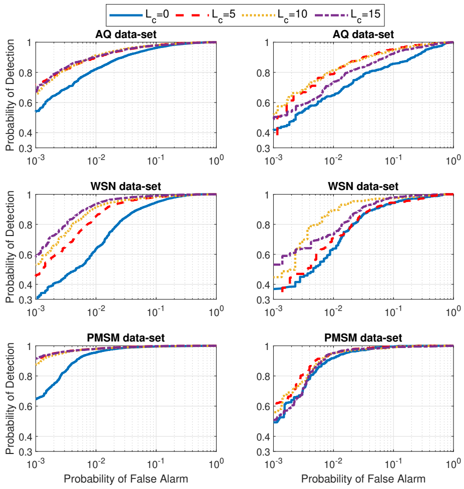
(a) Training set (b) Test set

Fig. 6: ROC curves of proposed SFDIA structure for all data-sets under bias and drift faults.



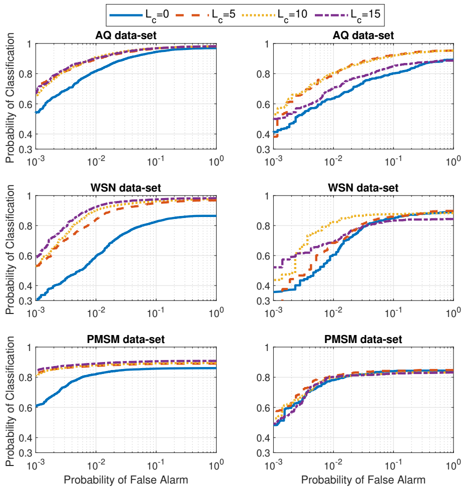
(a) Training set (b) Test set

Fig. 7: Detection performance of the classifier for different number of nodes per hidden layer N_c in terms of ROC on each data-set.



(a) Training set (b) Test set

Fig. 8: Detection performance of the classifier in configuration 2×15 for different number of previous time instants L_c in terms of ROC on each data-set.



(a) Training set (b) Test set

Fig. 9: Averaged classification (isolation) performance of the classifier in configuration 2×15 for different number of previous time instants L_c in terms of ROC on each data-set.

trainable parameters of the classifier are required to be updated through training phase over AQ and PMSM data-set[1].

The probabilities of detection and false-alarm are two important metrics for evaluating the performance of a detector. Accordingly, in Fig. 6, fault detection performance is investigated in terms of both metrics by using the well-known receiver operating characteristic (ROC) curves (i.e. by varying the threshold γ). Results highlight that, although the classifier is facing weak fault signals, it is still capable to detect them with a very high probability for negligible false-alarm probability. Detection probability of bias faults is noticeably

[1] The number of trainable parameters of the classifier is different for WSN data-set due to different Number of unreliable sensors ($N_U = 4$).

TABLE III: number of trainable parameters (weights and biases).

		Virtual Sensor[a]				Classifier[a]			
N_c,	H_c,	L_v				L_c			
N_v	H_v	0	5	10	15	0	5	10	15
5	1	31	131	231	331	60	185	310	435
	2	61	161	261	361	90	215	340	465
10	1	61	261	461	661	115	365	615	865
	2	171	371	571	771	225	475	725	975
15	1	91	391	691	991	170	545	920	1295
	2	331	631	931	1231	410	785	1160	1535

[a] No reliable sensor ($N_R = 0$) and $N_U = 5$ unreliable sensors considered for calculations.

higher than drift faults over different false alarm rates. This is originally due to the ramp up phase of drift faults which takes classifier more samples to detect faults. As illustrated in Fig. 6, WSN data-set has somewhat lower performance in comparison with the other two data-sets (in case of drift faults). It is mainly because of very weak fault levels on this data-set according to its sensors' variation domains (see Tab. II). Conversely, detection performance of proposed architecture under strong faults are significantly higher than the detection performance under weak faults as shown in Fig. 6, which highlights the importance of detection and isolation of weak faults.

The detection rate of the classifier with 5 and 15 nodes per hidden layer is assessed in Fig. 7 in case of drift faults. It is apparent from both train and test sets that 5 nodes per hidden layer are not enough for distilling relevant features from the data sequences. In general, the accuracy on test set is lower than the accuracy on train set since the classifier is optimized for the latter. Figs. 8 and 9 demonstrate the effect of using time-delayed samples on the classifier in the case of drift fault. There are certain improvements in detection performance and averaged classification (isolation) performance[2] when temporal correlation exists in sensor measurements. However, as it can be seen on both Fig. 8.(b) and 9.(b), the performance slightly reduces with increasing number of time delays ($L_c = 15$) due to the negligible temporal correlation between older samples and current sample in the measurements. Besides, in this scenario, increasing the window size should potentially lead to a performance improvement, however a larger number of nodes in the hidden layers might be required to handle properly the increased number of input nodes. Differently, with a fixed network structure, increasing the window size might in practice saturate the learning capability.

Figure 10 shows the performance in terms of "multi-class ROC" for each detected class for AQ data-set under drift faults, i.e. no failure and sensor-1 to sensor-5 failures. More specifically, each subfigure refers to a specific true sensor fault and reports the curves of the probability of classification for each possible fault (including the no-fault scenario represented with a dashed line) obtained through varying the selected threshold[3]. The probability of correct classification for all 5 sensors reaches $\approx 95\%$. Also, it is apparent how good detection and identification results are obtained at the expenses of reduced misclassification rates. Apart from misclassification with the none case, the case with NO_2 sensor failure being misclassified as a NMH sensor failure is the most difficult misclassification case to avoid in AQ data-set. In all data-sets, the results with bias faults are notably improved in comparison to those with drift fault[4].

There exists several criteria for setting the optimal threshold

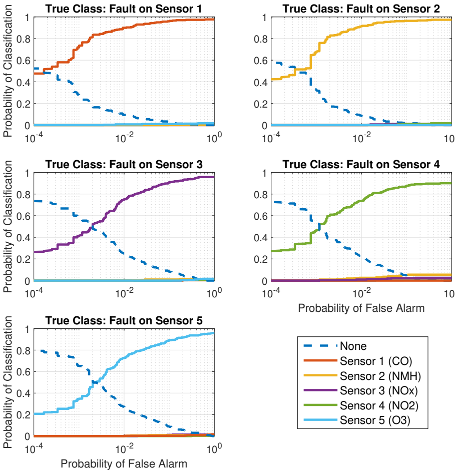


Fig. 10: Classification ROC curves for AQ data-sets under drift faults.

TABLE IV: Detection and classification accuracy based on Youden's index.

Data-set	Fault Type	γ	P_d (%)	P_f (%)	P_{di} [a] (%)	Sensors Classification Performance (%)				
						CO	NMH	NO$_x$	NO$_2$	O$_3$
AQ	Bias	0.109	95.2	1.1	**91.3**	94.8	99.8	93.0	68.5	99.9
	Drift	0.1345	93.1	7.0	90.6	95.3	96.0	89.1	86.5	86.2
						T1	T2	T3	T4	-
WSN	Bias	0.151	**97.3**	**0.2**	89.8	94.1	99.9	94.7	70.1	-
	Drift	0.213	95.0	2.1	86.0	92.7	95.4	83.7	72.4	-
						CT	V	C	MS	TRQ
PMSM	Bias	0.001	**99.9**	7.4	89.0	99.0	94.6	91.7	68.1	91.4
	Drift	0.107	97.0	2.0	81.1	90.2	76.4	83.3	65.7	90.0

[a] P_{di} = Averaged probability of correct classification on all sensors.

[2] Averaged classification performance is the average of correct classification probability on all sensors in data-set. Non-fault occurrence is considered as a separate class.

[3] Plots are not depicted with respect to the selected threshold, but with respect to the corresponding probability of false alarm. It is worth noticing that well-known confusion matrices may be obtained from these plots by selecting a desired point of operation (corresponding to a specific value of the numerical threshold γ providing the classifier output).

[4] Classification performance on different sensors of other two data-sets as well as bias faults are not shown for brevity.

TABLE V: Detection accuracy of the proposed architecture compared to the SVM classifier and the FCC technique on the test set.

Data-set	**Architecture**	**Metrics**	**Bias** (%)		**Drift** (%)	
			Weak	Strong	Weak	Strong
AQ		P_{f}	2.82	0.01	2.32	0.17
	SVM	P_{d}	79.2	98.0	70.4	88.8
	FCC	P_{d}	98.5	-	85.2	87.9
	Proposed	P_{d}	97.5	98.9	84.1	95.9
WSN		P_{f}	22.7	0.15	21.7	1.0
	SVM	P_{d}	95.9	98.5	88.2	90.3
	FCC	P_{d}	100	-	94.4	96.3
	Proposed	P_{d}	100	98.9	98.2	94.2
PMSM		P_{f}	0.05	0.06	0.11	0.15
	SVM	P_{d}	34.9	92.3	31.8	77.7
	FCC	P_{d}	15.9	99.7	25.0	50.8
	Proposed	P_{d}	58.1	99.8	56.0	96.2

value to maximize the probability of detection. In this study we selected Youden index J, i.e. maximization the vertical distance between the 45-degree line (equality line) and the point on the ROC curve [40]

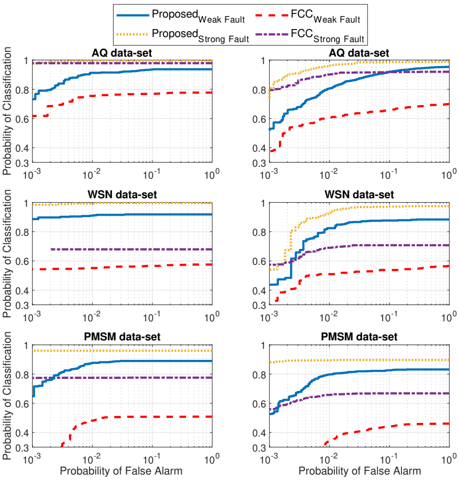
$$J = \max_{\gamma}(P_{\mathrm{d}} - P_{\mathrm{f}}). \tag{7}$$

where P_{d} is the probability of detection and P_{f} is the probability of false alarm.

Sensors classification performance on test sets of all data-sets with Youden index criteria are summarized in Tab. IV. Thresholds in Tab. IV are set by applying Youden index criteria to ROC curves from training sets. Next, all recorded probabilities are derived from test sets for obtained thresholds. On the whole, the achieved accuracy with bias faults is comparatively higher than drift faults. The best detection accuracy of 99.9% as well as very good detection accuracy of 97.3% with the lowest false alarm rate of 0.2% respectively obtained on PMSM and WSN data-sets under bias fault condition which shows excellent detection performance of the proposed SFDIA scheme. Moreover, good classification performance on most sensors is evident with highest average correct classification of 91.3%, with MS sensor on PMSM data-set as the hardest classification case.

C. Performance Comparison

Table V compares the proposed architecture with two state-of-the-art techniques previously outlined in Sec. II: (i) the SVM classifier [14] and (ii) the FCC NN [15] with 6 nodes. The SVM classifier has no control over the probability of false alarm since it does not have any threshold mechanism. Hence, to provide a fair comparison, we tuned the threshold on the proposed architecture and on the FCC technique to achieve the same probability of false alarm as the SVM classifier, and compared the probability of detection for all techniques in Tab. V. Apparently, the detection performance of the proposed architecture outperforms the SVM technique for all fault types. The performance gap between these two techniques in terms of detection accuracy becomes more evident under weak faults. More specifically, under weak drift fault for the PMSM data-set, the performance improvement in fault detection of the proposed architecture over the SVM technique is approximately 24.2%. The main reason lies in the fact that the SVM classifier takes raw-sensor data as input while the proposed architecture exploits the estimations of each sensor and feeds the residual data as input to the classifier which contains easy-to-interpret information about faults. The FCC technique exhibits similar detection performance as the proposed architecture over AQ and WSN data-sets, while on the PMSM data-set the proposed architecture turns to be better performing. In Tab. V, the detection accuracy of the FCC technique with respect to the corresponding probability of false alarm was not available for the WSN and AQ data-sets under strong bias faults (as can be seen also in Fig. 11(a)). It is worth mentioning that the detection performance on the training set resembles those shown for the test set in Tab. V.

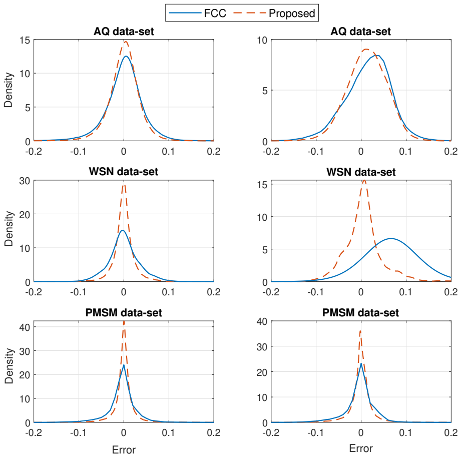


(a) Bias Fault (b) Drift Fault

Fig. 11: Averaged classification (isolation) performance comparison in terms of ROC for the test set on each data-set.

As for the isolation task, the proposed architecture achieves significant gains over the FCC technique as observed in terms of classification performance shown in Fig. 11. More specifically, the proposed architecture takes advantage of MLP classifier while the FCC technique merely uses a sliding window mechanism. The relevance of the proposed architecture as an effective SFDIA scheme is apparent.

Finally, as for the accommodation task, Fig. 12 compares the accuracy of the virtual sensors which reveals better estimation capability of the MLPs from the proposed architecture against the FCC NNs. The improvement is mainly due to the capability of the proposed technique to exploit temporal correlation. Finally, it is worth noticing that isolation and accommodation performances of the SVM technique cannot be compared due to its incapability to classify and estimate faulty sensors.



(a) Train Set (b) Test Set

Fig. 12: Accommodation performance comparison in terms of PDF of the error signals on each data-set.

VI. Conclusion

In this paper, we presented a three-stage SFDIA architecture with capability to adapt with different applications. The classifier at the third stage detects and isolates the faulty sensor from patterns within the input residual signals. The bank of estimators at the first stage allows to accommodate unreliable sensors by replacing the measurements from the identified faulty sensors. Estimators are also used at the second stage to derive the residual signals for the classifier. An extensive evaluation on three real-world data-sets from different applications indicated that the proposed SFDIA architecture attains high probability of detection and correct classification with low probability of false alarm in presence of weak bias and drift faults.

The same architecture allows large flexibility with the components in each layer (e.g. replacing the considered MLPs with RNNs), thus might achieve further performance improvements under specific circumstances. In addition, although not investigated in this work, the proposed architecture is potentially capable of handling multiple simultaneous faults, a feature to be considered in future works.

References

[1] 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 2020, accepted.*

[2] M. Wollschlaeger, T. Sauter, and J. Jasperneite, “The future of industrial communication: Automation networks in the era of the Internet of Things and Industry 4.0,” *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[3] S. Yin, S. X. Ding, and D. Zhou, “Diagnosis and prognosis for complicated industrial systems—Part I,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2501–2505, 2016.

[4] 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.

[5] H. Pham, J. Bourgeot, and M. E. H. Benbouzid, “Comparative investigations of sensor fault-tolerant control strategies performance for marine current turbine applications,” *IEEE Journal of Oceanic Engineering*, vol. 43, no. 4, pp. 1024–1036, 2018.

[6] K. Thiyagarajan, S. Kodagoda, L. Van Nguyen, and R. Ranasinghe, “Sensor failure detection and faulty data accommodation approach for instrumented wastewater infrastructures,” *IEEE Access*, vol. 6, pp. 56 562–56 574, 2018.

[7] S. Mandal, B. Santhi, S. Sridhar, K. Vinolia, and P. Swaminathan, “Nuclear power plant thermocouple sensor-fault detection and classification using deep learning and generalized likelihood ratio test,” *IEEE Transactions on Nuclear Science*, vol. 64, no. 6, pp. 1526–1534, 2017.

[8] S. Gutiérrez and H. Ponce, “An intelligent failure detection on a wireless sensor network for indoor climate conditions,” *MDPI Sensors*, vol. 19, no. 4, p. 854, 2019.

[9] 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.

