Computers in Industry 151 (2023) 103958

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.sciencedirect.com/journal/computers-in-industry

Updating digital twins: Methodology for data accuracy quality control using machine learning techniques

Fabio Rodríguez [a], William D. Chicaiza [b], Adolfo Sánchez [c], Juan M. Escaño [b,*]

[a] Departamento de Matemática Aplicada II, Universidad de Sevilla, Camino de los Descubrimientos s/n. 41092, Sevilla, Spain
[b] Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Camino de los Descubrimientos s/n., 41092 Sevilla, Spain
[c] Department of Mechanical, Biomedical, and Manufacturing Engineering, Munster Technological University, Bishopstown, Cork, Ireland

ARTICLE INFO

Keywords:
Adaptive digital twin
Neural network (NN)
Fuzzy inference system
Fault detection
Adaptive decision making

ABSTRACT

The Digital Twin (DT) constitutes an integration between cyber and physical spaces and has recently become a popular concept in smart manufacturing and Industry 4.0. The related literature provides a DT characterisation and identifies the problem of updating DT models throughout the product life cycle as one of the knowledge gaps. The DT must update its performance by analysing the variable data in real time of the physical asset, whose behaviour is constantly changing over time. The automatic update process involves a data quality problem, i.e., ensuring that the captured values do not come from measurement or provoked errors. In this work, a novel methodology has been proposed to achieve data quality in the interconnection between digital and physical spaces. The methodology is applied to a real case study using the DT of a real solar cooling plant, acting as a learning decision support system that ensures the quality of the data during the update of the DT. The implementation of the methodology integrates a neurofuzzy system to detect failures and a recurrent neural network to predict the size of the errors. Experiments were carried out using historical plant data that showed great results in terms of detection and prediction accuracy, demonstrating the feasibility of applying the methodology in terms of computation time.

1. Introduction

Data-driven smart manufacturing is one of the pillars of Industry 4.0, as a strategic part of the quest to maximise production by optimising its processes. Data acquisition is an essential element, as it records the behaviour and evolution of industrial processes throughout their life cycle. The evolution and innovation of the Industrial Internet of Things, smart sensors, different industrial communication protocols, and other technologies make it possible to acquire data in real time to improve decision making.

The Digital Twin (DT) concept has recently become popular in the Industry 4.0 sector. Grieves presented the conceptualisation of DT[1] as part of a study on Product life cycle Management (PLM), (Grieves and Vickers, 2017). Grieves presents all the elements that make up the DT: the physical entity, the virtual entity, and the interconnection of the data flow, which inextricably links and connects the two entities. Since the DT neologism, several articles have introduced different concepts; however, most of them share the same elements of the initial conceptualisation (Glaessgen and Stargel, 2012; Boschert and Rosen, 2016; Grieves and Vickers, 2017; Haag and Anderl, 2018; Saracco, 2019).

DT is based on high-fidelity models capable of accepting and incorporating modifications to their behaviour during their life cycle to realise the concept of context-awareness (Hribernik et al., 2021). The development of DT facilitates the analysis of behaviour in different physical situations, operations, and control methods, since it can be performed virtually, avoiding experimentation. Furthermore, the use of DT optimises production efficiency and energy savings in industrial processes (Pileggi et al., 2019; Onile et al., 2021). During the life cycle of an industrial asset, the physical entity is affected by continuous operation, leading to a decrease in performance. Therefore, an important requirement for the DT is the ability to update its performance at a certain frequency. The study is carried out with the aim of developing a sensor-based DT composed of computational models built on historical data from the physical counterpart. During the asset's life cycle, the

* Corresponding author.
E-mail addresses: frodriguex@us.es (F. Rodríguez), wchicaiza@us.es (W.D. Chicaiza), adolfo.sanchezdelpozofernandez@mtu.ie (A. Sánchez), jescano@us.es (J.M. Escaño).

[1] NASA formalised the Digital Twin concept as an integrated multiphysics, multiscale, probabilistic simulation of an as-built system that uses the best available physical models, sensor updates, fleet history, and real-time sensor data (Glaessgen and Stargel, 2012).

https://doi.org/10.1016/j.compind.2023.103958
Received 19 April 2022; Received in revised form 12 May 2023; Accepted 24 May 2023
Available online 15 June 2023

update is done using the more recent collected data of the physical part to readjust the models of the DT. When updating tasks are triggered, the following questions arise:

- What happens when the data that feed the DT are corrupted or contain incorrect values?
- When is it considered that a change in data should trigger the update of the digital twin?

A failure or simply a drift in a sensor can trigger the twin's models learning mechanism in an unintended way. As a result, retraining a DT requires a priori analysis of the real-time output data of the system. Therefore, an important first step is to detect the incorrect data to assess the decision to start the learning routine.

Sensor errors in industrial processes can be caused by noise, malfunction, or, occasionally, incorrect instrument handling. The application of fault detection techniques, such as predictive, diagnostic, and monitoring systems, maximises operational availability and reduces operational risks. These techniques are often enriched with real-world data to diagnose and detect failures with high efficiency. These systems integrate smart sensors and models based on artificial intelligence that are integrated into the DT of physical entities.

In this work, a case study of a solar cooling plant (SCP) has been chosen, for which there is already a DT, (Chicaiza et al., 2021). A system to detect faults in sensor output has been proposed as a tool to support the decision to retrain the DT. The proposed system is based on models that combine fuzzy logic techniques and NNs. The models classify whether or not a sensor fault occurs at a specific time instant and then predict the correct value for that sensor applying regression based on recurrent neural networks (RNNs) with long–short-term memory (LSTM) layers. Model-based fault detection systems are well known and widely used (Ding, 2008). Models are trained using a data collection built from historical and simulated SCP data. Following these ideas, a system capable of establishing retraining rules for the DT and capable of detecting erroneous behaviour in the SCP is obtained. The developed methodology is a key piece in the periodic updating process of the DT since it filters poor-quality data to be fed to the models composing the DT. Initially, it is assumed that only one sensor can fail in a same time unit, however, later in the article experiments show that is possible to detect failures in more than one sensor in a single time unit.

The rest of the article is organised as follows. Section 2 summarises some techniques used in fault detection based on operational data and highlights the contribution of the present work. Section 3 introduces the general workflow of the proposed data quality methodology and its contribution to the DT update. Section 4 describes the case studied in the scope of this work, that is, the plan from which the DT is obtained. A description of the training and validation data sets is given in Section 5. In Sections 6 and 7 the neurofuzzy detection and the offset prediction networks are depicted. Section 8 analyses the results of a complete implementation of the proposed methodology applied to the case studied, and a comparison with a similar approach is made. The article ends in Section 9, with the conclusions and future research work.

2. Related work

Today, failure detection during the life cycle of industrial components is an essential task to early apply the required maintenance activities to avoid possible breakdowns. Given the increase in the use of data to optimise industrial processes, there is extensive literature on the use of ANNs in the development of fault detection and classification applications, due to their ease of implementation and efficient self-learning capabilities.

In Tang et al. (2020), the authors process large mechanical data using a continuous wavelet transform and a deep convolutional neural network (CNN) model for fault classification and diagnosis in a hydraulic axial piston pump and use T-SNE to reduce characteristics and visualise classification. Different types of faults were precisely identified and high classification accuracy was achieved for the piston pump. However, the authors highlight the difficulty of the CNN model in handling time–frequency distributions. In Kang et al. (2021), a novel machine learning-based approach is used to automate the prediction of the remaining useful life of the equipment on continuous production lines. The proposed model applies normalisation and PCA techniques for data pre-processing; then, grid search for parameter optimisation and multilayer NNs are used for failure prediction. The results show that the performance of these techniques is affected by external factors such as dimension, noise level, and environmental variation. The work done in Heo and Lee (2018) uses deep neural networks (DNNs) for fault detection and classification in the Tennessee Eastman process (chemical process). The authors claim that different architectures of NN need to be tested to see if they fit better for fault detection and classification problems. In Jamil et al. (2015) the authors detect and classify the line to ground fault in a power transmission line, using voltage and current measurements as input to the neural scheme (feedforward NN with backpropagation learning method). A systematic study on the application of ANN and hybridised ANN models for the detection and diagnosis of photovoltaic faults was conducted in Li et al. (2021). Similarly, the work of Wang et al. (2020) employs CNNs, DNNs based on automatic encoders, and standalone SoftMax classifiers for the detection and classification of open-circuit high-voltage DC faults of modular multilevel converters (MMC-HVDCs). A survey on fault monitoring, detection, and classification in smart grid systems is presented in Labrador Rivas and Abrão (2020). In this research, a great number of machine learning techniques are presented to solve problems of this kind, such as models, ANN, DNN, support vector machine, and others (Labrador Rivas and Abrão, 2020); all yield great results. Furthermore, Ruiz-Moreno et al. (2022) combines multilayer perceptrons NNs with a defocusing stage to detect and distinguish mirror and flow rate faults in the Fresnel-solar field located at the Seville Engineering School, the same plant used in the experiments of the present work. The NN is used to detect faults and classify them into two main locations, and the defocusing stage is used to increase the accuracy of the network. This research is strongly related to the present work because both are dedicated to classifying different types of fault in the same solar plant.

The Adaptive Neuro-Fuzzy Inference System (ANFIS) proposed by Jang (1993) uses the learning capacity of ANN and FL to represent knowledge in an interpretable form. ANFIS is a widely used hybrid model in classification, prediction, and modelling problems, (Zhang and Morris, 1996; Camacho et al., 2019a; Escaño et al., 2020; Shah and Wang, 2021). ANFIS has the capability to construct an input–output mapping based on human knowledge in the form of fuzzy if–then rules, which can handle imprecise input data (uncertainty). Detection and classification of faults such as partial shading, series resistance, and shorted shunt diodes in photovoltaic arrays (PVA) are developed in Belaout et al. (2018) by fuzzy logic classifiers using experimental data and with an ANFIS to improve classification performance. The proposed technique was capable of correctly discriminating between five types of fault that occur in the PVA and demonstrated superiority over an ANN classifier.

The main contribution of this work is a novel methodology to address the data quality problem related to the DT update during its lifetime. The proposed methodology combines the techniques with better performance in the literature to feed high-quality data to the models composing the DT. The methodology is applied effectively to the particular case study of a DT of an SCP. Furthermore, the methodology can generally be applied to the data quality problem of similar sensor-based DT in other industrial scenarios.

In the specific SCP case study, the methodology integrates an ANFIS to detect and classify failures in sensor signals, with a deep RNN system to predict the offset error per sensor. The advantage of using the ANFIS once trained is to be able to observe the internal structure and to interpret the logical inference. The ANFIS technique was also

chosen because of its computationally fast performance in classification problems. On the other hand, the use of RNNs was appropriate to make offset predictions based on sensor signals over time. Finally, a comparison was made between the proposed methodology and the work in Ruiz-Moreno et al. (2022) in terms of the scope of both techniques and the accuracy of the fault detection results.

3. DT updating: Data quality methodology

The interconnection between the DT and the physical asset is the basis for the long-term use of the DT technology. The physical part retrieves data related to variables and processes using sensors, allowing the DT to update its behaviour throughout the asset's life cycle. The DT update is performed periodically by retraining the DT model with recent data from the physical counterpart. In this process, failures in the collected data lead to a gap between the digital and physical parts. As a result, the data quality problem is essential in the DT update process.

To address the data quality problem, an evaluation of different dimensions is needed, such as accuracy, confidence, completeness, data volume, and data reception time, as proposed in Klein and Lehner (2009). To be able to assess each dimension, it is necessary to combine a fault detection method with a mechanism that predicts what the data should be like. The methodology proposed in this work ensures data quality during the data exchange process based on historical data recovered by the sensor system.

