

A cascade neural network methodology for fault detection and diagnosis in solar thermal plants

Sara Ruiz-Moreno^{a,*}, Antonio J. Gallego^a, Adolfo J. Sanchez^b, Eduardo F. Camacho^a

^a Dept. de Ingeniería de Sistemas y Automática, University of Seville, Camino de los Descubrimientos, E-41092, Seville, Spain

^b Dept. of Mechanical, Biomedical and Manufacturing Engineering, Munster Technological University, Cork T12 P928, Ireland

ARTICLE INFO

Keywords:

Solar energy
Parabolic-trough collectors
Artificial intelligence
Fault detection
Fault diagnosis

ABSTRACT

Detecting and isolating faults in collector fields of solar thermal power plants is a crucial and challenging task. The system variables in the collector area are highly coupled, which can lead to a high misclassification rate. For this reason, it becomes necessary to combine knowledge of systems engineering with machine learning techniques that unravel the complex dynamics that govern the systems using historical data. Furthermore, the performance of a solar thermal plant is highly dependent on solar irradiance which changes during the day and is subject to perturbations caused by clouds and other atmospheric conditions. Detecting the fault requires using techniques that cope with the disturbances in solar irradiance.

In this work, real irradiance profiles with many types of clouds are used. First, a model-based fault detector is applied, obtaining an accuracy of over 89% for all test irradiances. Then, different machine learning techniques are compared: static neural networks with and without decoupling strategy, dynamic neural networks, dynamic neural networks in cascade, classification trees, random forests, radial basis function networks, and self-organizing maps. The combination of neural networks was the only method that obtained a total accuracy of over 73% and F1-scores over 50% for all the test irradiance profiles.

1. Introduction

For some years now, there has been a growing preoccupation with the environment and the reduction of emissions of polluting gases into the atmosphere. Consequently, the main concerns of the 21st century are climate change and energy security [1]. Currently, there is a constant development and expansion of clean and renewable energy sources, which play a very important role in curbing global warming and climate change, as well as in promoting economic growth [2]. Among renewable energy sources, solar energy is considered to be the cleanest [3].

This work focuses on solar thermal energy, which harnesses the thermal energy of the Sun to heat fluids. More specifically, parabolic trough collectors (PTCs) are a type of solar thermal device composed of parabolic mirrors that concentrate solar irradiance onto a tube to heat a fluid and then produce steam, generally to drive a turbine generator.

In addition to temperature regulation [4], a very important task in this type of plant is the correct detection of any type of failure and its characterization. This allows the application of any necessary mitigation, reconfiguration, and correction tasks [5], and facilitates

maintenance operations. Fault detection and diagnosis (FDD) [6] encompasses the set of techniques destined for these tasks. It is divided into fault detection, which consists of alarming about the appearance of a fault, and diagnosis, which provides certain information about it. Within the diagnosis techniques, one can find fault isolation, which provides information on the type and location of the fault, and fault identification, which determines its magnitude.

Machine learning (ML) is a research field dedicated to providing systems with the ability to learn in order to generate models and solve problems [7]. The use of ML techniques, and more specifically, artificial neural networks (ANNs), has been extended to a wide range of applications. One of them is FDD, where the ability to learn from historical records or to find relationships between data is exploited.

Numerous examples of ML algorithms applied to FDD can be found in the literature. For instance, Ahmadipour et al. [8] propose a fault detection and classification method for microgrids combining support vector machines with augmented Lagrangian particle swarm optimization and signal processing. Brown et al. [9] apply a k-nearest neighbors classifier for detecting faults and decision trees to classify them. In the work by Fuming et al. [10], deep neural networks are applied

* Corresponding author.

E-mail addresses: srmoreno@us.es (S. Ruiz-Moreno), agallego2@us.es (A.J. Gallego), adolfo.j.sanchez@ieee.org (A.J. Sanchez), efcamacho@us.es (E.F. Camacho).

<https://doi.org/10.1016/j.renene.2023.04.051>

Received 11 November 2022; Received in revised form 16 March 2023; Accepted 12 April 2023

Available online 25 April 2023

0960-1481/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

to wind turbines to detect multiple faults based on improved triplet loss. The work by Hussain et al. [11] uses radial basis functions for fault detection integrating two bidirectional input parameters in photovoltaics.

In the energy field, great advances are being made lately by applying FDD techniques to microgrids. It is worth mentioning the works by Bernardi et al. [12], which proposes a fault estimator with a fault-tolerant model predictive controller, or Marquez et al. [13], aimed at quantifying and mitigating faults with model predictive control reconfiguration. Concerning thermal solar plants, there are not many FDD applications, focusing mainly on detection or considering the collectors as a single subsystem, instead of distinguishing faults inside the collector. For example, Georgii et al. [14] present a software framework that selects the best fault detection method in solar heating systems based on previous data, and Faure et al. [15] propose a methodology for analyzing the effect of faults in solar thermal systems. An example of detection is the work by Correa-Jullian et al. [16], which uses a neural network to predict the temperature of the system and detect faults. Considering fault diagnosis, Zahra et al. [17] use a Takagi–Sugeno fuzzy model to estimate the state and generate residuals for diagnosis in a simplified solar field. An ML approach is made by Jiang et al. [18], who apply support vector machines to classify faults in solar water heaters. Finally, Gao and Su [19] implement an active disturbance rejection control in a PTC plant considering flow rate fluctuations and changes in collector parameters.

A previous work [20] proposed a methodology based on artificial neural networks and two decoupling strategies to detect and isolate three types of faults in PTCs: faults in the optical efficiency –associated with breakage, degradation, corrosion and coating of the mirror and the receiver–, flow rate –related to imbalances of flow-rate in the loop–, and thermal losses –related to dirt, wear, insulation and breakage of the pipes–. Lately, the neural networks were improved by introducing information from past inputs to detect faults in days with high variations in the irradiance due to clouds passing by [21]. The method detects sudden faults in the same day they appear, or incipient faults when the magnitude exceeds about 10%. The main contribution of this work is threefold. First, a model-based fault detection technique is developed by using the concentrated parameter model. Then, this work proposes a new technique to distinguish the three types of faults in days with clouds by means of a combination of neural networks. This is performed by implementing a neural selector acting as a weighter in cascade with the classification neural networks to give them more generalizability. In addition, this paper provides a comparison between the proposed method and traditional classification techniques under different circumstances, which validates the effectiveness of the proposed method.

This paper is organized as follows. Section 2 describes the PTC plant and provides the models of the system, as well as the flow rate controller. Then, Section 3 describes the proposed methodology with artificial neural networks. First, the challenge of isolating faults is illustrated, then the classification neural networks are described, and finally, the neural network combination is explained. A description of the rest of the classification techniques that were implemented is also provided. Section 4 presents the simulation results and comparative analysis, and finally, Sections 5 and 6 draw some discussion and conclusions.

2. System description

This section describes the solar plant and the physical models used to simulate and control it. Next, the controller used to regulate the flow rate is also presented.

PTC plants are composed of parabolic mirrors in series that form loops, as shown in Fig. 1. Each mirror receives the solar direct normal irradiance (DNI) and reflects it towards a pipe located at the focal line of its parabola. A heat transfer fluid (HTF) that circulates through the

Table 1
Description of parameters and variables.

Symbol	Description	Units
G	Collector aperture	m
$H_f(T)$	Convective heat transfer coefficient	W/(m ² °C)
A	Cross-sectional area	m ²
$\rho(T)$	Density	kg/m ³
$I(t)$	Direct solar irradiance	W/m ²
μ	Dynamic viscosity	mPa s
ϵ	Effective roughness	m
$q(t)$	Flow rate	l/s
$n_o(t)$	Geometric efficiency	–
g	Gravity of Earth	m/s ²
K_{opt}	Optical efficiency	–
L_p	Pipe length	m
η_{pump}	Pump efficiency	–
$C(T)$	Specific heat capacity	J/(kg °C)
v	Speed	m/s
$T(t)$	Temperature	°C
$H_f(T)$	Thermal loss coefficient	W/(m ² °C)
t	Time	s
S	Total area of the field	m ²
L	Tube diameter	m

Table 2
Description of subscripts.

Symbol	Description
a	Ambient
f	Fluid
m	Metal
in	Input
out	Output
$mean$	Mean between input and output
$loop$	Mean for the entire loop
ref	Reference
est	Estimated
mea	Measured

pipe is heated with solar rays. The fluid is then fed to a heat exchanger to produce steam and drive a turbine generator.