[10] L. Liu, G. Han, Y. He, and J. Jiang, “Fault-tolerant event region detection on trajectory pattern extraction for industrial wireless sensor networks,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2072–2080, 2020.

[11] S. De Vito, E. Massera, M. Piga, L. Martinotto, and G. Di Francia, “On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario,” *Sensors and Actuators B: Chemical*, vol. 129, no. 2, pp. 750–757, 2008.

[12] 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.

[13] W. Kirchgässner, O. Wallscheid, and J. Böcker, “Empirical evaluation of exponentially weighted moving averages for simple linear thermal modeling of permanent magnet synchronous machines,” in *IEEE 28th International Symposium on Industrial Electronics (ISIE)*, 2019, pp. 318–323.

[14] 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.

[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] P. Goupil, “Airbus state of the art and practices on FDI and FTC in flight control system,” *Elsevier Control Engineering Practice*, vol. 19, no. 6, pp. 524–539, 2011.

[17] S. Yin, B. Xiao, S. X. Ding, and D. Zhou, “A review on recent development of spacecraft attitude fault tolerant control system,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 5, pp. 3311–3320, 2016.

[18] 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.

[19] G. Heredia and A. Ollero, “Detection of sensor faults in small helicopter uavs using observer/Kalman filter identification,” *Hindawi Mathematical Problems in Engineering*, 2011.

[20] B. Pourbabaee, N. Meskin, and K. Khorasani, “Sensor fault detection, isolation, and identification using multiple-model-based hybrid Kalman filter for gas turbine engines,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1184–1200, 2015.

[21] C. Alippi, S. Ntalampiras, and M. Roveri, “Model-free fault detection and isolation in large-scale cyber-physical systems,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 1, pp. 61–71, 2017.

[22] 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 International Journal of Aerospace Engineering*, 2018.

[23] M. Carminati, G. Ferrari, R. Grassetti, and M. Sampietro, “Real-time data fusion and MEMS sensors fault detection in an aircraft emergency attitude unit based on Kalman filtering,” *IEEE Sensors Journal*, vol. 12, no. 10, pp. 2984–2992, 2012.

[24] E. Naderi and K. Khorasani, “Data-driven fault detection, isolation and estimation of aircraft gas turbine engine actuator and sensors,”

Mechanical Systems and Signal Processing, vol. 100, pp. 415 – 438, 2018.

[25] J. Gao, J. Wang, P. Zhong, and H. Wang, "On threshold-free error detection for industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2199–2209, 2017.

[26] M. R. Napolitano, Y. An, and B. A. Seanor, "A fault tolerant flight control system for sensor and actuator failures using neural networks," *Elsevier Aircraft Design*, vol. 3, no. 2, pp. 103–128, 2000.

[27] M. L. Fravolini, G. Campa, M. Napolitano, and Yongkyu Song, "Minimal resource allocating networks for aircraft SFDIA," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, vol. 2, 2001, pp. 1251–1256 vol.2.

[28] S. Gururajan, M. L. Fravolini, H. Chao, M. Rhudy, and M. R. Napolitano, "Performance evaluation of neural network based approaches for airspeed sensor failure accommodation on a small UAV," in *21st IEEE Mediterranean Conference on Control and Automation (MED)*, 2013, pp. 603–608.

[29] C. Lo, J. P. Lynch, and M. Liu, "Distributed reference-free fault detection method for autonomous wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 5, pp. 2009–2019, 2013.

[30] P. Tchakoua, R. Wamkeue, M. Ouhrouche, F. Slaoui-Hasnaoui, T. A. Tameghe, and G. Ekemb, "Wind Turbine Condition Monitoring: State-of-the-Art Review, New Trends, and Future Challenges," *Energies*, vol. 7, no. 4, pp. 1–36, April 2014.

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

[32] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[33] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 162–169, 2017.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] W. Kirchgässner, O. Wallscheid, and J. Böcker, "Deep residual convolutional and recurrent neural networks for temperature estimation in permanent magnet synchronous motors," in *IEEE International Electric Machines & Drives Conference (IEMDC)*, 2019, pp. 1439–1446.

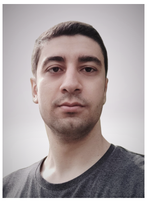
[36] H. Alwi, C. Edwards, and C. P. Tan, "Fault tolerant control and fault detection and isolation," in *Fault Detection and Fault-Tolerant Control Using Sliding Modes*. Springer, 2011, pp. 7–27.

[37] T. Muhammed and R. A. Shaikh, "An analysis of fault detection strategies in wireless sensor networks," *Elsevier Journal of Network and Computer Applications*, vol. 78, pp. 267–287, 2017.

[38] G. Campa, M. L. Fravolini, M. Napolitano, and B. Seanor, "Neural networks-based sensor validation for the flight control system of a B777 research model," in *IEEE American Control Conference (ACC)*, vol. 1, 2002, pp. 412–417.

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

[40] W. J. Youden, "Index for rating diagnostic tests," *Wiley Cancer*, vol. 3, no. 1, pp. 32–35, 1950.



Hossein Darvishi (GS'20) received the B.Sc. degree from the Kermanshah University of Technology, Iran, and the M.Sc. degree (*ranked first*) in telecommunications engineering from K.N. Toosi University of Technology, Iran, in 2016 and 2018, respectively. He is currently pursuing the Ph.D. degree in electronics and telecommunications with the Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway. His research interests include statistical signal processing, machine learning, Internet of Things and wireless sensor networks.

Domenico Ciuonzo (S'11-M'14-SM'16) is an Assistant Professor at University of Napoli Federico II. He holds a Ph.D. in Electronic Engineering from the University of Campania "L. Vanvitelli", Italy. Since 2011, he has been holding several visiting researcher appointments. Since 2014, he has been (Area) Editor of several IEEE, IET, and ELSEVIER journals. He is the recipient of the Best Paper award at IEEE ICCCS 2019, the 2019 Exceptional Service award from IEEE Aerospace and Electronic Systems Society, and the 2020 Early-Career Technical Achievement award from IEEE Sensors Council for sensor networks/systems. His research interests include data fusion, statistical signal processing, wireless sensor networks, the Internet of Things and machine learning.

Eivind R. Eide received his M.Eng. degree in Information and Computer Engineering from the University of Cambridge in 2018. He is currently working in Kongsberg Digital. He has been working on the use of machine learning for anomaly detection in IoT systems and on combining machine learning with first principle simulators for real time optimization in Digital Twins.

Pierluigi Salvo Rossi (SM'11) was born in Naples, Italy, in 1977. He received the Dr.Eng. degree (*summa cum laude*) in telecommunications engineering and the Ph.D. degree in computer engineering from the University of Naples "Federico II", Italy, in 2002 and 2005, respectively. He held visiting appointments at the Dept. Electrical and Computer Engineering, Drexel University, USA; at the Dept. Electrical and Information Technology, Lund University, Sweden; at the Dept. Electronics and Telecommunications, Norwegian University of Science and Technology (NTNU), Norway; and at the Excellence Center for Wireless Sensor Networks, Uppsala University, Sweden. From 2005 to 2008, he held postdoctoral positions with the Dept. Computer Science and Systems, University of Naples "Federico II", Italy; with the Dept. Information Engineering, Second University of Naples, Italy; and with the Dept. Electronics and Telecommunications, NTNU, Norway. From 2008 to 2014, he was an Assistant Professor (tenured in 2011) in telecommunications with the Dept. Industrial and Information Engineering, Second University of Naples, Italy. From 2014 to 2016, he was an Associate Professor in signal processing with the Dept. Electronics and Telecommunications, NTNU, Norway. From 2016 to 2017, he was a Full Professor in signal processing with the Dept. Electronic Systems, NTNU, Norway. From 2017 to 2019, he was a Principal Engineer with the Dept. Advanced Analytics and Machine Learning, Kongsberg Digital AS, Norway. Since 2019, he has been a Full Professor of statistical machine learning with the Dept. Electronic Systems, NTNU, Norway, and also the Director of IoT@NTNU. His research interests fall within the areas of communication theory, data fusion, machine learning, and signal processing. He serves as an Executive Editor for the IEEE COMMUNICATIONS LETTERS since 2019, an Area Editor for the IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY since 2019, an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS since 2019, and an Associate Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS since 2015. He was a Senior Editor from 2016 to 2019 and an Associate Editor from 2012 to 2016 of the IEEE COMMUNICATIONS LETTERS. He was awarded as an Exemplary Senior Editor of the IEEE COMMUNICATIONS LETTERS in 2018.

Paper 3

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

H. Darvishi, D. Ciuonzo and P. S. Rossi

2021 IEEE International Conference on Networking, Sensing and Control (ICNSC)

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).

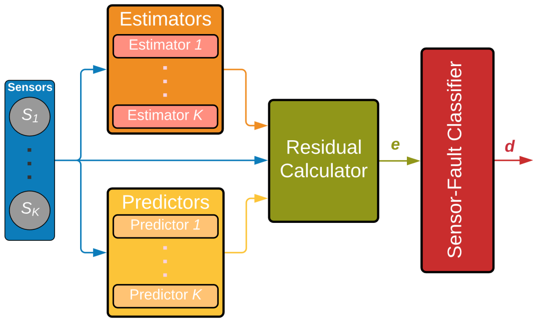


Figure 1: Block diagram of the proposed SFDIA architecture.

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.

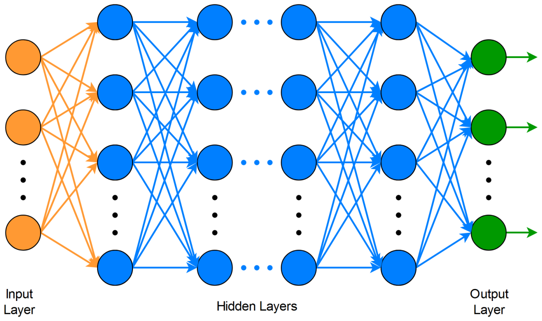


Figure 2: The generic structure of MLP.

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.

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

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.

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

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:

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

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.

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

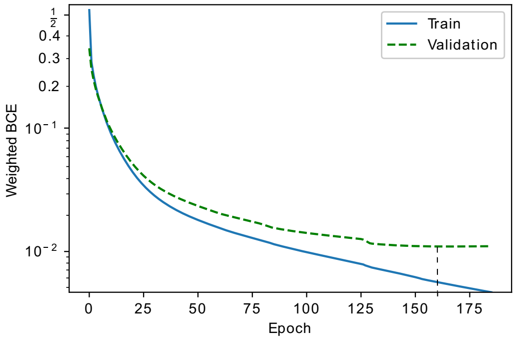


Figure 3: Weighted BCE loss of the classifier (cf. Eq. (5)) for training and validation sets during the training phase.

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.

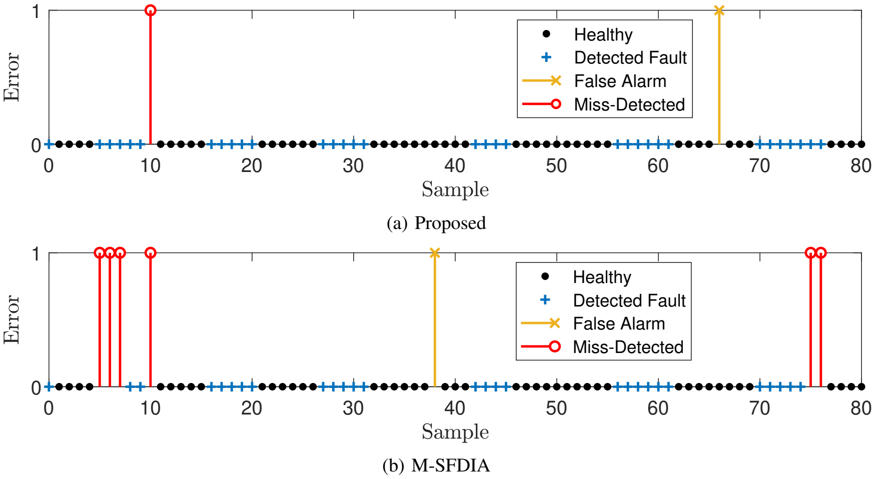
Fig. 5 illustrates detection and classification (i.e. detection plus isolation) performance by means of the corresponding ROC curves.[1] More specifically, the results show a clear performance improvement achieved by the proposed architecture w.r.t. the M-SFDIA architecture for *both* (*i*) *detection* 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

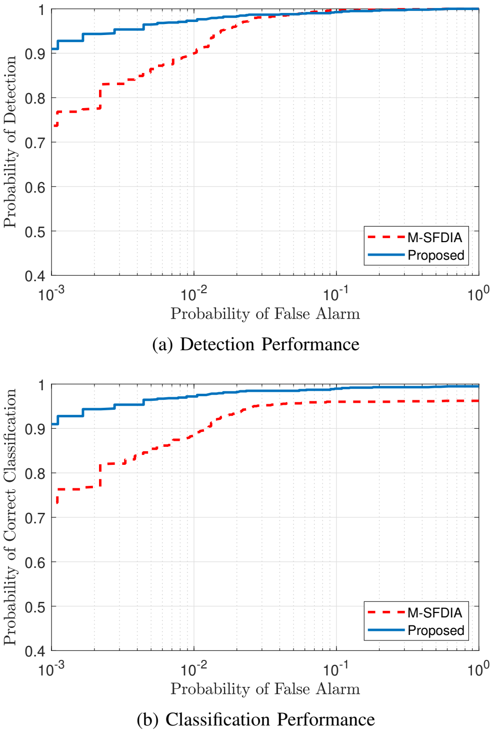
[1] Receiver operating characteristics (ROC) curves show the trade-off between the *probability of detection* (resp. *probability of correct classification*) and the *probability of false alarm* by varying γ, when assessing detection (resp. identification) performance. In detail, the *probability of 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

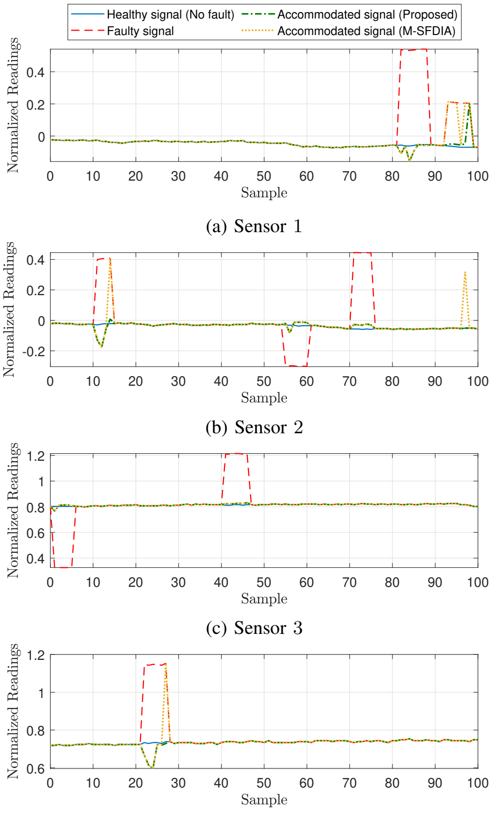
[1] Z. Yang, N. Meratnia, and P. Havinga, "An online outlier detection technique for wireless sensor networks using unsupervised quarter-sphere support vector machine," in *IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2008, pp. 151–156.

[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



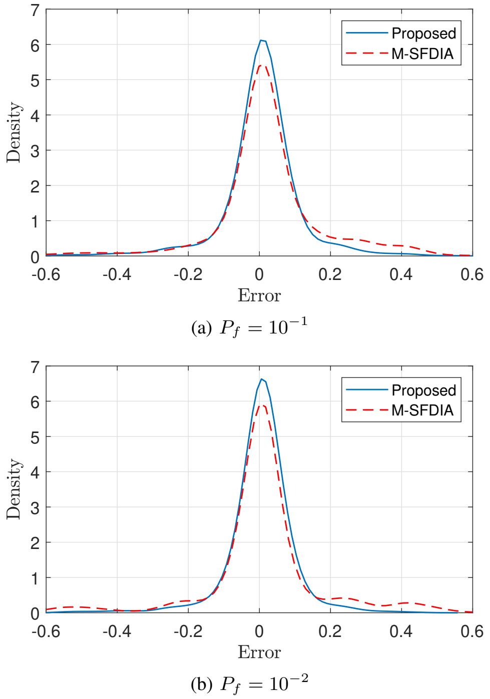
(a) Sensor 1

(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.