Initially, it is assumed that only one sensor can fail at a specific moment in time. This assumption is made since there is a low probability of a malfunction in more than one sensor in the same unit of time. In real scenarios, sensors that retrieve incorrect readings are quickly fixed or replaced. However, in Section 8 experimental results show the ability of the methodology to be adapted to detect failures in multiple sensors at the same time.

The methodology consists of two parallel flows: the Fault Detection System (FDS), which detects sensor failures, and the Offset Error Prediction (OEP), for measuring the severity of the detected failures. The algorithm 1 represents a pseudocode that describes the methodology workflow, where $s_1, s_2, \dots s_N$ are the measured values retrieved by the physical asset (sensor output), the variables $f_1, f_2, \dots, f_N$ are binary variables outputted by the FDS and $\hat{s}_1, \hat{s}_2, \dots \hat{s}_N$ are the predictions of the expected value of each sensor used to calculate the offset errors, where N is the number of sensors considered in the DT. The offset error of the ith sensor is calculated as the absolute difference between the sensor output and the prediction of the expected output, $|s_i - \hat{s}_i|$. As the FDS and OEP flows are independent, the execution efficiency of the methodology can be improved by making the detection and prediction flows run as parallel parts in the algorithm.

In this work, machine learning techniques were used to implement the methodology proposed for a specific case study: the DT of an SCP. Fig. 1 shows the specific architecture used in this work. A detector based on a neurofuzzy classifier with PCA projections has been developed for the FDS flow, while an RNN was trained for each sensor in the OEP flow. Therefore, discrepancies between measured signals and predicted signals can be compared, allowing the DT to keep only high-quality data for its posterior update.

The methodology can be implemented to contribute to the DT update process during its life cycle. The DT update process consists of a simple loop procedure that starts after a previously defined time interval. The values of every sensor reading are verified for faults during the time interval, storing fault-free readings to update the DT models at the beginning of each loop.

The proposed methodology is generic for working with similar DT systems that receive data from their physical counterpart through sensors. The methodology can be adapted to other systems by adding new sensors to the workflow. In addition, other data science techniques can be used in the fault detection and offset prediction process. In the particular case of the SCP, a detailed explanation of the sensors is given in the next Section.

Algorithm 1: Algorithm for the data quality methodology workflow. The input is the measured values of N sensors and the output is the fault detection list and the sensor offset error list. The FDS function detects sensors failures, while the OEP function predicts expected sensor values.

Data: $N, s_1, s_2, \dots, s_N$
Result: $\mathcal{F}$, $\mathcal{O}$
$thread_1$: $FDS(s_1, s_2, \dots, s_N)$;
$thread_2$: $OEP(s_1, s_2, \dots, s_N)$;
$f_1, f_2, \dots, f_N = await(thread_1)$;
$\hat{s}_1, \hat{s}_2, \dots \hat{s}_N = await(thread_2)$;
$\mathcal{F} = [f_1, f_2, \dots, f_N]$;
$\mathcal{O} = new\ List()$;
$i = 0$;
while $i < N$ **do**
  if $f_i = 1$ **then**
    $\mathcal{O}.add(|\hat{s}_i - s_i|)$;
  end
  else
    $\mathcal{O}.add(0)$;
  end
  $i = i + 1$;
end
return $\mathcal{F}, O$;

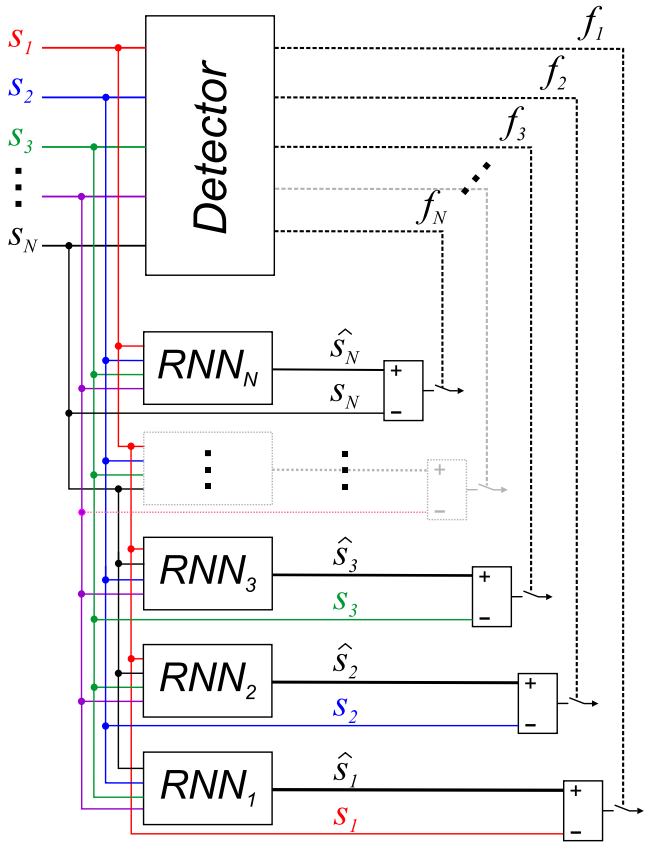


Fig. 1. Schematic diagram of the system architecture.

4. Case study: Cooling process through a solar plant

The plant under study is located on the roof of the main building of the School of Engineering of the University of Seville. The plant consists of three subsystems: a solar field, an absorption chiller machine, and

Fig. 2. Fresnel-type solar field.

a PCM storage tank. The Fresnel-type solar field consists of several mirrors that concentrate solar radiation onto a vacuum receiver tube. A heat transfer fluid (pressure water) inlet at one end of the receiver tube flows through it, increasing its temperature, and obtaining water at a higher temperature at the outlet.

The heat transfer fluid transfers the thermal energy to the absorption chiller machine, which operates on a double-effect thermodynamic absorption cycle. The hot side uses the water supplied by the solar field as a high-temperature source. The cold side chills a stream of water through different concentrations of Lithium-Bromide solution and evaporation/condensation heats. The chilled water on the cold side is used in the building's air-conditioning system, whereas the cooled water on the hot side is fed back to the Fresnel solar collector. More information on the plant can be found in Bermejo et al. (2010). In this work, only one part of the plant has been chosen: the Fresnel solar field.

4.1. Description of the Fresnel solar field

The main source of energy supply to increase the temperature of the heat transfer fluid in the SCP is the Fresnel solar field subsystem (see Fig. 2). It consists of a series of linear Fresnel collectors (LFCs) installed on the roof of the building with an East–West orientation ($Latitude = 37.4108972°$, $Longitude = -6.0006621°$). The solar field concentrates sun irradiation onto a 64-m receiver tube where the pressurised water circulates. In addition, it has a secondary reflector that optimises optical efficiency and improves the performance of the installation. A further description can be found in Robledo et al. (2011).

The field model utilised by LFC employs equations akin to those employed for a parabolic solar field, only differing in the use of a distributed parameter model (DPM) to calculate the thermodynamic properties of the fluid, the geometric efficiency, and the shading factor. The DPM considers the spatial distribution of the SCP, providing a better representation of the fluid heat transfer model (Camacho et al., 2019a,b). The DPM was used to simulate the solar field and observe the behaviour of the variation of the fluid outlet temperature (T_f). The physical equations of the dynamic model describe the energy balance as follows:

$$\begin{aligned} \rho_m C_m A_m \frac{\partial T_m}{\partial t} &= I K_{opt} n_o G - H_l (T_m - T_a) \dots \\ \dots &- l_p H_t (T_m - T_f) \\ l_p H_t (T_m - T_f) &= \rho_f C_f A_f \frac{\partial T_f}{\partial t} + \rho_f C_f q \frac{\partial T_f}{\partial l} \end{aligned} \tag{1}$$

Where m refers to the metal, f is the fluid, C and ρ are the specific heat capacity and density, respectively; A is the area of the cross section, I is the direct normal irradiance (DNI), K_{opt} is the optical efficiency, n_o is the geometric efficiency, G is the aperture, H_l is the global thermal loss coefficient, l_p is the perimeter of the absorption tube, and H_T is the coefficient of heat transmission (Gallego et al., 2020; Machado et al., 2022).

The DT of this solar field has been developed by using the dynamic model based on fuzzy inference systems (FIS) designed in Chicaiza et al. (2021). To train the FIS, real and artificial data generated from the differential equation model (1) were used, solved with the Euler forward integration method with the step 0.25 s. The integration time has been chosen to be small enough to avoid numerical instabilities (Gallego et al., 2020).

The resulting DT contains rules that capture the non-linear behaviour of the plant. One advantage of this technique is that more rules can be added once trained. New rules can be added by human experts and may include constraints, association between parameters, and behavioural predictions. The DT simulates the behaviour of the outlet temperature of the heat transfer fluid flowing in the absorption machine, which cools water for use in the air conditioning system of the University of Seville.

5. Preparation of operating data set

In the training process of the methodology models, an extensive collection of data corresponding to the historical output values of the solar plant sensors was used. The data set contains values from direct solar radiation (I), ambient temperature (T_{amb}), water flow rate (Q), inlet temperature (T_{in}), and outlet temperature (T_{out}) sensors. In addition to the historical data sensors, new data was generated using the model in Section 4.1.[2] The model (1) quite closely simulates the temperature dynamics of the Linear Fresnel Collector (LFC) system.

The resulting data set represents the fault-free sensor readings per minute for 14 summer days during 11 to 18 h, 6798 samples in total. The data set is divided into 12 days (5839 samples) for training and 2 days (959 samples) for validation. Each day contains data related to five variables: Inlet temperature, outlet temperature, ambient temperature, effective irradiance, and flow.

The DPM has been quite useful in exploring multiple scenarios under different conditions during the LFC solar field workflow. As a result, the detection and prediction algorithms are general enough to cover all scenarios. Fig. 3 shows the behaviour of the Fresnel-type solar field variables on one of the days.

The fault-free data were subjected to an outlier purification preprocess, where a smoothing technique based on Gaussian filters was performed for each variable in time. From the absolute error between the smoothed and real data, the mean m_i and the standard deviation std_i were calculated for the ith sensor, and the threshold value $t_i = m_i + 2std_i$. Finally, the values in the ith sensor with absolute error greater than t_i with respect to the smoothed data were considered outliers and were removed. Later, a second pre-processing step was performed to address the data completeness problem. The completeness problem arises when there is missing data in the sensors or when the outlier detection process deletes entries in the sequence of reading values of the sensors. In this case, a linear interpolation method was used to fill in the missing entries on the basis of its neighbouring values in the sequence.

For the fault detection task, more data with faults in the variables were included. Faults were simulated by adding positive and negative values to the variables in the operational data. This added value is what we consider an offset. In Ruiz-Moreno et al. (2022), the authors create offsets in the SCP flow sensor by adding and subtracting values in the interval $[0.5, 5]$ m^3/h. In this work, flow faults are created by adding and subtracting the offset of 2.5 m^3/h, which is approximately in the

[2] The entire set of data can found in https://github.com/fabio-rodriguez/SCP-dataset.

