Available online at www.sciencedirect.com

ScienceDirect

Fuzzy Sets and Systems 459 (2023) 43–66

FUZZY sets and systems

www.elsevier.com/locate/fss

Bayesian inference using an adaptive neuro-fuzzy inference system

Mohammed Knaiber, Leen Alawieh *

Mechanical Engineering Department, American University of Beirut, Beirut 1107 2020, Lebanon

Received 13 December 2020; received in revised form 1 July 2022; accepted 3 July 2022
Available online 15 July 2022

Abstract

For most Bayesian inference problems that are of interest, solving for the model parameter posterior probability distribution remains to be the main challenge. A typical technique entails using Markov chain Monte Carlo (MCMC), in order to obtain an approximation of the posterior distribution. However, one drawback of MCMC is that it usually requires solving the forward problem multiple times, which could be infeasible, if the forward model is computationally expensive. In this paper, a computationally cheap surrogate model is developed using the adaptive neuro-fuzzy inference system with a fuzzy c-means initialization (ANFIS-FCM). The optimal number of fuzzy rules is chosen via a proposed algorithm based on the pairing frequency clustering validity index. The effectiveness of the ANFIS-FCM model is tested on inferring reaction rate parameters in a 3-node network toy problem, which mimics idealized combustion reaction networks. Using multiple error measures, it is shown that the developed ANFIS-FCM model is capable of constructing accurate surrogate models at a minimal computational cost.
© 2022 Elsevier B.V. All rights reserved.

1. Introduction

The solution to a Bayesian inverse problem is a posterior probability distribution, which describes the probability of a model or a set of model parameters given a set of observations. However, for most Bayesian inverse problems of interest, the solution cannot be arrived at analytically, but is rather obtained statistically. Specifically, random sampling techniques, such as Markov chain Monte Carlo (MCMC), are implemented in order to obtain an approximation of the probability distribution. One drawback of MCMC though, is that it requires solving the forward model several times. This could be impracticable, if the forward model is computationally expensive and the posterior distribution is complex. There exist several approaches for tackling such challenges, some of which focus either on devising better sampling strategies [1–14], or on alleviating the cost of the forward model computations through developing reduced order models [15–21], or cheaper surrogate representations that can (locally or globally) approximate the forward model [22–34]. In this work, we focus on the surrogate representation approach.

Fuzzy logic, or merely computing with words (CW) technique [35,36], has recently gained substantial attention in the approximation of unknown nonlinear functions as seen in [37–44]. The significance of such a soft computing

* Corresponding author.
E-mail address: lalawieh@ur.rochester.edu (L. Alawieh).

https://doi.org/10.1016/j.fss.2022.07.001
0165-0114/© 2022 Elsevier B.V. All rights reserved.

technique is highlighted in its ability to reduce the mathematical complexity of highly nonlinear models. In particular, a fuzzy logic system performs the input-output mapping through an IF-THEN rule-based inference framework using linguistic labels and is based on human knowledge [45]. However, unlike artificial neural networks, fuzzy logic systems lack any learning capabilities by nature. This led to the coupling of the two frameworks in the form of an adaptive neuro-fuzzy inference system (ANFIS) [46,47], in an attempt to enhance the inference capabilities of fuzzy systems. Following the iterative learning process of ANFIS, not only is the implicitly constructed input-output mapping based on human knowledge (in the form of fuzzy IF-THEN rules), but also on stipulated input-output data pairs. This ANFIS-based representation of nonlinear relationships between selected inputs and given outputs is currently still an active research area with several applications, such as system prognosis [48,49], power grid applications [50–52], energy studies [53–56], healthcare services [57], chemical research [58–61], environmental projects [49,62–65], and industrial processes [66–74].

In this paper, we propose building an ANFIS surrogate model to replace expensive forward models, which should help speed up the on-line computations of MCMC. Fuzzy logic is known to be capable of approximating any continuous complex system (where a mathematical model is not known and/or there is a high degree of uncertainty) in a computationally efficient and accurate manner. We specifically chose ANFIS to build the surrogate model, mainly because it allows us to benefit from the capabilities of neural networks — which are widely used now in a wide range of regression problems [75] — while also being able to (1) easily inject uncertainty into the network, and (2) avoid some of the drawbacks that neural networks typically suffer from; namely the need for large training data sets and lots of manual tuning. We will show in what follows that ANFIS can produce accurate surrogate models using a small number of training samples, and we will develop an algorithm that automatically determines the optimal number of rules (thus eliminating the major element of manual tuning that ANFIS requires). Despite the obvious appeal of using ANFIS to build surrogate models for solving Bayesian inverse problems, to our knowledge, it has not yet been tried before. Therefore, we hope that this work will help provide a reference starting point for any such future efforts.

The rest of this paper is organized as follows. We begin by giving a brief background on surrogate models, fuzzy logic, and neural networks. The problem formulation is then presented in Section 2. Section 3 describes the ANFIS structure after highlighting the mathematically verified capabilities of ANFIS to approximate, to arbitrary accuracy, any real-valued and continuous function over a compact set. In Section 4, we introduce the regularized PFI algorithm for ANFIS-FCM regression models, which is used to identify the optimal number of IF-THEN rules for building the surrogate models. Implementation of the surrogate model and the assessment of the accuracy of the constructed ANFIS surrogate models is presented in Section 5, demonstrating the power of ANFIS surrogates in attaining a reliable approximation of posterior probability distributions in Bayesian inference problems. To allow for a direct comparison between the ANFIS surrogate model and another surrogate model generated using a probabilistic, kernel regression method (specifically, Gaussian process regression), we adopt the same Bayesian inverse network problem that was used to assess the accuracy of a Gaussian process regression approximation in [76]. Major conclusions are summarized in Section 6.

2. Brief background

In Uncertainty propagation and Bayesian inference for complex physical systems, statistical information is sought that describes the model output or model input, respectively. This usually requires a large number of expensive forward model evaluations, and is in most cases infeasible. One way to mitigate this issue is to use surrogate models that are cheaper to evaluate than the original model [1,77,78]. Surrogate models that rely on basis function expansions do not scale well as the number of inputs or the complexity of the model increases [78–80]. Kernel methods avoid this problem, as they can easily handle a possibly infinite number of basis functions [80]. However, some of the disadvantages of kernel methods is that they require a careful choice of training data, along with the underlying kernel function and its hyperparameters [80,81]. It is still an open research question of how one can appropriately choose the right training data and kernel function for complex problems. Moreover, for large training data sets, kernel methods become prohibitively expensive [80].

Neural networks have become the surrogate model of choice recently [78], since they are known to be capable of approximating any smooth function with arbitrary accuracy that scales only with the model complexity and not the dimensionality of the input space [75,79]. Moreover, they can handle large amounts of training data and there is no longer the need to determine an appropriate kernel function, thus avoiding the main drawbacks of kernel methods.

However, the disadvantage of neural networks is that they require large amounts of training data to yield accurate predictions, which constitutes a problem for cases where data is sparse or difficult to obtain (such as when the original model is computationally expensive). They also require careful heuristic tuning of the network architecture and its associated parameters [78,79].

Considering the dependence of the predictions from neural network surrogate models on the amount and quality of the training data and on the heuristically chosen network topology, network architecture and its parameters, one needs to be able to translate this aggregate dependence into an uncertainty in the resulting predictions — note that this uncertainty applies to surrogate models in general, and not just neural networks, but we focus our attention here on neural networks for the sake of brevity. Most uncertainty quantification methods for neural networks focus on addressing data uncertainty [82,83]. However, the issue of model inadequacy or uncertainty in the model itself, which is distinct from uncertainty in the data, is rarely addressed. Existing methods use Bayesian neural networks [84], which invoke a probabilistic formulation of the network weight parameters and tend to be computationally expensive. Approximations to Bayesian neural networks, such as Monte Carlo dropout, have been shown to be unreliable [85]. Other methods treat the network depth as a random variable in order to quantify the network prediction uncertainty [86]. Both methods, however, address only a single source of uncertainty without considering uncertainties in other aspects of the network selection (such as topology, architecture, parameters other than network depth and weights).

In contrast to probabilistic neural networks, fuzzy systems are capable of encapsulating different aspects of data and model uncertainty without having to make simplistic assumptions about the underlying source of randomness or uncertainty (such as the assumption of independent or Gaussian distributed random variables, as is commonly done in probabilistic treatments) [87]. They allow the formulation of a set of approximate mapping relationships between the space of possible input states and the corresponding set of possible outputs for cases where the system we are trying to model is complex, and the information available is not enough (sparse data) to allow us to fully understand or describe the system's behavior using an explicit functional relationship [87]. When a single output value is required, the resulting possible outputs of the fuzzy system are combined using some averaging or weighting method. It is their relatively low computational cost and their capability to seamlessly inject uncertainty / fuzziness into the inputs, mapping relationships, and outputs which gives fuzzy systems an edge over the other methods mentioned above. In this paper, we aim to demonstrate how fuzzy logic can be implemented to develop accurate surrogate models in situations where only a sparse set of input-output data is available, and where the underlying physical model is assumed to be unknown.

3. Problem formulation

3.1. Bayesian parameter inference

In order to predict the behavior of a complex physical system, given a certain set of inputs, a mathematical model approximating the system under study is typically solved. Such a forward problem could be concisely described by the following general relationship:

$$z = \mathcal{M}(\boldsymbol{x}) + \eta \tag{1}$$

where z represents the value of an observed data point ($z \in \mathbb{R}$), $\mathcal{M}$ represents the forward model which maps a real-valued ni-dimensional vector of parameters $\boldsymbol{x}$ into a relevant value in $\mathbb{R}$, such that $\mathcal{M} : \mathbb{R}^{ni} \rightarrow \mathbb{R}$. η is a random variable with a probability density p_η, which accounts for the difference between the model predictions and the actual observed data. As written above, it is assumed that the error is additive.

On the contrary, inverse problems infer the model inputs, $\boldsymbol{x}$, using a set of observations, z. By taking into account noise in the observations and the uncertainty in the input parameters, $\boldsymbol{x}$, a Bayesian inverse problem leads to a probability distribution for the input parameters.

In the Bayesian framework, the knowledge about the true values of the parameters $\boldsymbol{x}$ is expressed via a probability distribution, which represents the degrees of belief about the possible values of the parameters $\boldsymbol{x}$, before and after observing the data z. Following the Bayes' rule, this is expressed as:

$$\pi(\boldsymbol{x}|z) \propto \pi(z|\boldsymbol{x}) \times \pi(\boldsymbol{x}) \tag{2}$$

where $\pi(\boldsymbol{x})$ is the prior probability distribution before any data has been observed, $\pi(z|\boldsymbol{x})$ is the likelihood function describing the probability that a given set of input parameters, $\boldsymbol{x}$, give rise to the observed data, z. $\pi(\boldsymbol{x}|z)$ is the posterior probability distribution, which incorporates the information encoded by the observed data. For a compact representation of (2), we will hereafter denote the prior probability by $\pi_{\text{prior}}(\boldsymbol{x})$, the likelihood function by $L(\boldsymbol{x})$ and the posterior probability by $\pi_{\text{post}}(\boldsymbol{x})$. Accordingly, (2) becomes:

$$\pi_{\text{post}}(\boldsymbol{x}) \propto L(\boldsymbol{x}) \times \pi_{\text{prior}}(\boldsymbol{x}) \tag{3}$$

where:

$$L(\boldsymbol{x}) \equiv p(z|\boldsymbol{x}) = p_{\eta}(z - \mathcal{M}(\boldsymbol{x})) \tag{4}$$

As can be seen in (4), evaluating the likelihood function $p(z|\boldsymbol{x}) = p_{\eta}(z - \mathcal{M}(\boldsymbol{x}))$ requires the evaluation of $\mathcal{M}$ too. Hence, trying to approximate the solution to (3) using MCMC will require solving the forward model multiple times, which could become computationally prohibitive if the forward model is expensive. However, this hurdle could be overcome if we are able to replace $\mathcal{M}$, or alternatively the likelihood function, with a cheaper surrogate model. In this current work, we choose to employ ANFIS for the establishment of a cheap surrogate model that substitutes the likelihood function $L(\boldsymbol{x})$. Using ANFIS for building the surrogate model is mainly driven by its approximation and generality powers, as will be discussed in more details below. Moreover, we will show in what follows that ANFIS is capable of achieving a faithful approximation of the likelihood response surface.