Paper 4

Exploring a Modular Architecture for Sensor Validation in Digital Twins

H. Darvishi, D. Ciuonzo and P. S. Rossi

2022 IEEE Sensors

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.**

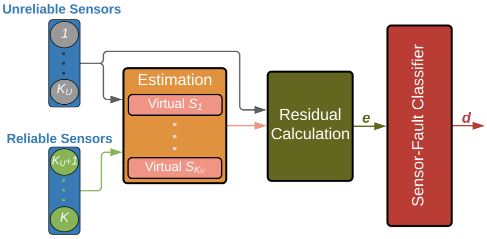


Fig. 1: Block diagram of the SFDIA system.

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.

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, $\boldsymbol{x}_{(-k)}[n]$, $k = 1, \dots, K_U$, 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.

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

[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. $\{a, a + 1, \dots, b\}$), 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:

$$\boldsymbol{\Delta}[n] \triangleq \left[(\hat{x}_1(n)-x_1(n)) \quad \cdots \quad (\hat{x}_{K_U}(n)-x_{K_U}(n))\right]^T \tag{2}$$

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 $d_k[n] \in [0,1],\ k=1,\dots,K_U$ denoting pseudo-probabilities of the sensors being faulty, namely

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

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 $f:\mathbb{R}^{i\times 1}\rightarrow\mathbb{R}^{j\times 1}$. 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 $\texttt{GRU-RS}^{\text{vs}}$).

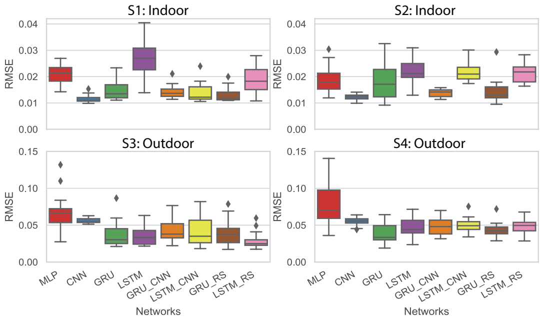
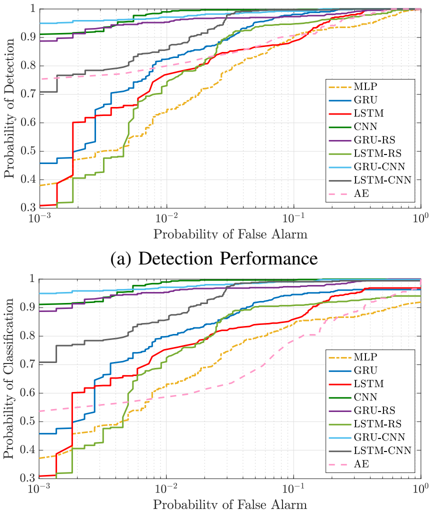


Fig. 2: Box-plot of estimation RMSE for each virtual sensor.



(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.

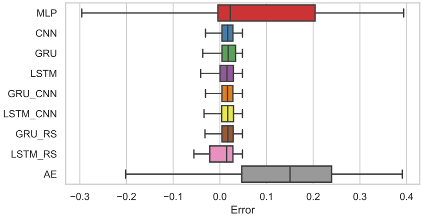


Fig. 4: Accommodation performance comparison in terms of averaged distribution of the error signals for the fixed false alarm probability of 10^{-2}.

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.

Paper 5

A Machine-Learning Architecture for Sensor Fault Detection, Isolation and Accommodation in Digital Twins

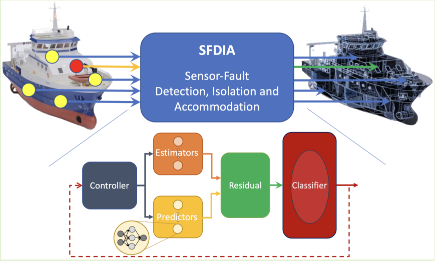
H. Darvishi, D. Ciuonzo and P. S. Rossi

IEEE Sensors Journal

A Machine-Learning Architecture for Sensor Fault Detection, Isolation and Accommodation in Digital Twins

Hossein Darvishi, *Student Member, IEEE*, Domenico Ciuonzo, *Senior Member, IEEE*,
and Pierluigi Salvo Rossi, *Senior Member, IEEE*

***Abstract*—Sensor technologies empower Industry 4.0 by enabling integration of in-field and real-time raw data into digital twins. However, sensors might be unreliable due to inherent issues and/or environmental conditions. This paper aims at detecting anomalies instantaneously in measurements from sensors, identifying the faulty ones and accommodating them with appropriate estimated data, thus paving the way to reliable digital twins. More specifically, a real-time general machine-learning-based architecture for sensor validation is proposed, built upon a series of neural-network estimators and a classifier. Estimators correspond to virtual sensors of all unreliable sensors (to reconstruct normal behaviour and replace the isolated faulty sensor within the system), whereas the classifier is used for detection and isolation tasks. A comprehensive statistical analysis on three different real-world data-sets is conducted and the performance of the proposed architecture is validated under hard and soft synthetically-generated faults.**



***Index Terms*—Digital twin, Fault diagnosis, Machine learning, Neural networks, Sensor validation.**

I. Introduction

DIGITAL TWINS (DTs) have recently emerged in several industrial applications and exploit Internet of Things (IoT) technology [1]. More specifically, most environments have been pervaded by the extensive use of spatially-distributed sensors, generating enormous amount of heterogeneous data over time, which requires advanced integrated solutions involving sensing, communication, and processing [2]–[4]. DTs represent one of the main products for building advanced analytics over such data and extract relevant information for prediction and effective control. DTs have been widely employed in various sectors such as industry [5], health care [6] and smart cities [7], [8], where their capabilities to visualize and treat with a perpetual stream of real-time sensor data is enabling new opportunities. Leveraging sensor data enables DTs to model system dynamics effectively for remote monitoring and controlling, for safety and risk analysis and for maintenance purposes. Since DTs rely on accurate sensor data, system performance may be affected severely by sensor failures. Sources of sensor faults are commonly found in: (i) *Hardware and inherited limitations* - sensors are electronic components and can collect inaccurate measurements or stop working without any indication due to low production quality, calibration issues, low battery level, end of life span, poor connections [9]; (ii) *Harsh environment* - in real-world scenarios, sensors can be deployed in inaccessible and unattended environments with possibility of unlikely situations which would hinder sensors performance [10]; (iii) *Malicious attacks* - faulty data can be injected by an attacker into a vulnerable system [11], [12].

A fault in a system refers to a complete (or partial) malfunction and manifests over a permanent (or transient) time span. As shown in Fig. 1, the most common types of sensor faults in a sensor network are defined (a detailed discussion of sensor faults is found in [13], [14]). Depending on the characteristics of sensor data, faults can be classified as following:

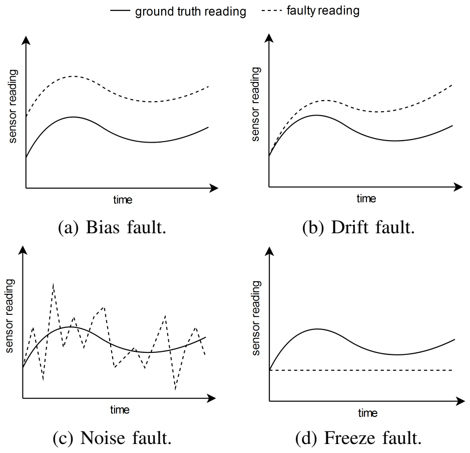
1) *Bias* fault: also known as offset fault, the deviation from nominal values is given by an additive constant bias;
2) *Drift* fault: sensor readings drift with a small slope from nominal values (drift faults are more subtle since they

This work was partially supported by the Research Council of Norway under the project SIGNIFY within the IKTPLUSS framework (project nr. 311902). Part of this work was presented at the IEEE International Conference on Networking, Sensing and Control (ICNSC) 2021.

H. Darvishi is with the Department of Electronic Systems, Norwegian University of Science and Technology, 7491 Trondheim, Norway, and with the Signal Processing Laboratory (LTS4), École polytechnique fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland (e-mail: hossein.darvishi@ntnu.no).

D. Ciuonzo is with the Department of Electrical Engineering and Information Technologies (DIETI), University of Naples “Federico II”, 80125 Naples, Italy (e-mail: domenico.ciuonzo@unina.it).

P. Salvo Rossi is with the Department of Electronic Systems, Norwegian University of Science and Technology, 7491 Trondheim, Norway, and with the Department of Gas Technology, SINTEF Energy Research, 7491 Trondheim, Norway (e-mail: salvorossi@ieee.org).



(a) Bias fault.

(b) Drift fault.

(c) Noise fault.

(d) Freeze fault.

Fig. 1: Types of sensor faults.

appear gradually over time and their effect is not very apparent);

3) *Noise* fault: an increased noise level in sensor readings (when noise power is much larger than usual, it is an indication of sensor malfunctioning).
4) *Freeze* fault: also known as stuck-at fault, the sensor readings stuck at a constant value (i.e. the variance of the readings becomes zero);

The impact of sensor faults would affect stability, reliability and accuracy of the system depending on the specific application. Hence, to fully utilize the expected properties of the DT, it is essential to continuously evaluate and amend sensor data. From this perspective, prompt **Sensor Fault Detection, Isolation and Accommodation (SFDIA)** is one key issue for deploying DTs while assuring reliable performance. SFDIA indeed consists of *three* parts:

- *fault detection*, i.e. determining sensor fault(s) within the system's sensor network;
- *fault isolation*, i.e. identifying specific faulty sensors and block their measurement feeding to DT;
- *fault accommodation*, i.e. feeding DT with some other replaced trustworthy data.

In what follows, related literature is reviewed by focusing on recent progress on sensor fault diagnosis and SFDIA approaches. It is worth highlighting that the following discussion leaves out the (huge) corpus of literature dealing with soft/virtual sensor design (see, for instance, the excellent survey [15]). Indeed, it should be noted the latter field is out of the scope of this paper, as soft/virtual sensors are usually meant to provide predictions only for analyzing, monitoring and/or controlling purposes (corresponding to the first layer of the proposed SFDIA architecture). Also, we emphasize that this work focuses on sensor faults only, i.e. the monitored physical process does not exhibit any anomaly while the measurement data do (e.g. errors in data acquisition and/or communication). Process fault detection and related analysis is beyond the scope of this work.

A. Related Work

In the last years, the main advancements in fault diagnosis technology have relied on the milestone concept of *redundancy* which embraces a wide spectrum of design solutions, e.g. redundancy can be accomplished by either *hardware* or *analytical* schemes. Within the class of *hardware-based approaches* (also referred to as physical-based approaches), multiple identical sensors (i.e. sensing the same physical parameter) along with a voting scheme (or more sophisticated techniques, see [16]) are employed to detect, isolate and accommodate sensor failures [17]–[19]. If the difference (namely, the residual signal) between the measured signal of a sensor and each other sensor in the set is considerably high, the aforementioned sensor is declared faulty and its data is replaced with those from the remaining (identical) sensors. For instance, the aforementioned assumptions apply to the case of homogeneous WSNs, where neighboring nodes are assumed to measure roughly the same parameter [16]. Conventional physical-redundancy approaches however cannot handle cases with simultaneous failures of identical sensors, as they do not capitalize the statistical dependence of measurements originating from other sensor types [17], [18]. Moreover, in many applications, it is impractical to implement these approaches due to space and/or weight and/or cost constraints [18].

Accordingly, it is not surprising that methods adopting *analytical redundancy* have gained increasing attention within the research on SFDIA [20]–[22]. Unlike physical redundancy, the latter approaches exploit correlations and functional relationships within the system instead of introducing additional (redundant) hardware. Still, it is worth highlighting that the above two philosophies *are not* mutually exclusive and hybrid approaches can be pursued toward the sophisticated design of fault-tolerant DTs. Analytical redundancy can be usually implemented by either *model-based* or *data-driven* techniques.

Model-based SFDIA have been mostly investigated in the context of power systems [23], e.g. using electrical dynamics equations [20] or Luenberger observers [24]. Some other methods have focused on the detection and accommodation of proportional-type faults in nonlinear systems [25], [26]. Unfortunately, those methods (a) usually result in high complexity, (b) require an explicit, application-dependent, formulation of the analytical redundancy relationship among sensors and (c) are seldom able to handle multiple sensor faults simultaneously. On the contrary, *data-driven* approaches relying on historical data have recently received large interest, starting from simpler methods (e.g. auto-regressive models with exogenous inputs (ARX) [27]) to more complicated (non-linear) learning approaches (e.g. random forest (RF) [28], support vector machines (SVMs) [29], [30] and NNs [31], [32]). Indeed, *data-driven* techniques *do not require exact knowledge* of the mathematical model for sensor fault diagnosis.

Specifically, SVM-based classification was one of the relevant attempts to *detect* sensor faults in WSNs, in both batch [29] and online forms [30], which showed relatively small computational costs, but limited performance. Successive works [33], [34] have also employed the SVM approach to allow *both* detection and identification of faults: a binary classifier was trained from the residuals of each sensor. Specifically, in the former case [33], the residual signals were generated by comparing the true measurements with a single (global) observer designed by including fault models. Conversely, in the latter case [34], a residual was obtained from each (correlated) sensor pair via an ARX model, thus providing multiple classification outputs for a given sensor then aggregated at a higher level.

A second important class of approaches for SFDIA relies on the well-known Autoencoder (AE) NN [12], [21], [35], [36]. Indeed, the AE is an unsupervised learning technique capable of learning and extracting hidden representations from raw data and it is thus suited for fault detection. Hence, once trained, the AE can provide a reconstructed estimate of the sensors' measurements, thus allowing straightforward computation of residuals (i.e. the difference between inputs and outputs of the AE). Specifically, an AE-based (aided by exogenous inputs) sensor validation scheme for a heating, ventilation and air conditioning system was proposed in NNs [36]. Detection and identification are simply performed by comparing overall and per-sensor residuals to a given threshold. A similar AE-based SFDIA method is presented in [21] for an air quality controlling system, with identification scheme performed via a more involved sensor validity index. In both works [21], [36] accommodation is simply performed by using the AE output associated to the sensor(s) declared as faulty. Differently, a more sophisticated proposal uses an additional denoising AE (a supervised learning technique) to perform the accommodation task [12], namely to clean faulty data. Despite their simplicity, AE-based SFDIA approaches can suffer however from degraded performance under weak-faults, as the latter type of faults does not considerably impact correlations in data.

Multi-layer perceptron (MLP) NNs (including variants) have also been proved to perform satisfactorily for a number of relevant sensor fault diagnosis tasks [22], [37], [38], including heavy-duty diesel engines' and aircrafts, based on a *sensor-centric viewpoint*. Indeed, in all the aforementioned works, *one MLP estimator per each sensor* is designed (solely on the basis of other sensors' measurements) and detection/identification is based on the evaluation of the residual vector. Accommodation is then performed by using the estimator(s) associated to the sensors declared as faulty. Specifically, the proposal in [37] adopts fully-connected cascade NNs (i.e. MLPs allowing direct connections across different hidden layers) for the sensor estimator design, while [22] considers a hybrid structure with a linear NN and resource allocation network (a variant of well-known radial basis function NN) for the same task. More recently, a plain MLP estimator (exploiting the sole spatial correlation among sensors) has been proven to provide reliable detection with low false-alarm rate as well [38].

A different rationale is pursued in [31], where a *single* Deep belief network (a Bayesian type of NNs) has been trained (in a supervised fashion) to detect a faulty condition whereas sensor identification is naively carried out based on the maximum deviation from data mean-value. Along the same lines, a general approach is presented to detect and identify sensor faults using either a single Recurrent NN (RNN) or an MLP [39] for predicting next-step measurements and comparing with actual ones. A disentanglement regularization term on the NN loss function is introduced to help the algorithm coping with propagation of faults to non-faulty sensors in the identification stage. Unfortunately, the accommodation stage is not taken into account in the above work. Interestingly, also a dynamic Bayesian network has succeeded in sensor fault detection and accommodation exploiting spatial and temporal correlations in the context of intelligent connected vehicles [40]. Still, its training difficulty (in terms of both parameter and structure learning) appears limiting in large-scale sensor systems.

