Environmental Modelling and Software 167 (2023) 105776

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft

Exploding the myths: An introduction to artificial neural networks for prediction and forecasting

Holger R. Maier [a,*], Stefano Galelli [b], Saman Razavi [c,d], Andrea Castelletti [e], Andrea Rizzoli [f], Ioannis N. Athanasiadis [g], Miquel Sànchez-Marrè [h], Marco Acutis [i], Wenyan Wu [j], Greer B. Humphrey [k]

[a] School of Architecture and Civil Engineering, The University of Adelaide, Australia
[b] Pillar of Engineering Systems and Design, Singapore University of Technology and Design, Singapore
[c] Institute for Water Futures, Mathematical Sciences Institute, Australian National University, Canberra, Australia
[d] Global Institute for Water Security, School of Environmental and Sustainability, University of Saskatchewan, Saskatoon, Canada
[e] Department of Electronics, Information, and Bioengineering, Politecnico di Milano, Milan, Italy
[f] Dalle Molle Institute for Artificial Intelligence (IDSIA USI-SUPSI), Switzerland
[g] Laboratory for Geo-information Science and Remote Sensing, And Wageningen Data Competence Center, Wageningen University & Research, Wageningen, the Netherlands
[h] Dept. of Computer Science, Intelligent Data Science and Artificial Intelligence Research Center (IDEAI-UPC), Universitat Politècnica de Catalunya (UPC), Catalonia, Spain
[i] Department of Agricultural and Environmental Science, University of Milan, Via Celoria 2, Milan, Italy
[j] Department of Infrastructure Engineering, The University of Melbourne, Australia
[k] South Australian Health and Medical Research Institute, Adelaide, Australia

A R T I C L E I N F O

Handling Editor: Daniel P Ames

Keywords:
Artificial neural networks
Deep learning
Introduction
Overview
Prediction
Forecasting
Environmental modelling
Good modelling practice

A B S T R A C T

Artificial Neural Networks (ANNs), sometimes also called models for deep learning, are used extensively for the prediction of a range of environmental variables. While the potential of ANNs is unquestioned, they are surrounded by an air of mystery and intrigue, leading to a lack of understanding of their inner workings. This has led to the perpetuation of a number of myths, resulting in the misconception that applying ANNs primarily involves "throwing" a large amount of data at "black-box" software packages. While this is a convenient way to side-step the principles applied to the development of other types of models, this comes at significant cost in terms of the usefulness of the resulting models. To address these issues, this inroductory overview paper explodes a number of the common myths surrounding the use of ANNs and outlines state-of-the-art approaches to developing ANNs that enable them to be applied with confidence in practice.

1. Introduction

Artificial Neural Networks (ANNs), sometimes also referred to as models for deep learning (Goodfellow et al., 2016; Razavi 2021), are a computational tool inspired by the structure and operation of the human brain. However, while early research into ANNs tried to understand and replicate the operation of the brain as closely as possible, the majority of subsequent research efforts have focused on the development and application of ANNs as a computational tool for forecasting and pattern recognition. In the field of environmental modelling, ANNs have been used primarily for "prediction and forecasting" in areas such as hydrology and water resources (Abrahart et al., 2012; ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000a, 2000b; Dawson and Wilby, 2001; Maier and Dandy, 2000a,b; Razavi and Araghinejad, 2009; Maier et al., 2010, Taormina et al., 2012; Wu et al., 2014), air quality and carbon emissions (Cabaneros et al., 2019; Ghalandari et al., 2021; Huber et al., 2021; Zhang et al., 2020), meteorology and climate science (Granata and Di Nunno, 2021; Samadianfard et al., 2020), environmental science and ecology (Gardner and Dorling, 1998; Zou et al., 2022), agriculture (Paudel et al., 2023; Zou et al., 2022), soil science (Schillaci et al., 2021) and renewable energy (Heydari et al., 2019; Kalogirou, 2001). Consequently, the context of

* Corresponding author.
E-mail address: holger.maier@adelaide.edu.au (H.R. Maier).

https://doi.org/10.1016/j.envsoft.2023.105776
Received 18 May 2023; Received in revised form 30 June 2023; Accepted 3 July 2023
Available online 5 July 2023
1364-8152/

this paper is forecasting and prediction. For ANN applications to other, less common, environmental modelling purposes, such as “diagnostic learning” and “scenario analysis”, the reader may refer to Razavi et al. (2022).