3.2. ANFIS-based regression problem

For any regression problem, let $\boldsymbol{x}$ be the input vector and y be the dependant scalar output. According to our Bayesian formulation, the vector $\boldsymbol{x}$ constitutes the unknown model parameters, while y is the log-likelihood (of base e) function. Hence,

$$L(\boldsymbol{x}) = \exp(y) \tag{5}$$

In addition, the i^{th} entry in the vector $\boldsymbol{x}$, denoted by x_i, symbolizes the i^{th} input component. Given that we are trying to approximate the expensive log-likelihood function $y = f(\boldsymbol{x})$ with a cheaper ANFIS built surrogate, we will denote the ANFIS approximation by $y_{ANFIS} = f_{ANFIS}(\boldsymbol{x}, \boldsymbol{p})$, where $\boldsymbol{p}$ represents the vector of parameters of the ANFIS model itself. Consequently,

$$\widetilde{L}(\boldsymbol{x}) = \exp(y_{ANFIS}) \tag{6}$$

which leads to an approximation of (3) such that:

$$\widetilde{\pi}_{\text{post}}(\boldsymbol{x}) \propto \widetilde{L}(\boldsymbol{x}) \times \pi_{\text{prior}}(\boldsymbol{x}) \tag{7}$$

where $\widetilde{\pi}_{\text{post}}(\boldsymbol{x})$ denotes the ANFIS-based approximation of the true posterior probability $\pi_{\text{post}}(\boldsymbol{x})$, at a given input point, $\boldsymbol{x}$. We will assume that $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{m_x}, \boldsymbol{\Sigma_x})$, such that the following holds:

$$\pi_{\text{prior}}(\boldsymbol{x}) = \det(2\pi\,\boldsymbol{\Sigma_x})^{-1/2} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{m_x})^T\,\boldsymbol{\Sigma_x}^{-1}(\boldsymbol{x} - \boldsymbol{m_x})\right) \tag{8}$$

Note that the ANFIS surrogates themselves approximate the log-likelihood (of base e) $\log_e(L(\boldsymbol{x}))$, as will be noticed in the equations that follow. Also note that $\boldsymbol{p}$ will be evaluated during the training phase of ANFIS.

4. ANFIS: adaptive neuro-fuzzy inference system

4.1. Universal approximator ANFIS

In [46,47], the authors proved the universal approximation feature of ANFIS. Using the Stone-Weierstrass theorem and given any real-valued and continuous function $y = f(\boldsymbol{x})$ over a compact set K, ANFIS was proven to be capable of approximating such a function y by y_{ANFIS} to any arbitrary accuracy. However, to guarantee the satisfaction of the Stone-Weierstrass theorem, there is a restriction on the type of membership functions that could be prescribed to the

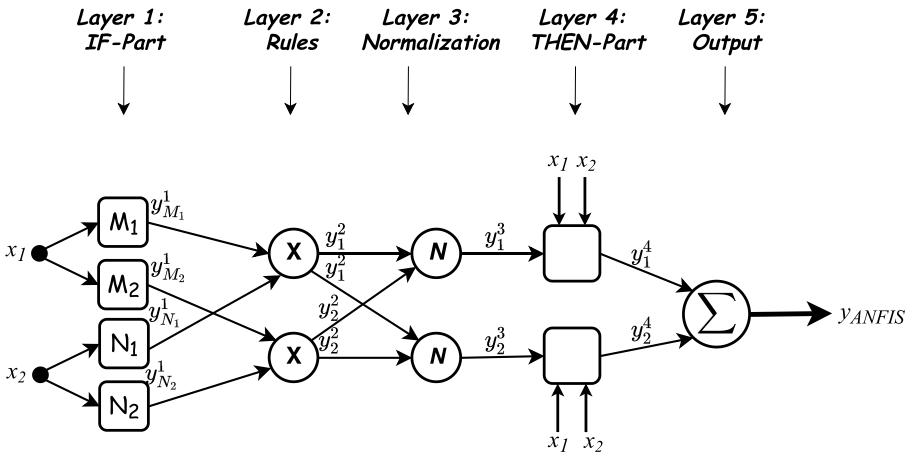

Fig. 1. A typical ANFIS architecture.

ANFIS input variables. Gaussian membership functions are among those that satisfy this requirement, and will thus be the type adopted in this paper.

The Stone-Weierstrass theorem demonstrates the universal approximation power of ANFIS as follows: $\forall \delta > 0$, $\forall y \in C(K)$, $\exists y_{ANFIS} \in S(K)$ s.t. $|y - y_{ANFIS}| < \delta \ \forall \boldsymbol{x} \in K$, where: $C(K)$ is the set of real-valued and continuous functions on a compact set K, $S(K)$ designates the set of all possible fuzzy systems on the same compact set K, y is the actual function to be approximated, and y_{ANFIS} is the y's approximation by ANFIS.

4.2. ANFIS structure

Any ANFIS model is constructed of five layers, where in each layer, the nodes represent the same mathematical operator. Layer 1 performs fuzzification of the signals of the input vector $\boldsymbol{x}$. This layer, that is also referred to by the IF-Part layer, computes the degrees to which every input x_i satisfies each of its fuzzy sets. Such degrees are usually called membership degrees. Layers 2–4 include a specific set of mathematical operations performed on the input data, until finally the output signals from layer 4 are summed up in layer 5.

To clarify the basis of the ANFIS structure, let us consider, as an illustrative example, the simple case of a typical fuzzy inference whose inputs are only x_1 and x_2, and y_{ANFIS} is its output. The corresponding 5 layers of the ANFIS model are illustrated in Fig. 1.

Every IF-THEN rule of any ANFIS model has the given structure: IF ≪premise≫, THEN ≪conclusion≫ Furthermore, for the sake of simplicity, let us suppose that the rule-base of the ANFIS example above contains only two fuzzy IF-THEN rules — in Section 4, an algorithm is proposed to select the optimal number of IF-THEN rules in ANFIS — and that the rules are of Sugeno's type, we then have:

Rule 1: IF x_1 is M_1 and x_2 is N_1, THEN
$f_1 = a_1 \times x_1 + b_1 \times x_2 + c_1$

Rule 2: IF x_1 is M_2 and x_2 is N_2, THEN
$f_2 = a_2 \times x_1 + b_2 \times x_2 + c_2$

where $\forall i \in \{1, 2\}$, M_i denotes a fuzzy set for x_1 and N_i indicates a fuzzy set for x_2. Numerous types of fuzzy sets exist, but in this paper, the Gaussian fuzzy sets are employed due to the reason stated in section 3.1. Note that the parameters a_i, b_i, and c_i, $\forall i \in \{1, 2\}$, constitute some of the entries of $\boldsymbol{p}$ which are computed during the training phase of the ANFIS model. The vector $\boldsymbol{p}$ also includes the parameters of the membership functions of the different input fuzzy sets. For instance, if M_i and N_i fuzzy sets have Gaussian membership functions $\forall i \in \{1, 2\}$, the mean m and the standard deviation σ of each of the Gaussian functions, will be among the elements in $\boldsymbol{p}$. Another point to clarify is that it is not necessary at all to have identical numbers of fuzzy sets for all inputs (as seen in this typical example).

As illustrated above, each fuzzy IF-THEN rule establishes a fuzzy implication relation between some input fuzzy sets (for example M_1 and N_1 in **Rule 1**) and their corresponding fuzzy outputs (for instance f_1 in **Rule 1**), so that no rules share the same premise parts. Hence, when each input admits two fuzzy sets, as in this typical example, then the maximum number of possible fuzzy IF-THEN rules is $2 \times 2 = 4$. However, it is not necessary to build an ANFIS model with such a maximum number of fuzzy IF-THEN rules. In fact, finding the optimal ANFIS model with the smallest number of IF-THEN rules is still an active research area. On the other hand, the conclusion part of each fuzzy IF-THEN rule in ANFIS is generally a first degree polynomial in $\boldsymbol{x}$. This linear formulation of the conclusion part has shown significant modeling accuracy and reliable computational performance, and will therefore be the form adopted in this paper.

During its training phase, ANFIS will have its parameters, $\boldsymbol{p}$, tuned using an optimization algorithm. Note again that the parameters $\boldsymbol{p}$ include the parameters of the conclusion part of all IF-THEN rules, and the parameters that are related to the fuzzy sets of the premise parts of all IF-THEN rules. In this work, the hybrid learning algorithm is the optimization algorithm that is used to tune the ANFIS parameters $\boldsymbol{p}$ during the training phase [47,50,70]. Briefly, the hybrid learning algorithm is a combination of the least squares method and the backpropagation algorithm [46,50,70]; the effectiveness of this hybrid algorithm has been widely endorsed in ANFIS research [50,62,70].

Unlike classical logic, the context of fuzziness declares the degree of membership of an element x_i of a fuzzy set β as a truth value between zero and one, represented by $\mu_\beta(x_i)$. Following the given notations and given the two input signals x_1 and x_2, the ANFIS computations start by evaluating the degrees of membership of x_1 and x_2 in M_i and N_i, respectively, according to $\mu_{M_i}(x_1) = e^{\frac{-(x_1-m)^2}{2\sigma^2}}$ for a Gaussian membership function with the mean m and the standard deviation σ. In what follows, every intermediate output of Fig. 1 that is generated by a node within any layer (except layer 5), holds the number of its generating layer as a superscript. The subscript of the output signal corresponds to the particular generating node. Subsequently, the outputs of layer 1 are:

$$y^1_{M_i} = \mu_{M_i}(x_1), \forall i \in \{1, 2\} \tag{9}$$

$$y^1_{N_i} = \mu_{N_i}(x_2), \forall i \in \{1, 2\} \tag{10}$$

where μ_{M_i} and μ_{N_i}, $\forall i \in \{1, 2\}$ are Gaussian membership functions, and the mean m and the standard deviation σ of each of these functions are among the elements in $\boldsymbol{p}$ which are tuned by the hybrid learning algorithm during the ANFIS training phase. As soon as the calculations finish at the level of layer 1, layer 2 initiates the evaluation of the firing strengths per rule node according to:

$$y^2_1 = y^1_{M_1} \times y^1_{N_1} \tag{11}$$

$$y^2_2 = y^1_{M_2} \times y^1_{N_2} \tag{12}$$

Then, layer 3 continues the operations by normalizing the firing strengths such that:

$$y^3_1 = \frac{y^2_1}{\Sigma^2_{k=1} y^2_k} \tag{13}$$

$$y^3_2 = \frac{y^2_2}{\Sigma^2_{k=1} y^2_k} \tag{14}$$

Finally, the outputs of layers 4 and 5 are calculated in accordance with the equations:

$$y^4_1 = y^3_1 \times f_1 = y^3_1 \times (a_1 \times x_1 + b_1 \times x_2 + c_1) \tag{15}$$

$$y^4_2 = y^3_2 \times f_2 = y^3_2 \times (a_2 \times x_1 + b_2 \times x_2 + c_2) \tag{16}$$

$$y_{ANFIS} = \Sigma^2_{k=1} y^4_k \tag{17}$$

An ANFIS model cannot be trained unless its vector of parameters $\boldsymbol{p}$ is initialized. ANFIS can be initialized using different techniques, including grid partitioning and fuzzy c-means clustering [48,53,54,58,60]. Fuzzy c-means clustering (FCM) is the technique that we adopt here. When the number of inputs of an ANFIS model is relatively large, grid partitioning aggregates the curse of dimensionality, and thus fosters an explosion of the number of fuzzy IF-THEN rules [48]. On the other hand, the construction of a relatively small number of fuzzy IF-THEN rules is one of the main advantages of using FCM for initializing ANFIS [48,60]. FCM clusters a training dataset into a user-defined

number of fuzzy groups or clusters G, through an iterative procedure [47,48] that aims to minimize a dissimilarity measure cost function. As a result, G corresponds to the number of fuzzy IF-THEN rules in the generated ANFIS regression model [88]. The initialized ANFIS via FCM is now ready for training, and is referred to as ANFIS-FCM in this paper.

Knowing that an accurate and computationally fast ANFIS-FCM is strongly desirable to possess, a minimal but necessary number of IF-THEN rules is always desirable to have in the ANFIS structure. While G determines the number of clusters in ANFIS-FCM models, G^* refers to the smallest possible value of G which best describes the data points. In what follows, we will introduce an algorithm that automatically determines the optimal number of fuzzy clusters G^*. The algorithm is based on the novel and attractive pairing frequency clustering validity index (PFI) [89]. We will start by briefly explaining the concept behind PFI, and then move on to introducing a useful variant of PFI.

5. Optimal number of rules in ANFIS-FCM regression models

5.1. Pairing frequency clustering validity index (PFI)

In the literature, many clustering validity indices have been addressed to evaluate the quality of induced partitions and thus, determine the optimal number of clusters [89–93]. However, the pairing frequency clustering validity index was presented as a novel and attractive clustering validity index in [89]. Following its computational procedure which detects G^* through logical reasoning and global information, by examining the different possible partitions, the authors in [89] illustrate the superior performance of PFI in comparison with various traditional clustering indices on multiple datasets. Based on this, G^* corresponds to the number of clusters whose PFI is maximum. The flowchart in Fig. 2, illustrates the method developed in [89] for selecting G^* using PFI.