Recently, the sensor-centric viewpoint in [22], [37], [38] has further been exploited to devise a *modular* SFDIA (M-SFDIA) method based on MLP NNs in [32], [41], with focus on supporting DTs. The proposed structure consists of a set of estimators (each associated to a sensor) providing residual signals as well as replacements (estimates) for faulty data. Therein a supervised classifier is trained to make detection & identification decisions upon the residual signals by leveraging their (possibly-nonlinear) relationships. An experimental analysis on three real-world data-sets has demonstrated satisfactory performance of M-SFDIA method. Although promising (from the estimators' design viewpoint), M-SFDIA architecture does not completely exploit the temporal correlations among sensors within the monitored system.

B. Paper Contribution

In view of the previous discussion, some proposals are restricted to a given vertical domain (e.g. aircraft [37], vehicle [34] or HVAC system [36] monitoring), thus *lacking a general formulation*. Secondly, part of the literature evaluates corresponding proposals on *private* (e.g. [39], [40]) or *simulated* (e.g. [28], [36], [37]) measurement data, thus precluding reproducibility and convincing evaluation, respectively. Thirdly, a number of the discussed works evaluate their proposals only on a *single fault type* (e.g. bias [21], [39] or drift [22]). Equally important, some architectures are only *limited to fault detection* [29], [30]. On the other hand, some recent proposals do not foresee all the three tasks in their original formulation, e.g. the identification and accommodation tasks in [12] and [39], respectively. Still, even when all three tasks can be carried out, in some cases *only spatial correlation* [22], [36], [38] is used to accommodate faulty measurements. Finally, some approaches have a *limited modularity* [12], [21], [39]. Accordingly, the *main contributions* of this article are summarized as follows:

- A *real-time* and *modular* data-driven SFDIA architecture is developed, fully exploring (viz. learning) *spatial* and *temporal* dependence in sensory data. The proposed architecture relies on *the novel use of a pair of regressors* for each sensor, performing *estimation* and *prediction* operations, respectively. In the former case, each estimator is leveraging readings from other sensors only to obtain a "virtual measurement". Conversely, each predictor plays a complementary role (to the estimator) by using only previous data from the sensor under consideration to obtain an analogous virtual measurement. Hence, their joint adoption enables the proposed architecture to ultimately exploit spatio-temporal correlation within the system, thus supporting nearly-instantaneous fault detection and isolation performance.

- The dissimilarity measured by predictors (resp. estimators) and measurements, referred to as *residual signals*, are then used as the perfect candidate for designing a reliable classifier able to perform both *fault detection* (i.e. whether there is a fault in the whole sensor set) and *identification* (i.e. which sensors are faulty).
- The proposed approach employs MLP NNs for both regression (estimation and prediction) and classification modules to capture and process analytical redundancy relations while keeping a *reasonable complexity* at the operational stage. In the latter case, a *multi-task* MLP NN (i.e. each sensor condition is seen as a binary classification task) is designed for detecting and (if any) identifying multiple faulty sensors via a *single* neural network.
- Moreover, classifier decisions, residual signals and virtual measurements are exploited by a *a specifically-designed controller* to make corrections on sensor models inputs and improve overall system performance both for detection and isolation tasks. Specifically, in a feedback loop, the controller is in charge of replacing corrupted input data and, consequently, avoiding propagation of faults throughout the architecture.
- The performance of the proposed SFDIA architecture is assessed on *three real-world (public) data-sets* [42]–[44] which are corrupted with (a) four relevant fault types (bias, drift, noise and freeze) and (b) different levels of faults (with special emphasis on weak faults, as they are more difficult to detect).
- The proposal is compared with two state-of-the-art machine-learning-based architectures [12], [41] from both performance (in terms of detection delay and probabilities of detection, false alarm, and correct identification, and accommodation error) and computational complexity (in terms of number of trainable parameters) standpoints.

The present work extends earlier conference paper [45], which (a) presented only an intermediate version of the proposed novel architecture (no controller block), (b) reported a significantly-smaller experimental analysis (focusing only on the WSN data-set [43]), (c) considered a smaller set of baselines in the comparison and (d) assessed the effectiveness of the SFDIA approach only on bias faults.

The remainder of this paper is structured as follows: in Sec. II, the proposed data-driven SFDIA architecture is presented and the functionalities of each block are illustrated; Sec. III describes the configuration of the NNs and the related training process; the description of the data-sets and the framework for fault generation are provided in Sec. IV; Sec. V presents and discusses the numerical performance of the proposed architecture in contrast with benchmarks from the current literature. Finally, concluding remarks and future directions of research are given in Sec. VI.

Notation - Lower-case bold letters indicate vectors; $\boldsymbol{I}_N$ denotes the null column vector of length N; $(\cdot)^T$ refers to the transpose operator, $\in$ is the set membership, and $\mathcal{O}(\cdot)$ denotes the Landau notation.

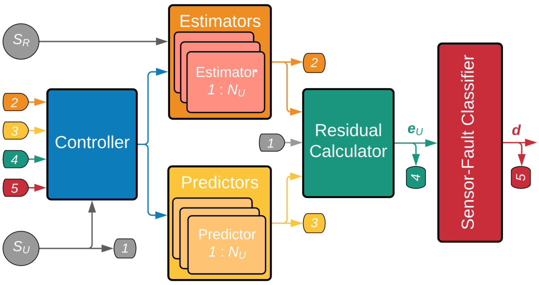


Fig. 2: Block diagram of the proposed SFDIA architecture.

II. SFDIA

The proposed method aims to exploit the full potential of *spatial* and *temporal* correlation among sensors in a system. Specifically, it is assumed that the sensors are divided into *two* sets: (i) the set of unreliable sensors $\mathcal{S}_U$, containing sensors that are vulnerable to faults; and (ii) the set of reliable sensors $\mathcal{S}_R$, which, depending on the working system, include sensors whose flawless functionality can be guaranteed [41]. This (ideal) level of reliability could be associated to: a meta-sensor modeling a group of identical sensors (enjoying hardware redundancy), high-quality sensors, a proper design and safe working environment, a device being at the middle of life span [46], or context measurement information which is assumed to have significantly higher reliability than the considered networked sensor system. In a more general sense, any reliable source of data correlated with the unreliable sensors could be included in the set of reliable sensors. In the following, without loss of generality, it is assumed $\mathcal{S}_U = \{1, \ldots, N_U\}$ and $\mathcal{S}_R = \{N_U + 1, \ldots, N\}$, where N_U and N denote the number of unreliable sensors and total number of sensors, respectively. Also, for compactness, N_R denotes the cardinality of the reliable set $\mathcal{S}_R$ (i.e. $N_R = N - N_U$).

A. Architectural Overview

The block diagram of the proposed SFDIA architecture is shown in Fig. 2. It consists of five building blocks (controller, estimators, predictors, residual calculator, classifier) arranged in *four layers*, whose function is explained as follows. The *first layer* contains two parallel blocks, the estimators block and the predictors block, each providing a virtual measurement for all the unreliable sensors in the system either regressed via other sensors' observations (i.e. the estimator) or based only on previous measurements of the same sensor under consideration (i.e. the predictor). The *second layer* is responsible for the computation of a discrepancy measure between the true and each calculated virtual measurement, usually in the form of a function of the residual signals. The *third layer* is fed with the aforementioned discrepancy measures and is able to perform a multidimensional classification to (a) detect a faulty condition and (b) identify the corresponding faulty sensors. Finally, the

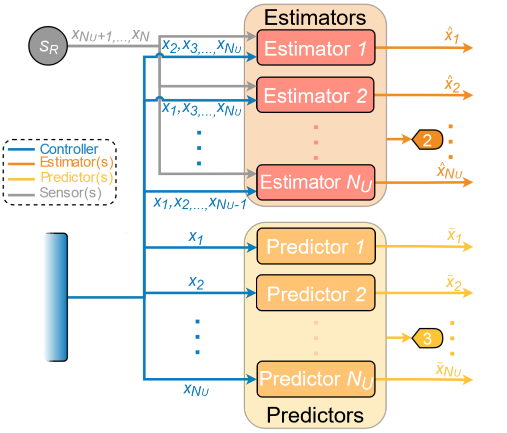


Fig. 3: Diagram detailing the estimators and predictors blocks.

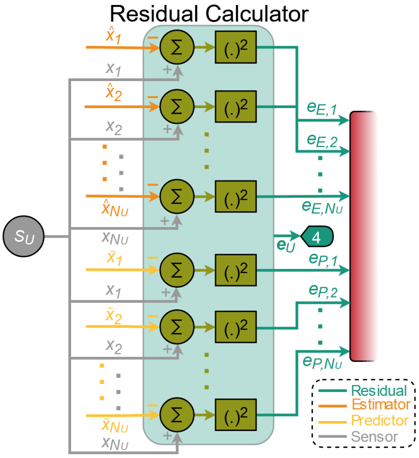


Fig. 4: Diagram detailing the residual calculator block.

fourth layer controls the inputs of the blocks in the first layer in order to preserve estimators and predictors accuracy, by avoiding error propagation.

The present architecture improves over the one proposed in [41] where the main novelty is the introduction of the controller and the predictors. Despite the addition of these two modules, it is worth remarking that the proposed architecture retains the advantages of *modularity* and *real-time implementation*. Indeed, regarding the former property, the proposed approach allows the implementation of diversified ML techniques for different modules and a more flexible deployment, also taking computational/hardware limitations into account. Differently, regarding the latter property, each of the proposed modules can be flawlessly implemented in real-time since they are all based on a sliding window implementation. Finally, given the adoption of MLP-based solutions for the estimators/predictors (Sec. II-B) and the classifier (Sec. II-D), the proposed implementation also retains simplicity. The following subsections detail each of the four layers constituting the proposed approach.

B. First Layer: Estimation & Prediction

The first layer aims to model the unreliable sensors within the system and is based on *two subsystems*: (a) a bank of estimators and (b) a bank of predictors.

As detailed in Fig. 3, the bank of **estimators** is composed of N_U estimators (each associated to an unreliable sensor), each providing the estimation $\hat{x}_s[n]$ of the measurement (at current time step n) from its corresponding unreliable sensor $s \in \mathcal{S}_U$. Each estimator receives as input the vector $\boldsymbol{x}_{(s)}$ collecting all existing sensors readings (from current time step n back to L_e previous time steps using a sliding window mechanism) except the one from the sensor to be estimated $\{\mathcal{S}_U \cup \mathcal{S}_R - s\}$, i.e.

$$\hat{x}_s[n] = f_s^{(H_v,N_v)}(\boldsymbol{x}_{(s)}[n], \ldots, \boldsymbol{x}_{(s)}[n-L_e]) \,, \qquad (1)$$

where $f_s^{(H_v,N_v)}(\cdot)$ denotes the function model of the MLP-based estimator for the sth sensor, being H_v and N_v the number of hidden layers and the number of neurons, respectively. Previous time samples are fed into the estimators in order to exploit the *temporal correlation* among the input signals.

The bank of **predictors** operates a complementary approach. Each of the N_U predictors provides a prediction $\tilde{x}_s[n]$ of the measurement (at current time step n) from its corresponding unreliable sensor $s \in \mathcal{S}_U$. Each predictor receives as input the readings $x_s[\cdot]$ of the sensor to be predicted (from previous time step $n-1$ back to L_p previous time steps using a sliding-window mechanism), i.e.

$$\tilde{x}_s[n] = g_s^{(H_v,N_v)}(x_s[n-1], \ldots, x_s[n-L_p]) \,, \qquad (2)$$

where $g_s^{(H_v,N_v)}(\cdot)$ denotes the function model of the MLP-based predictor for the sth sensor, again being H_v and N_v the number of hidden layers and the number of neurons, respectively.

C. Second Layer: Residual Evaluation

The second layer computes the square of residual signals i.e. the difference of sensors reading with their respective estimation or prediction values (see Fig. 4), namely

$$e_{E,s}[n] = (x_s[n] - \hat{x}_s[n])^2, \qquad (3)$$
$$e_{P,s}[n] = (x_s[n] - \tilde{x}_s[n])^2, \qquad (4)$$

for each unreliable sensor $s \in \mathcal{S}_U$. Residual signals are used as input to the classifier in the third layer as they contain effective information for fault classification. It is worth noticing that the proposed SFDIA architecture enjoys modularity and generality: thus other discrepancy measures (other than that used in Eqs. (3) and (4)) may be adopted *without any substantial change* in the subsequent layers.

D. Third Layer: Classification

An MLP **classifier**, meant to work in real-time, is used for fault *detection* and the *identification* of the faulty sensors, and its detailed structure is shown in Fig. 5. Denoting

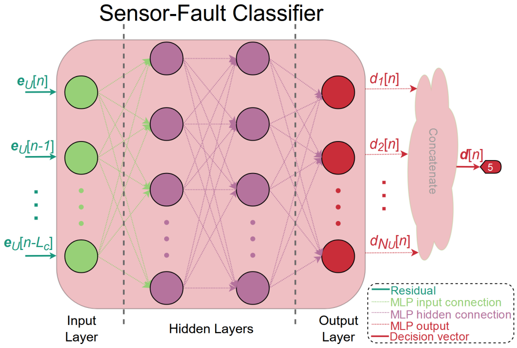


Fig. 5: Structure of the MLP-based classifier block.

$\boldsymbol{e}_U[n] = (e_{E,1}[n], \ldots, e_{E,N_U}[n], e_{P,1}[n], \ldots, e_{P,N_U}[n])^T$ the residual vector containing the residual signals of all N_U 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 $\boldsymbol{e}_U[n], \ldots, \boldsymbol{e}_U[n-L_c]$. Conversely, a decision vector $\boldsymbol{d}[n] = (d_1[n], d_2[n], \ldots, d_{N_U}[n])^T$ represents the output of the classifier and identifies which among the unreliable sensors are suspected to be in failure, i.e.

$$\boldsymbol{d}[n] = \boldsymbol{h}^{(H_c,N_c)}(\boldsymbol{e}_U[n], \ldots, \boldsymbol{e}_U[n-L_c]) , \qquad (5)$$

where $\boldsymbol{h}^{(H_c,N_c)}(\cdot)$ denotes the function model of the MLP-based classifier, being H_c and N_c are the number of hidden layers and the number of neurons of the classifier, respectively. More specifically, the sth entry of the decision vector, i.e. $d_s[n] \in [0,1],\ s = 1, \ldots, N_U$, represents a pseudo-probability for the sth unreliable sensor to be faulty. Apparently, $d_s[n] = 1$ (resp. $d_s[n] = 0$) represents the situation in which the system declares with maximum confidence the sth sensor to be faulty (resp. fault-free). As a consequence, a vector $\boldsymbol{d}[n] = \boldsymbol{0}_{N_U}$ indicates healthy operation of *all* the sensors within the system at time n.

Therefore, faulty sensors are identified via a threshold-based logic for each of the components of the decision vector. The considered threshold will be denoted γ in what follows. Principally, herein faulty sensors are detected and identified/isolated when the entries of the decision vector $\boldsymbol{d}[n]$ exceed the threshold γ. Specifically, $\max_{s=1}^{N_U} d_s[n] \gtrless \gamma$ is used for detection. Accordingly, for the identification task, the set of identified faulty sensors (denoted with $\mathcal{I}_U$) is obtained as $\mathcal{I}_U \triangleq \{s \in \mathcal{S}_U \,:\, d_s[n] > \gamma\}$.

It is worth mentioning that, from overall SFDIA system perspective, the measurements from the sensors declared faulty are *replaced* (viz. *accommodated*) with their corresponding *estimates* in order to preserve system utility.

E. Fourth Layer: Control

The role of the control block is to preserve the performance of the proposed SFDIA method when faults occur. Referring to Fig. 2, this block operates at the beginning of each time step, and controls inputs-outputs of both estimators and predictors regarding the latest residual signals and the decision vector $\boldsymbol{d}[n-1]$.