While the use of ANNs was initially restricted to the research domain, their use in industry has become increasingly common. Reasons for this include factors such as significant reductions in the cost of sensors and the corresponding increase in the availability of large data sets (Reis et al., 2015), as well as an increase in the complexity of the systems being modelled and the associated understanding of how these can be represented mathematically (e.g. Mount et al., 2016). Morever, the availability of many open-source software libraries—such as *scikit-learn* (Pedregosa et al., 2011) or *caret* (Kuhn, 2008) — and tailored implementations for GPUs, like TensorFlow (Abadi et al., 2016) and pytorch (Paszke et al., 2019), have made ANN implementations available to a wider group of users. Graduate and undergraduate courses in machine learning have also started to permeate curricula in environmental science and engineering, increasing their level of understanding and acceptance.

ANNs are also playing an increasingly important role in integrated environmental modelling. For example, they can act as surrogate- or meta- or emulation-models to replace all, or portions of, computationally expensive simulation models to facilitate repeated model applications, as is the case in optimisation and sensitivity analysis studies (e.g. Broad et al., 2015; Castelletti et al., 2012; Razavi et al., 2012). Another example can be drawn from climate science, where ANNs are being used increasingly to solve parameterization issues or other computationally-challenging aspects of climate models (Gentine et al., 2018). In addition, ANNs can be used as management models designed to replace a number of linked process-based models by a single input/output model, which can generally overcome some of the data integration and software platform incompatibility issues associated with integrated process-based models, therefore reducing model complexity and the potential for error accumulation. To this end, they can also be used for operationalizing digital twins in environmental applications (Pylianidis and Athanasiadis, 2022; Pylianidis et al., 2022).

Despite their widespread application, the inner workings of ANNs are often not well understood by environmental modellers, who are generally quite content to repeat the rhetoric commonly associated with the use of these types of models, such as “*ANNs work well because they learn from examples*”, “*complex ANNs are better able to deal with large amounts of information*”, “*ANNs are black boxes*” and “*ANNs need large amounts of data*”. While some of these statements are valid in their original computer science context, they are often misleading, or even untrue, in the context of environmental modelling (Maier and Dandy, 2000b). As a result of the perpetuation of these myths, the principles and level of rigor commonly applied to the development of other types of environmental models are generally not applied to the development of ANNs (Maier et al., 2010; Wu et al., 2014). Examples include the lack of consideration of the selection of appropriate model inputs, the way the available data should be divided into the subsets needed for model development, how to select an appropriate number of hidden nodes, how to deal with the issue of parameter identifiability, how to obtain confidence limits on predictions, how ANN models should be validated and how models should be deployed in operational environments (Cabaneros et al., 2019; Maier et al., 2010; Wu et al., 2014). This can have a detrimental impact on the quality and usefulness of the resulting models, potentially misleading users (e.g. Humphrey et al., 2017). It also makes it difficult to assess and compare research findings (e.g. see Wu et al., 2014) and to make significant research progress on the use of ANNs for environmental modelling (Abrahart et al., 2012).

In order to address the above issues, the purpose of this Introductory Overview on the use of ANNs for prediction and forecasting is to shed light on the inner workings of ANNs by exploding some of the common myths that exist about them and to present a state-of-the-art approach to their development. We believe this will assist those who are apprehensive and sceptical about using ANNs, those who are curious about using ANNs and those who have already used ANNs extensively, but have done so “blindly” without “looking under the hood” to fully understand how ANNs work and what potential implications this might have. This increased understanding and guidance is expected to overcome some of the perceived barriers to the adoption of ANNs and increase trust and confidence in their use and the results they produce.