The symbols in the illustrated PFI procedure correspond to the following [89]:

n	Number of training data points for FCM
G	Number of clusters for FCM
G^{upper}	Upper limit of G
G^{max}	Maximum possible value of G
G^{min}	Minimum possible value of G
$PFI(G)$	Computed PFI corresponding to G
G^*	Optimal number of clusters for FCM such that $G^* = \underset{G^{\min},\ldots,G^{\max}}{\arg\max}\ (\mathrm{PFI(G)})$

In addition the following matrix notations and mathematical operators apply:

$\mathbf{M}$	Membership matrix with dimensions $G \times n$
M_{ij}	Element of $\mathbf{M}$ in the i^{th} row and the j^{th} column
$\left[M_{ij}\right]_{\max}$	i^{th} row of $\mathbf{M}$ corresponding to the maximum element in the j^{th} column
t_j	Compact representation of $\left[M_{ij}\right]_{\max}$
$\mathbf{F}$	Pattern matrix with dimensions $n \times n$
F_{sk}	Element of $\mathbf{F}$ in the s^{th} row and the k^{th} column
$\mathbf{Q_g}$	Global pairwise pattern matrix with dimensions $n \times n$
$\mathbf{P_p}$	Pairwise pattern matrix with dimensions $n \times n$
$\mathbf{Q_g} \circ \mathbf{F}$	Hadamard product of matrices $\mathbf{Q_g}$ and $\mathbf{F}$
$\mathrm{S}(\mathbf{Q_g} \circ \mathbf{F})$	Sum of the Hadamard product of matrices $\mathbf{Q_g}$ and $\mathbf{F}$

In terms of its operation, as shown in Fig. 2, the PFI procedure starts by setting the values of G^{max}, G^{upper}, and G^{min} according to [89]. Then, the following collection of actions is executed repeatedly for every value of G ranging from G^{min} to G^{upper}: First, the matrix of training data is clustered into G clusters via FCM, so that the membership matrix $\mathbf{M}$ is built; then, the t_j value is computed for every column in $\mathbf{M}$, and finally, the corresponding pattern matrix $\mathbf{F}$ is established. The end of such loop computations launches the construction of the matrices $\mathbf{P_p}$ and $\mathbf{Q_g}$. Afterwards, the $PFI(G)$ index corresponding to every number of clusters, G, is assessed, which gives rise to the optimal number of clusters G^* among the values lying between G^{min} and G^{max}. At this level, G^* is chosen to be the number of clusters whose PFI, denoted by $PFI(G^*)$, is maximum. Hence, the PFI procedure ends by identifying the optimal number

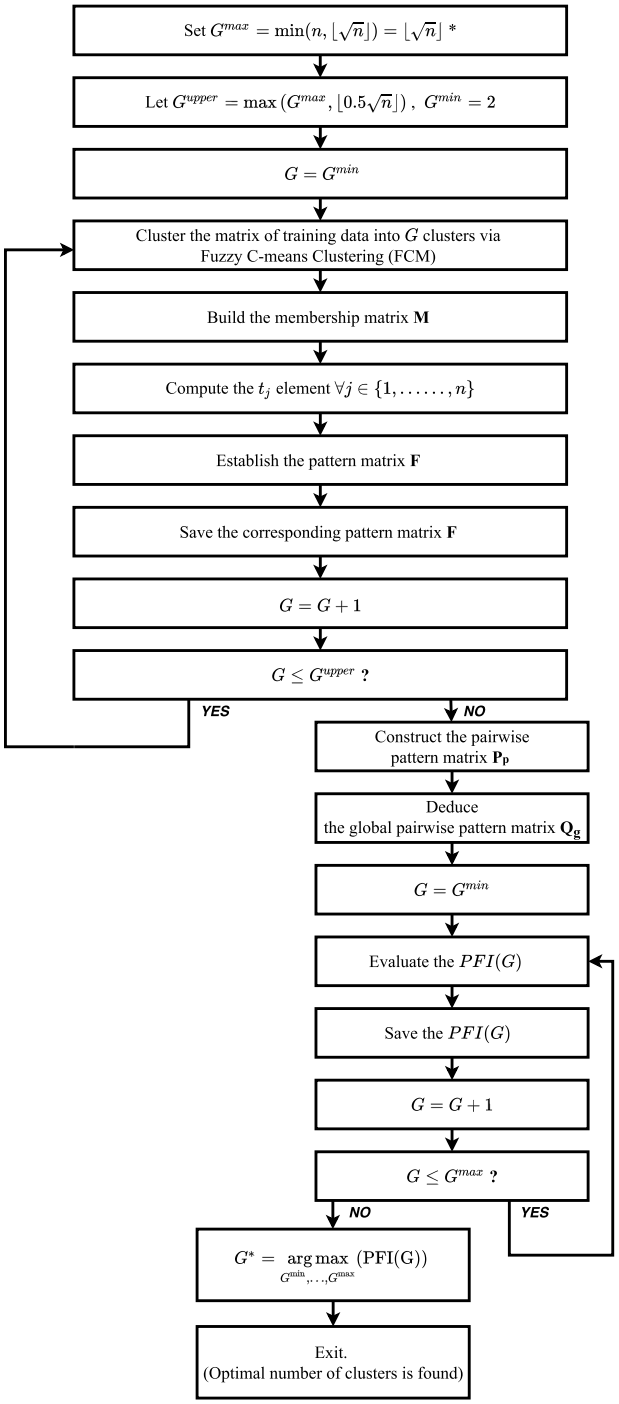

Fig. 2. Flowchart of the selection of the optimal number of clusters G* using PFI.

of clusters for this FCM-based clustering problem for the existing training data. The description of the mathematical quantities of Fig. 2 is demonstrated below:

$$\max \mathbf{M}_s = \max \{M_{is} : 1 \leq i \leq G\} \tag{18}$$

$$F_{sk} = \begin{cases} 1 - \frac{G}{G-1} \times |\max \mathbf{M}_s - \max \mathbf{M}_k| & \text{if } t_s = t_k \\ -\frac{G}{2G-2} \times \left(\max \mathbf{M}_s + \max \mathbf{M}_k - \frac{2}{G}\right) & \text{if } t_s \neq t_k \end{cases} \tag{19}$$

$$\mathbf{P_p} = \sum_{G=2}^{G^{upper}} \mathbf{F} \tag{20}$$

$$\mathbf{Q_g} = \mathbf{P_p} / \left(G^{upper} - 1\right) \tag{21}$$

$$PFI(G) = S\left(\mathbf{Q_g} \circ \mathbf{F}\right) \tag{22}$$

In (18), s represents the column's index. While this PFI procedure, as reported in [89], explores the most realistic clustering of the training dataset, it does not address the problem of the generalizing capabilities of the ANFIS-FCM regression model. Typically, the main goal ANFIS-FCM regression models are twofold; it should be capable of correctly fitting to the training dataset, while also accurately predicting the outputs of a separate yet unseen testing dataset. In the next section, we tackle this problem by developing a PFI-inspired algorithm, which is referred to hereafter as the regularized PFI procedure for ANFIS-FCM regression models. The algorithm seeks to select the optimal number of clusters G^* (or equivalently the optimal number of IF-THEN rules) for an ANFIS-FCM regression model by taking into account its fitting, as well as its generalizing capabilities. While the identification of the optimal number of IF-THEN rules for ANFIS-FCM regression models persists to be an extensive research area, our proposed algorithm presents a simple way of addressing such a fundamental ANFIS challenge.

5.2. Optimal number of rules via regularized PFI procedure for ANFIS-FCM regression models

Before diving deeply in this subsection, recall that this paper considers:

y_{ANFIS}	Estimated log-likelihood (of base e) via ANFIS-FCM surrogate model
y	Actual or desired log-likelihood (of base e)

Since our goal is to approximate the likelihood probability, all errors will be evaluated by considering the difference between the estimated $\exp(y_{ANFIS})$ and the desired $\exp(y)$. Starting from this stage, a list of notations is introduced below to address some error measures which will be used:

E, ε	An error measure and a user-defined error tolerance level
$\boldsymbol{AE}$	Vector of absolute errors
AE	Absolute error
MAE	Mean absolute error
$RMSE$	Root mean squared error
$\bullet_{train}, \bullet_{test}$	Measured error $\bullet$ on the training and the testing data respectively
$\bullet^M$	Maximum of the measured error $\bullet$

As mentioned before, in this subsection, a regularized PFI procedure is introduced to address the selection of an optimal number of clusters G^* (or equivalently the optimal number of IF-THEN rules) for ANFIS-FCM regression models. Fig. 3 shows the flowchart of the proposed PFI-based algorithm. As illustrated in Fig. 3, the proposed algorithm starts by executing the original PFI procedure and saving the resulting values of G with their corresponding PFI indices, $PFI(G)$, in the **PFI** matrix. For each row, each G and its $PFI(G)$ are loaded, and no G^* is computed at this level. After that, the rows of the **PFI** matrix are sorted in descending order of the PFI indices; this step makes use of the original PFI procedure. Then, the PFI values are normalized to [0, 1], and the initial G^* is set to a large number (such as ∞), so that the algorithm searches for the smallest but the most efficient value of G^* afterwards. In order to satisfy the ANFIS-FCM approximation requirements defined by the user, an error measure E is adopted by the algorithm with a user-defined error tolerance goal ε. Knowing that E helps the suggested algorithm to decide the value of G^* in a later stage, we use the maximum absolute testing error AE^M_{test} as our E measure. Note that, in our case, the absolute error is computed on the likelihood probability, since our goal in this paper is to attain an ANFIS-FCM surrogate model that accurately approximates the likelihood probability. Given that nv generally represents the number of testing (or validation) data points for each of our ANFIS-FCM regression models, we have:

$$\exp(y_{ANFIS_j}) = \exp(f_{ANFIS}(\boldsymbol{x}_j, \boldsymbol{p})) \tag{23}$$

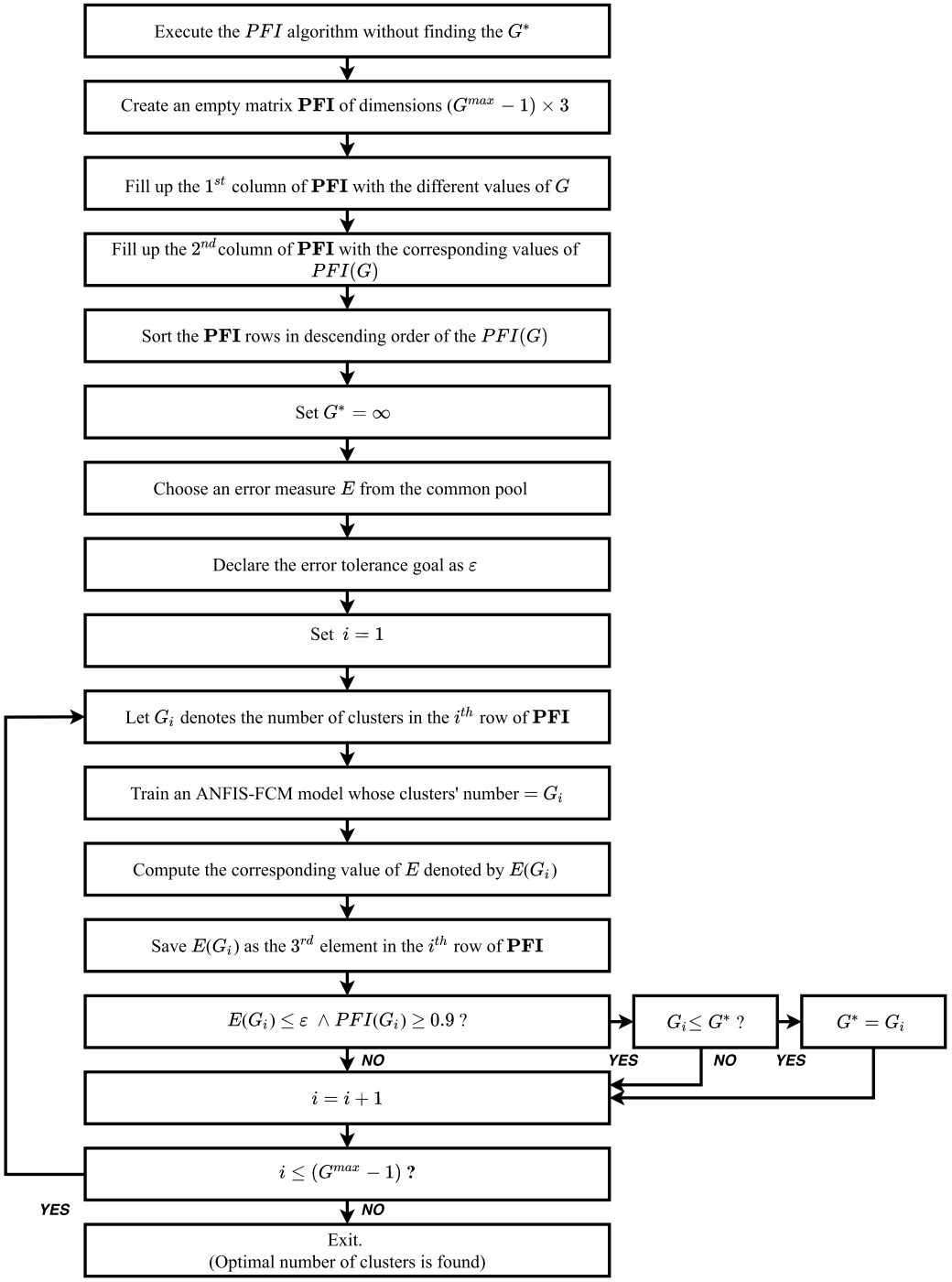

Fig. 3. Flowchart of the selection of the optimal number of clusters G* using the regularized PFI procedure for ANFIS-FCM regression models.

$$\exp(y_j)=\exp(f(\boldsymbol{x}_j)) \tag{24}$$

$$E=AE^{M}_{test}=\max_{\boldsymbol{x}_j}(|\exp(y_{ANFIS_j})\text{-}\exp(y_j)|) \tag{25}$$

where $\{\boldsymbol{x}_1,\boldsymbol{x}_2,.....,\boldsymbol{x}_j,....\}$ are input data points (testing data points in case of equation (25)).