The symbol $\phi_{E,s}$ (resp. $\phi_{P,s}$) denotes the average residual signal for the sth estimator (resp. predictor) computed with a moving average over a window of size L_r starting from time step $n-1$ while *excluding* the identified faulty time steps. The signal $\phi_{E,s}$ (resp. $\phi_{P,s}$) of the unreliable sensor s is used by the controller as a metric to define the estimation (resp. prediction) accuracy of the corresponding estimator (resp. predictor).

In the first step, after applying the proposed SFDIA scheme at time step $(n-1)$, the elements of the decision vector $\boldsymbol{d}[n-1]$ larger than a predefined threshold υ identify faulty sensors for the controller. Then, the following process will be conducted at the beginning of each time step n. To keep the discussion simple, we will generically refer to sth sensor as the one identified as faulty.

As for the **predictor controlling** scheme, if the estimator's average residual signal $\phi_{E,s}$ is smaller than a certain value τ (i.e. the system tolerable level of deviation), the estimator output $\hat{x}_s[n-1]$ *replaces* the respective sensor input $x_s[n-1]$ to the corresponding predictor. In other words, the predictor in Eq. (2) will be then fed as:

$$\tilde{x}_s[n] = g_s^{(H_v,N_v)}(\underbrace{\hat{x}_s[n-1]}_{\text{replacement}}, \ldots, x_s[n-L_p]), \qquad (6)$$

This logic is intended to use only those estimates whose quality is better than the faulty-data within the sth predictor.

As for the **estimator controlling** scheme, if the predictor's average residual signal $\phi_{P,s}$ smaller than both (i) the system tolerable level of deviation τ and (ii) $\phi_{E,s}$, the predictor output $\tilde{x}_s[n]$ is obtained and *replaces* the respective sensor input $x_s[n]$ (updates all estimators' input vectors except $\boldsymbol{x}_{(s)}[n]$) to the estimators. In other words, we have $\forall s_\star \in \mathcal{S}, s_\star \neq s$:

$$\hat{x}_{s_\star}[n] = f_{s_\star}^{(H_v,N_v)}(\underbrace{\tilde{\boldsymbol{x}}_{(s_\star)}[n]}_{\text{replacement}}, \ldots, \boldsymbol{x}_{(s_\star)}[n-L_e]) , \qquad (7)$$

where the vector $\tilde{\boldsymbol{x}}_{(s_\star)}[n]$ collects all existing sensors readings except for $s_\star$ and with sth reading being replaced by $\tilde{x}_s[n]$. Otherwise, if $\phi_{E,s}$ is smaller than the system tolerable level of deviation[1], the estimator output $\hat{x}_s[n]$ is obtained and replaces the respective sensor input $x_s[n]$ (updates all input vectors except $\boldsymbol{x}_{(s)}[n]$) to the estimators. Specifically, $\forall s_\star \in \mathcal{S}, s_\star \neq s$:

$$\hat{x}_{s_\star}[n] = f_{s_\star}^{(H_v,N_v)}(\underbrace{\hat{\boldsymbol{x}}_{(s_\star)}[n]}_{\text{replacement}}, \ldots, \boldsymbol{x}_{(s_\star)}[n-L_e]) , \qquad (8)$$

where the vector $\hat{\boldsymbol{x}}_{(s_\star)}[n]$ collects all existing sensors readings except for $s_\star$ and with sth reading being replaced by $\hat{x}_s[n]$. This logic is intended to replace the input faulty-data with estimates/predictions whose accuracy are better than the input faulty-data (i.e. $x_{(s)}[n]$) to all estimators (except the corresponding sensor s estimator). We highlight that, in all three cases, *no architectural modification* (i.e. varying input size for

[1] In other words, the corresponding estimator is providing better accuracy than the corresponding predictor, i.e. $\phi_{E,s} < \phi_{P,s}$ and $\phi_{E,s} < \tau$.

the estimators and predictors) *is required* for the blocks of the proposed SFDIA method.

Conversely, in the case of *no-fault detected*, this block merely slides the window forward in time to update both $\phi_{P,s}$ and $\phi_{E,s}$ by using the recent residual signals $\boldsymbol{e}_U[n-1]$.

A pseudo-code of the controlling block process is given in Algorithm 1. It is worth remarking that substitution of faulty inputs with either estimated or predicted values maintains estimators and predictors accuracy (by avoiding error propagation) and results in better accommodation performance as well as increased detection rate.

Algorithm 1 Controller

```
procedure CONTROLLER
    Input: $\boldsymbol{d}$, $\boldsymbol{e}_U$, and $x_s$ for all $s \in \mathcal{S}_U$;
    At starting of each time step $n$:
    for $s = 1 : N_U$ do                    ▷ Corresponds to $s \in \mathcal{S}_U$
        if $d_s[n-1] > \upsilon$ then        ▷ Identified faulty
            if $\phi_{E,s} < \tau$ then
                Feed $\hat{x}_s[n-1]$ instead of $x_s[n-1]$ to the prediction block as input;
            if $\phi_{P,s} < \phi_{E,s}$ and $\phi_{P,s} < \tau$ then
                Obtain $\tilde{x}_s[n]$ from Eq. (2);
                Feed $\tilde{x}_s[n]$ instead of $x_s[n]$ to the estimation block as input;
            else if $\phi_{E,s} < \tau$ then
                Obtain $\hat{x}_s[n]$ from Eq. (1);
                Feed $\hat{x}_s[n]$ instead of $x_s[n]$ to the estimation block as input;
        else                                 ▷ Identified healthy
            Update $\phi_{E,s}$, $\phi_{P,s}$ using $\boldsymbol{e}_U[n-1]$;
```

III. NNs Configuration

The MLP is a feed-forward layered NN made up of an input layer, an arbitrary number of hidden layers and an output layer, where neurons are interconnected in forward direction from the input to the output layer [47]. The MLP is a suitable NN for regression and classification tasks and is capable to model arbitrary non-linearities while exhibiting fine generalization on unseen data [38], [41]. MLP NNs in the proposed SFDIA architecture are trained using an optimization algorithm [48].

A. Estimators and Predictors

Each MLP-based **estimator** has been implemented with $(N-1)\cdot(L_e+1)$ inputs, H_v hidden layers with N_v neurons each, and a single output. Conversely, each MLP-based **predictor** has been implemented with L_p inputs, H_v hidden layers with N_v neurons each, and a single output. For both the estimators and predictors, the hyperbolic tangent has been selected as the activation function for the hidden layers, while the linear activation function has been selected for the output layer.

Training was accomplished using the Nesterov-accelerated adaptive moment estimation (Nadam) optimization algorithm [49] over real-world data-sets. The mean square error (MSE) loss function was considered as the relevant optimization metric for both the estimators and the predictors. More specifically, the MSE loss for sth estimator and predictor, respectively, is defined as

$$\mathcal{L}_{\mathrm{es},s}(\boldsymbol{\phi}_s) = \frac{1}{w}\sum_{j=0}^{w-1}(\hat{x}_s^j(\boldsymbol{\phi}_s) - x_s^j)^2 \tag{9}$$

$$\mathcal{L}_{\mathrm{pr},s}(\boldsymbol{\varphi}_s) = \frac{1}{w}\sum_{j=0}^{w-1}(\tilde{x}_s^j(\boldsymbol{\varphi}_s) - x_s^j)^2 \tag{10}$$

where w is the number of samples in each batch, $\boldsymbol{\phi}_i$ (resp. $\boldsymbol{\varphi}_i$) represents the vector of trainable parameters of the sth estimator (resp. predictor). Finally, $\hat{x}_s^j$ (resp. $\tilde{x}_s^j$) is the network output associated to the sth estimator (resp. predictor), while x_s^j denotes the true measurement (viz. the labeled sample) of sth sensor.

B. Classifier

The MLP-based **classifier** has been implemented with $2N_U\cdot(L_c+1)$ inputs, H_c hidden layers with N_c neurons each, and N_U outputs. The hyperbolic tangent has been selected as the activation function for the hidden layers, while a logistic (viz. sigmoid) activation function has been selected for each node in the output layer. In order to accomplish both detection & identification tasks, a loss capitalizing *multitask learning* is employed for training the classifier. Specifically, a *weighted sum* of the losses of the N_U binary (fault/no-fault) classification tasks associated with the unreliable sensors is minimized, i.e.

$$\mathcal{L}_{\mathrm{cl}}\left(\boldsymbol{\theta}_{\mathrm{shared}}, \{\boldsymbol{\theta}_s\}_{s=1}^{N_U}\right) \triangleq \sum_{s=1}^{N_U}\lambda_s\,\mathcal{L}_s\left(\boldsymbol{\theta}_{\mathrm{shared}}, \boldsymbol{\theta}_s\right) \tag{11}$$

In the above formula, the weight λ_s indicates the preference level of the sth task (i.e. detection of a fault at sth unreliable sensor). It is worth noticing that the multitask objective function allows the proposed classifier to solve multiple learning tasks *at once* (i.e. via a single NN). Accordingly, in the above expression, $\boldsymbol{\theta}_{\mathrm{shared}}$ represents the vector of *shared* parameters of the MLP common to all the N_U different tasks, whereas $\boldsymbol{\theta}_s$ indicates the vector of parameters which are *task-specific* for sth learning task.

In this work uniform weighting is adopted, i.e. $\lambda_s = 1/N_U$ for $s = 1, \ldots, N_U$, and a binary cross-entropy (BCE) loss function for *all* the N_U binary tasks $\mathcal{L}_1(\cdot), \ldots, \mathcal{L}_{N_U}(\cdot)$. The BCE loss for sth task is formally defined as

$$\mathcal{L}_s\left(\boldsymbol{\theta}_{\mathrm{shared}}, \boldsymbol{\theta}_s\right) = -\frac{1}{w}\sum_{j=0}^{w-1}\Big\{y_s^j \ln d_s^j(\boldsymbol{\theta}_{\mathrm{shared}}, \boldsymbol{\theta}_s) + \\ (1-y_s^j)\ln\left(1 - d_s^j(\boldsymbol{\theta}_{\mathrm{shared}}, \boldsymbol{\theta}_s)\right)\Big\} \tag{12}$$

where w is the number of samples in each batch. Furthermore d_s^j is the entry of classifier output associated to sth sensor, while y_s^j denotes (the $0/1$ representation of) the true fault status (viz. the labeled sample) of sth sensor. The same optimization algorithm (i.e. Nadam) as the estimators/predictors is employed for training the aforementioned MLP-based classifier.

C. Summary of the training phase

The whole training process of the proposed SFDIA architecture is summarized in Algorithm 2. In detail, the estimators and predictors (Sec. III-A) are trained *only* with healthy (fault-free) data, according to the inputs specified via Eqs. (1) and (2), respectively. A similar comment applies to the associated validation set for estimators and predictors.

Conversely, the classifier block (Sec. III-B) is also trained based on *faulty* training data, by including the controller and residual evaluation blocks in an open-loop fashion. Indeed, during the training process, the controller is given the classifier label set (i.e. the binary-valued vector pattern collecting true faulty/healthy condition for all the sensors) as input, in the place of the classifier decision vector. This is to avoid detrimental effects due to training instability of the classifier. However, since perfect identification provided by the label set may lead to overfitting, 25% of controller decisions are randomly dropped out to help the classifier generalize better during the training process. The corresponding validation set for the classifier block includes faulty measurements as well.

Algorithm 2 Training Process

```
procedure INITIALIZE
    Preparing training set and create a falsified copy;
    Random weights and biases for all networks;
    Set initial value of all other parameters to zero;
procedure ESTIMATORS AND PREDICTORS
    Input: healthy training set;                    ▷ Fault free
    while Epoch number $<$ Max epoch or Validation loss Not triggered do
        for each epoch do
            Calculate MSE;
            Update weights and biases using Nadam optimization;
            Calculate validation loss;
procedure CLASSIFIER
    Input: Falsified training set;
    while Epoch number $<$ Max epoch or Validation loss Not triggered do
        for each epoch do
            Obtain $\hat{x}_s$ $\tilde{x}_s$ for all $s \in \mathcal{S}_U$ with respect to the controller mechanism;
            Calculate residual signals;
            Feed residual signals to the classifier;
            Calculate weighted BCE;
            Update weights and biases using Nadam optimization;
            Calculate validation loss;
```

IV. Data-sets and Faults Setup

A. Data-sets Setup

For the sake of a complete evaluation, three real-world data-sets (similarly as [41]) have been employed to assess the proposed SFDIA architecture. Specifically, the **air quality (AQ)** data-set [42] includes readings from *five* chemical sensors (assumed to be unreliable, namely $N_U = 5$) which are complemented by measurements originating from humidity and temperature sensors (assumed to be reliable, i.e. $N_R = 2$). Such a sensor system is aimed at pollution-level evaluation in an Italian city. The second data-set is related to a **wireless sensor network (WSN)** with *four* unreliable sensors measuring indoor and outdoor humidity and temperature [43]. Labeled anomalies injected into the data-set were omitted and only the temperature readings of the multi-hop section of data-set are considered as unreliable readings ($N_U = 4$, $N_R = 0$) for our analysis. The last data-set includes multiple sensors on a **permanent-magnet synchronous motor (PMSM)** [44], [50]. Among the collected measurements[2], (i) coolant temperature, (ii) voltage and (iii) current (summation of q and d components), (iv) motor speed and (v) torque are included in the unreliable set $\mathcal{S}_U$ (thus $N_U = 5$), whereas the stator yoke temperature is assumed to belong to the reliable set $\mathcal{S}_R$ (thus $N_R = 1$).

Before feeding the data-sets to the proposed architecture, sensors readings in each data-set are normalized using min-max scaling on the training set to avoid polarization during the learning process. Finally, the entire rows containing missing values are ignored from the data-sets. Table I summarizes data-sets description.

TABLE I: Data-sets description. The reliable sensors in each data-set are highlighted in italic.

Data-set	Samples	N_U	N_R	Attributes
AQ	8991	5	2	**Multivariate**, **time-series**; carbon monoxide (CO), non-metanic hydrocarbons (NMH), nitrogen oxides (NO_x), nitrogen dioxide (NO_2) and ozone (O_3) gas concentrations, as well as measurements of *temperature* and *humidity*
WSN	4589	4	0	**Multivariate**, **time-series**; four temperature sensors: two indoor, two outdoor
PMSM	55000	5	1	**Multivariate**, **time-series**; coolant temperature, voltage and current (summation of q and d components), motor speed, torque and *stator yoke* temperature

B. Sensor Faults Modeling

The performance of the proposed SFDIA architecture is evaluated under transient faults.

Also, with the aim of adapting and examining the proposed architecture according to DTs' needs, *four different fault types* with varying severity levels were modeled, as detailed hereinafter. It is worth highlighting that the practice of modeling simulated faults superimposed to real data is a common practice in the evaluation of SFDIA systems (e.g. [12], [29], [39]), as (i) real faulty measurements are sporadic and very hard to obtain and (ii) simulated faults also allow quantifying accommodation performance. This is also to highlight the *generality* of the proposed architecture in accommodating diversified faulty conditions.

[2] The readings were sampled with 1.5 s-intervals and the first 55k readings were picked after sampling.

Bias faults: for each bias fault, a constant bias b injected to the normal operation data-sets for M consecutive samples as follows

$$x_{s,b}[n] = \begin{cases} x_{s,h}[n] + b \ , & 0 \leq n - m < M \\ x_{s,h}[n] \ , & \text{otherwise} \end{cases} \tag{13}$$

where $x_{s,h}[n]$ and $x_{s,b}[n]$ are the healthy and possibly-faulty reading of sensor $s \in \mathcal{S}_U$, respectively, under bias fault. Finally, m denotes the starting time instant of the fault.

Drift faults: as for drift fault, an additive term drifts to bias level b in M samples and remains for K samples ($M > K$), namely:

$$x_{s,d}[n] = \begin{cases} x_{s,h}[n] + \frac{b(n-m+1)}{M} \ , & 0 \leq n - m < M \\ x_{s,h}[n] + b \ , & M \leq n - m < M + K \\ x_{s,h}[n] \ , & \text{otherwise} \end{cases} \tag{14}$$

where $x_{s,d}[n]$ is the possibly-faulty reading of sensor $s \in \mathcal{S}_U$ under drift-type faults.