For simulation purposes, this work uses a model of the ACU-REX plant, which was located at the Plataforma Solar de Almería before dismantling and operated for more than 30 years [22]. It was a plant of 1 MW composed of 10 loops of 12 modules of 4 single-axis east–west aligned solar collectors. Each loop is 172 m long and has an active part of 142 m that receives solar irradiance and a passive part of 30 m. In this work, only one loop is considered, and the methodologies can be extended to more loops independently. The heat transfer fluid is Therminol 55 thermal oil, with density ρ_f and specific heat capacity C_f given by Eqs. (1) and (2). Tables 1 and 2 give the notation used in this document.

$$\rho_f = 903 - 0.672T_f \quad (1)$$

$$C_f = 1820 - 3.478T_f \quad (2)$$

2.1. Distributed parameter model

The system is simulated by computing the distributed parameter model. It is given by the partial differential Eqs. (3) and (4), which describe the energy balances in the pipe and the HTF with spatially distributed variables [4,23]. The model is discretized into 172 segments and an integration time of 0.25 s is used. The faults are modeled as multipliers. Three types of faults are taken into account, each for a term of the equations: $\alpha_{K_{opt}} < 1$ for faults in the optical efficiency, $\alpha_q \neq 1$ for faults in the flow rate, and $\alpha_{Hl} > 1$ for faults in the thermal losses. Values of these parameters equal to 1 indicate that the system is working without failure.

$$\rho_m C_m A_m \frac{\partial T_m}{\partial t} = \alpha_{K_{opt}} I K_{opt} n_o G - \alpha_{Hl} H_l G (T_m - T_a) - L H_l (T_m - T_f) \quad (3)$$

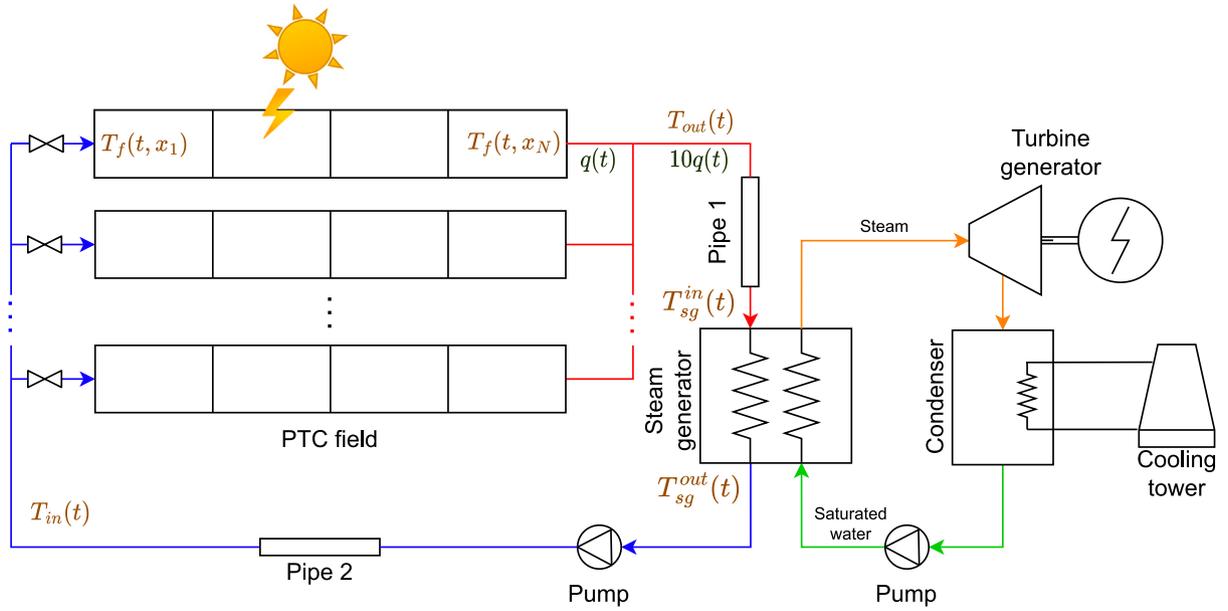


Fig. 1. Scheme of a parabolic-trough collector field.

Table 3
Parameters of the ACUREX plant.

Parameter	Value
ρ_m	7800 kg/m ³
C_m	550 J/kg °C
A_m	$2.4806 \cdot 10^{-4}$ m ²
G	1.82 m
L	$7.98 \cdot 10^{-2}$ m
A_f	$5.0671 \cdot 10^{-4}$ m ²
S	2672 m ²

$$\rho_f C_f A_f \frac{\partial T_f}{\partial t} + \alpha_q \rho_f C_f q \frac{\partial T_f}{\partial x} = LH_t(T_m - T_f) \quad (4)$$

The parameters of the system are collected in Table 3, and the coefficients of thermal loss H_l and convective heat transfer of the inner tube H_t are obtained with Eqs. (5) and (6), respectively.

$$H_l = 0.00249(T_f - T_a) - 0.06133 \quad (5)$$

$$H_t = q^{0.8}(2.17 \cdot 10^6 - 5.01 \cdot 10^4 T_f + 4.53 \cdot 10^2 T_f^2 - 1.64 T_f^3 + 2.1 \cdot 10^{-3} T_f^4) \quad (6)$$

The geometric efficiency n_o , also known as $\cos(\theta)$, is obtained with the relation between the radiation beam vector and the normal vector of the mirror. It depends on the collector dimensions, declination, hourly angle, Julian day, latitude, and solar hour [24,25]. The other type of efficiency considered in the collectors is optical efficiency K_{opt} , which takes into account factors such as reflectivity and soiling of the mirrors, tube absorptance or interception factor. Defects and dirt on the mirrors can lower the optical efficiency of the collectors, which is considered a plant fault. As Azouzoute et al. point out, dust accumulation alone can already decrease the optical efficiency of solar power plants by more than 30% in just one week [26,27].

2.2. Concentrated parameter model

A simpler description of the plant is provided by the concentrated parameter model – sometimes referred to as lumped parameter model –, which represents the internal energy variation of the fluid and is given by Eq. (7). The thermal capacity of the loop is $C_{loop} = L_{loop} \rho_f C_f A_f$,

and $P_{cp} = \rho_f C_f$. This model is used to implement a flow controller, described in Section 2.3.

$$C_{loop} \frac{dT_{out}}{dt} = \alpha_{K_{opt}} n_o K_{opt} S I - \alpha_q q P_{cp} (T_{out} - T_{in}) - \alpha_{H_l} H_l A (T_{mean} - T_a) \quad (7)$$

2.3. Flow control

The flow rate that circulates through the pumps is manipulated to maintain the outlet temperature around a reference temperature. A feedforward controller is implemented with a sample time of 39 s [28] and assuming that there are no faults in the system. The controller equation derives from the concentrated parameter model in steady-state, as shown in Eq. (8). The flow rate is kept between 0.2 l/s and 1.2 l/s.

$$q = \frac{n_o K_{opt} S I - H_l A (T_{mean} - T_a)}{P_{cp} (T_{ref} - T_{in})} \quad (8)$$

3. Proposed methodologies for FDD

In this work, an FDD methodology is implemented starting from the results obtained in previous works [21]. Three types of faults in the collector area of a PTC plant are detected and isolated (K_{opt} , q and H_l faults) using only the information from the available sensors. The inputs are the variables that the concentrated parameter model uses and are obtained from the following elements: inlet and outlet temperature sensors, ambient temperature sensor, pyrheliometer, flowmeter, and geometric efficiency estimation. This section describes the different classification techniques that were tested as FDD modules.

3.1. Residuals from model

The concentrated parameter model from Eq. (7) can be used to estimate the outlet temperature of the collector. This measurement can be used as a residual when compared to the temperature read by the thermometer (in this case, simulated using the distributed parameter model of Eqs. (3) and (4)). The estimated and read temperatures are filtered with a low-pass filter and the resulting residual is given by Eq. (9), where $T_{out,est}$ is the filtered estimated outlet temperature

and $T_{out,mea}$ is the filtered measured outlet temperature. Whenever the residual r_T surpasses a given threshold, the system triggers an alarm.

$$r_T = \frac{|T_{out,est} - T_{out,mea}|}{T_{out,mea}} \quad (9)$$

In these types of systems, the faults are strongly coupled [20] and it is a hard task to isolate them, but using residuals helps to detect them. Based on the concentrated parameter model, the next step is the use of artificial neural networks to isolate faults with the same inputs as variables there are in the concentrated parameter model.

3.2. Artificial neural networks

An artificial neural network [29] is a function approximator that is worldwide used to solve linear and nonlinear problems in which the solution is too complex or costly. An ANN is composed of different nodes that solve internally a linear regression problem, and the combination of nodes results in the resolution of a nonlinear regression problem. One of the most known ANNs is the multilayer perceptron (MLP), where the nodes (also referred to as neurons) are disposed in layers. There are three types of layers: inputs layers, hidden layers that transform the data, and output layers.

The output of each neuron is generally transformed into an active/non-active state using activation functions, being the most common ones the relay, sigmoid and hyperbolic tangent [30,31]. The ANNs in this work contain hyperbolic tangent sigmoid functions in all layers, except for the output, where a softmax function is used to scale the data in the range [0, 1].