2. Exploding the myths

We begin this Introductory Overview by exploding some commonly held myths because the way ANNs are perceived has significant flow-on effects on the approach that is taken to their development, and hence what the resulting models are and how they perform. While some of these myths and misconceptions have been the cause of a fascination with ANNs that has resulted in their widespread use, they have also resulted in an unhealthy obsession with certain aspects of the model development process, such as the utilisation of different ANN architectures and “training” algorithms, at the expense of the consideration of other, equally important, aspects, such as input variable selection, data splitting and validation. In addition, they have fostered “magical thinking” that has resulted in ANNs being perceived as having “special powers” that make them exempt from the consideration of issues that are considered important for other types of models (e.g. model parsimony). Consequently, it is important to understand what some of these common myths and misconceptions are, as well as their resulting implications, before presenting a state-of-the-art process for developing ANN models. A summary of these myths and their corresponding realities is given in Table 1 and discussed in more detail below.

2.1. Myths

Myth #1: ANNs are fundamentally different from other modelling approaches. Although ANNs “look” different from other types of models because their structure is inspired by the structure of the human brain (i.e. connected nodes in different layers – see Fig. 1), the way they operate is very similar to, if not identical, to other environmental modelling

Table 1
Common ANN myths and corresponding realities.

	MYTH	REALITY
1	ANNs are fundamentally different from other modelling approaches.	Like all other types of models, ANNs convert a set of model inputs into an output via a (complex) mathematical relationship.
2	ANNs are black boxes.	The mathematical relationship between model inputs and outputs is known.
3	ANNs need more data than other types of models.	ANNs often have lower data requirements than other types of models. They can deal very well with incomplete or missing data.
4	ANNs are different from other modelling approaches as they learn from examples.	As is the case for most other types of models, the unknown parameters of ANNs are obtained by calibration (which in artificial intelligence (AI) jargon is called training).
5	A disadvantage of ANNs is that they are a “prisoner” of data.	All types of models are a “prisoner” of the data used in their development. Recent development has shown that ANNs have a great capacity for generalization.
6	ANNs can be used with confidence if they perform well on an independent validation set.	Good performance on an independent validation set is insufficient - ANNs can only be used with confidence if they produce good answers “for the right reasons”.
7	ANNs perform better if they have more model inputs.	The inputs to ANNs have to be selected carefully to ensure good model performance.

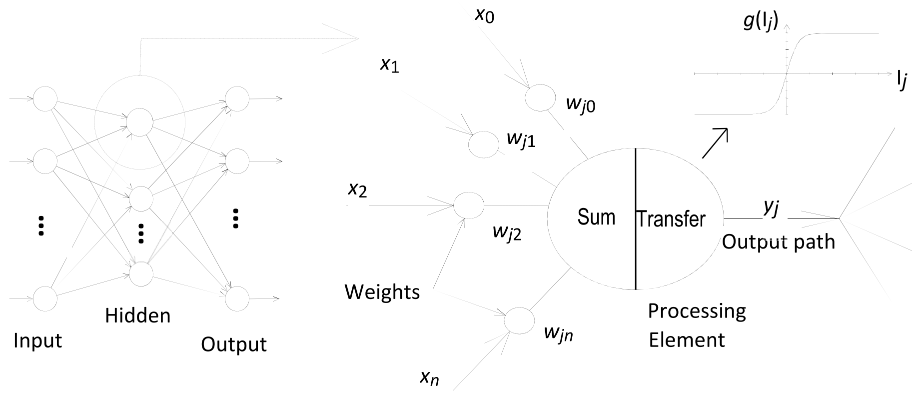

Fig. 1. Typical structure (left) and operation (right) of a Multi-Layer Perceptron.

approaches. As is the case with all models, ANNs predict/forecast variables of interest (dependent variables, model outputs) as a function of one or more model inputs (independent variables, predictors). This is done by converting model input(s) to model output(s) via a mathematical relationship (the functional form of the model) that is usually dependent on a number of unknown parameters. Consequently, ANNs, like all other models, can be represented by the following equation (Mount et al., 2016):

$$y = f(\theta, x) + \varepsilon \qquad \text{(Equation 1)}$$

where *y* is the model output(s) (dependent variable(s)), *x* is the model input(s) (independent variable(s)), θ is a set of model parameters, *f(.)* = is the functional form of the model converting the model input(s) into the model output(s) and *ε* is the random error associated with the model prediction.