Noise faults: in the latter case, zero-mean additive Gaussian noise $w[n] \sim \mathcal{N}(0, c)$ is added to the sensor measurement for M consecutive samples, i.e.:

$$x_{s,g}[n] = \begin{cases} x_{s,h}[n] + w[n] \ , & 0 \leq n - m < M \\ x_{s,h}[n] \ , & \text{otherwise} \end{cases} \tag{15}$$

where $x_{s,g}[n]$ is the possibly-faulty reading of sensor $s \in \mathcal{S}_U$ under noise-type faults and c is the variance of the noise.

Freeze faults: for freeze-type faults, sensor output stuck at previous reading for M consecutive samples as follows

$$x_{s,f}[n] = \begin{cases} x_{s,h}[m-1] \ , & 0 \leq n - m < M \\ x_{s,h}[n] \ , & \text{otherwise} \end{cases} \tag{16}$$

where $x_{s,f}[n]$ is the possibly-faulty reading of sensor $s \in \mathcal{S}_U$ under freeze-type faults.

V. Numerical Results

The effectiveness of the proposed architecture for detection, isolation and accommodation of sensor faults has been assessed by means of a comprehensive analysis conducted on the three previously-described real world data-sets. The following section first details the considered system setup and employed parameters, for the sake of reproducibility (Sec. V-A). Then, the working principle of the two relevant SFDIA baselines used for comparison is recalled (Sec. V-B). Finally, the SFDIA performance is reported and discussed (Sec. V-C).

A. System Setup and Parameters

Training and Evaluation Setup: MLP NNs within the proposed architecture were trained using the first 70% and 15% of samples of each data-set as train set and validation set, respectively. The rest ending 15% of samples of each data-set was used as test set for performance evaluation. A validation process based on *early stopping* method [51] was employed during the training phase to avoid over-fitting: the training process was stopped if the loss on the validation set had not decreased for 20 consecutive epochs or if the maximum number of epochs was reached[3].

Hyperparameter specification of proposed approach: As in [41], a similar configuration for the classifiers and the estimators was considered. More specifically, estimators and predictors with $H_v = 1$ hidden layer, $N_v = 10$ nodes per hidden layer and $L_v = L_p = 10$, along with a classifier with $H_c = 2$ hidden layers, $N_c = 15$ nodes per hidden layer and $L_c = 10$ were trained. Table II lists MLPs' configurations and corresponding hyper-parameters of the proposed architecture. In addition, the predefined thresholds τ and υ are set to 0.15 and 0.9 for the controller, respectively. The threshold τ needs to be adjusted with respect to the system tolerable level of deviation as well as the estimators/predictors accuracy, whereas threshold υ is selected heuristically according to the system performance on the validation set.

Random generation of synthetic faults: The four types of faults considered in this work are synthetically generated [12], [29], [39] according to the corresponding models detailed in Sec. IV-B on the top of the *real measurement data* described in Sec. IV-A. Unless otherwise stated, the fault absolute level b (with unbiased random positive and negative faults) and noise variance c are assumed uniformly distributed between 0.2 and 0.4 to represent weak fault signals. The fault length (M and K) is also assumed uniformly distributed between 3 and 11 consecutive samples to represent transient faults[4]. It is worth stressing that the uniform distribution choice for the fault level b (resp. the noise variance c) and the fault length (M and K) helps the classifier to generalize better without focusing on a specific fault level/length [37], [39]. To verify the robustness of the proposed architecture against simultaneous faults, *up to three concurrent faulty sensors* were considered for the (fault-)generation process.

Training phase of classifier block: Fig. 6 shows the evolution of the classifier loss function vs. number of epochs (during the training phase) on both training (solid lines) and validation (dashed lines) sets, under *bias* faults. Indeed, validation and training losses under other fault types resemble those shown under bias fault and are thus omitted for brevity. For completeness, both the weighted (multitask) BCE (cf. Eq. (11)) and the per-sensor BCE (cf. Eq. (12)) are reported in Figs. 6a and 6b, respectively. As evident from the curves, the training phase on WSN data-set stops after ≈ 260 epochs ("□" marker) by early-stopping mechanism as the validation loss stops decreasing. Conversely, the training phase on the other two data-sets stops after reaching the maximum number (400) of epochs.

B. Considered Baselines

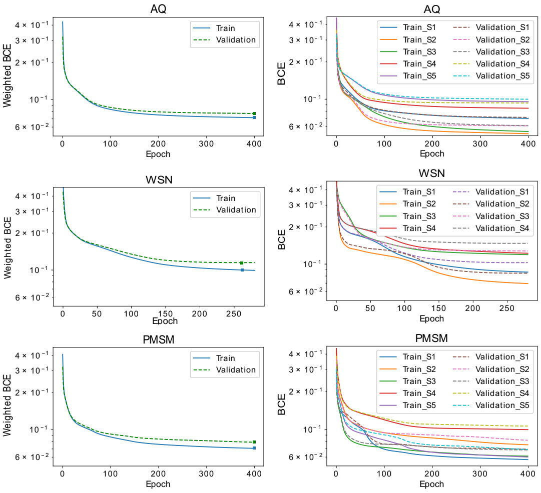
Results of the proposed approach in terms of detection, identification and accommodation performance are compared

[3] We implement the proposed architecture and other baselines using Keras Python API running on TensorFlow version 2.9.2 on MacBook pro M1 CPU 2.1-3.2 GHz with 16 GB memory.

[4] Under freeze fault, the fault length (M) is uniformly distributed between 100 and 400 consecutive samples due to smooth oscillating (WSN and PMSM) data-sets. Smaller fault lengths cause negligible faults on the working data-sets.

TABLE II: Configuration of the proposed architecture.

Parameter	Estimator	Predictor	Classifier
No. of input nodes	$(N-1)\cdot(L_v+1)$	L_p	$2N_U\cdot(L_c+1)$
No. of output nodes	1	1	N_U
No. of hidden layers	1	1	2
No. of nodes per hidden layer	10	10	15
Output activation	Linear	Linear	Sigmoid
Hidden layers activation	Tanh	Tanh	Tanh
Optimizer	Nadam	Nadam	Nadam
Loss function	MSE	MSE	BCE
Maximum epochs	400	400	400
Batch size	20 (50 for PMSM)	20 (50 for PMSM)	200
Learning rate	$4\cdot10^{-4}$ (10^{-3} for PMSM)	$4\cdot10^{-4}$ (10^{-3} for PMSM)	10^{-3}



(a) Weighted loss. (b) Per-sensor loss.

Fig. 6: Training and validation loss of the classifier during the training phase under bias fault.

with *two* state-of-the-art architectures: (i) M-SFDIA [41] and (ii) AE [12].

Similar to the proposed method, our previous M-SFDIA proposal is able to detect and isolate faulty sensors from patterns within the input residual signals. However, solely a bank of estimators is used to derive the residual signals, and to accommodate unreliable sensors in M-SFDIA method. Additionally, the controller block is absent in M-SFDIA. Furthermore, the original M-SFDIA's decision logic was designed to detect, isolate and accommodate only *up to one faulty sensor*. For this reason and for the sake of a fair comparison, the same decision logic as the proposed method was used (see Sec. II-D) to enable the M-SFDIA method to detect, isolate and accommodate multiple sensors simultaneously.

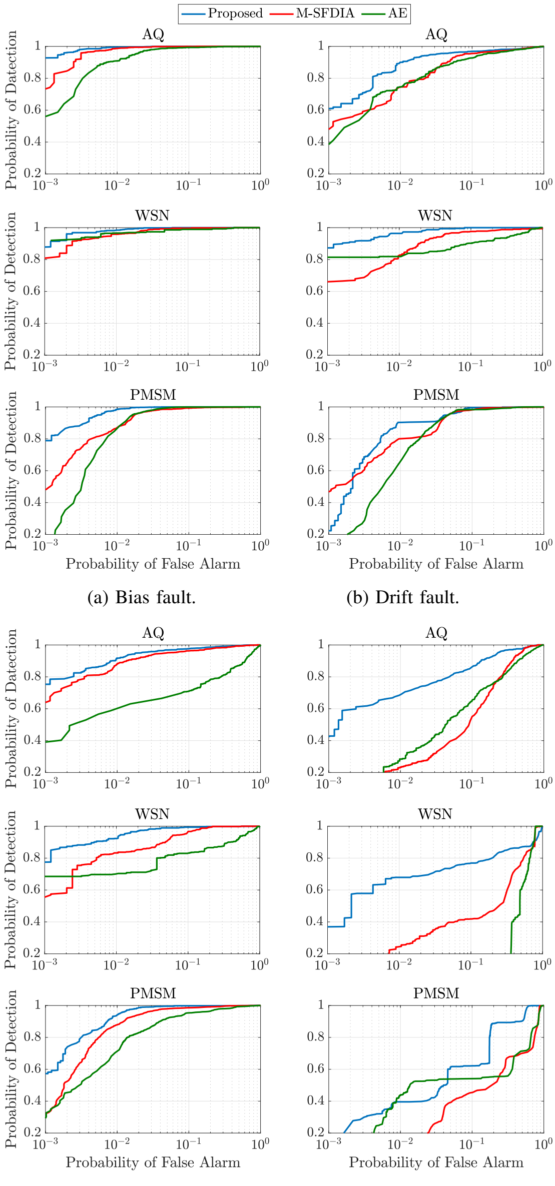
Conversely, the AE-based architecture devised in [12] is based on a *two-stage* approach. Specifically, the first stage is represented by a (standard) AE to learn data correlations among sensors, and detect anomalies (viz. faults) by tracking the MSE between input and output of the AE. As for the accommodation task, a second stage based on a (supervised) denoising AE is then used to clean faulty data. It is worth noticing that the identification task for AE architecture was not addressed in the original work [12]. Indeed, in the aforementioned AE-based method, the overall MSE of input and output (reconstructed) vector of the first AE is compared to a predefined threshold for fault detection only. As opposed to the aforementioned decision logic, herein (for the sake of a fair comparison) the squared error between the corresponding input and output *for each entry* (viz. unreliable sensor) is traced. Then, this error is compared with a predefined threshold σ, enabling the AE method to both detect & identify the faulty sensors[5]. Specifically, similar to the proposed method, $\max_{s=1}^{N_U} e_{AE,s}[n] \gtrless \sigma$ is used for detection, where $e_{AE,s}[n]$ is the squared error for the sth unreliable sensor. Accordingly, for the identification task, the set of identified faulty sensors is obtained as $\mathcal{I}_U \triangleq \{s \in \mathcal{S}_U \,:\, e_{AE,s}[n] > \sigma\}$.

C. Performance Analysis and Comparison

Fig. 7 illustrates *fault detection performance* in terms of probability of detection vs. probability of false alarm, i.e. showing the receiver operating characteristic (ROC) curves. In this case, a fault rate[6] $F_R = 0.1$ is considered. Also, ROC performance is reported *separately* for each of the three data-sets and for all four fault typologies considered. It is evident that the proposed architecture outperforms the two baselines for all four fault types. Specifically, the best detection rate is attained on AQ data-set when bias faults are present. Also, for all architectures, detection accuracy under bias faults appears to be generally higher than the other types of faults. Moreover, as can be seen, AE architecture fails to detect freeze faults on the WSN data-set. Indeed, drift and freeze faults are "trickier" to detect since they slowly appear in the system and have a less-appreciable effect on spatio-temporal correlations within the system.

[5] Numerical results (not shown for brevity) based on the original detection logic as [12], namely $\sum_{s=1}^{N_U} e_{AE,s}[n] \gtrless \sigma$ (and a matched identification logic, i.e. $\mathcal{I}_U \triangleq \{s \in \mathcal{S}_U \,:\, e_{AE,s}[n] > \sigma/N_U\}$) highlighted worse performance than the considered variant, due to the inability to cope with weak (and transient) faults.

[6] Fault rate refers to the ratio between the number of faulty and non-faulty samples.



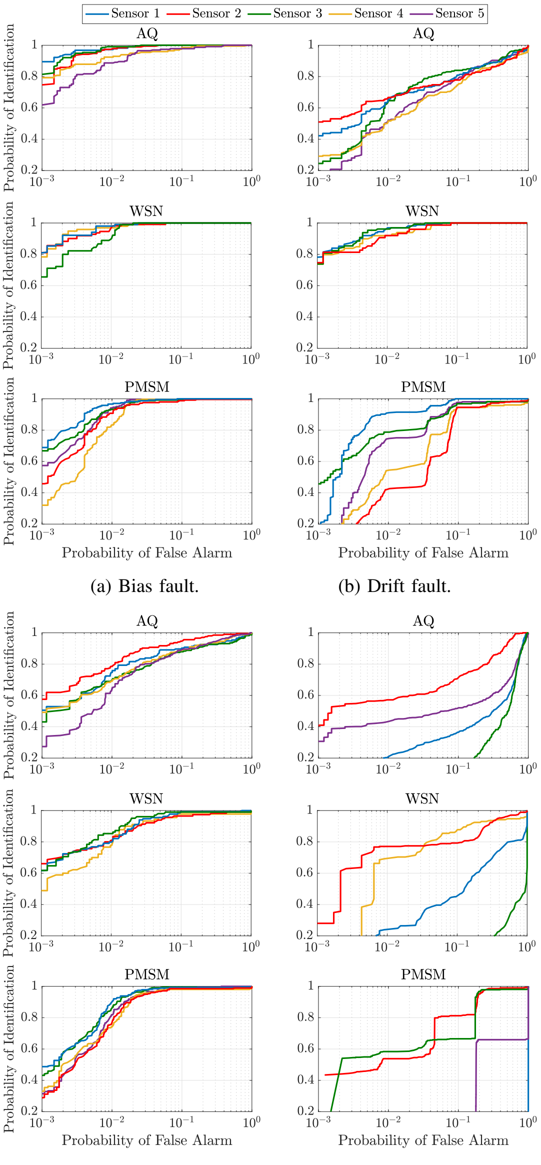
(a) Bias fault.

(b) Drift fault.

(c) Noise fault.

(d) Freeze fault.

Fig. 7: Detection performance in terms of ROC curves for all architectures over different fault types.



(a) Bias fault.

(b) Drift fault.

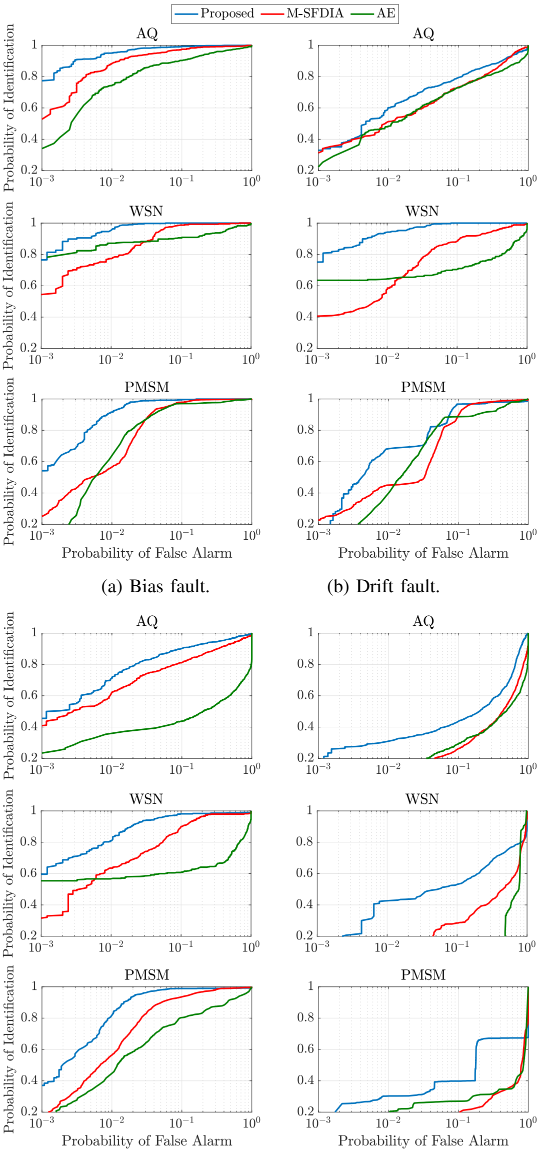
(c) Noise fault.