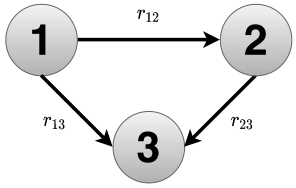

Fig. 4. Schematic diagram of the 3-node network model.

Note that E can be any error measure from the common pool (AE, MAE, or $RMSE$ as measured on either the training or testing data points). ε is set to 9×10^{-2} in this current research work. After choosing the appropriate E and setting ε, the regularized PFI algorithm trains, per each iteration i, an ANFIS-FCM surrogate model using the fixed training data and the value of G from the i^{th} row of the sorted **PFI**; this value is shortly referred to as G_i. Subsequently, the associated E to the current value of G_i, with the notion $E(G_i)$, is calculated too. For each G_i, G^* is not updated unless the given conditions are met simultaneously: $PFI(G_i) \geq 0.9$ and $E(G_i) \leq \varepsilon$ and $G_i \leq G^*$. Note that the value 0.9 is chosen to guarantee obtaining an ANFIS-FCM regression model which captures well the nature of the training dataset without losing the required generalization capabilities. Thus, the algorithm repeats per iteration the three simple logical tests ($PFI(G_i) \geq^? 0.9$ and $E(G_i) \leq^? \varepsilon$ and $G_i \leq^? G^*$) while searching for G^*. When the PFI test is performed for all G_i values, the proposed algorithm ends. Consequently, the regularized PFI procedure for ANFIS-FCM regression models returns the value G^* which guarantees the smallest number of clusters (equivalent to the least number of IF-THEN rules) that simultaneously have a high PFI index (≥ 0.9) and an acceptable error measure ($\leq \varepsilon$) (in our case, $AE^M_{test} \leq \varepsilon$). Following the identification of G^* via the proposed regularized PFI procedure, we will move on to demonstrating the implementation of ANFIS-FCM surrogates on different complex physical models.

6. Implementation

In [76], the authors proposed a new algorithm based on Gaussian Process (GP) regression, in order to efficiently sample a non-Gaussian posterior probability distributions of Bayesian inverse problems. One drawback of GP regression, though, is that its computational expense increases with the size of the training data. In this work, our goal is to demonstrate how ANFIS-based surrogates can achieve comparable approximation accuracy with little computational cost. To this end, we adopt the 3-node network problem explored in [76] as our benchmark testbed. However note that, unlike [76], we use noise-free synthetic data, so the obtained posterior probability distributions in this work will be slightly different from those obtained in [76].

6.1. 3-Node network model

We are going to test our ANFIS surrogate model on the 3-node network model illustrated in Fig. 4. For each node, the reaction rate r is evaluated via the equation:

$$r_{ij;K} = \mathrm{C}_{ij;K} \times A_{ij} \times e^{-\beta_K E_{ij}} \tag{26}$$

where A represents the pre-exponential factor, E denotes the activation energy, β is the thermodynamic inverse temperature, and C is the concentration of the reaction species. In (26), the node across which a reaction is occurring is indicated by the ij indices, while the index K represents the particular experimental conditions under which the reactions is taking place. All reactions are assumed to be irreversible, and to proceed sequentially starting at node 1 and ending at node 3. The observed end-output of each experiment corresponds to the total time required by the slowest reaction pathway to terminate, whereby the time for any reaction pathway is given by the sum of the inverse of the reaction rate across each of the nodes involved in that pathway. For instance, the reaction pathway $\{1 \rightarrow 2 \rightarrow 3\}$ has a total reaction time t given by: $t = 1/r_{12} + 1/r_{23}$. Tables 1 and 2 summarize the specific experimental conditions and reaction rate parameter values that we will use in order to carry out our Bayesian inference exercise. The values in the tables have been taken from [76].

Table 1
Adopted experiments for the Bayesian inference problem tackling the detection of the uncertain reaction rate parameters of the 3-node network model [76].

Experiment K	C_{12}	C_{13}	C_{23}	β
1	10	0.5	10	0.01
2	20	0.5	20	0.1
3	0.5	20	2	0.01
4	2	30	0.5	0.1
5	2	20	0.5	0.01
6	5	20	5	0.01
7	0.5	30	0.5	0.1

Table 2
True values of the E_{ij} and A_{ij} parameters of the 3-node network model [76].

Nodes	(1,2)	(1,3)	(2,3)
E	5	2	1
A	1	3	2

Table 3
General description of the distinct ANFIS surrogate models.

ANFIS-FCM surrogate models	
Parameter	*Description*
Selection method of the number of FCM clusters	Regularized PFI procedure for ANFIS-FCM regression models
Number of inputs	2 or 6 in current research
Input membership functions	Gaussian (“gaussmf”)
Number of outputs	1 (log-likelihood)
Output membership functions	Linear
Training algorithm	Hybrid learning algorithm
Maximum number of training epochs	10000
Training error goal	0
Initial training step size	0.01 (default)
Step size decrease rate	0.9 (default)
Step size increase rate	1.1 (default)

6.2. Structure of ANFIS-FCM surrogates

The structure of three constructed ANFIS surrogate models is described in this section, and the performance of each of the surrogates is numerically evaluated and interpreted. Each of these surrogate models addresses the approximation of a posterior probability distribution resulting from the Bayesian inference of reaction rate model parameters in the 3-node reaction network described above. Table 3 summarizes the shared common parameters of the three distinct ANFIS models.

For each ANFIS-FCM surrogate model, the number of training data points, n, is selected by systematically decreasing such number until good learning and prediction abilities are recorded. A distinct testing dataset of 40000 data points is used to check the generalization capabilities of each of the constructed surrogate models. The execution of the different computational algorithms is carried out in Matlab on a Windows 7 machine having Intel(R) i5-2430M CPU at 2.40 GHz and 8.0 GB of RAM.

We will investigate building a surrogate model for three different inference cases, where we will refer to each case by Model k where $k \in \{1, 2, 3\}$. For each case, FCM generates the initial fuzzy inference system with an appropriate (non-optimal) number of clusters, while the proposed regularized PFI procedure determines the optimal number of

clusters or IF-THEN rules. Each Model k with ni inputs is trained using a specific number of training data points, where each training data point is denoted by $(x_1, x_2, ..., x_{ni}, y)$. x_i denotes a parameter model input that we are seeking to infer, while y is the corresponding log-likelihood model output. Note that all the surrogate models investigated here approximate the natural logarithm of the targeted likelihood (log-likelihood).

6.2.1. 2-D model 1

For Model 1, the input vector is $(A'_{12}, A'_{23})^T$ and $ni = 2$. For this model and all the models that follow, the apostrophe sign $\bullet'$ denotes a normalized variable which lies in the interval [0,1], the superscript $\bullet^T$ describes the transpose operator, and ni represents the number of model inputs. Model 1 is trained using 225 grid points that fall within a maximum distance of 3 actual standard deviations of the (Gaussian) prior probability distribution of the input parameters that we seek to infer. In this model and those that follow, the Gaussian prior distribution is taken to be of the form shown in Eq. (8). By making use of the regularized PFI procedure for ANFIS-FCM regression models, the optimal number of IF-THEN rules in Model 1 is detected to be 5.

6.2.2. 2-D model 2

Model 2 is supplied with $(E'_{12}, E'_{23})^T$ input vector. This model is trained with 117 MCMC data points sampled from the prior probability distribution of the input parameters for this model. Following the regularized PFI procedure, the optimal number of IF-THEN rules is detected to be 4 for Model 2. By comparing Model 1 and Model 2, we can notice that as the number of training data increases, the number of membership functions increases as well.

6.2.3. 6-D model 3

Model 3 accepts the input variables E'_{12}, A'_{12}, E'_{23}, A'_{23}, E'_{13}, and A'_{13} simultaneously, such that $ni = 6$. 60 MCMC data points, sampled from the Gaussian prior probability distribution of the input parameters, are used to train Model 3. Following the regularized PFI-based procedure, the optimal number of membership functions turns out to be 3. As expected, since the number of training data points for Model 3 is less than that of Models 1 and 2, the number of membership functions for Model 3 is also smaller than both of them.

6.3. Regularized PFI procedure results

As observed in the previous subsection, the execution of the regularized PFI procedure determined the optimal number of clusters for every ANFIS-FCM surrogate model. As mentioned in the previous section, it is important to determine the optimal number of clusters in order to reduce the computational expense of ANFIS, since the number of clusters not only determines the number of fuzzy IF-THEN rules, but also it represents the number of membership functions per input/output. Figs. 5–6 show the variation of PFI and AE^M_{test} as a function of the number of clusters for Models 1 and 2. The results for Model 3 show a similar trend, and have thus been omitted for brevity. As can be seen from the figures, the optimal number of clusters determined by the regularized PFI algorithm, $G^* = 5$ and $G^* = 4$ for Models 1 and 2 respectively, satisfies the inequalities: $PFI(G^*) \geq 0.9$ and $AE^M_{test}(G^*) \leq \varepsilon$. Increasing the number of clusters beyond this point does not result in a measurable improvement in the accuracy of the surrogate model. The effectiveness of this choice will be further demonstrated in Sections 6.5–6.6. Note that due to the very high AE^M_{test} values for $G = 3, 5$ in Model 2, the results for these G values were omitted from Fig. 6 for clarity.

In the next subsection, the obtained optimal number of clusters G^* per Model k, where $k \in \{1, 2, 3\}$, is used to create the suitable ANFIS-FCM surrogate model for each inference case.

6.4. Detailed layout of built ANFIS-FCM surrogates

Having determined the optimal number of membership functions for each model, we now move on to optimizing the ANFIS-FCM model parameters, $\boldsymbol{p}$, using the hybrid learning algorithm. As mentioned previously, $\boldsymbol{p}$ contains the parameters of the Gaussian membership functions and the parameters of the conclusion parts of each IF-THEN rule.

Tables 4 and 5 summarize the obtained optimal parameters of the fuzzy input membership functions for each model case, where the following notation holds (notice that Model 1 and Model 2 utilize only M^k_i and N^k_i because each of them admits 2 inputs only):

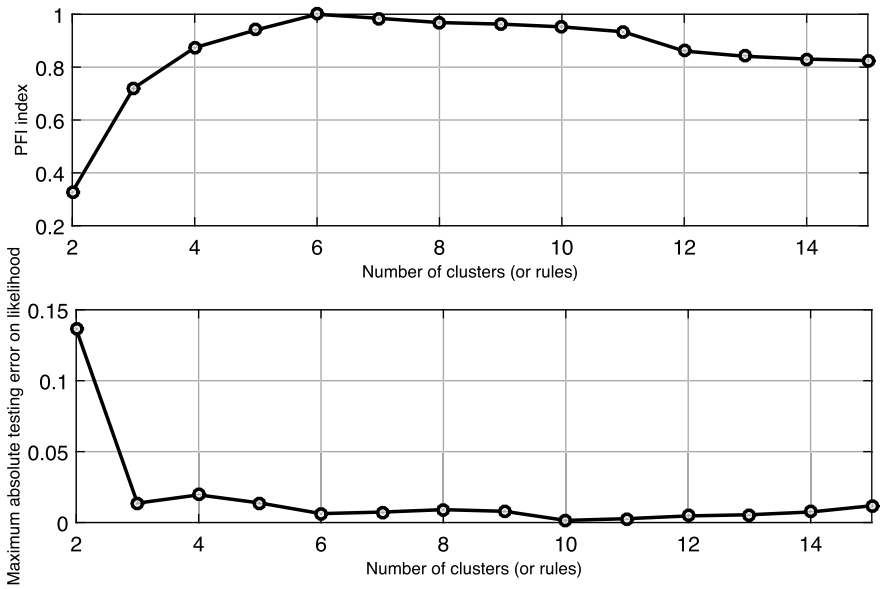

Fig. 5. The variation of PFI and Maximum absolute testing error AE^{M}_{test} as function of the number of clusters using the regularized PFI procedure for Model 1.