Each node is characterized by some weights that are obtained during the training of the ANN with backpropagation [32]. This algorithm adapts the weights by computing the error of each neuron iteratively from the output to the input. In this work, the scaled conjugate gradient algorithm [33] is used. To train the neural network, it performs a search along conjugate directions for fast, low memory convergence. The structure and the parameters of the neural networks are selected by trial and error in an iterative process until obtaining the desired results.

The inputs to the neural network are scaled in the range $[-1, +1]$ and divided into three subsets: training set (for adjusting the parameters), validation set (for readjusting hyperparameters) and test set (for estimating the performance with new data). The outputs are codified with one-hot encoding. There are four outputs: one for the nonfaulty case and the rest of each type of fault: $Y(k) = [y_0(k), y_1(k), y_2(k), y_3(k)]$ for each sample k , where $y_i \in \{\text{Faultless}, K_{opt} \text{ fault}, q \text{ fault}, H_f \text{ fault}\}$. The classifications are made by high values in the output corresponding to the winning class.

Initially, in the previous work [20], static neural networks were employed, using the values of $X(k) = [T_{in}(k), T_{out}(k), dT_{out}(k)/dT, T_a(k), I(k), q(k), n_o(k)]$ as inputs. These ANNs were tested by simulation in clear days, obtaining an accuracy over 87% and over 95% with a decoupling strategy. This strategy consisted of analyzing the flow rate dynamics or defocusing the first collector when necessary. When taking into account days with large clouds, the results worsen significantly to less than 50%. For this reason, [21] considers the dynamics of the system by introducing delays at the inputs and the accuracy augments to more than 73%.

3.3. Neural network combination

The previous work [21] did not take into account that each neural network performs better for one type of irradiance profile. This is because on days with many clouds it is more convenient to apply a different delay to the inputs than when the day has very little variation.

This paper proposes a methodology based on two levels of neural networks in cascade, as shown in Fig. 2. There are several dynamic ANNs trained with the same dataset, but each one with different inputs.

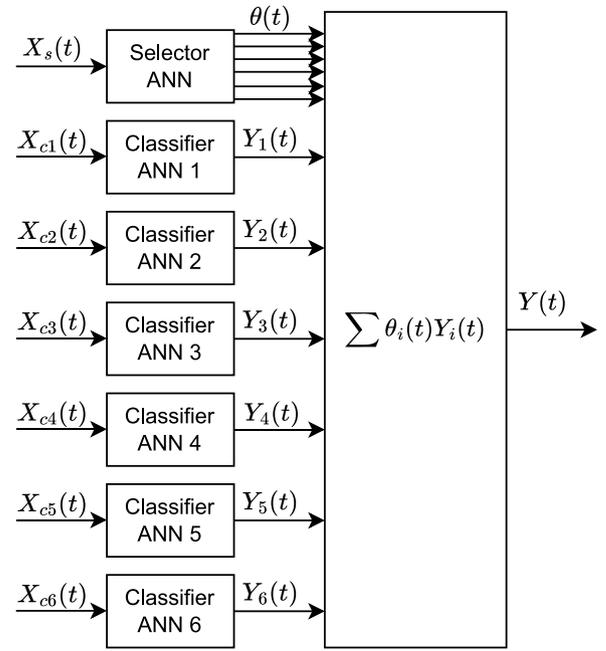


Fig. 2. Scheme of the cascade neural network with two levels. The score of each class is the weighted sum of the scores given by each classifier ANN, with the weights provided by the selector ANN.

An external ANN reads the current and past irradiance values and decides the best combination of neural networks to apply at each instant. The signal that corresponds to each fault is the weighted sum of the outputs of the classifier ANNs with the weights given by the selector ANNs.

The selected classification neural networks have the following inputs for every instant k , where $\bar{x}(k-j : k-i)$ is the mean value of x between $k-i$ and $k-j$ and $\bar{x}^w(k-j : k-i)$ is the weighted mean between $k-i$ and $k-j$.

- Inputs to classifier 1: $X_{c1}(k) = [T_{in}(k), T_{out}(k), dT_{out}(k)/dT, T_a(k), I(k), q(k), n_o(k)]$
- Inputs to classifier 2: $X_{c2}(k) = [\bar{T}_{in}(k-5 : k-1), \bar{T}_{in}(k-20 : k-6), \bar{T}_{out}(k-5 : k-1), \bar{T}_{out}(k-15 : k-6), \bar{T}_{out}(k-30 : k-16), T_a(k-5 : k-1), \bar{I}(k-5 : k-1), \bar{I}(k-20 : k-6), \bar{q}(k-3 : k-1), \bar{q}(k-10 : k-4), n_o(k)]$
- Inputs to classifier 3: $X_{c3}(k) = [T_{in}(k), T_{out}(k), T_{out}(k-1), T_{out}(k-15), T_{out}(k-30), T_a(k), I(k), q(k), q(k-1), q(k-15), n_o(k)]$
- Inputs to classifier 4: $X_{c4}(k) = [T_{in}(k), T_{out}(k), \bar{T}_{out}(k-5 : k-1), \bar{T}_{out}(k-15 : k-6), \bar{T}_{out}(k-30 : k-16), T_a(k), I(k), \bar{I}(k-5 : k-1), \bar{I}(k-20 : k-6), q(k), \bar{q}(k-3 : k-1), \bar{q}(k-10 : k-4), n_o(k)]$
- Inputs to classifier 5: $X_{c5}(k) = [T_{in}(k), \bar{T}_{in}(k-5 : k-1), \bar{T}_{in}(k-20 : k-6), T_{out}(k), \bar{T}_{out}(k-5 : k-1), \bar{T}_{out}(k-15 : k-6), \bar{T}_{out}(k-30 : k-16), T_a(k), I(k), \bar{I}(k-5 : k-1), \bar{I}(k-20 : k-6), q(k), \bar{q}(k-3 : k-1), \bar{q}(k-10 : k-4), n_o(k)]$
- Inputs to classifier 6: $X_{c6}(k) = [T_{in}(k), \bar{T}_{in}(k-5 : k-1), \bar{T}_{in}(k-20 : k-6), \bar{T}_{in}(k-20 : k-1), T_{out}(k), \bar{T}_{out}(k-5 : k-1), \bar{T}_{out}(k-15 : k-6), \bar{T}_{out}(k-30 : k-16), \bar{T}_{out}(k-30 : k-1), T_a(k), I(k), \bar{I}(k-5 : k-1), \bar{I}(k-20 : k-6), \bar{I}(k-20 : k-1), q(k), \bar{q}(k-3 : k-1), \bar{q}(k-10 : k-4), \bar{q}(k-10 : k-1), n_o(k)]$

With regard to the selector ANN, different experiments were carried out by combining the output of the classifier ANNs. Based on the previous experiments [21], different combinations were tested. The most relevant experiments are the following, selected on the basis of which obtained the best metric results in each type of test irradiance profile. The main idea was to combine the best properties of the static ANN (classifier 1) and other dynamic ANNs:

- Combination 1: classifiers 1 and 2.
- Combination 2: classifiers 1, 2, 3 and 4.
- Combination 3: classifiers 1 and 6.

The selector neural network receives present and past values of the irradiance and outputs a weight to apply at the output of each classifier neural network. This way, it tries to predict the reliability of every classifier ANN. The following criteria were followed:

- Type 1: The inputs are $X_{s1}(k) = [I(k), \sigma(I(k-2:k)), \sigma(I(k-7:k)), \sigma(I(k-22:k))]$. It has an output for each associated classifier. For the outputs, it takes the output of the classifier that corresponds to the real fault and divides it over the sum of all classifier outputs corresponding to that fault.
- Type 2: The inputs are $X_{s2}(k) = [I(k), \sigma(I(k-2:k)), \sigma(I(k-7:k)), \sigma(I(k-22:k))]$. It has an output for each associated classifier. The output is 1 whenever the output of the corresponding classifier gives more than 0.75 to the correct fault.
- Type 3: The inputs are $X_{s3}(k) = [I(k), \sigma(I(k-2:k)), \sigma(I(k-7:k)), \sigma(I(k-22:k))]$. It has an output for each associated classifier. The output is 1 if the corresponding classifier gave the highest value to the correct fault.
- Type 4: The inputs are $X_{s4}(k) = [I(k), I(k-2), I(k-7), I(k-22)]$. It has an output for each associated classifier. The output is 1 whenever the output of the corresponding classifier gives more than 0.75 to the correct fault.

The activation functions are tangent sigmoid functions, except for a softmax in the last layer. The neural networks are trained until achieving a minimum gradient of 10^{-6} , 6 validation checks, or 4000 epochs.

3.4. Classification trees

A classification and regression tree (CART) is an algorithm that predicts the values of a variable. It is based on a decision tree constructed from historical data [34]. CARTs are widely used in the literature. For example, Sánchez et al. [35] use a regression tree to estimate solar radiation in a solar plant, and Said et al. [36] optimizes the performance of a solar flat plate collector with bayesian optimization and regression trees.