(d) Freeze fault.

Fig. 8: Identification (isolation) performance in terms of ROC curves for the proposed architecture over different fault types. Sensor numbers refer to sensor indices.

Delving into real-time performance of SFDIA architectures, in Tab. III a *detection delay analysis*[7] for fixed false alarm rate of 10^{-2} is reported. Specifically, the *expected detection delay* is evaluated, defined as the average number of samples needed by an SFDIA architecture to detect a faulty sensor. The latter delay is indeed another important indicator of the SFDIA framework performance, which has a crucial effect on DTs functionality. In the experiments, the fault rate is set to $F_R = 0.5$ to generate a sufficient number of fault events allowing to obtain a reliable estimate of the aforementioned metric. Results highlight that the proposed architecture achieves the *lowest*

[7] Every span of simultaneous faults is considered as a unified fault.



(a) Bias fault. (b) Drift fault.

(c) Noise fault. (d) Freeze fault.

Fig. 9: Averaged identification (isolation) performance in terms of ROC curves for all architectures over different fault types.

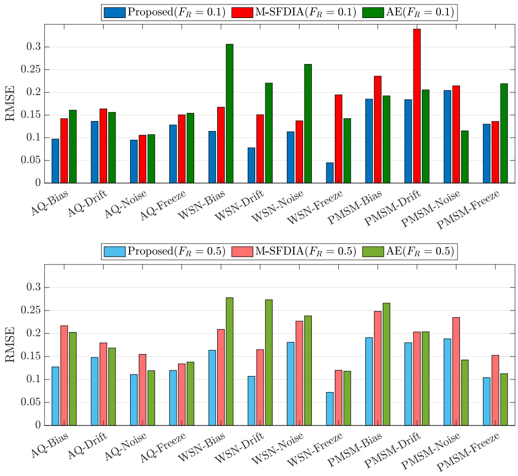


Fig. 10: Comparison of accommodation performance in term of RMSE ($P_f = 10^{-2}$).

detection delay in comparison to the state-of-the-art for all data-sets and fault types considered. Specifically, the average detection delay for the proposed architecture is confined below 1 sample (except for the AQ data-set with drift fault-types), whereas the other two architectures *always require a longer span to detect fault(s) within the system*. The most evident performance difference is observed on the PMSM data-set for drift faults (boldface in Tab. III). Indeed, in the latter case, the proposed architecture detects weak faults on average after 0.67 samples whilst MSFDIA and AE architectures take on average 3.40 and 8.30 samples to detect the same faults, respectively. The reported difference corresponds to a *faster detection for our proposal* of more than 5× and 12× than the MSFDIA and AE architectures, respectively. The reduced detection delay of the proposed architecture is mainly due to the *joint* exploitation of estimation and prediction blocks (cf. Sec. II-B), as they provide complementary (residual) information for the classifier.

TABLE III: Detection delay Analysis. Results refer to bias and drift faults and are in the format avg. (± std.) delayed samples obtained for a fault rate $F_R = 0.5$.

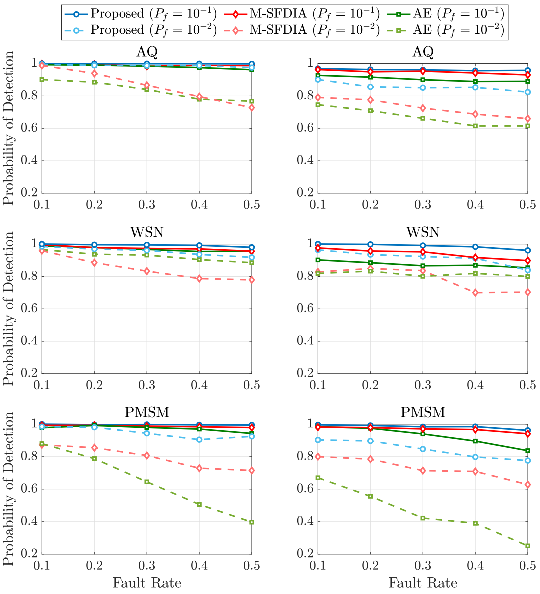
Data-set	Fault type	Proposed	M-SFDIA	AE
AQ	Bias	0.06 (± 0.30)	0.39 (± 1.11)	0.50 (± 1.33)
	Drift	1.77 (± 1.69)	2.33 (± 2.09)	4.04 (± 2.84)
WSN	Bias	0.28 (± 0.91)	0.84 (± 1.47)	0.31 (± 1.08)
	Drift	0.72 (± 1.12)	2.12 (± 2.14)	2.73 (± 2.12)
PMSM	Bias	0.10 (± 0.53)	1.24 (± 1.94)	3.61 (± 3.62)
	Drift	**0.67 (± 1.21)**	**3.40 (± 2.68)**	**8.30 (± 4.85)**

Fig. 8 shows the *identification performance* from the individual sensor perspective for the proposed architecture under different fault types. The probability of identification refers to the probability that SFDIA architecture correctly isolates the corresponding faulty sensor(s), where the averaged value is the average probability of identification over all unreliable sensors in each data-set. Apparently, different sensors undergo different performances, mostly depending on the level of spatio-temporal correlation (implicitly) providing the available redundant information within the system. The corresponding sensor-averaged identification performance (under the same fault rate) is depicted in Fig. 9. Here in Fig. 8 and 9, the proposed architecture performs even better over other methods since it manages to reduce fault propagation within the architecture itself and avoid functionality degradation using the controlling block. Replacing faulty sensors with their estimates or predictions by the controller provides the classifier with easier interpretative residual signals.

The *accommodation performance* in terms of root mean square error (RMSE) is shown in Fig. 10, where fault rates $F_R \in \{0.1, 0.5\}$ are considered. Herein the term *error* means the difference between sensor healthy values before adding the fault and the accommodated values provided by the SFDIA architecture (or the original values, in the case of an undetected/unidentified fault). First of all, it is apparent that the proposed architecture outperforms the M-SFDIA architecture by presenting more accurate replacements for faulty data. The reason is that the proposed architecture relies on a combined estimator/predictor pair for each sensor and a controller block to continuously improve the accommodation performance by modifying their inputs based on the decision vector obtained from the classifier in a closed-loop fashion. Conversely, the M-SFDIA architecture does not take advantage of these excessive data. Finally, the proposed architecture outperforms AE-based SFDIA on all the three available data-sets (except for PMSM-Noise), with the higher improvement (viz. RMSE reduction) in the case of WSN data-set.

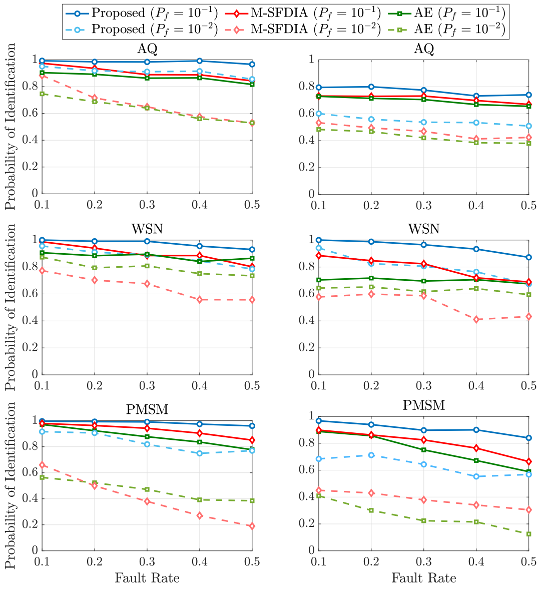
The rest of analysis specifically focuses on bias and drift faults as they well represent sudden (*hard*) faults and slowly appearing (*soft*) faults, respectively. The *impact of different fault rates on the detection and (averaged) isolation performance* is assessed in Figs. 11 and 12, respectively. In the above cases, two relevant false-alarm probability values are considered, namely $P_f = 10^{-1}$ and $P_f = 10^{-2}$. As expected, both detection and identification results reveal that higher fault rates have a negative impact on the architecture overall performance, as well as the considered baselines. Still, while the proposed architecture is capable to preserve its detection and isolation performance by incurring a milder detection/identification loss, both AE and M-SFDIA architectures exhibit a higher degradation with the fault rate. This outcome is mostly due to the estimators and predictors limiting the impact of fault propagation within the proposed architecture. For instance, referring to PMSM data-set, drift faults and $P_f = 10^{-2}$, a (harsh) fault-rate condition equal to 0.3 leads to a detection probability ≈ 0.4 (resp ≈ 0.7) for AE (resp. M-SFDIA). This corresponds to a 30% (resp. 10%) decrement w.r.t. a fault-rate scenario equal to 0.1. On the contrary, our architecture attains a detection probability ≈ 0.85 in the same harsh condition, with a corresponding degradation (w.r.t. fault-rate equal 0.1) equal to 0.05.

Figs. 13 and 14 compare *the performance trend of different architectures under versus the fault level b*. Clearly, detection and isolation performance for all architectures under strong faults are higher than the case of weaker faults. However, this in turn motivates the importance of developing techniques suited for weak faults. Results demonstrate a clear advantage of the proposed architecture over other architectures for different fault levels, with performance improvement being



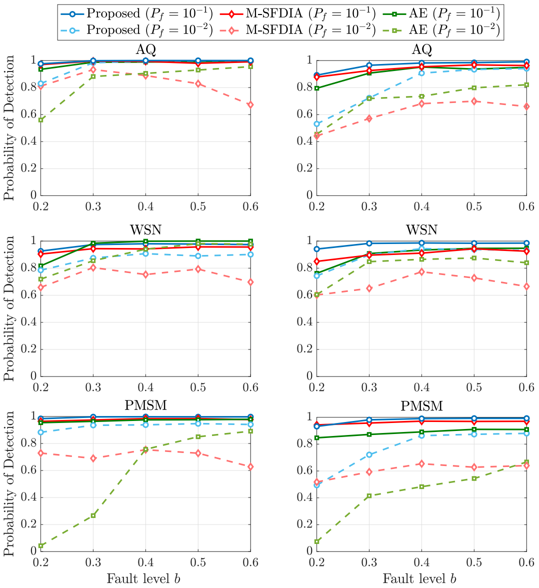
(a) Bias fault. (b) Drift fault.

Fig. 11: Impact of different fault rate on the detection accuracy.



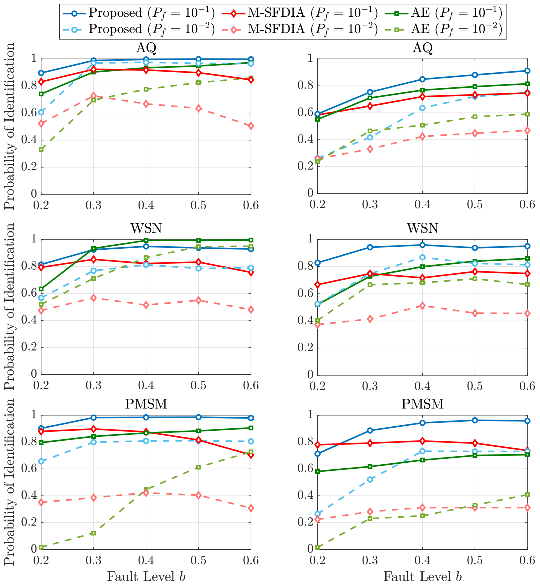
(a) Bias fault. (b) Drift fault.

Fig. 12: Impact of different fault rates on the averaged identification accuracy.



(a) Bias Fault (b) Drift Fault

Fig. 13: Impact of different fault level (b) on the detection accuracy.



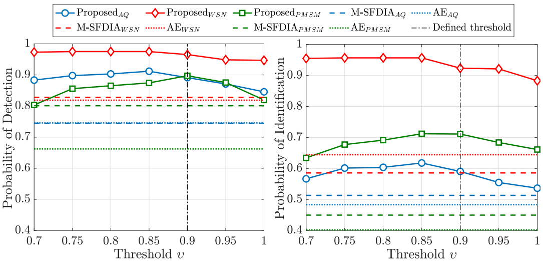
(a) Bias fault. (b) Drift fault.

Fig. 14: Impact of different fault level (b) on the averaged identification accuracy.

extremely evident under weak faults. For instance, referring to the case of bias faults with $b = 0.2$ on the PMSM dataset and assuming $P_f = 10^{-2}$, the proposed architecture achieves correct-identification probability of 0.9 while the AE architecture is below 0.1. The AE architecture mostly exploits change detection in the correlation structure of the signals and weak faults might have a negligible impact from this perspective. Conversely, the combined use of estimators, predictors and residual processing employed by the proposed architecture is able to detect & isolate these "low-observable" faults. Moreover, as the fault level increases, the proposed architecture is overtaking the M-SFDIA architecture since the proposed method mitigates propagation of strong faults within the architecture by means of the controller block.

To deepen the investigation of the controller block, a *sensitivity analysis* was also performed, focusing on detection and identification performance of the proposed architecture, by varying the threshold υ during the test phase. More specifically, Fig. 15 shows the detection and identification performance of the proposed method with respect to the threshold υ. To better apprehend the impact of the threshold υ, the detection and identification performance of the state-of-the-art counterparts were reported as a lower bound. Results highlight quite smooth performance trends on the three datasets with respect to the threshold υ. Interestingly, predefined threshold $\upsilon = 0.9$ based on the validation set is pretty near to the optimum value on the test set.

Finally, to have a finer-grained view of the three architectures for detection & isolation tasks, Fig. 16 reports their



(a) Detection Sensitivity (b) Identification Sensitivity

Fig. 15: Impact of threshold (υ) on the detection and identification accuracy ($P_f = 10^{-2}$). Threshold $\upsilon = 1$ associated to a zero-effect of the controller (i.e. off-circuit controller).

decision outcomes for a time-segment long 50 samples taken from the PMSM data-set under bias fault ($P_f = 10^{-2}$). Specifically, for each time index n, "$\circ$" symbol denotes the actual (true) faulty sensors, whereas "None" is used in the case of a healthy system. Then, for each architecture, the miss-detected faults (denoted with red "$*$" symbol) and the false-alarms (i.e. sensors erroneously declared as faulty by the architecture when the system is healthy, with blue "$\times$" symbol) are highlighted. Finally, when each SFDIA architecture declares a detection, the corresponding identified faults are reported with a green "$+$" symbol. The most eminent point in Fig. 16 is that, by resorting to the proposed architecture, *only one fault remained*

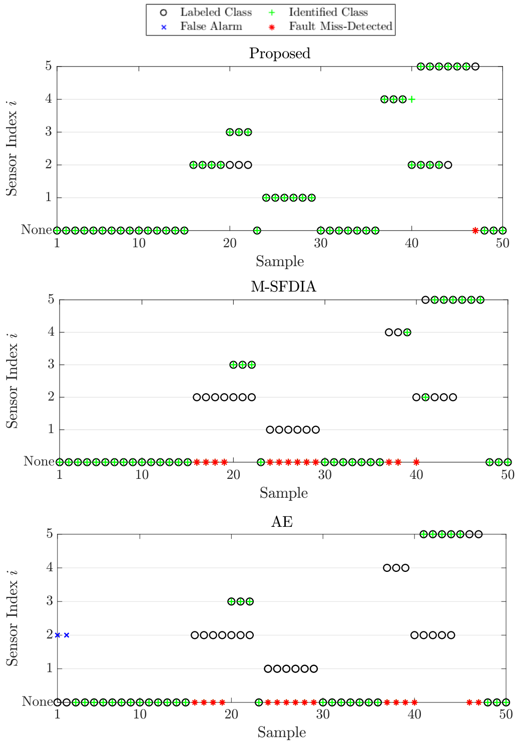


Fig. 16: Visualization of fault classification for all architectures on PMSM data-set.

undetected whereas M-SFDIA and AE architectures miss-detected 13 and 16 out of 24 faulty samples, respectively. As mentioned earlier, the proposed architecture attains better prompt detection & identification performance with respect to its counterparts. For instance, according to Fig. 16, the latter two architectures were only capable to identify only one faulty sensor for the given snapshot when simultaneous faults occurred, while the proposed architecture successfully identified most of them.