For multi-layer perceptron (MLP) ANNs (Werbos, 1975), which are the most widely used ANN for prediction and forecasting (Maier et al., 2010; Wu et al., 2014), the functional form of the relationship converting model input(s) to output(s) is determined by how the nodes in the input layer (i.e. the model inputs) are connected to the nodes in the output layer (i.e. the model output(s)) via hidden nodes and transfer functions (see Fig. 1). Application of Equations 2 and 3 to a particular network configuration (i.e. number of inputs, outputs and hidden layers and nodes) results in a specific mathematical relationship between model inputs and outputs that is a function of a number of unknown model parameters (i.e. the connection weights and bias). *Consequently, while the way in which the functional form is arrived at may be different to other modelling approaches (*e.g. *physically-based models), ANNs are fundamentally the same as other types of models used for prediction and forecasting.*

$$\textbf{Input to } \boldsymbol{j}\textbf{th node}: I_j = \sum_{i=0}^{n} w_{ji} x_i \text{ summation} \qquad \text{(Equation 2)}$$

$$\textbf{Output from } \boldsymbol{j}\textbf{th node}: y_j = g(I_j) \text{ transfer} \qquad \text{(Equation 3)}$$

Myth #2: ANNs are black boxes. The fact that the functional relationship between ANN model inputs and outputs is usually represented as a diagram (Fig. 1), rather than an explicit set of equations, gives rise to the perception that ANNs are black boxes. However, as mentioned above, *the mathematical relationship between model inputs and outputs is known for any given model configuration*. In fact, some ANN configurations correspond to well-known statistical models.

For example, the application of Equations (2) and (3) to the simplest configuration of a MLP, which consists of a network with one input, one output, no hidden layers and a linear transfer function with a slope of 1, results in a linear regression model (Fig. 2) (Maier and Dandy, 2000b). In this case, the connection weight (w) and bias (b) are equal to the slope and intercept of the regression line, respectively. By including additional nodes in the input layer, the artificial neural network model corresponds to a multiple linear regression model. The dimensionality and degree of non-linearity of the model can be changed further by the inclusion of more hidden layers, different numbers of nodes in each of these and different types of non-linear transfer functions at each node. While this increases the complexity of the mathematical relationship, making it more difficult to interpret, the mathematical relationship between model inputs and outputs is still known explicitly.

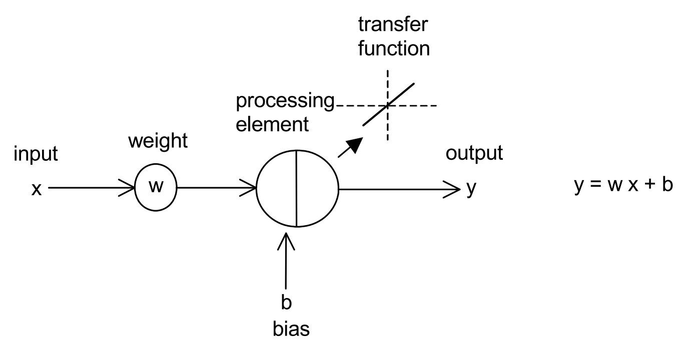

Fig. 2. Artificial neural network representation of a linear regression model.

Therefore, the internals of ANNs are clear and explicit, but because of the often *massively parallel* nature of the working of neurons, it is more difficult to interpret and explain how and why an ANN generates an output in response to a given input. This difficulty arises primarily since environmental modellers are used to modular modelling configurations (as opposed to parallel), which can be understood and interpreted with less effort. Modular configurations break down the underlying system into smaller modules based on prior knowledge, each of which is in charge of modelling a particular process. While such 'reductionism' makes model applications easier in terms of interpretability and explainability, this may limit opportunities to model complex, and possibly yet unknown, relationships. Therefore, ANNs can and should be used with due diligence to ensure their efficacy is accompanied by transparency and explainability (e.g. Binder et al., 2016; Humphrey et al., 2017; Lipton, 2018; Paudel et al., 2023; Zeiler and Fergus, 2014).