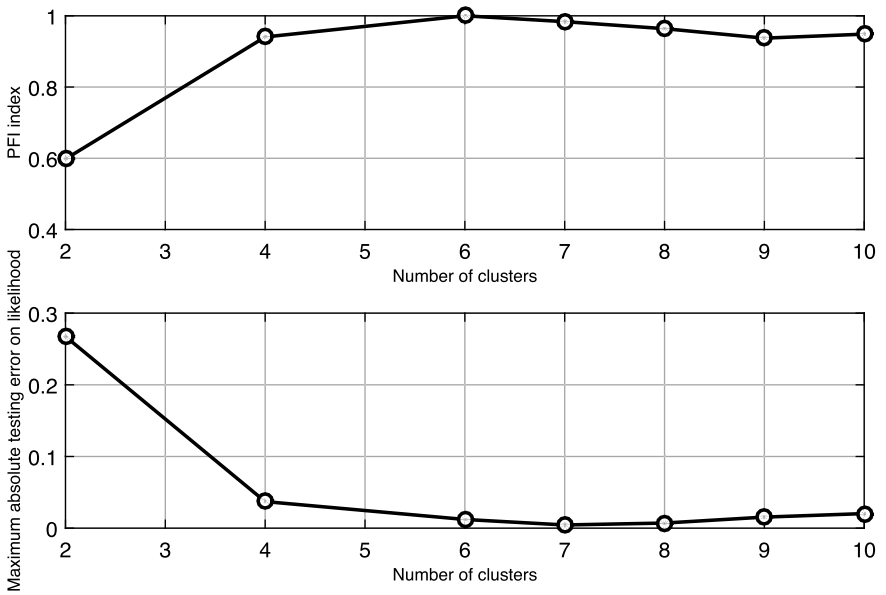

Fig. 6. The variation of PFI and Maximum absolute testing error AE^{M}_{test} as function of the number of clusters using the regularized PFI procedure for Model 2.

Table 4
Tuned parameters of the input membership functions in Model 1 and Model 2.

$k = 1, 2$	Model 1 $\{(m_i^1, \sigma_i^1)\}$	Model 2 $\{(m_i^2, \sigma_i^2)\}$
$\{M_1^k, N_1^k\}$	$\{(0.2110, 0.5355), (0.2896, 0.5135)\}$	$\{(0.1011, 0.5194), (0.0634, 0.5303)\}$
$\{M_2^k, N_2^k\}$	$\{(0.1955, 0.5985), (0.2427, 0.5042)\}$	$\{(0.0875, 0.4245), (0.0486, 0.4347)\}$
$\{M_3^k, N_3^k\}$	$\{(0.2111, 0.5967), (0.3034, 0.5079)\}$	$\{(0.0942, 0.4735), (0.0565, 0.5142)\}$
$\{M_4^k, N_4^k\}$	$\{(0.0575, 0.1738), (0.0719, 0.0325)\}$	$\{(0.0943, 0.4149), (0.0498, 0.4989)\}$
$\{M_5^k, N_5^k\}$	$\{(0.2051, 0.6364), (0.2641, 0.5154)\}$	—

Table 5
Tuned parameters of the input membership functions in Model 3.

$k=3$	Model 3 $\{(m_i^3, \sigma_i^3)\}$
$\{M_1^k, M_2^k, M_3^k\}$	$\{(0.1130, 0.5009), (0.1066, 0.4342), (0.0476, 0.5796)\}$
$\{N_1^k, N_2^k, N_3^k\}$	$\{(0.0337, 0.2609), (0.0298, 0.3049), (0.0115, 0.3079)\}$
$\{P_1^k, P_2^k, P_3^k\}$	$\{(0.2828, 0.4987), (0.2516, 0.4103), (0.1150, 0.3189)\}$
$\{Q_1^k, Q_2^k, Q_3^k\}$	$\{(0.1450, 0.4311), (0.1381, 0.4113), (0.0601, 0.4099)\}$
$\{S_1^k, S_2^k, S_3^k\}$	$\{(0.1463, 0.7236), (0.1367, 0.6172), (0.1279, 0.4113)\}$
$\{T_1^k, T_2^k, T_3^k\}$	$\{(0.1115, 0.3104), (0.1056, 0.3842), (0.0980, 0.3663)\}$

Table 6
Tuned linear output membership functions in Model 1 and Model 2.

$k=1,2$	Model 1	Model 2
f_1^k	$+175.7419A'_{12} + 76.6084A'_{23} - 40.8204$	$-127.9316E'_{12} - 78.6896E'_{23} - 0.6478$
f_2^k	$+7.0528A'_{12} + 10.7703A'_{23} + 3.8919$	$+198.6326E'_{12} + 59.6281E'_{23} - 146.7267$
f_3^k	$+20.3930A'_{12} - 27.2766A'_{23} - 90.1663$	$+33.0437E'_{12} + 41.3949E'_{23} + 115.5273$
f_4^k	$+78.4821A'_{12} + 46.4550A'_{23} - 42.2846$	$-1.3350E'_{12} + 13.5553E'_{23} - 88.1983$
f_5^k	$-42.5511A'_{12} - 62.1272A'_{23} + 36.0810$	—

Table 7
Tuned linear output membership functions in Model 3.

$k=3$	Model 3
f_1^k	$-3.8480E'_{12} - 333.7062A'_{12} + 11.1319E'_{13} + 137.8962A'_{13}$ $+63.7738E'_{23} - 74.1780A'_{23} - 26.9320$
f_2^k	$+75.5718E'_{12} - 336.5887A'_{12} + 19.9218E'_{13} - 41.3688A'_{13}$ $+71.0713E'_{23} - 75.3734A'_{23} + 70.6550$
f_3^k	$-86.3773E'_{12} + 274.9081A'_{12} + 11.8898E'_{13} + 59.2876A'_{13}$ $+22.9472E'_{23} - 95.1118A'_{23} - 53.0760$

$M_i^k : i^{th}$ fuzzy set of 1^{st} input x_1 in ANFIS-FCM Model k
$N_i^k : i^{th}$ fuzzy set of 2^{nd} input x_2 in ANFIS-FCM Model k
$P_i^k : i^{th}$ fuzzy set of 3^{rd} input x_3 in ANFIS-FCM Model k
$Q_i^k : i^{th}$ fuzzy set of 4^{th} input x_4 in ANFIS-FCM Model k
$S_i^k : i^{th}$ fuzzy set of 5^{th} input x_5 in ANFIS-FCM Model k
$T_i^k : i^{th}$ fuzzy set of 6^{th} input x_6 in ANFIS-FCM Model k

while $k \in \{1, 2, 3\}$.

The couplet (m_i^k, σ_i^k) stands for the mean and the standard deviation of the Gaussian membership function of the i^{th} fuzzy set of an input in ANFIS-FCM Model k. For instance, Model 1 admits a membership function for each of the 5 fuzzy sets per input, so the means and the standard deviations of the third membership function associated with the two fuzzy sets, $\{M_3^1, N_3^1\}$, are denoted as $\{(0.2111, 0.5967), (0.3034, 0.5079)\}$, where $(m_3^1, \sigma_3^1) = (0.2111, 0.5967)$ corresponds to the set M_3^1. For every ANFIS-FCM model, the collection of the means and the standard deviations of the different input membership functions constitute what is generally described as the premise parameters.

Tables 6 and 7 summarize the optimal parameters obtained for the IF-THEN fuzzy output functions for each model, whereby each IF-THEN rule in an ANFIS-FCM Model k obeys the following general layout:

Rule i: IF x_1 is M_i^k AND x_2 is N_i^k, THEN
$f_i^k = a_i \times x_1 + b_i \times x_2 + c_i$

Accordingly, the log-likelihood output, via any activated rule in the ANFIS-FCM surrogate models, can be linearly expressed in terms of the normalized inputs.

6.5. Approximated versus real posterior probabilities

Having laid out the structure of the ANFIS-FCM surrogate models that we are going to use and optimized all of the parameters associated with each one, we are now ready to train the surrogate models using the training data from each Model k, where $k \in \{1, 2, 3\}$. We will then compare the resulting approximated joint posterior probabilities of the unknown model parameters, which are $(A_{12}, A_{23})^T$, $(E_{12}, E_{23})^T$, and $(E_{12}, A_{12}, E_{23}, A_{23}, E_{13}, A_{13})^T$ for Models 1, 2, and 3 respectively, with the real posterior distributions.

For qualitative comparisons, the approximated joint posterior probabilities will be plotted against the actual joint posterior probabilities. For a more quantitative measure of the qualities of the approximations, the absolute testing and training errors $AE = |\exp(y_{ANFIS_j}) - \exp(y_j)|$ will be evaluated per model for every testing and training point $\boldsymbol{x}_j$. Note that since Model 3 is 6-D and thus hard to inspect visually, only the quantitative comparisons will be provided for the approximations associated with Model 3. Finally, the computational time to build and implement each of the ANFIS-FCM surrogate models will be discussed in Section 6.6.

6.5.1. First case study: 2-D Model 1

Model 1 involves inferring the posterior probability distributions of parameters A_{12} and A_{23}. Fig. 7 shows a comparison of the actual (bottom left) versus the approximated (bottom right) posterior probability contours for Model 1, along with training data points that were used to train the ANFIS-FCM surrogate model, the Gaussian prior distribution, π_{prior}, (top left) and the initial ANFIS approximation prior to training (top right). As can be seen from the figure, the ANFIS-FCM surrogate model is capable of considerably narrowing down the support of the prior distribution even before training, and eventually accurately reproducing the actual posterior distribution after training.

Fig. 8 shows a box plot of the absolute training error as a function of the input parameter A_{12} computed over the 225 training points, while Fig. 9 shows a histogram of the absolute testing error computed over 40,000 test points. As can be seen from Fig. 8, the majority of the individual values of AE_{train} per training data point $\boldsymbol{x}_j$ are located around zero, with the exception of a few outliers with values that don't exceed 6×10^{-3}. The maximum observed error and the highest dispersion of error values seems to occur around the mode of the posterior distribution ($A_{12} = 1$), which is expected because errors around the mode are weighted more strongly than errors further away per our definition of the absolute error (see Eq.(25)). From Fig. 9, we can notice that in addition to the mapping accuracy of the ANFIS-FCM surrogate model, it also has good generalization capabilities. Testing the trained surrogate model against 40,000 distinct and new data points generates a low likelihood absolute testing error, AE_{test}, which ranges between 0.000345 and 0.0134. Specifically, more than 67% of the testing points show a value of AE_{test} that is close to 0.000345, while 83% have AE_{test} values that are less than 0.002.

6.5.2. Second case study: 2-D Model 2

Model 2 involves inferring the posterior probability distributions of parameters E_{12} and E_{23}. Fig. 10 shows a comparison of the actual (bottom left) and the ANFIS-FCM approximated (bottom right) joint posterior probability distributions for Model 2, along with MCMC-selected training points, the Gaussian prior distribution, π_{prior}, (top left) and the initial ANFIS approximation prior to training (top right). Again, as can be seen from the figure, the initial untrained ANFIS surrogate is capable of gearing the posterior towards the correct location of the mode of the posterior distribution, and eventually accurately reproducing the actual posterior distribution after training. Notice that despite having many of the training points lying outside the support of the real posterior distribution, ANFIS is still capable of producing an accurate approximation of the posterior. This seems to indicate that, contrary to many other types of surrogate models, the quality of the training data points is of minor importance to ANFIS surrogate models.

Fig. 11 shows a box plot of the absolute training error, AE_{train}, as function of E_{12} computed over the 117 training data points. As can be seen from the figure, AE_{train} values range from 0 to 0.04, with the smallest error occurring near E_{12} values that have a higher concentration of training data points.