D. Complexity Analysis

As the final stage of the numerical comparison, the proposed approach is compared with the considered baselines in terms of the relevant computational complexity involved, by looking at both the (i) training and (ii) operational (testing) phases.

Regarding the *training phase*, the number of trainable parameters associated with each architecture is summarized in Tab. IV. Trainable parameters refer to weights and biases of each NN to be learned during the training phase (through stochastic gradient descent by resorting to the back-propagation technique) in the architecture. Clearly, the number of trainable parameters grows with the complexity of the (sensor) system to be accommodated, with the higher complexity associated with AQ data-set on all three architectures. Also, the info in the table highlights that the proposed architecture has a *comparable complexity with M-SFDIA while enjoying shorter training times than the considered AE*. Furthermore, thanks to the modularity granted by the proposed approach, different blocks of the considered architecture (e.g. estimators and predictors) could be trained in a parallel fashion on distributed (e.g. cloud) architectures.

Regarding the *testing phase*, the assumption of an equal number of hidden layers ($H_v = H_c = H_J$), time delays ($L_v = L_p = L_c = L_J$) and nodes per hidden layer ($N_v = N_c = N_J$) is made, as considered in [41], where index J refers to the joint value. Additionally, the impact of the activation functions is neglected (for simplicity). Accordingly, the computational complexity of the operational phase is analyzed in terms of the well-known big-$\mathcal{O}$ (Landau's) notation. First, it is worth recalling that the computational complexity of M-SFDIA approximately equals $\mathcal{O}(L_J N_U^2 N_J + L_J N_R N_U N_J + H_J N_U N_J^2)$ for one input sample [41]. Furthermore, the complexity cost of each predictor in the proposed architecture is approximately $\mathcal{O}(L_J N_J)$. Accordingly, the overall computational complexity of the proposed architecture approximately equals the M-SFDIA architecture. Indeed, the complexity is mainly dominated by the computational cost of the estimators and of the classifier, which is almost equal in both architectures [41]. Indeed, the impact of the residual and controller block operations is negligible in the overall cost.

TABLE IV: Number of NNs' trainable parameters.

Data-set	N_U	N_R	Proposed			M-SFDIA		AE	
			Est.	Pre.	Clf.	Est.	Clf.	AE	Denoising-AE
AQ	5	2	681	121	1985	681	1160	16945	16945
			In total = 5995			4565		33890	
WSN	4	0	351	121	1639	351	979	5470	5470
			In total = 3527			2383		10940	
PMSM	5	1	571	121	1985	571	1160	12250	12250
			In total = 5445			4015		24500	

For instance, for AQ data-set the computational complexity for estimators block is $\mathcal{O}(3510)$, for the classifier block[8] is $\mathcal{O}(1740)$ and for the predictors block is $\mathcal{O}(550)$. This results in a total computational complexity of $\mathcal{O}(5{\cdot}10^3)$ and $\mathcal{O}(6{\cdot}10^3)$ for M-SFDIA and the proposed architecture, respectively. Still, the proposed architecture attains substantially higher overall SFDIA performance at the expense of a manageably higher complexity (see Sec. V-C).

Conversely, for AE-based architecture (which is made of two similar AEs with a three-fold compression factor [12]), the computational complexity is $\mathcal{O}((452/81) \cdot (L_J \cdot (N_U + N_R))^2)$. Accordingly, in the peculiar case of AQ data-set, the computational complexity of the aforementioned architecture is approximately $\mathcal{O}((452/81) \cdot (10 \cdot (5+2))^2) \approx \mathcal{O}(3 \cdot 10^4)$ for one input sample. As a result, the complexity of the AE-based architecture appears to be considerably higher than that incurred by the proposed approach.

[8] The computational complexity for the classifier in M-SFDIA architecture is $\mathcal{O}(990)$.

VI. Conclusion

This article presented a four-layer architecture for SFDIA based on MLP NNs. Our contribution represents a stepping stone towards the development of (modular) DTs based on sensor systems/networks in IoT contexts. The (four) designed layers consist of estimation&prediction, residual, classification and controlling blocks. The classifier block at the heart of the architecture is in charge of detecting and identifying faulty sensors based on residual signals provided by estimators and predictors. Moreover, a controlling block is placed to track the classifier's decision output in order to boost overall system performance. This is accomplished by stopping fault propagation chain at the first layer by modifying estimators and predictors inputs with respect to the classifier's decision.

The proposed method was trained and tested on *three* real-world and publicly-available data-sets (i.e. [41], [42], [50]) for the sake of a complete and reproducible assessment. For the sake of generalization, four types of faults were considered in this study: bias, drift, noise and freeze. The proposed architecture yielded notably higher detection and isolation performance compared to the state-of-art M-SFDIA [41] and AE [12] architectures, for *all* four fault types. Moreover, the proposed architecture was shown to enjoy robustness against different fault rates while other architectures' performances were affected considerably.

Future works will focus on (i) the study of DTs for sensors operating under channel uncertainty, (ii) the design of SFDIA architectures which scale well with the number of sensors, (iii) the investigation of reinforcement-learning algorithms for optimized controller design and (iv) the application of explainable artificial-intelligence algorithms' in interpreting (and improving) the proposed SFDIA approach. Finally, more sophisticated NN approaches (e.g. convolutional NNs, RNNs) for each SFDIA module we will be also investigated with the intent of improving detection, identification and accommodation performance under specific circumstances while meeting the operational deployment constraints.

References

[1] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21 980–22 012, 2020.

[2] Z. Chang, S. Liu, X. Xiong, Z. Cai, and G. Tu, "A survey of recent advances in edge-computing-powered artificial intelligence of things," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[3] L. Chettri and R. Bera, "A comprehensive survey on internet of things (iot) toward 5g wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2020.

[4] S. Cui, F. Farha, H. Ning, Z. Zhou, F. Shi, and M. Daneshmand, "A survey on the bottleneck between applications exploding and user requirements in iot," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[5] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.

[6] J. Zhang, L. Li, G. Lin, D. Fang, Y. Tai, and J. Huang, "Cyber resilience in healthcare digital twin on lung cancer," *IEEE Access*, vol. 8, pp. 201 900–201 913, 2020.

[7] A. Francisco, N. Mohammadi, and J. E. Taylor, "Smart city digital twin–enabled energy management: Toward real-time urban building energy benchmarking," *Journal of Management in Engineering*, vol. 36, no. 2, p. 04019045, 2020.

[8] N. Mohammadi and J. E. Taylor, "Smart city digital twins," in *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, pp. 1–5.

[9] 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.

[10] 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.

[11] X. Luo, Y. Li, X. Wang, and X. Guan, "Interval observer-based detection and localization against false data injection attack in smart grids," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 657–671, 2021.

[12] 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.

[13] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava, "Sensor network data fault types," *ACM Trans. Sen. Netw.*, vol. 5, no. 3, Jun. 2009.

[14] T. Muhammed and R. A. Shaikh, "An analysis of fault detection strategies in wireless sensor networks," *Elsevier Journal of Network and Computer Applications*, vol. 78, pp. 267 – 287, 2017.

[15] Y. Jiang, S. Yin, J. Dong, and O. Kaynak, "A review on soft sensors for monitoring, control and optimization of industrial processes," *IEEE Sensors Journal*, 2020.

[16] M. M. Gharamaleki and S. Babaie, "A new distributed fault detection method for wireless sensor networks," *IEEE Systems Journal*, vol. 14, no. 4, pp. 4883–4890, 2020.

[17] E. Dubrova, "Hardware redundancy," in *Fault-Tolerant Design*. Springer, 2013, pp. 5–86.

[18] A. A. Amin and K. Mahmood-Ul-Hasan, "Advanced fault tolerant air-fuel ratio control of internal combustion gas engine for sensor and actuator faults," *IEEE Access*, vol. 7, pp. 17 634–17 643, 2019.

[19] S. Yin, B. Xiao, S. X. Ding, and D. Zhou, "A review on recent development of spacecraft attitude fault tolerant control system," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 5, pp. 3311–3320, 2016.

[20] 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.

[21] J. Loy-Benitez, Q. Li, K. Nam, and C. Yoo, "Sustainable subway indoor air quality monitoring and fault-tolerant ventilation control using a sparse autoencoder-driven sensor self-validation," *Elsevier Sustainable Cities and Society*, vol. 52, p. 101847, 2020.

[22] 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.

[23] M. Ruba, R. O. Nemes, S. M. Ciornei, and C. Martis, "Simple and robust current sensor fault detection and compensation method for 3-phase inverters," *IEEE Access*, vol. 8, pp. 34 820–34 832, 2020.

[24] 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.

[25] C. Sun and Y. Lin, "Adaptive output feedback compensation for a class of nonlinear systems with actuator and sensor failures," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–10, 2021.

[26] J. Zhang, S. Li, and Z. Xiang, "Adaptive fuzzy output feedback event-triggered control for a class of switched nonlinear systems with sensor failures," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 5336–5346, 2020.

[27] C. Lo, J. P. Lynch, and M. Liu, "Distributed reference-free fault detection method for autonomous wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 5, pp. 2009–2019, 2013.

[28] P. Liu, Y. Zhang, H. Wu, and T. Fu, "Optimization of Edge-PLC-Based fault diagnosis with random forest in Industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9664–9674, 2020.

[29] 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.

[30] Guo Su, D. Fang, Sun Jian, and Li Fengmei, "Sensor fault detection with online sparse least squares support vector machine," in *Proceedings of the 32nd Chinese Control Conference*, 2013, pp. 6220–6224.

[31] S. Mandal, B. Santhi, S. Sridhar, K. Vinolia, and P. Swaminathan, "Nuclear power plant thermocouple sensor-fault detection and classification using deep learning and generalized likelihood ratio test," *IEEE Transactions on Nuclear Science*, vol. 64, no. 6, pp. 1526–1534, 2017.

[32] 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.

[33] J. Gao, J. Wang, P. Zhong, and H. Wang, "On threshold-free error detection for industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2199–2209, 2018.

[34] K. Jeong, S. B. Choi, and H. Choi, "Sensor fault detection and isolation using a support vector machine for vehicle suspension systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3852–3863, 2020.

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

[36] M. Elnour, N. Meskin, and M. Al-Naemi, "Sensor data validation and fault diagnosis using auto-associative neural network for HVAC systems," *Journal of Building Engineering*, vol. 27, p. 100935, 2020.

[37] 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.

[38] 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 International Journal of Aerospace Engineering*, 2018.

[39] 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 Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2020.

[40] H. Zhang, Q. Zhang, J. Liu, and H. Guo, "Fault detection and repairing for intelligent connected vehicles based on dynamic Bayesian network model," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2431–2440, 2018.

[41] H. Darvishi, D. Ciuonzo, E. R. Eide, and P. Salvo Rossi, "Sensor-fault detection, isolation and accommodation for digital twins via modular data-driven architecture," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4827–4838, February 2021.

[42] S. De Vito, E. Massera, M. Piga, L. Martinotto, and G. Di Francia, "On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario," *Sensors and Actuators B: Chemical*, vol. 129, no. 2, pp. 750–757, 2008.

[43] 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.

[44] W. Kirchgässner, O. Wallscheid, and J. Böcker, "Deep residual convolutional and recurrent neural networks for temperature estimation in permanent magnet synchronous motors," in *IEEE International Electric Machines & Drives Conference (IEMDC)*, 2019, pp. 1439–1446.

[45] H. Darvishi, D. Ciuonzo, and P. Salvo Rossi, "Real-time sensor fault detection, isolation and accommodation for industrial Digital Twins," in *IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 2021.

[46] P. Tchakoua, R. Wamkeue, M. Ouhrouche, F. Slaoui-Hasnaoui, T. A. Tameghe, and G. Ekemb, "Wind Turbine Condition Monitoring: State-of-the-Art Review, New Trends, and Future Challenges," *MDPI Energies*, vol. 7, no. 4, pp. 1–36, April 2014.

[47] F. Rosenblatt, *The perceptron: a theory of statistical separability in cognitive systems (Project Para)*. Cornell Aeronautical Laboratory, 1958.

[48] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

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

[50] W. Kirchgässner, O. Wallscheid, and J. Böcker, "Empirical evaluation of exponentially weighted moving averages for simple linear thermal modeling of permanent magnet synchronous machines," in *IEEE 28th International Symposium on Industrial Electronics (ISIE)*, 2019, pp. 318–323.

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

Hossein Darvishi (GS'20) received the B.Sc. degree from the Kermanshah University of Technology, Iran, and the M.Sc. degree (*ranked first*) in telecommunications engineering from K.N. Toosi University of Technology, Iran, in 2016 and 2018, respectively. Since 2020, he has been pursuing the Ph.D. degree in Electronics and Telecommunications with the Department of Electronic Systems, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. He is currently a visiting researcher at the Signal Processing Laboratory (LTS4) in the Electrical Engineering Institute of the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. He is affiliated with the Nordic Industrial IoT Hub (HI2OT) and the SIGNIFY project (NTNU and SINTEF's research and connectivity program in sensor validation solutions for digital twins of safety-critical systems). He is a Reviewer for reputable journals including IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS, and IEEE SENSORS JOURNAL. His research interests include statistical signal processing, machine learning, Internet of Things and wireless sensor networks.

Domenico Ciuonzo (S'11-M'14-SM'16) is an Assistant Professor at University of Napoli Federico II. He holds a Ph.D. from the University of Campania "L. Vanvitelli", Italy. Since 2011, he has been holding several visiting researcher appointments. He is currently a Technical Editor for the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS and an Executive Editor for the IEEE COMMUNICATIONS LETTERS. He is the recipient of two Best Paper awards (IEEE ICCCS 2019 and Elsevier Computer Networks 2020), the 2019 Exceptional Service award from IEEE AESS, the 2020 Early-Career Technical Achievement award from IEEE SENSORS COUNCIL for sensor networks/systems and the 2021 Early-Career Award from IEEE AESS for contributions to decentralized inference and sensor fusion in networked sensor systems. His research interests include data fusion, statistical signal processing, wireless sensor networks, the Internet of Things and machine learning.

Pierluigi Salvo Rossi (SM'11) was born in Naples, Italy, in 1977. He received the Dr.Eng. degree (*summa cum laude*) in telecommunications engineering and the Ph.D. degree in computer engineering from the University of Naples "Federico II", Italy, in 2002 and 2005, respectively. He is currently a Full Professor and the Deputy Head with the Department of Electronic Systems, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. He is also a part-time Research Scientist with the Department of Gas Technology, SINTEF Energy Research, Norway.

Previously, he worked with the University of Naples "Federico II", Italy, with the Second University of Naples, Italy, with NTNU, Norway, and with Kongsberg Digital AS, Norway. He held visiting appointments with Drexel University, USA, with Lund University, Sweden, with NTNU, Norway, and with Uppsala University, Sweden.

His research interests fall within the areas of communication theory, data fusion, machine learning, and signal processing. Prof. Salvo Rossi was awarded as an Exemplary Senior Editor of the IEEE COMMUNICATIONS LETTERS in 2018. He is (or has been) in the Editorial Board of the IEEE SENSORS JOURNAL, the IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS, the IEEE COMMUNICATIONS LETTERS and the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.

Paper 6

IoT-based Monitoring in Carbon Capture and Storage Systems

A. Chawla, Y. Arellano, M. V. Johansson, H. Darvishi, K. Shaneen, M. Vitali, F. Finotti, P. S. Rossi

IEEE Internet of Things Magazine

This paper is not included due to copyright
available in IEEE Internet of Things Magazine (IoTM) 2022 ;Volum 5.(4) s. 106-111
https://doi.org/10.1109/IOTM.001.2200175 and
https://hdl.handle.net/11250/3045952

Paper 7

Deep Recurrent Graph Convolutional Architecture for Sensor Fault Detection, Isolation and Accommodation in Digital Twins

H. Darvishi, D. Ciuonzo and P. S. Rossi

Submitted to *IEEE Sensors Journal*

ISBN 978-82-326-7007-9 (printed ver.)
ISBN 978-82-326-7006-2 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU
Norwegian University of
Science and Technology