A CART is a combination of if-else rules. First, the root node is defined, together with the explanatory variables of the model. Also, impurity, which is a measure of the ability of a variable to lead to the correct final prediction, must be selected. Each variable is divided into all its possible split points. If the data is continuous, different thresholds must be tested. For each split point, two child nodes are created, the impurity is calculated and the best branch is selected. The process is repeated until completing the tree.

In this work, different classification trees were created to detect and diagnose faults in the solar plant. The inputs to the CARTs are the same ones as to the ANNs. The following hyperparameters are optimized to minimize the cross-validation loss with bayesian optimization: maximum number of splits, minimum number of leaf node observations, the split criterion and the number of predictors to select at random for each split. For all the classifiers trained, the best slip criterion was the deviance or cross-entropy, given by Eq. (10).

$$dev = - \sum_i p(i) \log_2 p(i) \quad (10)$$

where $p(i)$ is the observed fraction of classes with class i that reach the analyzed node.

3.5. Random forests

Random forests (RF) is another technique based on decision trees [37]. It combines multiple individual CARTs to obtain different non-correlated models. Each tree votes for a classification for the incoming data and the most voted class is selected. The trees are constructed randomly and each one is trained with a random subset of the data (with repetition) and with a random subset of features. Some recent examples are the works by Chai and Zhao [38], which apply them for fault classification in chemical processes, and Dhibi et al. [39] use them for fault classification in PV systems.

The training process is similar to that of the classification tree, with the difference of the bootstrap: first, a random subset is selected. Then, a few random variables are taken to select the one that separates the data further. That one is used as a root. Then, the process is repeated for the rest of the variables, creating a new branch at the time. Once a tree is obtained, a new subset is created and all the steps are repeated. This is performed until obtaining hundreds of trees, which is called a forest.

The process followed for training the RF is analogous to that for the CARTs, making tests with different numbers of trees. In this case, the split criterion was the Gini's diversity index, given by Eq. (11).

$$gdi = 1 - \sum_i p^2(i) \quad (11)$$

3.6. Radial basis functions

The radial basis function (RBF) network is a type of feedforward neural network. It is a function whose value depends on the distance to a center, and the transfer functions are different from those of the multilayer perceptron, usually with the normal distribution. Their value depends on the distances between a vector associated with each neuron and the input vectors. The output neurons have purelin transfer functions. These neural networks have one hidden layer with a high number of neurons. Chouhan et al. [40] use RBF to identify and classify plant leaf diseases, and Hussain et al. [11] apply them for fault detection.

In an iterative process, a neuron is added to the network until reaching the desired mean squared error goal. In this type of neural network, the input to the radial basis neuron is the distance between its weight vector w and the input vector x multiplied by the bias b , as in Eq. (12).

$$a = \exp(-(\|w - x\|b)^2) \quad (12)$$

The spread must be selected, which measures how much the neurons span. It is given by Eq. (13).

$$spread = \frac{b}{-\log(0.5)} \quad (13)$$

As these neural networks compute the distance between input vectors and weight vectors, to speed up the learning process, the training is performed on a subset of the original dataset. Two tests were performed: a random selection and a set based on the self-organizing map (SOM) as a data reduction technique.

3.6.1. Self organizing maps

The self-organizing map [41] is a type of neural network that reorganizes high-dimensional data into a low-dimensional matrix. Each neuron constitutes a node of the SOM topology and has an associated weight. The SOM weights constitute new prototypes that represent the input space.

In this work, a SOM is used for generating a small dataset to train the RBF. The SOMs were trained in batches of 10 000 data and using hexagonal and squared topologies with a maximum of 200 epochs.

4. Results

This section presents the results obtained with the residuals and each one of the classifiers. The training of the classifiers and the simulations were carried out in MATLAB® R2020b with Intel® Core™ i7-9700F CPU at 3.00 GHz and 16 GB RAM using the distributed parameter model. The classifiers for FDD are trained with the Deep Learning Toolbox and Statistics and Machine Learning Toolbox. To compare the results, the following measures are obtained:

- Accuracy: the rate of correct classes over true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \quad (14)$$

- Precision: the rate of correct positives over all positive-assigned instances.

$$Pre = \frac{TP}{TP + FP} \quad (15)$$

- Recall: the rate of correct positives over all positive instances.

$$Rec = \frac{TP}{TP + FN} \quad (16)$$

- F1-score: the harmonic mean of precision and recall.

$$F1 = 2 \cdot \frac{Rec \cdot Pre}{Rec + Pre} \quad (17)$$

To train the neural network, a dataset with different clouds was used (except for the static ANNs, which were trained with static data [20]). The dataset contains 13 462 h of simulations between hours 10:00 and 17:59 h, a total of 1 259 256 instances, without control and controlling with temperature references between 220 °C and 300 °C. The faults values are as follows: optical efficiency faults in the range [0.1, 0.9], negative and positive additive flow rate faults in the range [0.1, 0.5], and thermal loss faults in the range [1.1, 2]. For each fault type, eight different fault values homogeneously selected inside the range were used to obtain a comprehensive sampling. The number of data of each type of fault has been selected to balance the data so that each class has the same amount of data. Although the dataset is obtained from simulations, real values of irradiance, ambient temperature and inlet temperature were used. The ambient temperature range is [17, 44] °C and the inlet temperature range is [44, 218] °C (although the lowest inlet temperature values are later removed, since they correspond to start-ups). The irradiance values obtained by the pyrheliometer are in the range [0, 988] W/m². The information from the sensors is taken every 39 s and the data is subsequently interpolated. Measurement noise is not taken into account, as it is considered an existing disturbance in the real system that the neural networks must cope with when they are implemented. If there were missing data during operation, an estimator could be used [42]. The training profiles are those of Fig. 3. To obtain the accuracies in real time, the three new profiles of Fig. 4 were used and different simulations were performed with faults (in the same range as with the training dataset, but with different magnitudes) appearing at different times. The tests profiles correspond to a sunny day (profile 1), a day with many clouds (profile 4), and a medium-irradiance day with one medium cloud (profile 9).

By using the concentrated parameter model to detect faults in different simulations, the F1-scores (F1) and fault accuracies (ACC) have been obtained. A low-pass filter was used at the output of the temperature estimator. A time constant of 30 min in the filter and a threshold of 1% in the residuals were set after different tests with the training dataset, where an F1-score of 80.85% and an accuracy of 93.35% were obtained, as shown in Table 4. Once selected the time constant and threshold, the results for each validation profile are shown in Table 5, together with the average time to compute the residual. The overall F1-score and accuracy are both over 88%. Analyzing the results for each profile individually, in cloudy days reductions of less than 2%

Table 4

Training results of the detection with the concentrated parameter model for different thresholds and time constants.

Threshold (%)	Filter (min)	F1 (%)	ACC (%)
2	60	5.78	80.02
1	60	70.52	90.63
0.5	60	43.69	52.45
2	30	15.64	81.16
1	30	80.85	93.35
0.5	30	42.34	46.75
2	10	50.16	85.32
1	10	65.33	82.54
0.5	10	37.15	33.03

Table 5

Results of the detection with the concentrated parameter model.

Profile	F1 (%)	ACC (%)	Time (s)
1,4,9	88.06	89.29	1.091·10 ⁻⁵
1	87.72	89.06	1.072·10 ⁻⁵
4	86.77	88.19	1.202·10 ⁻⁵
9	89.66	90.63	9.982·10 ⁻⁶

are observed, even reaching an accuracy of 90% for profile 9. Profile 4 is the one that obtains slightly lower results since it is the one with the largest amount of clouds.

The training parameters and results of the classification neural networks are detailed in [20,21]. To train the selector neural networks, the dataset was divided into training (70%), validation (15%) and test subsets (15%). Note that the test subset is different from the test irradiance profiles, as the first is a subset of the training dataset and the latter contains data from completely different days. For each combination of classifiers, different architectures of selector ANN were tested by trial and error. Table 6 gathers the training times and accuracies of the best architecture for each selector ANN, described with the number of neurons in each layer separated by a dash. The highest accuracies for every combination are obtained with the first selector type, which uses the weighted sum of the classifiers. The best combination takes classifiers 1, 2, 3 and 4.

To train the CARTs, the datasets are divided into a training subset (85%) and a test subset (15%). The inputs to the trees are the same as to the classifier ANNs described earlier with coinciding identifiers (tree 1 has the same inputs as ANN classifier 1). The resulting trees use a minimum leaf size of 1 and the deviance criterion. Table 7 shows the training results of the most relevant trees. The dynamic trees (trees 2–6) have similar values of accuracy, whereas the accuracy of the static one (tree 1) is expectedly lower. To apply the trees in real-time, the output is codified with one-hot encoding to obtain four variables in the range [0,1], each one corresponding to a type of situation (no fault, K_{opt} fault, q fault and H_l fault).