Fig. 12 shows a histogram of the absolute testing error computed over 40,000 test points. Again, the ANFIS-FCM surrogate model displays good generalization capabilities for Model 2 as can be seen from the low likelihood absolute testing error, whereby AE_{test} ranges from a minimum of 0.000948 to a maximum of 0.03697. In particular, more than

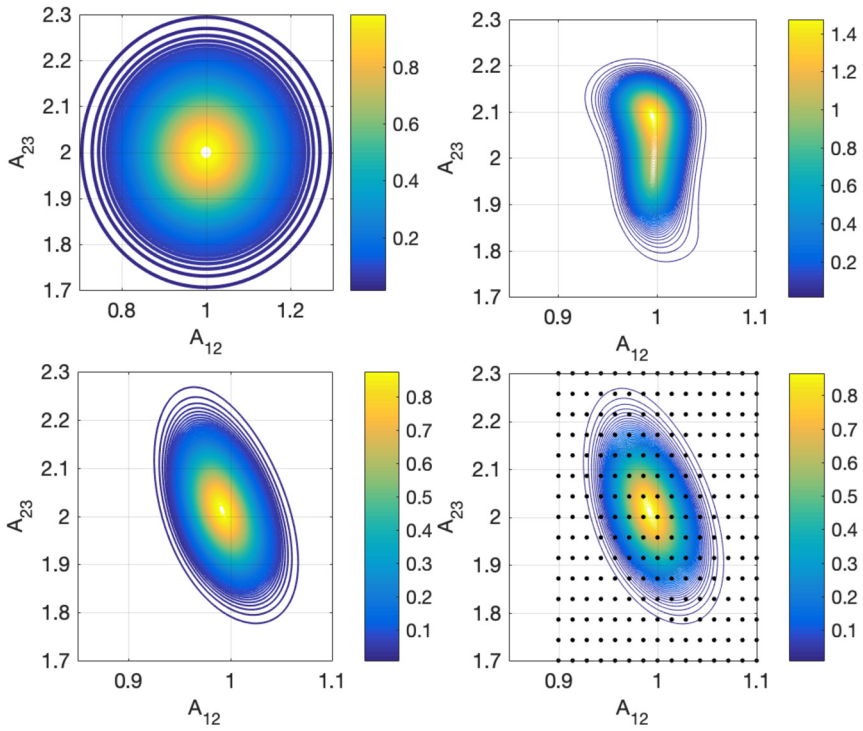

Fig. 7. Prior Gaussian probability distribution (top left) and the initial posterior probability distribution using the untrained ANFIS surrogate model (top right) for Model 1, whose parameters are A_{12} and A_{23}. Actual (bottom left) and trained ANFIS-approximated (bottom right) posterior probabilities. Black dots denote the training data points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

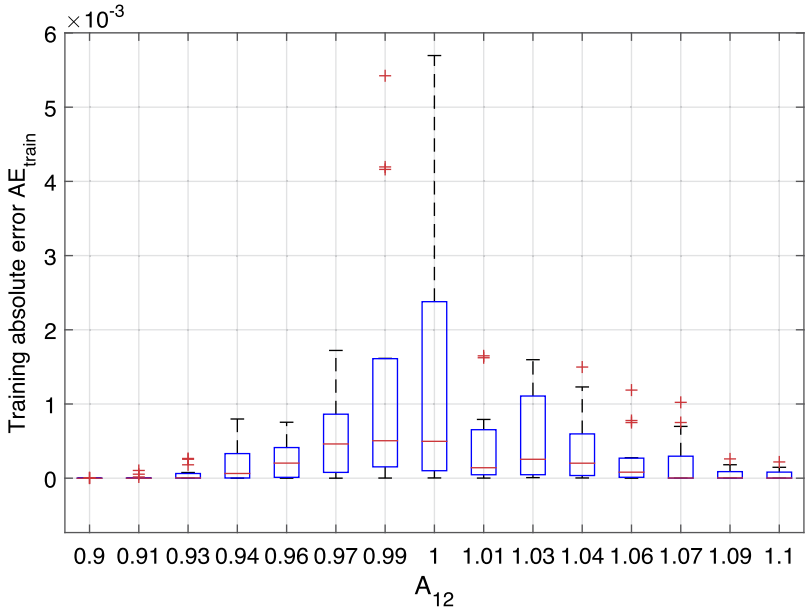

Fig. 8. Box plot of the training absolute error AE_{train} as measured on the likelihood for Model 1 over 225 training points.

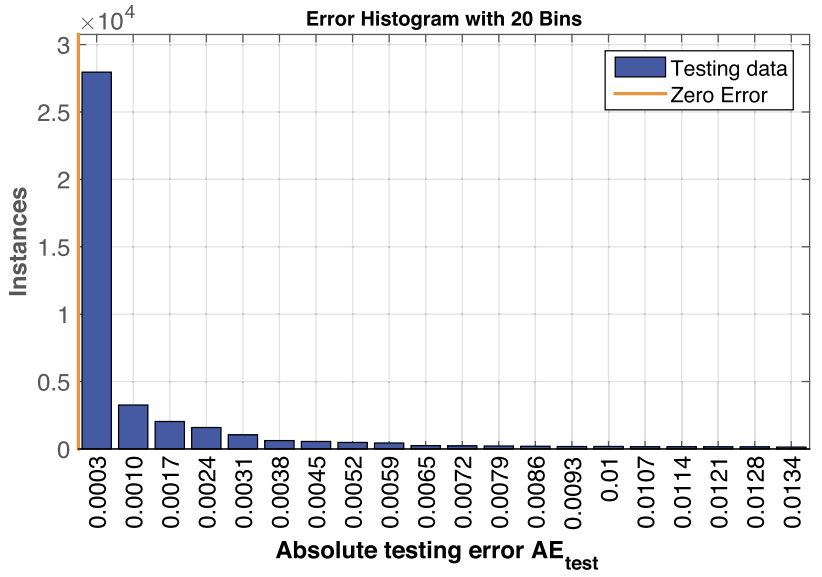

Fig. 9. Likelihood error histogram showing the number of testing data points associated with every absolute testing error value AE_{test} for Model 1 over 40,000 test points.

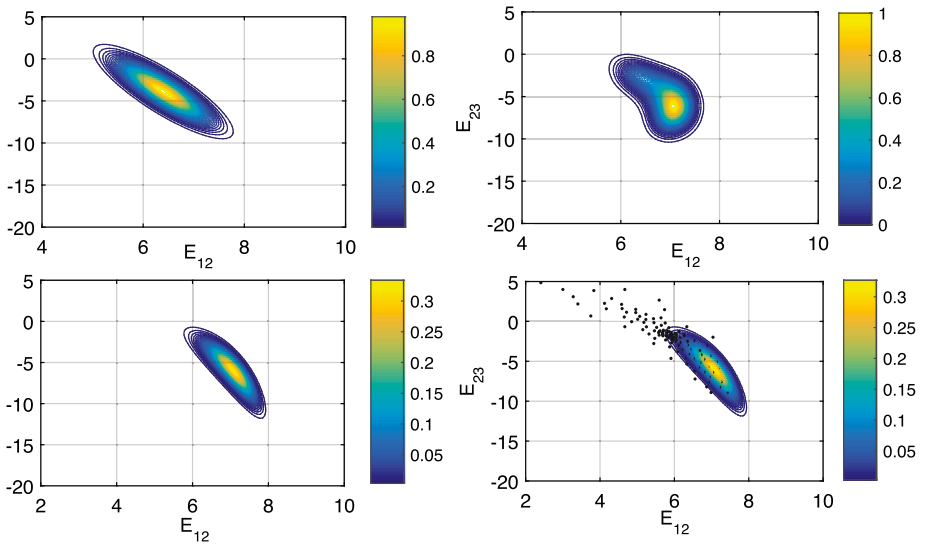

Fig. 10. Prior Gaussian probability distribution (top left) and the initial posterior probability distribution using the untrained ANFIS surrogate model (top right) for Model 2, whose parameters are E_{12} and E_{23}. Actual (bottom left) and trained ANFIS-approximated (bottom right) posterior probabilities. Black dots denote the training data points.

99.4% of the testing data points have AE_{test} values that are less than 0.003, while 99.15% of the testing data points contribute to AE_{test} values that are less than 0.001.

6.5.3. Third case study: 6-D Model 3

This part explores the input/output mapping and the generalization capacities of ANFIS-FCM surrogate Model 3, which is a 6D problem with inputs $(E_{12}, A_{12}, E_{23}, A_{23}, E_{13}, A_{13})^T$. As mentioned previously, only quantitative error measure plots will be presented for this model due to the high dimensionality of the problem.

Fig. 13 shows a box plot of the absolute training error as a function of the input E_{23} computed over 60 MCMC-selected training data points. As can be noticed, values of AE_{train} range from 0 to 0.055, with the majority having

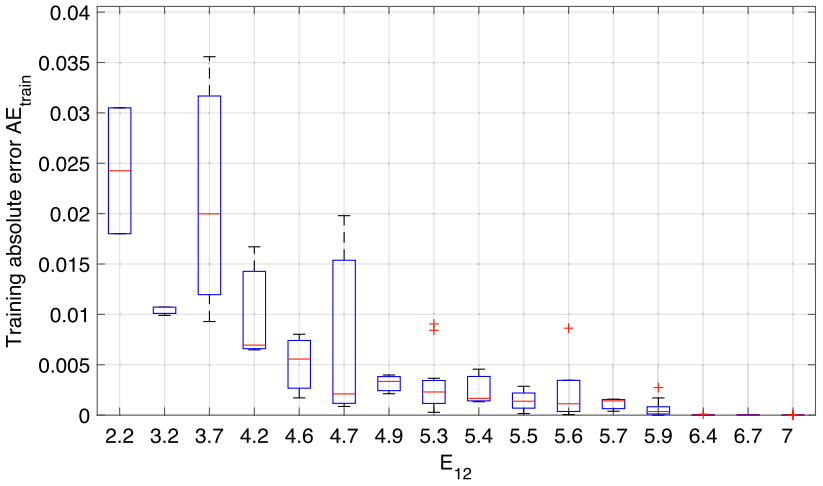

Fig. 11. Box plot of the training absolute error AE_{train} as measured on the likelihood for Model 2 computed over 117 training points.

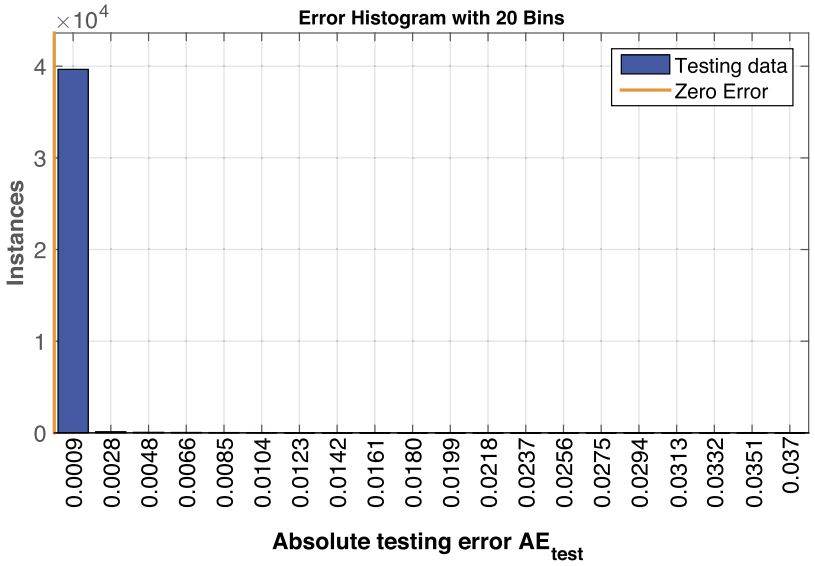

Fig. 12. Likelihood error histogram showing the number of testing data points associated with every absolute testing error value AE_{test} for Model 2 computed over 40,000 test points.

values as small as 0.01. This again demonstrates the good mapping capabilities of the ANFIS-FCM surrogate 6D model.

Fig. 14 shows a histogram of the absolute testing error computed over 40,000 distinct and new data points. As can be seen from the figure, AE_{test} ranges from 4.84×10^{-10} to 1.89×10^{-8}, with 95% of the test points attaining an error measure as small as 4.84×10^{-10}. This strongly demonstrates the good generalization capabilities of the surrogate model, even for high-dimensional problems.

6.6. Performance indicators of the ANFIS surrogates

In this section, we further evaluate the performance of each of the three different ANFIS-FCM surrogate models presented above using various statistical indicators. Specifically, based on the actual and the predicted likelihood probabilities, we compute the Root Mean Square Error ($RMSE$), the Mean Absolute Error (MAE), and the Maximum Absolute Error (AE^M) for each model as follows:

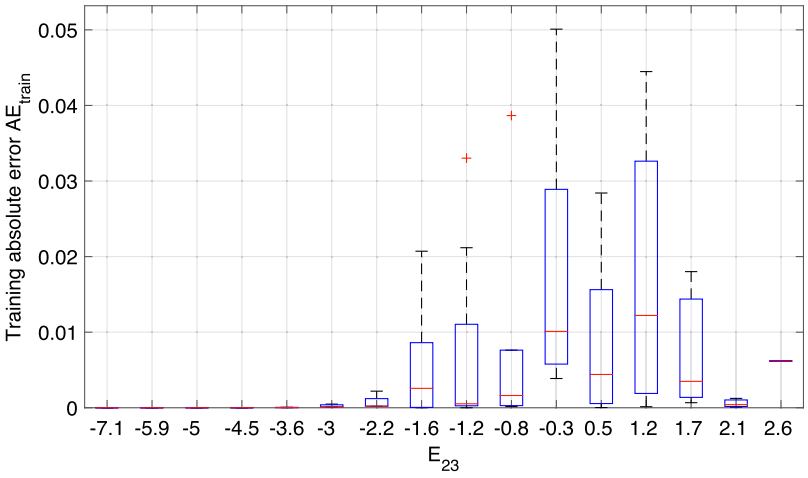

Fig. 13. Box plot of the training absolute error AE_{train} as measured on the likelihood for Model 3 computed over 60 training data points.