Myth #3: ANNs need more data than other types of models. In order to bust the myth that ANNs need more data than other types of models, it is important to recognise that data are used in the development of the vast majority of models, irrespective of model type (Table 2). For all model types, the quality of the developed model increases if the "length" of the available data (i.e. the number of available data points, or "examples") increases, as this increases the chances that different types of "events" or "patterns" (e.g. extremes) are contained in the data used for

Table 2
Influence of data (D) or process understanding (PU) in determining the inputs, structure and parameters of different modelling methods (Adapted from Mount et al. (2016)).

Model Type	Inputs (x)	Structure ($f(.)$)	Parameters (θ)
Data-Driven (e.g. ANNs)	PU/D	D	D
Conceptual (Flexible Structure)	PU	PU/D	D
Conceptual (Fixed Structure)	PU	PU	D
Physically Based	PU	PU	PU/D

model development. Consequently, having greater “data length” is beneficial for the development of all types of models, although this also depends on degree of informativeness of the data (e.g. whether the data contain repeated examples of similar information, such as medium-flow events, or different types of information, such as low-, medium- and high-flow events) (Gupta et al., 2014; Singh and Bárdossy, 2012; Noshad et al., 2021). However, the impact of “data length” is likely to be greater for data-driven models, such as ANNs and traditional statistical regression approaches, as for these types of models, data are generally used to assist with the selection of appropriate model inputs, the determination of an appropriate model structure, as well as values of the unknown model parameters. In contrast, for physically-based models, data are generally only used for the determination of unknown model parameters.

However, when assessing the *amount* of data required for model development, “data length” is only one consideration. The other consideration is the number of variables for which data are needed (i.e. “data breadth/width”). As the structure of more physically-based models is dictated by underlying physical processes, so is the number of variables for which data are required. However, this is not the case for ANNs, which can be flexibly scaled to any available dataset, thereby making best use of the available information. In other words, while the structure of physically-based models dictates data requirements, the structure of ANNs is adjusted based on which data are actually available. Under certain circumstances, ANNs can even work with incomplete data or variables that contain only proxy information about the processes of interest, whereas physically-based models are constrained within their rigid parametrizations that dictate their data demand (see Pylianidis et al., 2022). *Consequently, the “data breadth” required for the development of ANNs is generally less than that required for the development of more physically-based models.*

Myth #4: ANNs are different from other modelling approaches as they learn from examples. In computer science, ANNs are different from rule-based approaches in that they “learn from examples”, rather than requiring all conceivable outcomes to be pre-specified, for example through a set of if-then rules, as is the case with expert systems (see Maier and Dandy, 2000b). However, the vast majority of model types used for prediction and forecasting “learn from examples”, as the values of their unknown model parameters (see Equation (1)) are adjusted incrementally to enable model outputs to better match corresponding measured example outputs. This process is termed model calibration and is typically facilitated by a formal optimisation process. Consequently, as the “training” of ANN models entails the incremental adjustment of unknown model parameters with the aid of an optimisation algorithm to minimise the error between model outputs and a set of corresponding historical data, it should be referred to as model “calibration”, rather than “training”, to avoid confusion. *Recognition of the equivalence of “ANN training” and “model calibration” assists with demystifying the ANN training process and opens the door to applying any of the methods and algorithms that are used for the calibration of other environmental models to ANNs.*

Myth #5: A disadvantage of ANNs is that they are a “prisoner” of data. The primary argument in support of this myth arises from a comparison of ANNs with physically-based models. Such models embed domain knowledge via differential and parameterization equations, as well as their given ranges or values, based on prior observations or experiments. Such domain knowledge may help physically-based models to extrapolate beyond available data, for example by preserving mass balance within a control volume (Razavi, 2021). However, this possible benefit may be at the trade-off of losing data requirement flexibility, and the extent of this benefit still depends on the available data. More broadly, as discussed previously, the quality of any model is a function of “data length”, as models can only be expected to perform well when they interpolate within the range of the data used for their development (Chen et al., 2022; Zheng et al., 2022). *Consequently, all models are generally a “prisoner” of the data used for their development to some extent – this is not a particular disadvantage of ANNs relative to other types of models.*