Analogously to the training of classification trees, random forests are trained with a minimum leaf size of 1 and a maximum size equal to the data size. The split criterion is the mean squared error. Table 8 shows the training results for the different forests trained. Forests of 25, 50, 100 and 200 trees are used. All trees obtain similar accuracies greater than 99%, which could mean that the models are overfitting. Although in random forests many trees are trained at the same time, training times are shorter than with classification trees, because in this case no parameter optimization is performed. Even so, the times and accuracies increase as the number of trees in each forest increases.

To train the RBF networks, two tests are carried out: using random subsets of 10 000 obtained from the training data, and training a SOM to represent the training data. Table 9 gathers the parameters used to train the different SOMs. The column Dim. indicates the number of nodes in each dimension (only bidimensional SOMs are trained), the column Neig. refers to the initial number of neighbors used to adapt the weights, and the topology can be hexagonal (Hex) or squared (Grid).

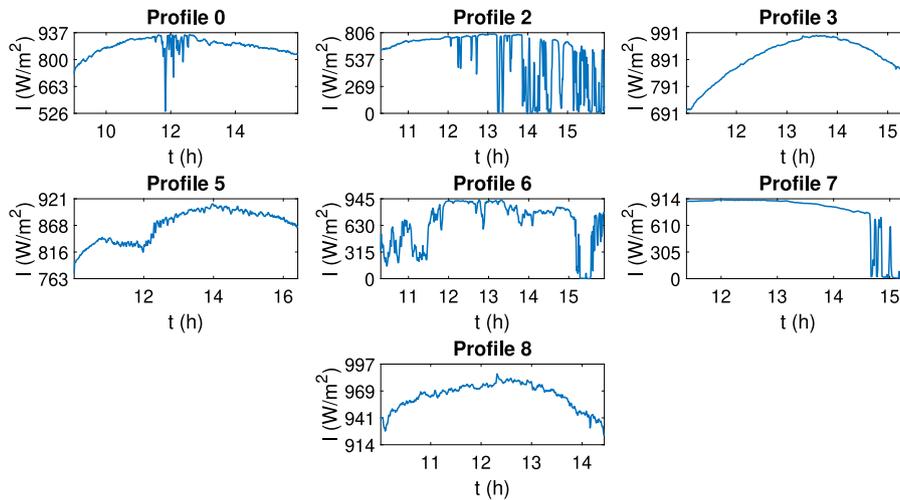


Fig. 3. Irradiance profiles used for training.

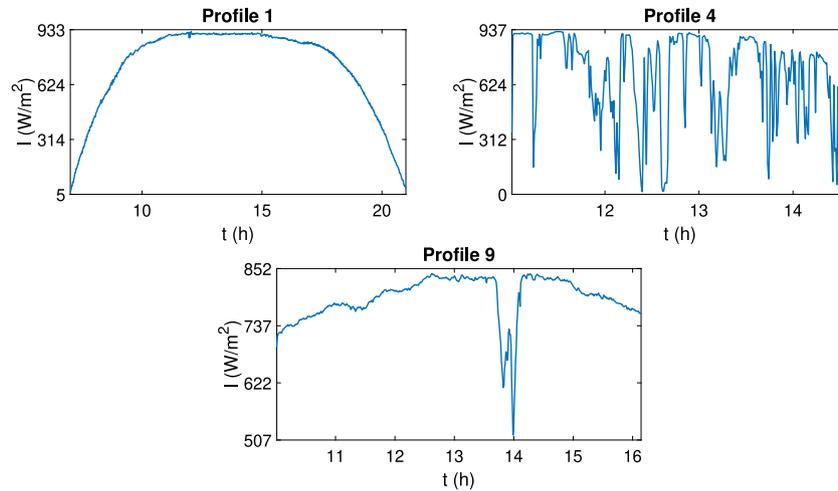


Fig. 4. Irradiance profiles used for testing.

Table 6
Training accuracies of the selector neural networks.

Combination	Type	Neurons	ACC train (%)	ACC valid. (%)	ACC test (%)	Tr. time (min)
1	1	400-200	73.76	73.80	74.20	15.84
1	2	400-200	69.84	69.81	70.19	19.88
1	3	400-200	68.10	68.13	68.23	18.38
1	4	80-40-20	61.17	61.39	61.08	21.00
2	1	200-100-40	75.00	74.85	75.17	44.66
2	2	200-100-40	73.99	73.98	74.07	67.24
2	3	80-40-20	45.08	45.25	45.12	22.24
2	4	200-100-40	62.82	62.66	62.54	130.02
3	1	200-100-40	73.19	73.48	73.15	4.62
3	2	80-40-20	72.92	72.10	72.82	8.51
3	3	400-200	68.21	68.33	68.35	15.43
3	4	200-100-40	72.99	72.97	73.04	7.83

The datasets for the RBF networks are divided into training (85%) and test (15%) subsets. The goal is set to 0.001. Different values of spread were tested. The training parameters and results are shown in Table 10. Low accuracies are obtained for all the classifiers.

All classifiers are applied following the same methodology. The FDD system is constantly reading data from the sensors with a sampling time of 39 s (the same as the controller). All the time, the system is performing classifications (which are filtered to avoid misclassifications and reject disturbances) and only alarms when the established limits

are exceeded. The alarm is only turned off at the end of each day, when the failure is assumed to be repaired.

For each of the classifiers, different tests are performed. 1152 one-day simulations are carried out using the irradiance profiles of Fig. 4, reference temperatures between 200 °C and 300 °C, and faults occurring at different times. The accuracies, F1-scores and mean computation times of the best classifiers of each type are presented in Table 11. To analyze their performance in different situations, the results are shown for the entire set of simulations and for each validation profile

Table 7
Training accuracies of the classification trees.

Tree	Data	Max. splits	ACC train (%)	ACC test (%)	Tr. time (min)
1	1	192 343	96.96	94.43	16.45
2	2	240 524	98.25	96.82	13.92
3	3	230 011	98.28	96.61	14.53
4	4	1 131 933	98.45	97.13	14.39
5	5	319 493	98.44	97.13	11.14
6	6	610 629	98.59	97.41	11.13

Table 8
Training accuracies of the random forests.

Forest	Data	Trees	ACC train (%)	ACC test (%)	Tr. time (min)
1	1	25	99.79	92.40	3.04
2	1	50	99.88	99.69	3.27
3	1	100	99.91	99.69	6.61
4	1	200	99.92	99.69	13.08
5	2	25	99.97	95.36	2.78
6	2	50	99.99	99.99	2.96
7	2	100	100.0	99.99	5.91
8	2	200	100.0	99.99	11.55
9	3	25	99.98	95.64	2.01
10	3	50	100.0	99.99	2.88
11	3	100	100.0	99.99	5.73
12	3	200	100.0	99.99	11.46
13	4	25	99.97	95.64	2.26
14	4	50	100.0	99.99	2.85
15	4	100	100.0	99.99	5.66
16	4	200	100.0	99.99	11.29
17	5	25	99.96	95.33	2.14
18	5	50	99.99	99.98	2.80
19	5	100	100.0	99.98	5.55
20	5	200	100.0	99.98	11.16
21	6	25	99.97	95.72	1.72
22	6	50	100.0	99.99	3.02
23	6	100	100.0	99.99	6.00
24	6	200	100.0	99.99	11.98

Table 9
Parameters of the SOMs.

SOM	Data	Dim.	Neig.	Topology	Tr. time (h)
1	1	32,32	3	Hex	12.90
2	1	32,32	3	Grid	12.61
3	1	32,32	1	Hex	13.17
4	1	32,32	1	Grid	3.72
5	1	100,100	3	Hex	55.09
6	1	100,100	3	Grid	55.04
7	1	100,100	1	Hex	55.12
8	1	100,100	1	Grid	55.35
9	2	100,100	3	Grid	9.79
10	3	100,100	3	Grid	9.77
11	4	100,100	3	Grid	10.34
12	5	100,100	3	Grid	10.85
13	6	100,100	3	Grid	12.01

individually. All classifiers are trained with the same data except for the static ANN, which only used sunny days. For the neural network combination, the classifications are filtered with a time constant of 60 min and the alarm is triggered when an output exceeds 0.5. The trees and forests use a time constant of 90 min and a threshold of 0.5 and, finally, the RBF networks use a time constant of 60 min and a limit of 1.

The accuracies obtained in the simulations are much lower than the ones obtained when training the classifier. This is because the training was offline and only the instantaneous value of the fault was taken (which is not admissible in the actual application because it is preferable to take longer to detect a failure than to get many false alarms), while in simulation the faults are read at the end of each day. In addition, even if the classifications are filtered, it is very difficult to decouple the failures, especially on cloudy days, when the system is very unstable. In general, the classifiers obtain higher accuracies on a

Table 10
Training accuracies of the RBFs.