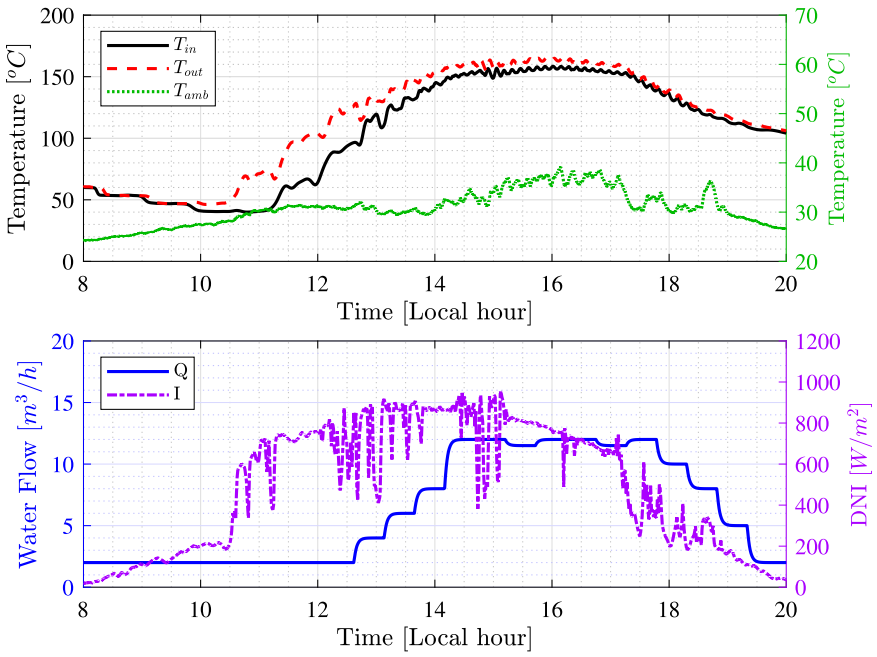


Fig. 3. Outlet fluid temperature performance.

Table 1
Induced failures in the different variables of the solar field.

Variables	Individual faults added Unit	$\mathbf{F}_{pos}$	$\mathbf{F}_{neg}$
Q	[m³/h]	+2.5	−2.5
I	[W/m²]	+150	−150
T_{in}	[°C]	+6	−6
T_{out}	[°C]	+6	−6

middle of the interval, and coincides with the 25% of the mean flow value throughout the data. Following this idea, the offsets for the rest of the sensors were fixed in 25% of the mean value of the respective sensor in the data set. Fuzzy detectors have to classify the data into three different classes: normal operation, false positive, and false negative in the aforementioned sensors, as shown in Table 1.

Faults induced by measurements of solar irradiation and water flow are shown in Fig. 4. Faults have been added in such a way that no more than one fault can occur at the same instant of time. It is intended to build a 1-fault-tolerant system that, having sensor readings at a given instant of time, can detect the faulty sensor if there is one.

6. Fuzzy detector

A fault detection method is developed using a neurofuzzy classifier with PCA projections. The classifier is based on an adaptive neurofuzzy inference system (ANFIS) to obtain a digital model capable of detecting faults in the variables measured by the target sensors of the Fresnel solar field. From the measured values, the detector (Fig. 5) must detect faults in the variable(s), when present, by assigning the binary variables $F_q, F_{T_{out}}, F_{T_{in}}, F_{T_{ext}}, F_{I_{ef}}$ with a value 1 (CORRECT) when there is a fault or 0 (INCORRECT) when there is no fault.

6.1. Principal components analysis

Some preliminary work has been done on the data set (including offsets) to train the neurofuzzy classifiers. First, a normalisation process is added to the learning set using Min–Max feature scaling so that all values fall in the range $[0, 1]$. The measurements of each sensor have a different scale, which can affect the learning process due to

Table 2
Set of variables for each fuzzy classifier: (Th) represents the thermal leap, that is, the difference between the outlet and inlet temperature, and (H_t) represents the coefficient of metal–fluid heat transmission.

Variables	Fuzzy classifiers Group
Solar radiation Flow	$\mathbf{D}_I = [T_{amb}, I, Th]$
Flow	$\mathbf{D}_Q = [T_{out}, Th^1, Q]$
Inlet temp	$\mathbf{D}_{T_{in}} = [Q, Th, T_{in}]$
Outlet temp	$\mathbf{D}_{T_{out}} = [H_t^2, Th, T_{out}]$

inconsistencies. It is solved by the normalisation process, thus avoiding the different nature and magnitude of the variables. Later, the fault detection data set was divided into three sets.

- (**class**$_{Norm}$): data without failures
- (**class**$_{F_{pos}}$): data with positive offset failure
- (**class**$_{F_{neg}}$): data with negative offset failure

The design of the fuzzy classifier for each variable uses different groups of variables. A Principal Component Analysis (PCA) is performed on each group of variables to obtain a covariance matrix $\mathbf{PCA}_v^c$ that is used to obtain the projection of each class in the space of the first principal component. The groups formed for each fuzzy classifier are shown in Table 2.

The PCA algorithm is applied offline to the entire training set ($\mathbf{S}_{Trn}$), obtaining the total covariance matrix $\mathbf{PCA}_v^{Total}$. Additionally, the PCA is calculated for each class composing the training set.

$$\mathbf{S}_{Trn} = \begin{bmatrix} \mathbf{D}_v^{Norm} \\ \mathbf{D}_v^{F_{pos}} \\ \mathbf{D}_v^{F_{neg}} \end{bmatrix} \tag{2}$$

The PCA showed that, for each $\mathbf{PCA}_v^c$, the first principal component contained more than 99% of the variability. Therefore, only this component is used to represent the data in one dimension without an appreciable loss of accuracy.

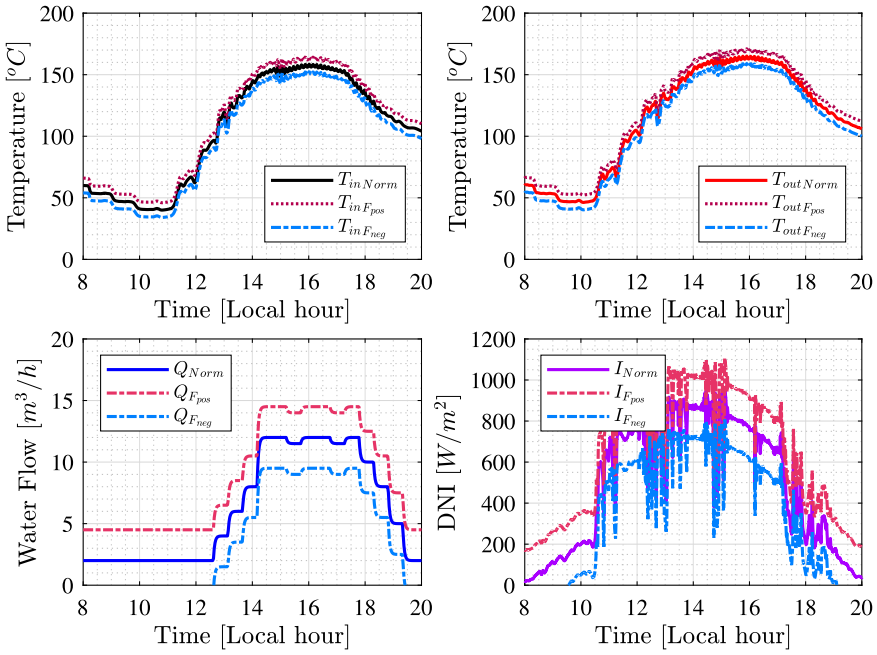


Fig. 4. Faults in the sensor measurement for a summer day.

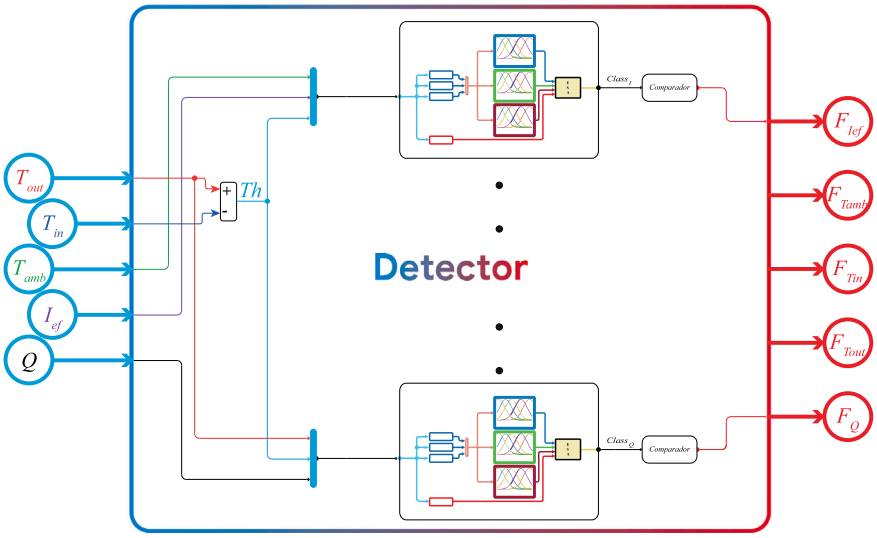


Fig. 5. Failures detector schematic. Sensors measurements feed the neuro-fuzzy classifiers. The neuro-fuzzy classifier's outputs are the detection of the failures where the subindex indicates the variable being detected.

6.2. Obtaining projected prototypes on the main components

After PCA calculation, the matrices $\mathbf{PCA}_{\upsilon}^{c} \rightarrow \left\{\mathbf{P}_{\upsilon}^{Norm}, \mathbf{PCA}_{\upsilon}^{Fpos}, \mathbf{PCA}_{\upsilon}^{Fneg}\right\}$ and $\mathbf{PCA}_{\upsilon}^{Total}$ referred to as *load matrices* were obtained. In the matrices, the classes $c \rightarrow \left\{Norm,\ Fpos,\ Fneg\right\}$ indicate: normal operation, false positive, false negative; and the subindex υ is the sensor. These matrix are used to obtain the projection of the data, since they contain the coefficients of the principal components of each variable.

The prototype **P** for a class is given by its projection on a single axis in space, that is, each class $\mathbf{D}_{\upsilon}^{c}$ is projected onto the first principal component of the different classes. This projection is the product between the class $\mathbf{D}_{\upsilon}^{c} \rightarrow \left\{\mathbf{D}_{\upsilon}^{Norm}, \mathbf{D}_{\upsilon}^{Fpos}, \mathbf{D}_{\upsilon}^{Fneg}\right\}$ and the first principal component of the corresponding matrix $\mathbf{PCA}_{\upsilon}^{c}$.

$$ {}^{Norm}\mathbf{P}_{\upsilon}^{c} = \mathbf{D}_{\upsilon}^{Norm} \times \mathbf{PCA}_{\upsilon}^{c} \tag{3a} $$

$$ {}^{F_{pos}}\mathbf{P}_{\upsilon}^{c} = \mathbf{D}_{\upsilon}^{F_{pos}} \times \mathbf{PCA}_{\upsilon}^{c} \tag{3b} $$

$$ {}^{F_{neg}}\mathbf{P}_{\upsilon}^{c} = \mathbf{D}_{\upsilon}^{F_{neg}} \times \mathbf{PCA}_{\upsilon}^{c} \tag{3c} $$

where $\mathbf{P}_{\upsilon}^{c} \rightarrow \left\{\mathbf{P}_{\upsilon}^{Norm}, \mathbf{P}_{\upsilon}^{Fpos}, \mathbf{P}_{\upsilon}^{Fneg}\right\}$ is a set of matrices containing the different projections obtained from the variables that constitute each class of each data group $\mathbf{D}_{\upsilon}^{c}$. Furthermore, a general prototype ${}^{g}\mathbf{P}_{\upsilon}^{c}$ is obtained for each class by the product with the matrix $\mathbf{PCA}_{\upsilon}^{Total}$.