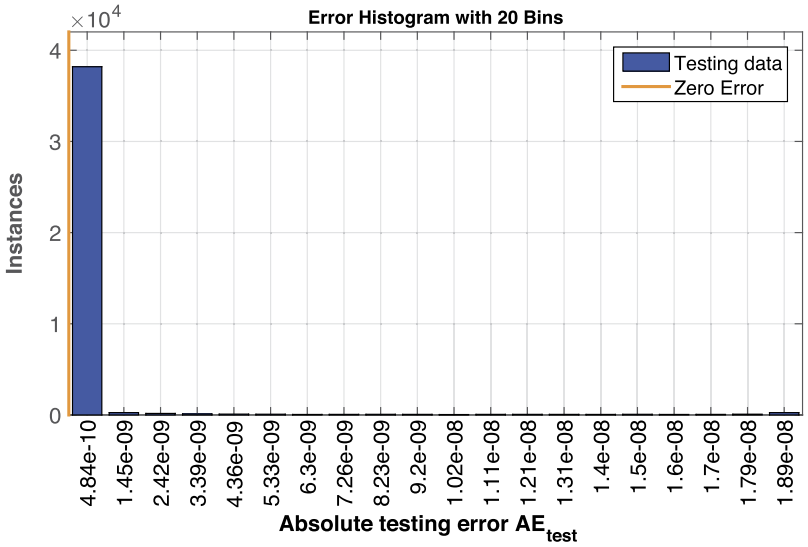

Fig. 14. Likelihood error histogram showing the number of testing data points associated with every absolute testing error value AE_{test} for Model 3 computed over 40,000 test points.

$$RMSE = \sqrt{\frac{1}{np}\sum_{j=1}^{np}\Big(\exp(y_{ANFIS_j}) - \exp(y_j)\Big)^2} \tag{27}$$

$$MAE = \frac{1}{np}\sum_{j=1}^{np}\Big(\Big|\exp(y_{ANFIS_j}) - \exp(y_j)\Big|\Big) \tag{28}$$

$$AE^M = \max_{\boldsymbol{x}_j}(|\exp(y_{ANFIS_j}) - \exp(y_j)|) \tag{29}$$

where each data point is symbolized by $\boldsymbol{x}_j$ and the total number of data points is denoted by np.

The learning capabilities of the different ANFIS-FCM surrogate models are evaluated by computing the above performance metrics using the training data points ($RMSE_{train}$, MAE_{train}, ...), whereas their extrapolation capabilities are evaluated by computing the error measures on the testing data points ($RMSE_{test}$, MAE_{test}, ...). The same training and testing data points that were used in Section 6.5 are used here as well.

Table 8
Performance indicators of the built ANFIS-FCM surrogate models on the training datasets.

Training set				
Model	$RMSE_{train}$	AE^{M}_{train}	MAE_{train}	*time*
Model 1	0.0027	0.006	0.0013	34.77 s
Model 2	0.0063	0.0356	0.0026	11.59 s
Model 3	0.0138	0.0501	0.0069	17.23 s

Table 9
Performance indicators of the built ANFIS-FCM surrogate models on the testing datasets.

Testing set				
Model	$RMSE_{test}$	AE^{M}_{test}	MAE_{test}	*time*
Model 1	0.0026	0.0138	0.0011	0.08 s
Model 2	0.0012	0.0371	9.1532e-05	0.06 s
Model 3	2.4385e-09	1.9377e-08	4.3640e-10	0.13 s

Tables 8 and 9 show the computed error metrics for each of the models on the training and testing data points, respectively. The total compute time for training and testing is also shown for each of the models. As can be noticed from the tables, all of the reported error measures are on the order of 10^{-2} or less, which demonstrates the fidelity of all of the built ANFIS surrogate models. It is interesting to note that while Models 2 and 3 feature, on average, higher training errors than Model 1, they both tend to have lower testing errors, despite the fact that the inference problem associated with Models 2 and 3 is more complex than Model 1. This could be due to the larger flexibility (higher training error) of Models 2 and 3 compared to Model 1. As regards the compute time, we can notice that, as expected, the training compute time is consistently larger than the testing compute time. Moreover, the compute time in both the testing and training stages is longest for Model 1 and shortest for Model 2 due to the larger number of training parameters for the former compared to the latter. Note that even though Model 3 has the smallest number of membership functions / rules per input, it has more input parameters which consequently leads to a larger overall number of input membership functions, and thus a larger overall number of training parameters.

7. Conclusion

In this work, we employed adaptive neuro-fuzzy inference systems (ANFIS) in order to build surrogate models for Bayesian inverse problems, which to our knowledge, has not been tried before. Prior to training, the ANFIS models were initialized via FCM, while the optimal number of IF-THEN rules was determined via a regularized PFI algorithm that was inspired by the pairing frequency clustering validity index PFI. The effectiveness of the built ANFIS-FCM surrogate models was tested on three different inference scenarios in a 3-node network toy problem [76], which mimics idealized combustion reaction networks. Using a variety of performance indicators, based on the actual and the predicted likelihood probabilities, it was shown that the resulting surrogates yielded acceptable approximations of the log-likelihood function with maximum absolute errors that did not exceed 0.0501. Moreover, the compute time during both the training and testing phases for all the models was shown to be at most 35 s, which is a testament of the efficiency of the ANFIS-based strategy.

A future research direction involves employing ANFIS-based surrogates on more complex Bayesian inverse problems in which the uncertain rate parameters correspond to real multi-node fossil fuel combustion chains. This would also include performing a quantitative comparison between the accuracy and computational efficiency of ANFIS-based surrogates, and other state-of-the-art techniques. It would also be interesting to explore in future works the possible use of higher-order Takagi–Sugeno fuzzy surrogates, which use higher-order (> 1) IF-THEN conclusion parts. Such higher-order Takagi–Sugeno fuzzy surrogates would resemble a local Taylor series expansion of the nonlinear likelihood function, which might be able to more accurately capture possible severe non-linearities in the likelihood function.

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.

Acknowledgement

This work has been supported by MSFEA startup funds at the American University of Beirut.

References

[1] James Martin, Lucas C. Wilcox, Carsten Burstedde, Omar Ghattas, A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion, SIAM J. Sci. Comput. 34 (3) (2012) A1460–A1487.
[2] Matthias Morzfeld, Marcus S. Day, Ray W. Grout, George Shu Heng Pau, Stefan A. Finsterle, John B. Bell, Iterative importance sampling algorithms for parameter estimation, SIAM J. Sci. Comput. (2018).
[3] Heikki Haario, Marko Laine, Antonietta Mira, Eero Saksman, Dram: efficient adaptive MCMC, Stat. Comput. 16 (4) (2006) 339–354.
[4] J. Andrés Christen, Colin Fox, Markov chain Monte Carlo using an approximation, J. Comput. Graph. Stat. 14 (4) (2005) 795–810.
[5] J. Andrés Christen, Colin Fox, et al., A general purpose sampling algorithm for continuous distributions (the t-walk), Bayesian Anal. 5 (2) (2010) 263–281.
[6] Amit Apte, Martin Hairer, A.M. Stuart, Jochen Voss, Sampling the posterior: an approach to non-gaussian data assimilation, Physica D 230 (1–2) (2007) 50–64.
[7] Gareth O. Roberts, Richard L. Tweedie, et al., Exponential convergence of Langevin distributions and their discrete approximations, Bernoulli 2 (4) (1996) 341–363.
[8] Paul Dostert, Yalchin Efendiev, Thomas Y. Hou, Wuan Luo, Coarse-gradient Langevin algorithms for dynamic data integration and uncertainty quantification, J. Comput. Phys. 217 (1) (2006) 123–142.
[9] Yalchin Efendiev, Thomas Hou, Wuan Luo, Preconditioning Markov chain Monte Carlo simulations using coarse-scale models, SIAM J. Sci. Comput. 28 (2) (2006) 776–803.
[10] J.B.M.B.J. Berger, A. Dawid, D.H.A. Smith, M. West, Markov chain Monte Carlo-based approaches for inference in computationally intensive inverse problems, in: Bayesian Statistics 7: Proceedings of the Seventh Valencia International Meeting, Oxford University Press, 2003, page 181.
[11] Radford M. Neal, et al., MCMC using hamiltonian dynamics, in: Handbook of Markov Chain Monte Carlo, vol. 2(11), 2011.
[12] John Geweke, Hisashi Tanizaki, On Markov chain Monte Carlo methods for nonlinear and non-gaussian state-space models, Commun. Stat., Simul. Comput. 28 (4) (1999) 867–894.
[13] Mark Girolami, Ben Calderhead, Riemann manifold Langevin and Hamiltonian Monte Carlo methods, J. R. Stat. Soc., Ser. B, Stat. Methodol. 73 (2) (2011) 123–214.
[14] Tiangang Cui, Kody J.H. Law, Youssef M. Marzouk, Dimension-independent likelihood-informed MCMC, J. Comput. Phys. 304 (2016) 109–137.
[15] Tan Bui-Thanh, Karen Willcox, Omar Ghattas, Model reduction for large-scale systems with high-dimensional parametric input space, SIAM J. Sci. Comput. 30 (6) (2008) 3270–3288.
[16] Tan Bui-Thanh, Karen Willcox, Omar Ghattas, Parametric reduced-order models for probabilistic analysis of unsteady aerodynamic applications, AIAA J. 46 (10) (2008) 2520–2529.
[17] S.R. Arridge, J.P. Kaipio, V. Kolehmainen, M. Schweiger, E. Somersalo, T. Tarvainen, M. Vauhkonen, Approximation errors and model reduction with an application in optical diffusion tomography, Inverse Probl. 22 (1) (2006) 175.
[18] Jingbo Wang, Nicholas Zabaras, Using bayesian statistics in the estimation of heat source in radiation, Int. J. Heat Mass Transf. 48 (1) (2005) 15–29.
[19] Gianluigi Rozza, Dinh Bao Phuong Huynh, Anthony T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, Arch. Comput. Methods Eng. 15 (3) (2007) 1.
[20] Roger G. Ghanem, Pol D. Spanos, Stochastic finite element method: response statistics, in: Stochastic Finite Elements: a Spectral Approach, Springer, 1991, pp. 101–119.
[21] Olivier Le Maître, Omar M. Knio, Spectral Methods for Uncertainty Quantification: with Applications to Computational Fluid Dynamics, Springer Science & Business Media, 2010.
[22] Salman Habib, Katrin Heitmann, David Higdon, Charles Nakhleh, Brian Williams, Cosmic calibration: constraints from the matter power spectrum and the cosmic microwave background, Phys. Rev. D 76 (8) (2007) 083503.
[23] Katrin Heitmann, Martin White, Christian Wagner, Salman Habib, David Higdon, The coyote universe. I. Precision determination of the nonlinear matter power spectrum, Astrophys. J. 715 (1) (2010) 104.
[24] Velamur Asokan Badri Narayanan, Nicholas Zabaras, Stochastic inverse heat conduction using a spectral approach, Int. J. Numer. Methods Eng. 60 (9) (2004) 1569–1593.
[25] Dave Higdon, James Gattiker, Brian Williams, Maria Rightley, Computer model calibration using high-dimensional output, J. Am. Stat. Assoc. 103 (482) (2008) 570–583.
[26] Marc C. Kennedy, Anthony O'Hagan, Bayesian calibration of computer models, J. R. Stat. Soc., Ser. B, Stat. Methodol. 63 (3) (2001) 425–464.