RBF	Data	Spread	ACC train (%)	ACC test (%)	Tr. time (min)
1	1	10	56.02	53.87	21.71
2	1	25	53.80	51.09	191.08
3	1	50	51.51	48.18	20.56
4	1	75	53.03	50.92	17.98
5	1	100	53.57	51.82	17.90
6	1	150	51.58	50.05	17.99
7	2	100	53.13	51.05	20.53
8	3	100	52.01	51.01	19.41
9	4	100	52.54	51.06	21.01
10	5	100	54.11	52.91	23.47
11	6	100	54.01	52.89	24.99
12 (SOM 1)	1	10	47.28	43.24	0.77
13 (SOM 1)	1	25	46.63	41.30	0.66
14 (SOM 1)	1	50	45.01	41.28	0.64
15 (SOM 2)	1	10	47.25	42.41	0.61
16 (SOM 2)	1	25	46.86	41.94	0.65
17 (SOM 2)	1	50	45.28	42.30	0.63
18 (SOM 3)	1	10	25.64	28.49	0.08
19 (SOM 3)	1	25	26.13	29.34	0.06
20 (SOM 3)	1	50	25.64	28.49	0.08
21 (SOM 4)	1	10	42.90	41.67	0.61
22 (SOM 4)	1	25	40.52	42.23	0.65
23 (SOM 4)	1	50	38.37	40.76	0.63
24 (SOM 5)	1	75	52.22	46.30	19.88
25 (SOM 5)	1	100	52.17	46.25	18.42
26 (SOM 5)	1	150	50.85	45.19	18.00
27 (SOM 6)	1	75	50.85	44.91	18.07
28 (SOM 6)	1	100	50.87	44.96	17.94
29 (SOM 6)	1	150	50.87	44.95	17.94
30 (SOM 7)	1	75	42.29	41.60	18.17
31 (SOM 7)	1	100	42.18	41.52	17.95
32 (SOM 7)	1	150	41.84	41.24	17.99
33 (SOM 8)	1	75	48.56	48.61	18.26
34 (SOM 8)	1	100	48.56	48.61	19.04
35 (SOM 8)	1	150	48.14	48.23	18.45
36 (SOM 9)	2	75	51.51	48.89	2.25
37 (SOM 10)	3	75	53.53	53.00	2.18
38 (SOM 11)	4	75	50.03	46.83	2.24
30 (SOM 12)	5	75	52.36	50.63	2.30
40 (SOM 13)	6	75	51.41	49.52	2.33

sunny day without clouds (profile 1) since the system is more stable. The methodologies that perform best are the dynamic ANN and the combination of ANNs. Although dynamic ANN achieves slightly higher accuracy, the ANN combination allows to improve the results on cloudy days, achieving F1-scores of over 50% for the three profiles.

As an example, Fig. 5 shows the temperatures, irradiance and flow rate obtained with profile 9 and a reference temperature of 250 °C when causing a 0.7 fault in the optical efficiency at 12:00. The dashed line shows the output temperature that would be obtained if there was no fault. The FDD results with the selected ANN combination are shown in Fig. 6, where the alarm is triggered before 13:00.

5. Discussion

A model-based fault detector and eight different classifiers have been compared for fault detection and isolation: a static neural network, a static neural network with a decoupling strategy, a dynamic neural network, a 2-layer combination of dynamic neural networks, a classification tree, a random forest, a radial basis function network, and a combination of radial basis function network with a self-organizing map. The static ANN is trained on sunny days, while the rest are trained with real irradiance profiles corresponding to days with different types of clouds. From the results in Table 11, the following observations can be extracted:

- The concentrated parameter model obtains accuracies and F1-scores near 90% for all irradiance profiles and requires a negligible time compared to the sampling time.

Table 11
Simulation results of the selected classifiers.

Profile	Classifier	F1-score (%)				Acc (%)	Time (s)
		Faultless	K_{opt} fault	q fault	H_i fault		
1,4,9	Static ANN [20]	48.82	55.18	47.34	47.58	49.57	$2.19 \cdot 10^{-3}$
1,4,9	Static ANN + decoupling [20]	49.61	51.26	47.58	53.33	50.35	$7.48 \cdot 10^{-4}$
1,4,9	Dynamic ANN [21]	88.21	82.56	65.53	55.17	73.35	$2.24 \cdot 10^{-4}$
1,4,9	ANN combination 2, 200-100-40, type 1	77.24	84.71	62.15	69.65	73.09	$7.58 \cdot 10^{-4}$
1,4,9	Tree 4	44.41	76.95	55.52	38.21	53.39	$5.82 \cdot 10^{-3}$
1,4,9	Forest 1	45.63	76.55	33.86	33.86	51.13	1.25
1,4,9	RBF 5	24.44	80.36	42.56	53.69	52.52	$4.88 \cdot 10^{-3}$
1,4,9	RBF 30 (SOM 12)	37.69	77.56	41.61	32.52	48.44	$5.03 \cdot 10^{-3}$
1	Static ANN	98.41	88.77	76.57	88.48	88.28	$2.30 \cdot 10^{-4}$
1	Static ANN + decoupling	98.46	91.8	75.43	86.51	88.28	$2.87 \cdot 10^{-4}$
1	Dynamic ANN	85.11	81.21	62.20	55.26	71.88	$2.50 \cdot 10^{-4}$
1	ANN combination 2, 200-100-40, type 1	71.14	84.26	68.42	68.89	73.18	$8.14 \cdot 10^{-4}$
1	Tree 4	60.65	80.90	59.70	58.97	64.58	$5.88 \cdot 10^{-3}$
1	Forest 1	65.79	82.98	66.67	59.76	68.23	1.24
1	RBF 5	46.89	83.24	22.22	49.62	53.13	$5.09 \cdot 10^{-3}$
1	RBF 30 (SOM 12)	50.26	80.45	27.03	–	47.66	$5.29 \cdot 10^{-3}$
4	Static ANN	–	49.11	40.00	10.81	34.11	$2.11 \cdot 10^{-4}$
4	Static ANN + decoupling	–	38.42	13.59	29.63	28.13	$1.48 \cdot 10^{-3}$
4	Dynamic ANN	84.97	82.21	63.60	38.76	68.75	$2.02 \cdot 10^{-4}$
4	ANN combination 2, 200-100-40, type 1	59.85	80.70	50.94	74.19	66.67	$7.01 \cdot 10^{-4}$
4	Tree 4	27.64	72.36	50.83	35.92	48.70	$5.54 \cdot 10^{-3}$
4	Forest 1	–	68.39	42.76	–	37.24	1.26
4	RBF 5	–	73.14	48.83	52.11	49.48	$4.75 \cdot 10^{-3}$
4	RBF 30 (SOM 12)	–	71.05	42.9	–	40.36	$4.74 \cdot 10^{-3}$
9	Static ANN	–	9.90	40.77	1.90	26.30	$2.16 \cdot 10^{-4}$
9	Static ANN + decoupling	–	28.57	44.34	33.82	34.64	$4.77 \cdot 10^{-4}$
9	Dynamic ANN	94.12	84.26	73.55	65.09	79.43	$2.20 \cdot 10^{-4}$
9	ANN combination 2, 200-100-40, type 1	93.20	89.14	70.66	65.45	79.43	$7.60 \cdot 10^{-4}$
9	Tree 4	43.03	78.26	57.32	1.67	46.88	$5.96 \cdot 10^{-3}$
9	Forest 1	48.98	76.92	45.16	–	47.92	1.25
9	RBF 5	–	84.00	46.67	59.54	54.95	$4.82 \cdot 10^{-3}$
9	RBF 30 (SOM 12)	27.03	82.41	46.47	61.75	57.29	$5.05 \cdot 10^{-3}$

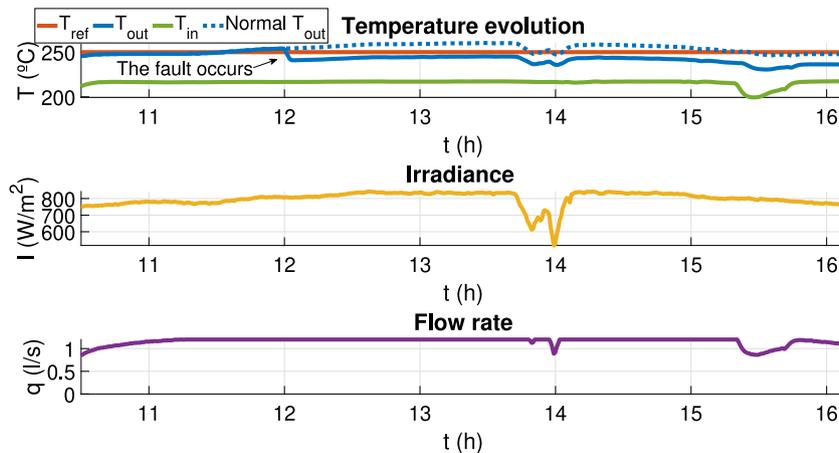


Fig. 5. Temperatures, irradiance and flow rate evolution from the experiment with test profile 9 and a fault of 0.7 in the efficiency of the collectors after hour 12:00 using the selected cascade ANN (ANN combination 2200-100-40, type 1).