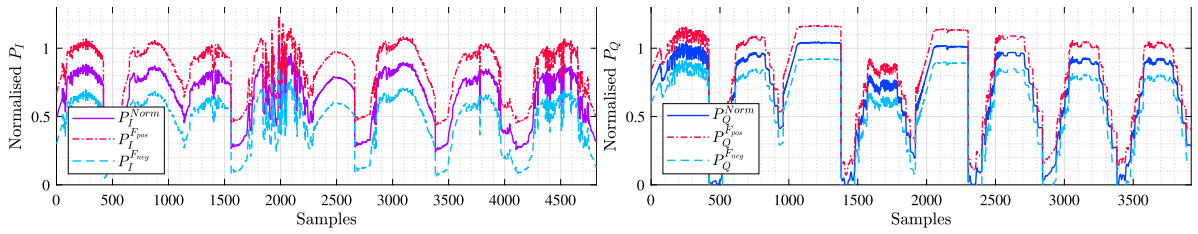
$$ {}^{g}\mathbf{P}_{\upsilon}^{c} = \mathbf{D}_{\upsilon}^{c} \times \mathbf{PCA}_{\upsilon}^{Total} \tag{4} $$

Finally, the training set used to detect sensor faults consists of the prototypes of each of the classes and their general prototype for each ANFIS that will form the neural classifier.

$$ \mathbf{Trn}_{\upsilon}^{Norm} = \left[{}^{Norm}\mathbf{P}_{\upsilon}^{Norm}\ \ {}^{Norm}\mathbf{P}_{\upsilon}^{F_{pos}}\ \ {}^{Norm}\mathbf{P}_{\upsilon}^{F_{neg}}\ \ {}^{g}\mathbf{P}_{\upsilon}^{Norm}\right], \tag{5a} $$

$$ \mathbf{Trn}_{\upsilon}^{F_{pos}} = \left[{}^{F_{pos}}\mathbf{P}_{\upsilon}^{Norm}\ \ {}^{F_{pos}}\mathbf{P}_{\upsilon}^{F_{pos}}\ \ {}^{F_{pos}}\mathbf{P}_{\upsilon}^{F_{neg}}\ \ {}^{g}\mathbf{P}_{\upsilon}^{F_{pos}}\right], \tag{5b} $$

$$ \mathbf{Trn}_{\upsilon}^{F_{neg}} = \left[{}^{F_{neg}}\mathbf{P}_{\upsilon}^{Norm}\ \ {}^{F_{neg}}\mathbf{P}_{\upsilon}^{F_{pos}}\ \ {}^{F_{neg}}\mathbf{P}_{\upsilon}^{F_{neg}}\ \ {}^{g}\mathbf{P}_{\upsilon}^{F_{neg}}\right], \tag{5c} $$



(a) Training set **I** Prototypes (b) Training set **Q** Prototypes

Fig. 6. Prototypes of the training set for the irradiance and water flow variable.

Table 3
ANFIS architecture parameters.

Description	ANFIS			
MF type:	*Gaussian*			
Optimisation method:	*hybrid*			
Output MF type:	*linear*			
$\mathbf{FIS}_v^n$	$\mathbf{I}^n$	$\mathbf{Q}^n$	$\mathbf{T}_{\mathbf{in}}^n$	$\mathbf{T}_{\mathbf{out}}^n$
Number MFs:	3	3	3	3
Number rules:	3	3	3	3
Influence range:	0.60	0.80	0.80	0.80
Epoch number:	500	200	500	200

The prototypes for the validation and check sets were obtained following the same procedure. Fig. 6 shows the prototypes obtained for the training sets for the variables irradiance and water flow, respectively. The figure represents the data for the different classes used to train each FIS. These classes are well separated in the principal component space.

6.3. Fault detection classifier design

An ANFIS model was used to define $\mathbf{class}_v^n$, using the prototypes of each class as input and the general prototype as output. In this way, several ANFIS models were obtained to estimate the prediction for each class. The Subtractive Clustering (SC) method was used in the training of ANFIS structures, (Chiu, 1994). The parameters of each ANFIS layer are updated using a hybrid learning method in Jang (1993), which combines gradient descent to optimise the antecedent and least squares parameters to determine the linear parameters of the consequence. The parameters for each ANFIS architecture are the same for each $\mathbf{FIS}_v^n$ model. These parameters are shown in Table 3 for all variables.

Since faults are induced offsets, the Root Mean Square Error (RMSE) is similar for each $\mathbf{FIS}_v^n$ model obtained on each variable. Therefore, the models $\mathbf{FIS}_v^{Norm}$, $\mathbf{FIS}_v^{F_{pos}}$ and $\mathbf{FIS}_v^{F_{neg}}$ have similar error indices $RMSE_{Trn}$ and $RMSE_{Chk}$, but differ from the models **FIS** for each variable. Table 4 shows the errors obtained in the learning process of each model.

After obtaining the $\mathbf{FIS}_v^n$ models, the class to which the new data belong is determined. Therefore, a cost function was defined that determines which of the $\mathbf{FIS}_v^n$ models reaches the real prototype.

$$J^i = \left\| P^*_{real_v} - P^i_{FIS_v^n} \right\|_2^2 \tag{6}$$

where $P^*_{real_v}$ represents the real prototype of the data to be classified and $P^i_{FIS_v^n}$ represents the prototype estimated by the $\mathbf{FIS}_v^n$ model. To determine the class of data: failure or not, an Exhaustive Search Algorithm (**ESA**) has been used, see Algorithm 2. This algorithm evaluates the cost function with the output value provided by each model $\mathbf{FIS}_v^n$ and determines the class by direct comparison, choosing the one that minimises the cost function. The model with minimum J^i_{min} defines the class to which the new data belong.

Table 4
RMSE index obtained in the learning process for each ANFIS developed.

$\mathbf{RMSE}_{\mathbf{min}}$	ANFIS models developed		
	$\mathbf{FIS}_I^{Norm}$	$\mathbf{FIS}_I^{F_{pos}}$	$\mathbf{FIS}_I^{F_{neg}}$
$RMSE_{Trn}$	$32.2497e^{-3}$	$32.2497e^{-3}$	$32.2497e^{-3}$
$RMSE_{Chk}$	$31.3575e^{-3}$	$31.3575e^{-3}$	$31.3575e^{-3}$
	$\mathbf{FIS}_Q^{Norm}$	$\mathbf{FIS}_Q^{F_{pos}}$	$\mathbf{FIS}_Q^{F_{neg}}$
$RMSE_{Trn}$	$11.4656e^{-3}$	$11.4655e^{-3}$	$11.4647e^{-3}$
$RMSE_{Chk}$	$10.4081e^{-3}$	$10.4074e^{-3}$	$10.3992e^{-3}$
	$\mathbf{FIS}_{T_{in}}^{Norm}$	$\mathbf{FIS}_{T_{in}}^{F_{pos}}$	$\mathbf{FIS}_{T_{in}}^{F_{neg}}$
$RMSE_{Trn}$	$2.0375e^{-3}$	$2.0375e^{-3}$	$2.0375e^{-3}$
$RMSE_{Chk}$	$1.8326e^{-3}$	$1.8326e^{-3}$	$1.8326e^{-3}$
	$\mathbf{FIS}_{T_{out}}^{Norm}$	$\mathbf{FIS}_{T_{out}}^{F_{pos}}$	$\mathbf{FIS}_{T_{out}}^{F_{neg}}$
$RMSE_{Trn}$	$2.1218e^{-3}$	$2.1218e^{-3}$	$2.1218e^{-3}$
$RMSE_{Chk}$	$2.0142e^{-3}$	$21.0719e^{-3}$	$47.3219e^{-3}$

Algorithm 2: Exhaustive Searching Algorithm

```
Result: class
initialisation: class_v = 0, J_class = ∞;
for i=1:i ≤ n:i++ do
    J^i = ‖P*_real_v − P^i_FIS_v^n‖_2^2;
    if J^i < J_class then
        J_class = J^i;
        class_v = i;
    end
end
```

6.3.1. Neuro Fuzzy classifier

The classifier is based on several ANFIS structures that detect measurement failures in one of the solar field variables. Fig. 7 shows the structure of the Neuro-Fuzzy classifier. The first principal component $\mathbf{PCA}_v^n$ obtained from each $\mathbf{class}_v^n$ calculated previously in Section 6.1 remains as fixed parameters and serves to define the new prototypes in the dimension of each class, where v defines the variable sensors. When the normalised data set is multiplied by the first principal component $\mathbf{PCA}_v^{F_{pos}}$, the prototype for the class $\mathbf{class}_v^{F_{pos}}$ is obtained. Multiplying the data by $\mathbf{PCA}_v^{Norm}$ gives the projection in $\mathbf{class}_v^{Norm}$. The input data passes through the first component $\mathbf{PCA}_v^{F_{neg}}$ and is projected into $\mathbf{class}_v^{F_{neg}}$, thus obtaining the corresponding prototype.

The calculated prototypes form the input vector that evaluates the $\mathbf{FIS}_v^n$ models. Finally, the data are passed through the first principal component $\mathbf{PCA}_v^{Total}$, obtaining the real prototype $P^*_{real_v}$, and the resulting class in the **ESA** is determined.

After performing the detection process and determining $class_v^n$ to which the incoming data set belongs, the Neuro-Fuzzy classifier will output an integer in the range $[1-3]$ where $(class_v^{Norm} \in 1, class_v^{F_{pos}} \in 2, class_v^{F_{neg}} \in 3)$ which will be the input of the comparative block. The comparative block consists of a rule (IF $class_v^n = 1$ THEN $F_v = 0$ ELSE $F_v = 1$)that identifies the value $class_v^n$ and assigns to the variable F_v a value of 1 when there is a defect or 0 when there is no

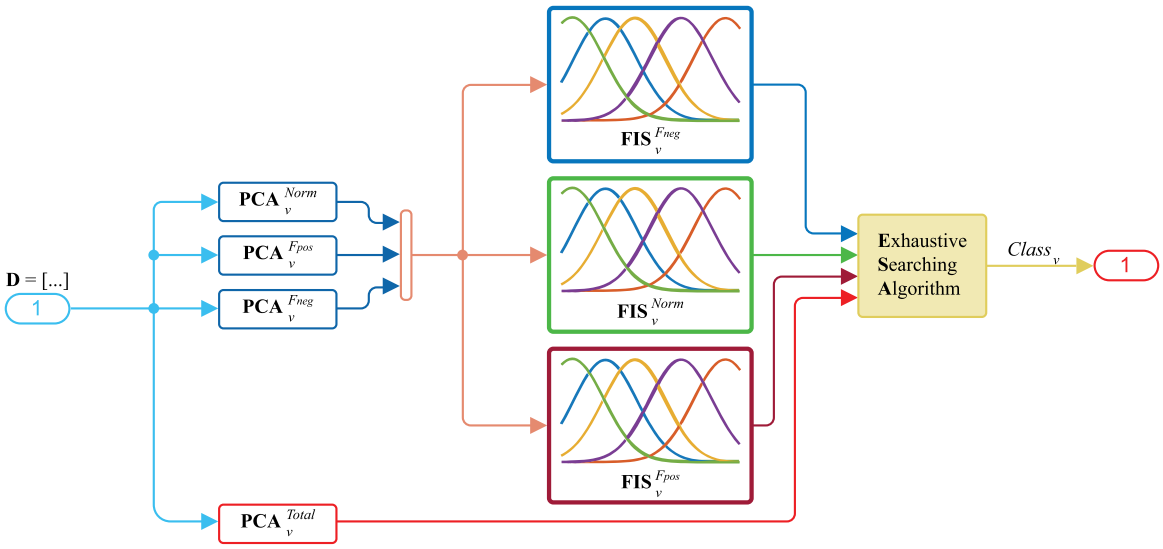


Fig. 7. Neuro-Fuzzy classificator.