[27] Marc C. Kennedy, Clive W. Anderson, Stefano Conti, Anthony O'Hagan, Case studies in gaussian process modelling of computer codes, Reliab. Eng. Syst. Saf. 91 (10–11) (2006) 1301–1309.
[28] Ilias Bilionis, Nicholas Zabaras, Bledar A. Konomi, Guang Lin, Multi-output separable gaussian process: towards an efficient, fully bayesian paradigm for uncertainty quantification, J. Comput. Phys. 241 (2013) 212–239.
[29] B.J.N. Blight, L. Ott, A bayesian approach to model inadequacy for polynomial regression, Biometrika 62 (1) (1975) 79–88.
[30] J.R. Koehler, A.B. Owen, 9 computer experiments, in: Handbook of Statistics, vol. 13, 1996, pp. 261–308.
[31] Max D. Morris, Toby J. Mitchell, Donald Ylvisaker, Bayesian design and analysis of computer experiments: use of derivatives in surface prediction, Technometrics 35 (3) (1993) 243–255.
[32] Jeremy Oakley, Anthony O'hagan, Bayesian inference for the uncertainty distribution of computer model outputs, Biometrika 89 (4) (2002) 769–784.
[33] Anthony O'Hagan, John Frank Charles Kingman, Curve fitting and optimal design for prediction, J. R. Stat. Soc. B (1978) 1–42.
[34] Patrick R. Conrad, Youssef M. Marzouk, Natesh S. Pillai, Aaron Smith, Accelerating asymptotically exact MCMC for computationally intensive models via local approximations, J. Am. Stat. Assoc. 111 (516) (2016) 1591–1607.
[35] L.A. Zadeh, Fuzzy logic = computing with words (cw), IEEE Trans. Fuzzy Syst. 4 (2) (1996).
[36] Kevin M. Passino, Stephen Yurkovich, Michael Reinfrank, Fuzzy Control, vol. 42, Citeseer, 1998.
[37] Jinzhu Peng, Rickey Dubay, Adaptive fuzzy backstepping control for a class of uncertain nonlinear strict-feedback systems based on dynamic surface control approach, Expert Syst. Appl. 120 (2019) 239–252.
[38] Harold R. Chamorro, Ivan Riano, Reinhard Gerndt, Ivan Zelinka, Francisco Gonzalez-Longatt, Vijay K. Sood, Synthetic inertia control based on fuzzy adaptive differential evolution, Int. J. Electr. Power Energy Syst. 105 (2019) 803–813.
[39] Hui Ma, Qi Zhou, Lu Bai, Hongjing Liang, Observer-based adaptive fuzzy fault-tolerant control for stochastic nonstrict-feedback nonlinear systems with input quantization, IEEE Trans. Syst. Man Cybern. Syst. 49 (2) (2018) 287–298.
[40] Yongming Li, Kangkang Sun, Shaocheng Tong, Observer-based adaptive fuzzy fault-tolerant optimal control for siso nonlinear systems, IEEE Trans. Cybern. 49 (2) (2018) 649–661.
[41] Jianwei Xia, Jing Zhang, Wei Sun, Baoyong Zhang, Zhen Wang, Finite-time adaptive fuzzy control for nonlinear systems with full state constraints, IEEE Trans. Syst. Man Cybern. Syst. 49 (7) (2018) 1541–1548.
[42] Hao Yan, Yongxiang Xu, Feiyang Cai, He Zhang, Weiduo Zhao, Chris Gerada, Pwm-vsi fault diagnosis for a pmsm drive based on the fuzzy logic approach, IEEE Trans. Power Electron. 34 (1) (2018) 759–768.
[43] Maolong Lv, Simone Baldi, Zongcheng Liu, The non-smoothness problem in disturbance observer design: a set-invariance-based adaptive fuzzy control method, IEEE Trans. Fuzzy Syst. 27 (3) (2019) 598–604.
[44] M. Saied, M. Knaiber, H. Mazeh, H. Shraim, C. Francis, Bfa fuzzy logic based control allocation for fault-tolerant control of multirotor uavs, Aeronaut. J. 123 (1267) (2019) 1356–1373.
[45] Bikash Pal, Balarko Chaudhuri, Robust Control in Power Systems, Springer Science & Business Media, 2006.
[46] J.-S.R. Jang, Anfis: adaptive-network-based fuzzy inference system, IEEE Trans. Syst. Man Cybern. 23 (3) (1993) 665–685.
[47] Jyh-Shing Roger Jang, Chuen-Tsai Sun, Eiji Mizutani, Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [book review], IEEE Trans. Autom. Control 42 (10) (1997) 1482–1484.
[48] Hao Liu, Jian Chen, Daniel Hissel, Hongye Su, Short-term prognostics of pem fuel cells: a comparative and improvement study, IEEE Trans. Ind. Electron. 66 (8) (2018) 6077–6086.
[49] Wei Chen, Mahdi Panahi, Khabat Khosravi, Hamid Reza Pourghasemi, Fatemeh Rezaie, Davoud Parvinnezhad, Spatial prediction of groundwater potentiality using anfis ensembled with teaching-learning-based and biogeography-based optimization, J. Hydrol. 572 (2019) 435–448.
[50] Dragan Mlakić, Hamid Reza Baghaee, Srete Nikolovski, A novel anfis-based islanding detection for inverter-interfaced microgrids, IEEE Trans. Smart Grid 10 (4) (2018) 4411–4424.
[51] Jaipal Saroha, Mukhtiar Singh, Dinesh Kumar Jain, Anfis-based add-on controller for unbalance voltage compensation in a low-voltage microgrid, IEEE Trans. Ind. Inform. 14 (12) (2018) 5338–5345.
[52] Abbas Fotouhi, Daniel J. Auger, Karsten Propp, Stefano Longo, Lithium–sulfur battery state-of-charge observability analysis and estimation, IEEE Trans. Power Electron. 33 (7) (2017) 5847–5859.
[53] Sinvaldo Rodrigues Moreno, Leandro dos Santos Coelho, Wind speed forecasting approach based on singular spectrum analysis and adaptive neuro fuzzy inference system, Renew. Energy 126 (2018) 736–754.
[54] Ling Zou, Lunche Wang, Li Xia, Aiwen Lin, Bo Hu, Hongji Zhu, Prediction and comparison of solar radiation using improved empirical models and adaptive neuro-fuzzy inference systems, Renew. Energy 106 (2017) 343–353.
[55] Mortaza Aghbashlo, Soleiman Hosseinpour, Meisam Tabatabaei, Mohamad Mojarab Soufiyan, Multi-objective exergetic and technical optimization of a piezoelectric ultrasonic reactor applied to synthesize biodiesel from waste cooking oil (wco) using soft computing techniques, Fuel 235 (2019) 100–112.
[56] Iftikhar Ali, Fiona Cawkwell, Edward Dwyer, Stuart Green, Modeling managed grassland biomass estimation by using multitemporal remote sensing data–a machine learning approach, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 10 (7) (2016) 3254–3264.
[57] David Achanccaray, Christian Flores, Christian Fonseca, Javier Andreu-Perez, A p300-based brain computer interface for smart home interaction through an anfis ensemble, in: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2017, pp. 1–5.
[58] Hadi Abdollahi, Amir H. Mohammadi, Hossein Safari, Milad Maghsoudi Akbari, Accurate density prediction for aqueous Na-K-Mg-Ca-Li-Cl-Br-SO4 solution at high-pressure and high-temperature conditions, J. Mol. Liq. 234 (2017) 209–219.
[59] Mostafa Mostafaei, Prediction of biodiesel fuel properties from its fatty acids composition using anfis approach, Fuel 229 (2018) 227–234.
[60] Mashallah Rezakazemi, Amir Dashti, Morteza Asghari, Saeed Shirazian, H2-selective mixed matrix membranes modeling using anfis, pso-anfis, ga-anfis, Int. J. Hydrog. Energy 42 (22) (2017) 15211–15225.
[61] Alireza Baghban, Jafar Sasanipour, Pouya Haratipour, Mehdi Alizad, Masih Vafaee Ayouri, Anfis modeling of rhamnolipid breakthrough curves on activated carbon, Chem. Eng. Res. Des. 126 (2017) 67–75.

[62] Ashkan Nabavi-Pelesaraei, Shahin Rafiee, Seyed Saeid Mohtasebi, Homa Hosseinzadeh-Bandbafha, Kwok-wing Chau, Comprehensive model of energy, environmental impacts and economic in rice milling factories by coupling adaptive neuro-fuzzy inference system and life cycle assessment, J. Clean. Prod. 217 (2019) 742–756.
[63] Abbas Mardani, Yee Van Fan, Mehrbakhsh Nilashi, Robert E. Hooker, Seckin Ozkul, Dalia Streimikiene, Nanthakumar Loganathan, A two-stage methodology based on ensemble adaptive neuro-fuzzy inference system to predict carbon dioxide emissions, J. Clean. Prod. 231 (2019) 446–461.
[64] Jing Liang, Xiaoxu Liu, Kuo Liao, Soil moisture retrieval using uwb echoes via fuzzy logic and machine learning, IEEE Int. Things J. 5 (5) (2017) 3344–3352.
[65] Zaher Mundher Yaseen, Isa Ebtehaj, Hossein Bonakdari, Ravinesh C. Deo, Ali Danandeh Mehr, Wan Hanna Melini Wan Mohtar, Lamine Diop, Ahmed El-Shafie, Vijay P. Singh, Novel approach for streamflow forecasting using a hybrid anfis-ffa model, J. Hydrol. 554 (2017) 263–276.
[66] Maida Bárbara Reyes Rodríguez, Jorge Laureano Moya Rodríguez, Cristiano Hora Fontes, Determination of design and operation parameters of a surface condenser using an adaptive neuro-fuzzy inference system, Int. J. Heat Mass Transf. 138 (2019) 17–24.
[67] Jun Wu, Yongheng Su, Yiwei Cheng, Xinyu Shao, Chao Deng, Cheng Liu, Multi-sensor information fusion for remaining useful life prediction of machining tools by adaptive network based fuzzy inference system, Appl. Soft Comput. 68 (2018) 13–23.
[68] Hui Yang, Yating Fu, Dianhui Wang, Multi-anfis model based synchronous tracking control of high-speed electric multiple unit, IEEE Trans. Fuzzy Syst. 26 (3) (2017) 1472–1484.
[69] Christos T. Krasopoulos, Minos E. Beniakar, Antonios G. Kladas, Multicriteria pm motor design based on anfis evaluation of ev driving cycle efficiency, EEE Trans. Transp. Electrification 4 (2) (2018) 525–535.
[70] Jino Mathew, James Griffin, Miltiadis Alamaniotis, Stratis Kanarachos, Michael E. Fitzpatrick, Prediction of welding residual stresses using machine learning: comparison between neural networks and neuro-fuzzy systems, Appl. Soft Comput. 70 (2018) 131–146.
[71] Di Wu, Huabin Chen, Yiming Huang, Yinshui He, Minghua Hu, Shanben Chen, Monitoring of weld joint penetration during variable polarity plasma arc welding based on the keyhole characteristics and pso-anfis, J. Mater. Process. Technol. 239 (2017) 113–124.
[72] Yajun Zhang, Tianyou Chai, Hong Wang, Dianhui Wang, Xinkai Chen, Nonlinear decoupling control with anfis-based unmodeled dynamics compensation for a class of complex industrial processes, IEEE Trans. Neural Netw. Learn. Syst. 29 (6) (2017) 2352–2366.
[73] Alireza Zendehboudi, Xianting Li, Baolong Wang, Utilization of ann and anfis models to predict variable speed scroll compressor with vapor injection, Int. J. Refrig. 74 (2017) 475–487.
[74] Christos T. Krasopoulos, Minos E. Beniakar, Antonios G. Kladas, Robust optimization of high-speed pm motor design, IEEE Trans. Magn. 53 (6) (2017) 1–4.
[75] Chris M. Bishop, Neural networks and their applications, Rev. Sci. Instrum. 65 (6) (1994) 1803–1832.
[76] Leen Alawieh, Jonathan Goodman, John B. Bell, Iterative construction of gaussian process surrogate models for bayesian inference, J. Stat. Plan. Inference 207 (2020) 55–72.
[77] Benjamin Peherstorfer, Karen Willcox, Max Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, SIAM Rev. 60 (3) (2018) 550–591.
[78] Rohit K. Tripathy, Ilias Bilionis, Deep uq: learning deep neural network surrogate models for high dimensional uncertainty quantification, J. Comput. Phys. 375 (2018) 565–588.
[79] Christopher M. Bishop, et al., Neural Networks for Pattern Recognition, Oxford University Press, 1995.
[80] C.M. Bishop, Pattern Recognition and Machine Learning, Information Science and Statistics, Springer, 2006.
[81] Youssef M. Marzouk, Habib N. Najm, Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems, J. Comput. Phys. 228 (6) (2009) 1862–1902.
[82] Andrey Malinin, Mark Gales, Predictive uncertainty estimation via prior networks, Adv. Neural Inf. Process. Syst. 31 (2018).
[83] Balaji Lakshminarayanan, Alexander Pritzel, Charles Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, Adv. Neural Inf. Process. Syst. 30 (2017).
[84] Martin Ferianc, Partha Maji, Matthew Mattina, Miguel Rodrigues, On the effects of quantisation on model uncertainty in bayesian neural networks, in: Uncertainty in Artificial Intelligence, PMLR, 2021, pp. 929–938.
[85] Andrew Foong, David Burt, Yingzhen Li, Richard Turner, On the expressiveness of approximate inference in bayesian neural networks, Adv. Neural Inf. Process. Syst. 33 (2020) 15897–15908.
[86] Javier Antorán, James Allingham, José Miguel Hernández-Lobato, Depth uncertainty in neural networks, Adv. Neural Inf. Process. Syst. 33 (2020) 10620–10634.
[87] T.J. Ross, Fuzzy Logic with Engineering Applications, fourth edition, John Wiley & Sons (US), 2017.
[88] Quang Thinh Tran, Sy Dzung Nguyen, Tae-Il Seo, Algorithm for estimating online bearing fault upon the ability to extract meaningful information from big data of intelligent structures, IEEE Trans. Ind. Electron. 66 (5) (2018) 3804–3813.
[89] Hongyan Cui, Kuo Zhang, Yajun Fang, Stanislav Sobolevsky, Carlo Ratti, Berthold K.P. Horn, A clustering validity index based on pairing frequency, IEEE Access 5 (2017) 24884–24894.
[90] Weina Wang, Yunjie Zhang, On fuzzy cluster validity indices, Adv. Fuzzy Sets Syst. 158 (19) (2007) 2095–2117.
[91] Isak Gath, Amir B. Geva, Unsupervised optimal fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. 11 (7) (1989) 773–780.
[92] Malay K. Pakhira, Sanghamitra Bandyopadhyay, Ujjwal Maulik, Validity index for crisp and fuzzy clusters, Pattern Recognit. 37 (3) (2004) 487–501.
[93] Malay K. Pakhira, Sanghamitra Bandyopadhyay, Ujjwal Maulik, A study of some fuzzy cluster validity indices, genetic clustering and application to pixel classification, Adv. Fuzzy Sets Syst. 155 (2) (2005) 191–214.