- As the static ANN – with and without decoupling strategy – was only trained with sunny days, it is the one that obtains higher accuracy and F1-scores for profile 1. Moreover, this shows that in days with low dynamics, it is not necessary to use past data to isolate the faults. However, the performance worsens when testing it on cloudy days (profiles 4 and 9) and it confuses clouds with some type of failure.
- The results with the decoupling strategy are better than with only the ANN, but the decoupling is also dependent on the weather conditions since the dynamic of the irradiance produces changes in the residence time of the fluid in the collectors and false positives.
- The dynamic ANN is the one with the highest accuracy (73.35%), but the F1-score for the H_i fault is lower than 40% with profile 4. This is the most challenging day, as it contains large and fast clouds during the whole simulation. For a sunny day (profile 1), the accuracy is lower than with the previous classifiers, but the improvement is substantial in general terms. This highlights the difficulty of detecting faults when clouds are present and the need to take past data into account.
- Although the total accuracy with the ANN combination is slightly lower than with only one dynamic network, this decrease is practically negligible, and it succeeds in obtaining no F1-scores

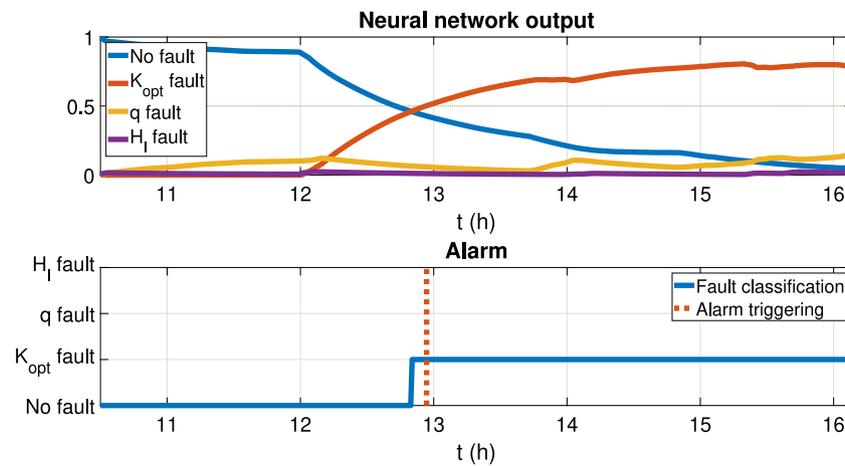


Fig. 6. Classifications for the experiment with test profile 9 and a fault of 0.7 in the efficiency of the collectors after hour 12:00 using the selected cascade ANN (ANN combination 2200-100-40, type 1).

below 50% for any profile. This makes this method the best option if the sky is not expected to be completely sunny.

- The tree shown in the table was the one that performed better in training and also in the online simulations. The results are much worse than with dynamic ANNs because it is much more difficult to represent nonlinear relationships with these methods, and increasing the number of trees can lead to overfitting. However, a tree with dynamic data obtains higher accuracy than a static ANN.
- The results with the random forests are similar to those with the trees since the methodology is an extension of the CART. However, in days with clouds, it loses the ability to distinguish days without clouds and its capabilities to isolate H_i faults worsen.
- The RBF networks are not good for representing nonlinear data too, and there is not much difference in using dynamic data, obtaining the best online results with RBF 5, which used static information. The results are similar to those with the trees and forests. The results do not improve when combining it with a SOM. The best results are obtained with dynamic data and a medium spread, but the obtained accuracy is the lowest one.
- Regarding computation times, all of them are suitable for the sampling time of 39 s, being the random forest the most demanding, as it contains 50 models in 1.

6. Conclusions

This work has analyzed different approaches for fault detection and isolation in parabolic through collectors. First, the use of the concentrated parameter model is proposed to generate residuals and detect faults in the plant. Then, eight different classifiers are compared to detect and isolate three types of faults: optical efficiency, flow rate, and thermal losses.

The challenge in the application of FDD techniques in thermal plants lies in two main facts: first, the system variables are highly coupled in the collector area, and then, the system performance depends strongly on the weather conditions. This study highlights the difficulty of decoupling faults, especially on cloudy days, when most classifiers obtain accuracies under 50%. The model-based detector obtained accuracies near 90% for sunny and cloudy days. Among the classifiers tested, the dynamic neural networks are the ones that provide the best results, with accuracies over 70%. Moreover, the combination of ANNs in two layers, one for classifying the fault and the other for combining the information given by several classifiers, allows not only to obtain high accuracy but to augment the true positives for each type, with all the F1-scores over 50%.

Given the great FDD performance of the static ANN with the irradiance profile 1, a possible improvement of the results might come from the use of this ANN on days when weather forecasts predict a complete absence of medium and large clouds. For all other weather conditions, the best option is to apply the combination of the selector network with the classifier network.

Future lines of research come from the application of the methodology to large-scale plants, extending the FDD system to all loops in the plant. In addition, it is intended to extend fault detection to the rest of the solar field subsystems and to analyze the effect of defocusing the collectors on false alarms in the optical efficiency.

CRediT authorship contribution statement

Sara Ruiz-Moreno: Conceptualization, Methodology, Software, Writing – original draft. **Antonio J. Gallego:** Conceptualization, Methodology, Software, Writing – review & editing. **Adolfo J. Sanchez:** Conceptualization, Methodology, Writing – review & editing. **Eduardo F. Camacho:** Supervision, Writing – review & editing, Project administration, Funding acquisition.

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.

Acknowledgments

This research has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (OCONTSOLAR, grant agreement No 789051) and the Spanish Ministry of Science, Innovation, and Universities (Grant FPU20/01958).

References

- [1] S. Suman, Hybrid nuclear-renewable energy systems: A review, *J. Clean. Prod.* 181 (2018) 166–177, <http://dx.doi.org/10.1016/j.jclepro.2018.01.262>.
- [2] A. Azam, M. Rafiq, M. Shafique, H. Zhang, J. Yuan, Analyzing the effect of natural gas, nuclear energy and renewable energy on GDP and carbon emissions: A multi-variate panel data analysis, *Energy* 219 (2021) 119592, <http://dx.doi.org/10.1016/j.energy.2020.119592>.
- [3] W. Ajbar, A. Parrales, A. Huicochea, J. Hernández, Different ways to improve parabolic trough solar collectors' performance over the last four decades and their applications: A comprehensive review, *Renew. Sustain. Energy Rev.* 156 (2022) 111–947, <http://dx.doi.org/10.1016/j.rser.2021.111947>.