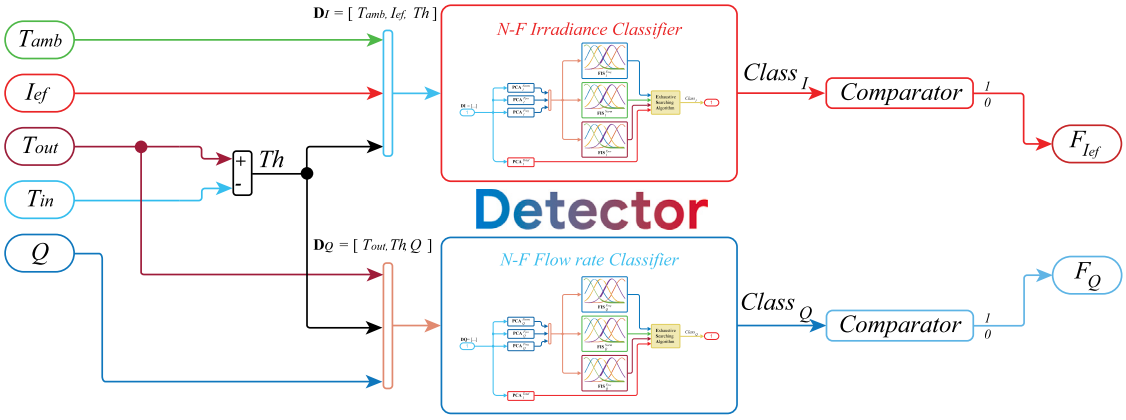


Fig. 8. Internal architecture of the Detector.

defect. This process is shown in the internal architecture of the detector in Fig. 8.

As discussed in Section 2, an advantage of ANFIS is the possibility of interpreting the internal structure inference system. In this case, we have observed that the differences between $\mathbf{FIS}_v^{Norm}$ and $\mathbf{FIS}_v^{Fneg}$ and $\mathbf{FIS}_v^{Fpos}$ are only in the independent term of the consequent of each. This process avoids training with failures and to provide a sensitivity adjustment parameter (the increment or decrement value of the independent term).

7. NN regressors

In this section, NN-based models are presented to predict the correct value of a faulty sensor at any specific time unit. The offset between the input value and the prediction indicates the type of sensor discrepancy. Networks use data about fault-free sensors to make regression predictions. It is assumed that a failure occurs in a single sensor in the system at a specific time unit. Two types of NN are presented:

1. The DNN is used to predict the correct value of one sensor given the other values of the fault-free sensors.
2. RNN using LSTM layers is used to predict the correct value of one sensor given the previous states of fault-free sensors.

7.1. Regression DNN

A regression DNN architecture is presented for sensor prediction. The same architecture is used to train four networks, each of which is used to predict *irradiance*, *water flow rate*, *inlet temperature* and *outlet temperature* in sensors. The architecture consists of a network with six dense hidden layers, of which the first four have 40 units, 5th has 20 units, and 6th has 10 units. All hidden units have the RELU activation function. The output layer has a single unit with a linear activation function. The linear activation function allows the network to perform the regression process to predict the numerical values of the faulty sensor. Fig. 9 represents the proposed architecture.

The four networks learn from the training and data set in Section 5. Before training, the data were normalised to mean 0 and deviation 1. With this network, faulty sensors are predicted from the values of the other fault-free sensors in the same instant of time. PCA is applied to the predictor values and the principal components that explain a variance greater than 0.9 are retained. The number of units in the input layers of each network depends on the number of principal components chosen.

The networks are trained with Adam optimiser using a learning rate of 0.001, with parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1e^{-7}$ and without weight decay. The learning process is stopped after 2000 epochs with parameters: *batch_size* = 10 and *patience* = 100 *epochs*. The loss function to minimise is the Mean Absolute Error (MAE), defined by:

$$MAE = \frac{\sum_{i=1}^{N} |\hat{y} - y_i|}{N} \tag{7}$$

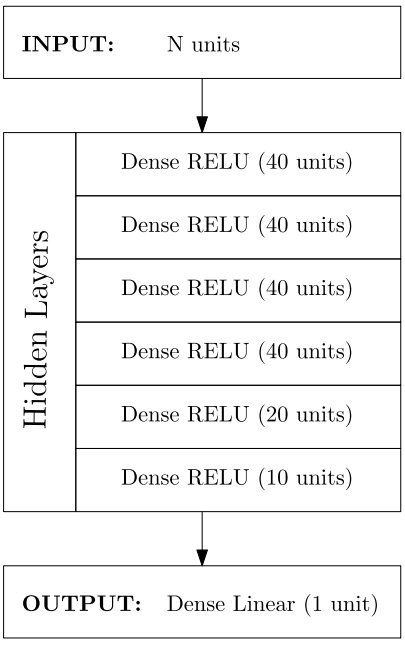


Fig. 9. Regression DNN architecture.

where N is the number of sensors, y_i is the sensor value in one reading and the values $\hat{y}$ are the predictions of the sensor output. Then $|\hat{y} - y_i|$ is the absolute prediction error of the sensor ith.

For all networks, the k-fold technique was used in the training process with $k = 4$ folds. Of the four models returned by k-keep, the one with the lowest MAE in the validation set is chosen. The use of other DNN architectures was attempted without significant improvement in these results. Smaller networks do not show to be accurate in the regression for MAE measures, whereas larger architectures show signs of overfitting.

7.2. Regression RNN

As a second approach, an RNN is created using LSTM layers. RNN architectures are popular when there exists a time dependence on the data. LSTM layers are used to avoid convergence problems when training these types of networks. In this case, the correct value of a faulty sensor is predicted given the values of the fault-free sensor from the last k sensor readings.

To predict a defective sensor from the other values in the training data set, a specific RNN architecture was implemented, Fig. 10. The architecture consists of two LSTM layers with 40 units each, two dense layers with 20 and 10 units, and the RELU activation function. The output layer has only one unit with a linear activation function to make the numerical prediction of the correct value for the faulty sensor. The recurrent layers at the beginning of the network are used to extract features from consecutive states k assuming the time dependence of the data. From these features, the last two fully connected layers predict the expected value of each sensor using regression.

Five LSTM networks were created, one for each sensor. As in the previous case, the training and validation data were normalised to mean 0 and deviation 1, and PCA was applied to the predictive sensor values, keeping the principal components that explain a variance greater than 0.9. Later, the data set is converted to temporal sequences using the parameter t, which indicates how many time instants in the past the network considered for the prediction. For example, if $t = 5$ the network will use for prediction the principal components of the fault-free variables in the current time instant plus the components from the last 5 time instants. The shape of the input of each network depends on the number of principal components and the value chosen for t, Fig. 10.

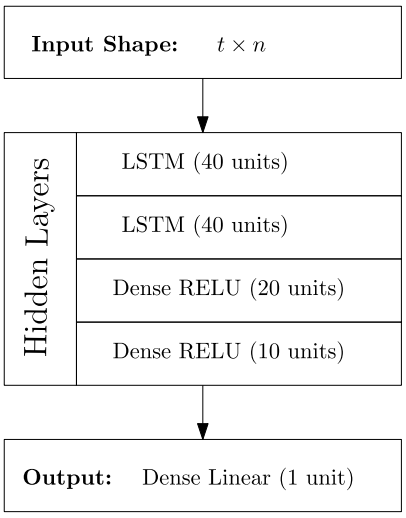


Fig. 10. RNN architecture using LSTM layers.

As in DNN networks, the LSTM networks are trained with Adam optimiser with parameters: $learning\,rate = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-7$ and without weight decay. The learning process is stopped after 2000 epochs with parameters: $batch\_size = 10$ and $patience = 100$ $epochs$. The loss function to minimise is the MAE.

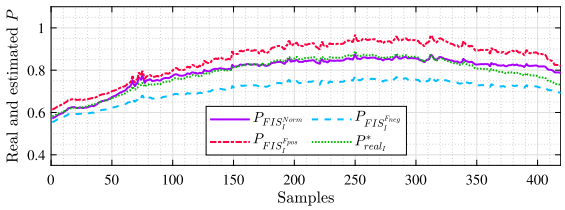
The k-fold technique was used in the training process with $k = 4$ folds. Of the four models returned by k-keep, the one with the lowest MAE on the validation set was chosen. Other architectures for the recurrent network were studied without a significant improvement in precision in the validation set.

8. Results

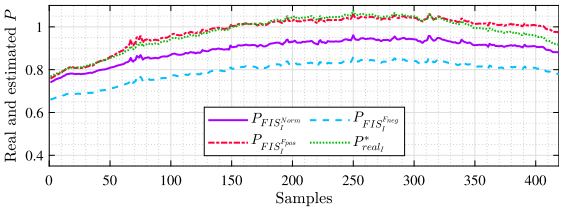
In this section, simulation results of a complete implementation of the methodology applied to the SCP are presented. The simulation integrates two branches: the NF system for fault detection and the RNN for offset error prediction. Experiments are presented for each specific branch and for both branches combined. The test data set used in the simulation is different from the training and validation data used to fit the machine learning models and consists of real data from the plant. Furthermore, a discussion of the results obtained is presented, including a comparison with a similar approach in the literature.

In the detection branch experiments, the NF was tested with real data from the plant. The input vectors are created by taking the principal components of the sensor readings in the test data to evaluate each $\mathbf{FIS}_v^n$ model. The first evaluation was carried out considering positive faults, negative faults, and fault-free values in the solar radiation sensor. If the data do not contain solar radiation faults, the values estimated by $\mathbf{FIS}_\mathbf{I}^\mathbf{Norm}$ follow the real prototype that characterises $class_I^{Norm}$, Fig. 11(a). If the input data are positive for failure, the model $\mathbf{FIS}_\mathbf{I}^\mathbf{F_{pos}}$ estimates the prototype $class_I^{F_{pos}}$, as shown in Fig. 11(b). Finally, if negative faults are considered, the model that best estimates the real prototype is $\mathbf{FIS}_\mathbf{I}^\mathbf{F_{neg}}$ that characterises $class_I^{F_{neg}}$, Fig. 11(c). For evaluation, ESA performs a direct comparison to assign the correct class to the values estimated by $\mathbf{FIS}_\mathbf{I}^\mathbf{n}$, which produces great results.

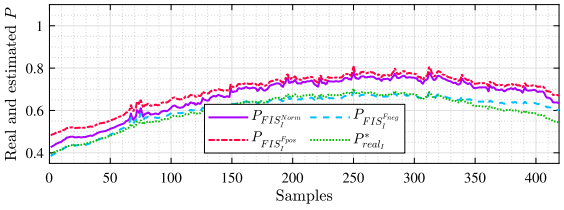
As can be seen in Fig. 11, the solar radiation pattern differs during the day and at night. The decrease in irradiance due to the absence of sunshine makes the classification less accurate, Fig. 12. In this case, it is assumed that the operation of the plant changes to another mode in which the solar collector is not used. Table 5 shows the confusion matrix with the general classification results for the radiation sensor.



(a) *Norm* Prototypes

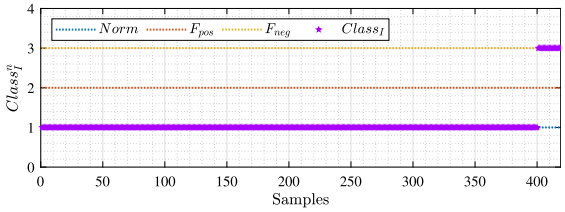


(b) F_{pos} Prototypes

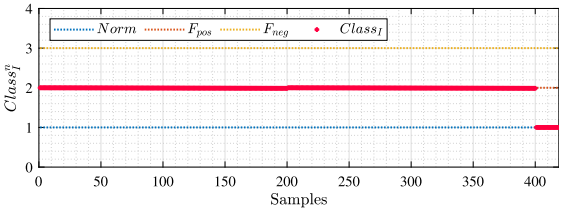


(c) F_{neg} Prototypes

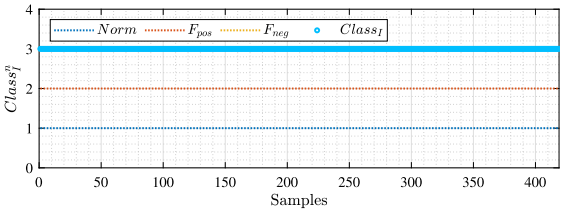
Fig. 11. Prototype obtained by $\mathbf{FIS_I^n}$ of the solar radiation.



(a) Classification with normal data

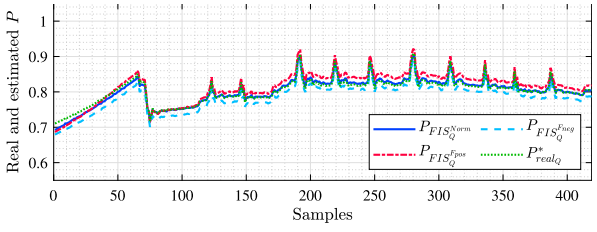


(b) Classification with positive fault data

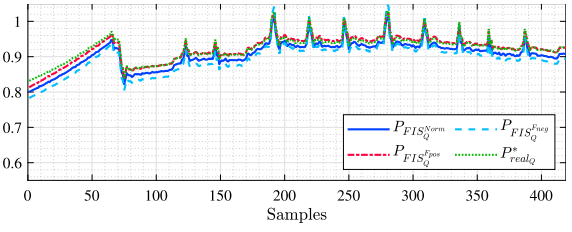


(c) Classification with negative failure data

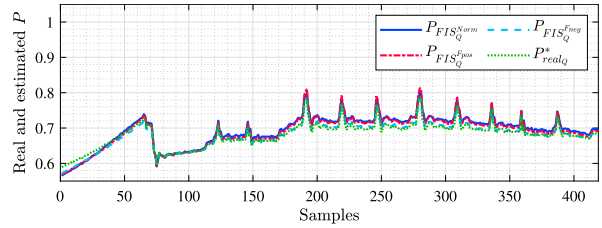
Fig. 12. Class assignment by means of ESA with different **I** data.



(a) *Norm* Prototypes

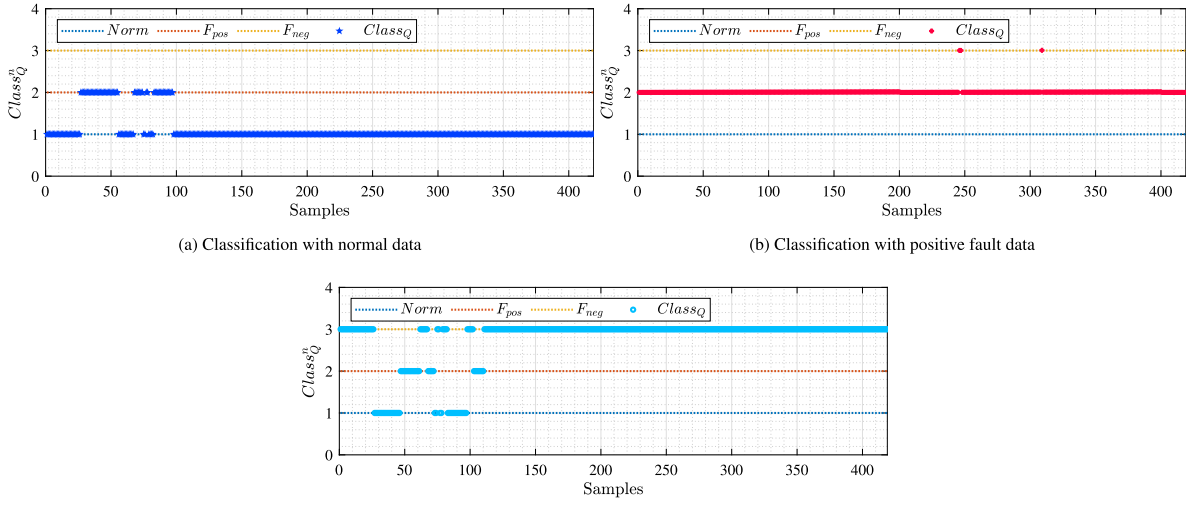


(b) F_{pos} Prototypes



(c) F_{neg} Prototypes

Fig. 13. Prototype obtained by $\mathbf{FIS_Q^n}$ of the flow rate.



(a) Classification with normal data

(b) Classification with positive fault data

(c) Classification with negative failure data

Fig. 14. Class assignment by means of ESA with different **Q** data.

Table 5
Confusion matrix **I**.

		Predicted Classes		
		$\mathbf{F_{neg}}$	Norm	$\mathbf{F_{pos}}$
Actual	$\mathbf{F_{neg}}$	419	0	0
	Norm	0	400	19
	$\mathbf{F_{pos}}$	0	19	400
	Precision	100%	95.47%	95.47%
	Recall	100%	95.47%	95.47%
	Accuracy	96.98%		

Table 6
Confusion matrix **Q**.

		Predicted classes		
		$\mathbf{F_{neg}}$	Norm	$\mathbf{F_{pos}}$
Actual	$\mathbf{F_{neg}}$	352	39	28
	Norm	0	366	53
	$\mathbf{F_{pos}}$	3	0	416
	Precision	84.01%	87.35%	99.28%
	Recall	99.15%	90.37%	83.70%
	Accuracy	90.21%		

Table 7
Results of Neuro-Fuzzy Classifiers.

NF classifier	Flow rate failure data.		
	$class_U^{Norm}$	$class_U^{F_{pos}}$	$class_U^{F_{neg}}$
Water Flow:	34	23	362
Inlet Temp:	390	24	5
Solar radiation:	402	0	17
NF Classifier	Outlet temperature failure data.		
	$class_U^{Norm}$	$class_U^{F_{pos}}$	$class_U^{F_{neg}}$
Water Flow:	393	23	3
Outlet Temp:	1	418	0
Solar radiation:	380	0	39

Table 8
Resulting mean and deviation of the MAE for the DNN and RNN networks in the validation set.

Cases	I	Q	$\mathbf{T_{in}}$	$\mathbf{T_{out}}$
DNN	10.70 ± 14.76	0.97 ± 1.48	12.73 ± 14.14	13.88 ± 15.25
$LSTM\ \ t = 1$	14.31 ± 13.77	1.07 ± 1.43	14.65 ± 14.01	15.88 ± 14.82
$LSTM\ \ t = 5$	13.71 ± 12.70	1.11 ± 1.61	13.51 ± 13.06	13.44 ± 12.58
$LSTM\ \ t = 10$	11.90 ± 12.00	0.93 ± 0.91	14.31 ± 13.50	$\mathbf{12.59 \pm 11.50}$
$LSTM\ \ t = 15$	$\mathbf{9.60 \pm 9.37}$	$\mathbf{0.89 \pm 1.20}$	$\mathbf{10.82 \pm 10.53}$	13.63 ± 11.61

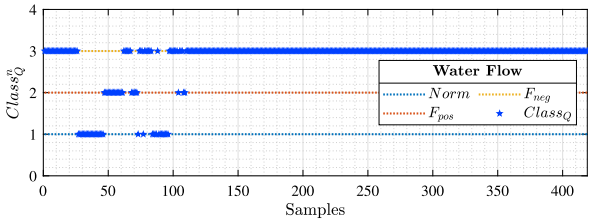
Similarly, experiments were performed considering positive faults, negative faults, and fault-free values on the water flow sensor, Fig. 13. The assignment of classes is shown in Fig. 14 and the respective confusion matrix is presented in Table 6. Results show that the NF system is capable of correctly classifying, as shown in Fig. 15, a flow failure. Furthermore, the detector is fed data that contain faults in both the output temperature and the flow sensors to test the accuracy of the model, as can be seen in Table 7. Simulations demonstrated the great performance of the NF detector in classifying faults presented in two sensors at the same time, Fig. 16.

Regarding the offset error prediction branch, a study on the precision of the DNN and RNN architectures was developed. For each sensor, positive and negative offsets were simulated taking the rest of the sensor to evaluate the networks. Table 8 shows the performance of the DNN and RNN architectures in the test set. In the case of RNN, a study was carried out on the parameter t, which is the number of states in the past that the RNN uses to make predictions. In the study, the recurrent network used 1, 5, 10, and 15 states in the past to check if there was a relationship between the current and previous states of the physical entity. In simulation results, the RNN architecture outperformed the DNN, proving the learning capabilities of the recurrent network to detect patterns from data series. Furthermore, the results indicate that the precision of the prediction increases as the value of t increases, reaching the best precision when $t = 15$. In this way, the relation in time of the sensor values is demonstrated.

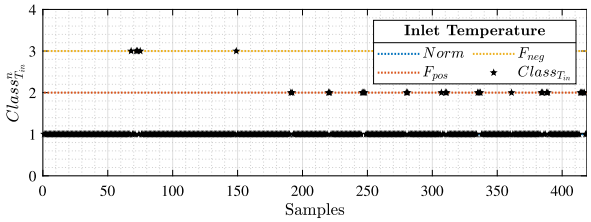
8.1. Integration of the fault detection system and the offset error prediction

When neurofuzzy detection models are joined with recurrent prediction models, a full implementation of the proposed methodology is obtained. The entire system has the ability to detect defective sensors in the SCP and measure the offset of failure, Algorithm 1. The implementation of the entire system can be found on Github.[3] In the code, the detection and prediction models are executed in parallel, maximising the time in the computation. The experiments were performed with a 3.20 GHz processor with 16.0 GB of RAM. The programming language used was Python 3.8.

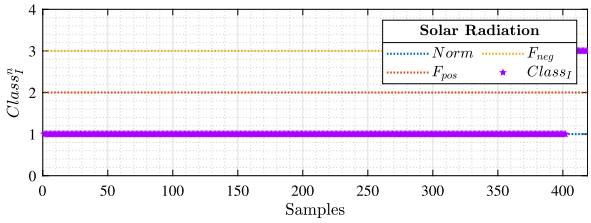
[3] https://github.com/fabio-rodriguez/dataqualitySCP



(a) Classification of flow rate with flow rate failure

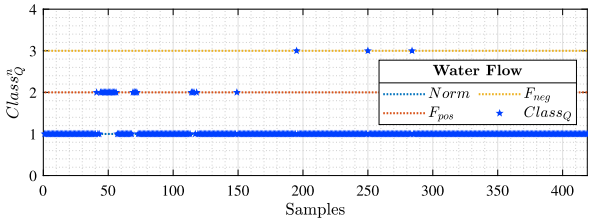


(b) Classification of inlet temperature with flow rate failure

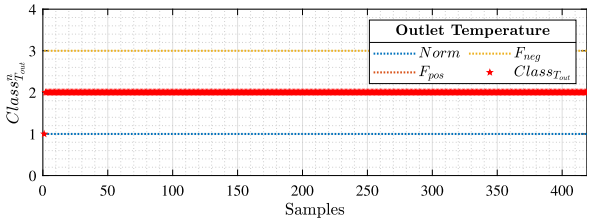


(c) Classification of irradiance with flow rate failure

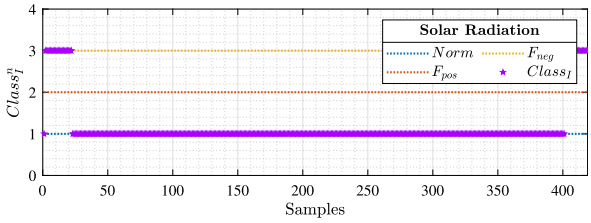
Fig. 15. Classification with flow rate failure data.



(a) Classification of flow rate with outlet temperature failure



(b) Classification of outlet temperature with outlet temperature failure



(c) Classification of irradiance with outlet temperature failure

Fig. 16. Classification of irradiance with outlet temperature failure.

For testing the proposed methodology, a simulation was executed with real data corresponding to a working day of the SCP, from 11:00 am to 6:00 pm. Data failures were induced by simulating offsets. Fig. 17 shows the result of the proposed methodology applied to the SCP, using the RNN architecture in the offset prediction with a value of 15 for the parameter t, which reached better results in previous experiments.