- [4] E.F. Camacho, M. Berenguel, F.R. Rubio, *Advanced control of solar plants*, Springer, 1997, <http://dx.doi.org/10.1007/978-1-4471-0981-5>.
- [5] J. Marquez, A. Zafra-Cabeza, C. Bordons, M.A. Ridaio, A fault detection and reconfiguration approach for MPC-based energy management in an experimental microgrid, *Control Eng. Pract.* 107 (2021) 104695, <http://dx.doi.org/10.1016/j.conengprac.2020.104695>.
- [6] D. Miljković, *Fault detection methods: A literature survey*, in: 2011 Proceedings of the 34th International Convention MIPRO, 2011, pp. 750–755.
- [7] C. Janiesch, P. Zschech, K. Heinrich, Machine learning and deep learning, *Electr. Mark.* 31 (3) (2021) 685–695, <http://dx.doi.org/10.1007/s12525-021-00475-2>.
- [8] M. Ahmadipour, M.M. Othman, R. Bo, Z. Salam, H.M. Ridha, K. Hasan, A novel microgrid fault detection and classification method using maximal overlap discrete wavelet packet transform and an augmented Lagrangian particle swarm optimization-support vector machine, *Energy Rep.* 8 (2022) 4854–4870, <http://dx.doi.org/10.1016/j.egy.2022.03.174>.
- [9] D. Brown, J. Fritz, M. Strube, Simulation-trained AI-system for two-stage fault detection and diagnosis of rolling bearings in industrial applications, in: 2021 International Conference on Control, Automation and Diagnosis, ICCAD, IEEE, 2021, pp. 1–6, <http://dx.doi.org/10.1109/ICCAD52417.2021.9638761>.
- [10] F. Qu, J. Liu, X. Liu, L. Jiang, A multi-fault detection method with improved triplet loss based on hard sample mining, *IEEE Trans. Sustain. Energy* 12 (1) (2021) 127–137, <http://dx.doi.org/10.1109/TSTE.2020.2985217>.
- [11] M. Hussain, M. Dhimish, S. Titarenko, P. Mather, Artificial neural network based photovoltaic fault detection algorithm integrating two bi-directional input parameters, *Renew. Energy* 155 (2020) 1272–1292, <http://dx.doi.org/10.1016/j.renene.2020.04.023>.
- [12] E. Bernardi, M.M. Morato, P.R. Mendes, J.E. Normey-Rico, E.J. Adam, Fault-tolerant energy management for an industrial microgrid: A compact optimization method, *Int. J. Electr. Power Energy Syst.* 124 (2021) 106–342, <http://dx.doi.org/10.1016/j.ijepes.2020.106342>.
- [13] J. Marquez, A. Zafra-Cabeza, C. Bordons, Fault quantification and mitigation method for energy management in microgrids using MPC reconfiguration, *IFAC-PapersOnLine* 55 (6) (2022) 575–582, <http://dx.doi.org/10.1016/j.ifacol.2022.07.190>.
- [14] M. Georgii, C. Schmelzer, H. Braas, J. Orozaliev, K. Vajen, A flexible software framework for self-adapting algorithm-based fault detection and diagnosis in solar heating systems, in: *ISES Solar World Congress 2017*, 2017, <http://dx.doi.org/10.18086/swc.2017.19.05>.
- [15] G. Faure, M. Vallée, T. Tran-Quoc, N. Lamaison, C. Paulus, *A methodology to analyse fault effect on large solar thermal system behaviour*, 2018.
- [16] C. Correa-Jullian, J.M. Cardemil, E. López Droguett, M. Behzad, Assessment of deep learning techniques for prognosis of solar thermal systems, *Renew. Energy* 145 (2020) 2178–2191, <http://dx.doi.org/10.1016/j.renene.2019.07.100>.
- [17] T. Zahra, L.M. Mourad, A.H. Ahmed, Robust fuzzy sliding mode observer for faults detection in solar power plant application, *Instrum., Mesure., Métrol.* 19 (4) (2020) 281–287, <http://dx.doi.org/10.18280/i2-m.190405>.
- [18] S. Jiang, M. Lian, C. Lu, S. Ruan, Z. Wang, B. Chen, SVM-DS fusion based soft fault detection and diagnosis in solar water heaters, *Energy Explor. Exploit.* 37 (3) (2019) 1125–1146, <http://dx.doi.org/10.1177/0144598718816604>.
- [19] X. Gao, Z. Su, Active disturbance rejection control for a parabolic trough solar field, in: 2021 4th International Conference on Control and Computer Vision, 2021, pp. 159–164, <http://dx.doi.org/10.1145/3484274.3484299>.
- [20] S. Ruiz-Moreno, A.J. Sanchez, A.J. Gallego, E.F. Camacho, A deep learning-based strategy for fault detection and isolation in parabolic-trough collectors, *Renew. Energy* 186 (2022) 691–703, <http://dx.doi.org/10.1016/j.renene.2022.01.029>, URL <https://www.sciencedirect.com/science/article/pii/S0960148122000283>.
- [21] S. Ruiz-Moreno, A.J. Gallego, A.J. Sanchez, E.F. Camacho, Deep learning-based fault detection and isolation in solar plants for highly dynamic days, in: 2022 International Conference on Control, Automation and Diagnosis, ICCAD, 2022, pp. 1–6, <http://dx.doi.org/10.1109/ICCAD55197.2022.9853987>.
- [22] J.M. Escaño, A.J. Gallego, A.J. Sánchez, L.J. Yebra, E.F. Camacho, Nonlinear fuzzy model predictive control of the TCP-100 parabolic trough plant, in: 19th World Congress of the International Fuzzy Systems Association (IFSA), 12th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT), and 11th International Summer School on Aggregation Operators, AGOP, Atlantis Press, 2021, pp. 235–241, <http://dx.doi.org/10.2991/asum.k.210827.032>.
- [23] R. Carmona, *Analisis, modelado y control de un campo de colectores solares distribuidos con sistema de seguimiento en un eje* (Ph.D. thesis), University of Seville, Spain, 1985.
- [24] D.Y. Goswami, F. Kreith, J.F. Kreider, *Principles of Solar Engineering*, CRC Press, 2000, pp. 912–918, <http://dx.doi.org/10.1201/b18119>.
- [25] A.J. Gallego, L.J. Yebra, E.F. Camacho, A.J. Sánchez, Mathematical modeling of the parabolic trough collector field of the TCP-100 research plant, in: Proceedings of EUROSIM 2016, SIMS 2016, Vol. 142, Linköping University Electronic Press, 2018, pp. 912–918, <http://dx.doi.org/10.3384/ecp17142912>.
- [26] A. Azouzoute, A.A. Merrouni, M. Garoum, E.G. Bennouna, A. Ghennioui, M.E. Ydrissi, The impact of optical soiling losses on the electrical production of CSP power plant, in: AIP Conference Proceedings, Vol. 2123, AIP Publishing LLC, 2019, 020090, <http://dx.doi.org/10.1063/1.5117017>.
- [27] A. Azouzoute, A.A. Merrouni, M. Garoum, E.G. Bennouna, Soiling loss of solar glass and mirror samples in the region with arid climate, *Energy Rep.* 6 (2020) 693–698, <http://dx.doi.org/10.1016/j.egy.2019.09.051>, The 6th International Conference on Energy and Environment Research - Energy and environment: challenges towards circular economy.
- [28] A.J. Sánchez, A.J. Gallego, J.M. Escaño, E.F. Camacho, Temperature homogenization of a solar trough field for performance improvement, *Sol. Energy* 165 (2018) 1–9, <http://dx.doi.org/10.1016/j.solener.2018.03.001>.
- [29] O.I. Abiodun, A. Jantan, A.E. Omolara, K.V. Dada, N.A. Mohamed, H. Arshad, State-of-the-art in artificial neural network applications: A survey, *Heliyon* 4 (11) (2018) e00938, <http://dx.doi.org/10.1016/j.heliyon.2018.e00938>.
- [30] P. Ramachandran, B. Zoph, Q.V. Le, Searching for activation functions, 2017, <http://dx.doi.org/10.48550/arXiv.1710.05941>, arXiv preprint arXiv:1710.05941.
- [31] A.D. Jagtap, G.E. Karniadakis, How important are activation functions in regression and classification? A survey, performance comparison, and future directions, 2022, <http://dx.doi.org/10.48550/arXiv.2209.02681>, arXiv preprint arXiv:2209.02681.
- [32] T.P. Lillcrap, A. Santoro, L. Marris, C.J. Akerman, G. Hinton, Backpropagation and the brain, *Nat. Rev. Neurosci.* 21 (6) (2020) 335–346, <http://dx.doi.org/10.1038/s41583-020-0277-3>.
- [33] M.F. Møller, *A scaled conjugate gradient algorithm for fast supervised learning*, *Neural Netw.* 6 (4) (1993) 525–533.
- [34] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, *Classification and Regression Trees (CART)*, Wadsworth, Montrey, CA, USA, 1984, <http://dx.doi.org/10.1201/9781315139470>.
- [35] A.J. Sánchez, J.M. Escaño, A.J. Gallego, E.F. Camacho, Fault tolerant MPC of a solar trough field based on classification and regression trees, in: 2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol), 2016, pp. 152–157, <http://dx.doi.org/10.1109/SYSTOL.2016.7739743>.
- [36] Z. Said, P. Sharma, L.S. Sundar, V.D. Tran, et al., Using Bayesian optimization and ensemble boosted regression trees for optimizing thermal performance of solar flat plate collector under thermosyphon condition employing MWCNT-Fe₃O₄/water hybrid nanofluids, *Sustain. Energy Technol. Assess.* 53 (2022) 102708, <http://dx.doi.org/10.1016/j.seta.2022.102708>.
- [37] L. Breiman, Random forests, *Mach. Learn.* 45 (1) (2001) 5–32, <http://dx.doi.org/10.1023/A:1010933404324>.
- [38] Z. Chai, C. Zhao, *Enhanced random forest with concurrent analysis of static and dynamic nodes for industrial fault classification*, *IEEE Trans. Ind. Inform.* 16 (1) (2019) 54–66.
- [39] K. Dhibi, R. Fezai, M. Mansouri, M. Trabelsi, A. Kouadri, K. Bouzara, H. Nounou, M. Nounou, Reduced kernel random forest technique for fault detection and classification in grid-tied PV systems, *IEEE J. Photovolt.* 10 (6) (2020) 1864–1871, <http://dx.doi.org/10.1109/JPHOTOV.2020.3011068>.
- [40] S.S. Chouhan, A. Kaul, U.P. Singh, S. Jain, Bacterial foraging optimization based radial basis function neural network (BRBFNN) for identification and classification of plant leaf diseases: An automatic approach towards plant pathology, *IEEE Access* 6 (2018) 8852–8863, <http://dx.doi.org/10.1109/ACCESS.2018.2800685>.
- [41] T. Kohonen, *The Self-Organizing Maps*, Springer-Verlag, Berlin/Heidelberg, Germany, 1995.
- [42] A.J. Gallego, E.F. Camacho, Estimation of effective solar irradiation using an unscented Kalman filter in a parabolic-trough field, *Sol. Energy* 86 (2012) 3512–3518, <http://dx.doi.org/10.1016/j.solener.2011.11.012>.