In the simulation, faults are simulated in all sensors with the restriction that only one sensor fails at the same time. In each reading, every sensor entry is verified for faults using the NF classifiers, and offsets are calculated using the RNN outputs. In this particular experiment, the historical record is subsequently filled with readings that contain failures, which obviously leads to bad predictions. To overcome this issue, when a sensor is detected to be faulty, the historical record takes the prediction of the expected value made by the RNN, so the historical record is filled with fault-free values and approximations of the faulty values. In this realistic simulation, errors in detection and prediction are carried out throughout the work day. Despite these drawbacks, Table 9 shows promising results in terms of fault detection accuracy and offset prediction error.

In real-world system applications, users must create their own definition of sensor failure. For example, taking the parameters n and α, n representing a number of states and α representing a probability value ($0 \leq \alpha \leq 1$). A definition of system failure could be the proportion of faulty states found in consecutive n states; if this proportion is greater than α, then the current state is declared faulty. This kind of concept can avoid the point where the system detects occasional outliers, such as sensor noise.

Table 9
Detection accuracy (DA) and mean prediction error (MPE) in offset prediction for a working day using the parameter $t = 15$ for RNN.

Sensor	DA	MPE
Irradiance	0.89	34.56 ± 68.19
Flow	0.83	0.61 ± 1.07
Inlet Temp.	0.83	1.35 ± 2.50
Outlet Temp.	0.88	1.62 ± 2.66

8.2. Comparison and discussion

The methodology and results presented are compared with a similar approach in the literature. In Ruiz-Moreno et al. (2022), a multilayer

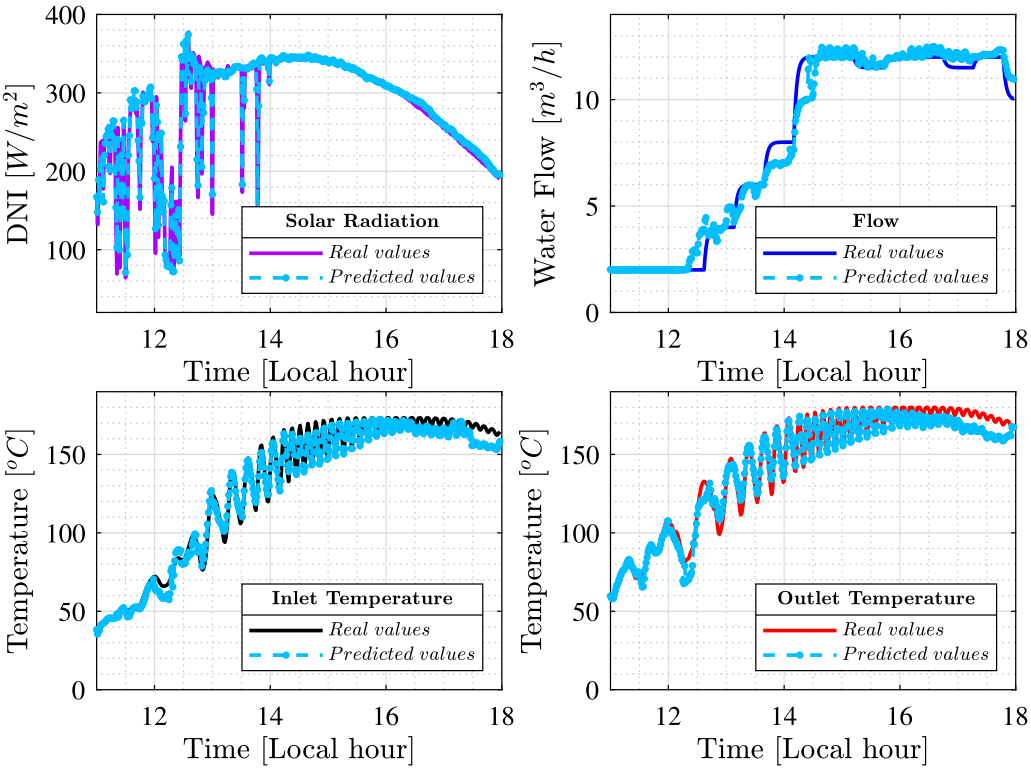


Fig. 17. LSTM networks prediction with $t = 15$.

perceptron network is used to detect and isolate faults in real time in the Fresnel solar field. The faults are grouped into two main categories: those related to the mirrors and their reflectivity and faults in the flow rate. To improve the performance of the network, a decoupling stage is proposed in which the collectors are defocused and the flow rate is decoupled from reflexion and collector efficiency. Great results are shown in the fault classification process, reaching a prediction accuracy of 94.2% in the test set.

Several differences between the present work and Ruiz-Moreno et al. (2022) can be highlighted. Unlike the quoted article, the methodology proposed in this work is a general-purpose system that can be adapted to similar sensor-based DTs in industry, and can allow classification between N types of failure. Furthermore, the proposed methodology is not only a failure classification system, but also a system that allows the variable under suspicion to be compared with its expected value, which allows for a multidimensional assessment of the data quality. One application can be the prediction of displacement errors, which is useful to give an idea of the severity of sensor damage.

Regarding the data set, in Ruiz-Moreno et al. (2022) the authors used simulated data and considered low dynamics with smooth changes throughout the day, while the proposed approach uses a data set that combines real SCP data with simulated data, in which dynamic behaviour is considered. Furthermore, the authors did not assume failures in temperature readings, while we considered failures in any of the plant sensors. All of this makes the system predictors developed more reliable and realistic.

In terms of classification accuracy, the authors compared several network architectures, testing with different numbers of layers and units per layer. The model that retrieved the best results consisted of three hidden layers of 200, 100, and 50 units, respectively. However, the NF detection system proposed in this work requires a relatively low number of epochs to accurately fit the data set and allows the use of expert knowledge to improve the detection of faults. These facts make it easier to update and upgrade the NF detector during the plant life cycle. The performance of the NF system, on the more realistic data set that has been used, is greater than 90% for each sensor, so both techniques are competitive. Furthermore, in this work, a simulation was performed with real data for a working day of the SCP and considers the error carried during the predictions; however, results with precision over 80% were obtained. Finally, unlike Ruiz-Moreno et al. (2022), in the present work a study on the time dependence of the data was carried out. For the prediction of the offset, an RNN was applied that demonstrates that the historical record of the sensors can be treated as time sequences.

The approach presented in that work detects faults at the level of the feedforward controller, since, as can be seen in the results presented, if a fault is added to this flow rate, it produces a change in the output temperature, and the same happens if the fault occurs at the K_{opt} (Optical efficiency). The experiments with which the NN is validated are carried out together with the distributed-parameter model presented in the article, which gives a constant reference temperature. There are a total of 972 experiments with different initial conditions and in a time interval of 11 am–18 pm. Once a failure is detected, an alarm is triggered on that particular day, advising that maintenance is required.

There is a major difference between the approach presented in this paper in relation to the comparative work, since the approach presented in this paper focusses on the detection of incorrect data in sensor measurements, and the comparative approach detects faults in the controller. Furthermore, the presented approach has the ability to differentiate whether the injection was positive or negative through the NF-detector. A particularity of the work with which it is compared presents failures in the variable K_{opt} that affects the product $(I * n_o * K_{opt})$, which is part of the physical equation model (PDE) with which T_{out} of the heat transfer fluid is determined. In this context, this failure has been extrapolated as if it were the failure of the solar radiation sensor, so the percentage that represents the failure in K_{opt} has been obtained and added to I to compare whether *detector* is able to distinguish between no failure, failure in irradiance, and failure in flow rate.

Table 10
Confusion matrix from Ruiz-Moreno et al. (2022).

		Predicted classes		
		Faultless	K_{opt}	Q
Actual	*Faultless*	324	0	0
	K_{opt}	0	280	44
	Q	0	57	267
	Precision	100%	83.09%	85.85%
	Recall	100%	86.42%	82.41%
	Accuracy	89.61%		

Table 11
Confusion matrix of NF-Detector.

		Predicted classes		
		Faultless	I	Q
Actual	*Faultless*	324	0	0
	I	0	324	0
	Q	0	0	324
	Precision	100%	100%	100%
	Recall	100%	100%	100%
	Accuracy	100%		

As can be seen in the confusion tables (Tables 10 and 11), the proposed detector, with the same operating conditions as the one presented by Ruiz-Moreno et al. (2022), has a prediction capacity of 100%. The main difference lies in the variables used to detect flow and irradiance failures in Table 2. This selection also allows for simultaneous fault detection.

9. Conclusion

In this work, a methodology is proposed to collect high-quality data from sensor readings in a sensor-based DT during its life cycle. The data collected helps in the periodic updating process of the models composing the DT. The methodology can be applied to similar sensor-based DT case studies with different numbers and types of sensors. Furthermore, data science techniques can be improved to detect failures in multiple sensors at the same time and can be easily integrated into the workflow.

The proposed methodology was applied to a real DT of a SCP. In this case study, a fuzzy classifier-based evaluator was developed to detect possible failures in the data provided by solar field sensors, and examples of failure detection in solar radiation and flow water sensors were depicted. The evaluator is made up of several Takagi–Sugeno fuzzy inference systems that classify each sensor reading into three possibilities: positive fault, negative fault, and fault-free. The detection system has shown excellent results in terms of accuracy and sensitivity rates, outperforming a similar approach in the literature.

In general, there were great results in predicting the error offset of the sensors. The study in the parameter t demonstrated the recurrent character inherent to the variables involved in the SCP, making the recurrent LSTM-based network an ideal candidate for the learning procedure. Machine learning techniques have been shown to have excellent computational time performance, taking less than a second to detect failures in each sensor reading. However, a study is needed on the impact of increasing the value of the parameter t to optimise the accuracy of the networks. Future approaches can be dedicated to improve the prediction of offsets by conducting this study.

In future work, it is proposed to complete a quality evaluation system, assessing several dimensions (accuracy, confidence, completeness, data volume, and timeliness). It is also interesting in future work to extend this methodology to the rest of the critical variables of the solar field that will have a physical-virtual interaction with its complete DT. The solar field has different modes of operation depending on different circumstances. The evaluation of the quality of the data will allow for the auto-configuration of the DT to be done with the appropriate context. In addition, machine learning techniques will be improved to detect failures that occur in multiple sensors at the same time. It will also be a future research to determine detector fitting parameters for each of the variables, exploiting the observed property of change only in the independent term of the resulting ANFIS. This will allow not to train with faulty data each time the detection is evaluated with more or less threshold deviation.

Although the work initially focusses on the data quality of one variable, subsequent tests are carried out in which it is observed that the developed system, using the proposed methodology, is capable of detecting multiple errors in several sensors simultaneously with good performance. These cases are shown in Figs. 15 and 16 as well as in Table 7. A simultaneous failure in two variables implies, due to the architecture, a defect that is difficult to evaluate; there is a dependency on the other variables in the evaluation system. The simultaneous detection capability of the proposed detector opens a path for future research in which simultaneous detection and evaluation of several faults can be obtained.

CRediT authorship contribution statement

Fabio Rodríguez: Investigation, Methodology, Formal analysis, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **William D. Chicaiza:** Investigation, Methodology, Formal analysis, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Adolfo Sánchez:** Conceptualization, Methodology, Investigation, Supervision, Visualization, Writing – original draft, Writing – review & editing. **Juan M. Escaño:** Conceptualization, Methodology, Investigation, Resources, Funding acquisition, Project administration, Supervision, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The authors want to thank the European Commission for funding this work under Project DENiM. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 958339.

Appendix. Adaptive neuro-fuzzy inference system, ANFIS

The classifier based on ANFIS (Jang, 1993), is based on fuzzy TS systems (Takagi and Sugeno, 1985) and is composed of rules of type j:

$$\texttt{IF } x_1 \texttt{ is } F_{1j} \texttt{ and } x_2 \texttt{ is } F_{2j} \texttt{ and } x_i \texttt{ is } F_{ij} \texttt{ ,}$$

$$\texttt{THEN} : f_j(x) = g_{0j} + g_{1j}x_1 + \cdots + g_{ij}x_i$$

where $g_{ij} \in \Re$ are parameters, x_i are inputs, f_j output, respectively, for each rule, and F_{ij} represents the fuzzy sets defined by Gaussian membership functions (MF) of the type:

$$\mu_{F_{ij}}(x_i) = \frac{1}{1 + \left[\left(\frac{x_i - c_{ij}}{a_{ij}}\right)^2\right]^{b_{ij}}} \qquad \text{(A.1)}$$

$$\mu_{F_{ij}}(x_i) = e^{-\frac{1}{2}\left(\frac{x_i - c_{ij}}{a_{ij}}\right)^2}$$

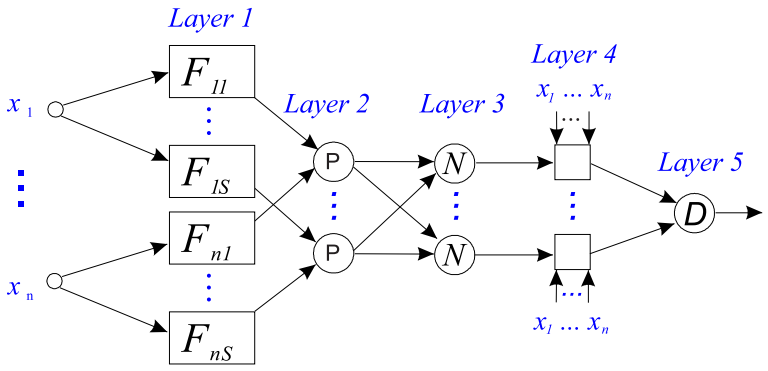


Fig. A.18. Fuzzy Neural Network (Jang, 1993).

where a_{ij}, b_{ij}, c_{ij} is the set of parameters used to vary the MFs. The value of the function $\mu_{F_{ij}}$ takes for a given x_i is known as the degree of membership of x_i for the fuzzy set F_{ij}. Fig. A.18 presents an ANFIS as an example with n input variables and one output variable. The first layer is made up of the membership functions of each A_{ij}, defined by the membership degree.

$$\mu_{F_{ij}} : x_i \in \mathbb{R} \longmapsto \mu_{F_{ij}}(x_i) \in \mathbb{R} \tag{A.2}$$

The output of each node i is $\mu_{F_{ij}}(x_i)$, the degree of membership of x_i. For the definition of these membership functions, Gaussian functions are used. The second layer features nodes labelled Π that implement fuzzy engine inference.

$$\omega_j(x) = \mu_{F_{1j}}(x_1) \cdot \mu_{F_{2j}}(x_2) \cdot \cdots \cdot \mu_{F_{nj}}(x_n) \tag{A.3}$$

Or

$$\omega_j(x) = min\{\mu_{F_{1j}}(x_1), \mu_{F_{2j}}(x_2), \dots, \mu_{F_{nj}}(x_n)\} \tag{A.4}$$

The third layer normalises the inference motor. The output of each node of this layer is:

$$a_i(x) = \frac{\omega_i(x)}{\sum_{i=1}^{N} \omega_i(x)} \tag{A.5}$$

N is the number of rules of the system. The fourth layer has adaptive nodes, and the fifth layer is the defuzzyfication node. For TS systems, the output could be the following:

$$\sum_{i=1}^{N} a_i(x) \cdot f_i(x) = \frac{\sum_{i=1}^{N} \omega_i(x) \cdot f_i(x)}{\sum_{i=1}^{N} \omega_i(x)} \tag{A.6}$$

References

Belaout, A., Krim, F., Mellit, A., Talbi, B., Arabi, A., 2018. Multiclass adaptive neuro-fuzzy classifier and feature selection techniques for photovoltaic array fault detection and classification. Renew. Energy 127, 548–558. http://dx.doi.org/10.1016/j.renene.2018.05.008.

Bermejo, P., Pino, F.J., Rosa, F., 2010. Solar absorption cooling plant in Seville. Sol. Energy 84 (8), 1503–1512. http://dx.doi.org/10.1016/j.solener.2010.05.012.

Boschert, S., Rosen, R., 2016. Digital twin—The simulation aspect. In: Hehenberger, P., Bradley, D. (Eds.), Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and their Designers. Springer International Publishing, Cham, pp. 59–74. http://dx.doi.org/10.1007/978-3-319-32156-1_5.

Camacho, E.F., Gallego, A.J., Escaño, J.M., Sánchez, A.J., 2019a. Hybrid nonlinear MPC of a solar cooling plant. Energies 12 (14), http://dx.doi.org/10.3390/en12142723.

Camacho, E.F., Gallego, A.J., Sanchez, A.J., Berenguel, M., 2019b. Incremental state-space model predictive control of a fresnel solar collector field. Energies 12 (1), 1–23. http://dx.doi.org/10.3390/en12010003.

Chicaiza, W.D., Sánchez, A.J., Gallego, A.J., Escaño, J.M., 2021. Neuro-fuzzy modelling of a linear fresnel-type solar collector system as a digital twin. In: Joint Proceedings of the 19th World Congress of the International Fuzzy Systems Association (IFSA), the 12th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT), and the 11th International Summer School on Aggregation Operators. AGOP, Atlantis Press, pp. 242–249. http://dx.doi.org/10.2991/asum.k.210827.033.

Chiu, S., 1994. Fuzzy model identification based on cluster estimation. J. Intell. Fuzzy Systems 2, 267–278. http://dx.doi.org/10.3233/IFS-1994-2306.

Ding, S.X., 2008. Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools. Springer Science & Business Media.

Escaño, J.M., Ridao-Olivar, M.A., Ierardi, C., Sánchez, A.J., Rouzbehi, K., 2020. Driver behavior soft-sensor based on neurofuzzy systems and weighted projection on principal components. IEEE Sens. J. 20 (19), 11454–11462. http://dx.doi.org/10.1109/JSEN.2020.2995921.

Gallego, A.J., Sánchez, A.J., Berenguel, M., Camacho, E.F., 2020. Adaptive UKF-based model predictive control of a Fresnel collector field. J. Process Control 85, 76–90. http://dx.doi.org/10.1016/j.jprocont.2019.09.003.

Glaesssgen, E., Stargel, D., 2012. The digital twin paradigm for future NASA and U.S. Air Force vehicles. In: 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. http://dx.doi.org/10.2514/6.2012-1818.

Grieves, M., Vickers, J., 2017. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In: Kahlen, F.-J., Flumerfelt, S., Alves, A. (Eds.), Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches. Springer International Publishing, Cham, pp. 85–113. http://dx.doi.org/10.1007/978-3-319-38756-7_4.

Haag, S., Anderl, R., 2018. Digital twin – proof of concept. Manuf. Lett. 15, 64–66. http://dx.doi.org/10.1016/j.mfglet.2018.02.006, Industry 4.0 and Smart Manufacturing.

Heo, S., Lee, J.H., 2018. Fault detection and classification using artificial neural networks. IFAC-PapersOnLine 51 (18), 470–475. http://dx.doi.org/10.1016/j.ifacol.2018.09.380, 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018.

Hribernik, K., Cabri, G., Mandreoli, F., Mentzas, G., 2021. Autonomous, context-aware, adaptive digital twins—state of the art and roadmap. Comput. Ind. 133, 103508. http://dx.doi.org/10.1016/j.compind.2021.103508.

Jamil, M., Sharma, S.K., Singh, R., 2015. Fault detection and classification in electrical power transmission system using artificial neural network. SpringerPlus 4, 1–13. http://dx.doi.org/10.1186/s40064-015-1080-x.

Jang, J.S., 1993. ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. 23 (3), 665–685. http://dx.doi.org/10.1109/21.256541.

Kang, Z., Catal, C., Tekinerdogan, B., 2021. Remaining useful life (RUL) prediction of equipment in production lines using artificial neural networks. Sensors 21 (3), http://dx.doi.org/10.3390/s21030932.

Klein, A., Lehner, W., 2009. Representing data quality in sensor data streaming environments. J. Data Inf. Qual. 1 (2), http://dx.doi.org/10.1145/1577840.1577845.

Labrador Rivas, A.E., Abrão, T., 2020. Faults in smart grid systems: Monitoring, detection and classification. Electr. Power Syst. Res. 189, 106602. http://dx.doi.org/10.1016/j.epsr.2020.106602.

Li, B., Delpha, C., Diallo, D., Migan-Dubois, A., 2021. Application of artificial neural networks to photovoltaic fault detection and diagnosis: A review. Renew. Sustain. Energy Rev. 138, 110512. http://dx.doi.org/10.1016/j.rser.2020.110512.

Machado, D.O., Sánchez, A.J., Gallego, A.J., de Andrade, G.A., Normey-Rico, J.E., Bordons, C., Camacho, E.F., 2022. Split-range control for improved operation of solar absorption cooling plants. Renew. Energy 192, 361–372. http://dx.doi.org/10.1016/j.renene.2022.04.064.

Onile, A.E., Machlev, R., Petlenkov, E., Levron, Y., Belikov, J., 2021. Uses of the digital twins concept for energy services, intelligent recommendation systems, and demand side management: A review. Energy Rep. 7, 997–1015. http://dx.doi.org/10.1016/j.egyr.2021.01.090.

Pileggi, P., Verriet, J., Broekhuijsen, J., van Leeuwen, C., Wijbrandi, W., Konsman, M., 2019. A digital twin for cyber-physical energy systems. In: 2019 7th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems. MSCPES, pp. 1–6. http://dx.doi.org/10.1109/MSCPES.2019.8738792.

Robledo, M., Escaño, J.M., Núñez, A., Bordons, C., Camacho, E.F., 2011. Development and experimental validation of a dynamic model for a fresnel solar collector. IFAC Proc. Vol. 44 (1), 483–488. http://dx.doi.org/10.3182/20110828-6-IT-1002.03252, 18th IFAC World Congress.

Ruiz-Moreno, S., Gallego, A.J., Sanchez, A.J., Camacho, E.F., 2022. Fault detection and isolation based on deep learning for a fresnel collector field. IFAC-PapersOnLine 55 (6), 563–568.

Saracco, R., 2019. Digital twins: Bridging physical space and cyberspace. Computer 52, 58–64. http://dx.doi.org/10.1109/MC.2019.2942803.

Shah, J., Wang, W., 2021. An evolving neuro-fuzzy classifier for fault diagnosis of gear systems. ISA Trans. http://dx.doi.org/10.1016/j.isatra.2021.05.019.

Takagi, T., Sugeno, M., 1985. Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst. Man Cybern. SMC-15 (1), 116–132. http://dx.doi.org/10.1109/TSMC.1985.6313399.

Tang, S., Zhu, Y., Yuan, S., Li, G., 2020. Intelligent diagnosis towards hydraulic axial piston pump using a novel integrated CNN model. Sensors 20 (24), http://dx.doi.org/10.3390/s20247152.

Wang, Q., Yu, Y., Ahmed, H.O.A., Darwish, M., Nandi, A.K., 2020. Fault detection and classification in MMC-HVDC systems using learning methods. Sensors 20 (16), http://dx.doi.org/10.3390/s20164438.

Zhang, J., Morris, J., 1996. Process modelling and fault diagnosis using fuzzy neural networks. Fuzzy Sets and Systems 79 (1), 127–140. http://dx.doi.org/10.1016/0165-0114(95)00295-2, Neuro-Fuzzy Techniques and Applications.