Myth #6: ANNs can be used with confidence if they perform well on an independent validation set. After the completion of the model calibration (training) process, it is considered good practice to check the predictive performance of the calibrated model on an independent validation set, which is a subset of the available model development data that has not been used as part of the model development (i.e. calibration and/or selection) process. By comparing the predictive performance of the model on the calibration and validation subsets, an assessment can be made as to whether the model has overfit (merely “memorised”) the calibration data, which is the case if validation performance is markedly worse than calibration performance. If this is not the case, it is assumed that the model can be used with confidence for its intended purpose.

However, given that the functional relationship between model inputs and outputs of ANNs is not constrained by any known underlying physical processes, the fact that the model has not overfit to the calibration data is no guarantee that it has captured a representative relationship between the model inputs and outputs. In fact, there are many examples of calibrated ANN models with similar validation performance that have captured very different relationships between model inputs and outputs, some of which agree with known system understanding, while others do not (Humphrey et al., 2017). Consequently, the fact that ANNs perform well on an independent validation set is no guarantee that they can be used with confidence in practice, as any model that behaves in a way that is counter to physical system understanding is unlikely to be trusted. As a result, ANNs not only need to perform well on an independent validation set, but they need to do so “for the right reasons” (Li et al., 2015c). In other words, *a given validation performance needs to be accompanied by verification that the calibrated ANN model does so while conforming to any known understanding of the physics of the relationship being modelled* (Humphrey et al., 2017). Hybrid ANNs that include process-related knowledge in the form of physical constraints in loss functions, parameter values, or even model structure, may tackle such limitations and allow for a new generation of models that reduces model variance by removing physically inconsistent solutions, generally without affecting their bias (e.g. Hunter et al., 2018; Karpatne et al., 2017; Kingston et al., 2005a).

Myth #7: ANNs perform better if they have more model inputs. There is a common misconception that ANNs perform better when they have more inputs (it should be noted that this refers to more “data breadth”, rather than “data length” - providing greater “data length” is definitely a good idea) (see Myth #3). However, the inclusion of a larger number of model inputs has a number of negative consequences, especially as a larger number of inputs also results in an increase in the number of connection weights (i.e. parameters to tune during model calibration) and, more generally, higher degrees of freedom. This makes it more difficult to identify the combination of parameter values that minimises the calibration error, reducing both the computational efficiency and likely success of the calibration process (see Zhu et al., 2022a) (see Myth #4). Consequences of this include increased difficulty in being able to assess true model performance, and hence which model structure (i.e. number of hidden nodes) is most appropriate, as it is unclear whether the relative performance of models with different numbers of hidden nodes is due to model structure or the inability to identify the combination of model parameters that maximises model performance for a network with a given number of hidden nodes. In addition, the inclusion of a larger number of inputs increases the likelihood of the presence of inputs that provide irrelevant (e.g. not related to the model output) or redundant (e.g. correlated with each other) information, which means that different combinations of parameter values are more likely to result in similar model performance. This causes parameter non-uniqueness (Guillaume et al., 2019), increasing the likelihood that there are different calibrated models that have similar validation performance, but have captured very different relationships between model inputs and outputs (see Myth #6). *Given the potential negative consequences of including irrelevant or correlated inputs, the inputs to ANNs have to be selected carefully to ensure good model performance* (Galelli et al., 2014).

2.2. Implications

The implications of the above myths are far-reaching, as they can present barriers to the adoption of ANNs and lead to poor model-development practices that result in models that do not stand up to rigorous examination, thereby decreasing trust and confidence in the use of ANNs. As shown in Fig. 3, busting Myths #1 and #2 is vital to ensuring this is not the case. The realisation that *ANNs are not fundamentally different* from other types of models, and are therefore not black boxes, is key to modellers not viewing ANNs as having "special powers" that can magically transform a large amount of data into robust and accurate predictions. The recognition that the inner workings of ANNs are essentially the same as those of other models makes a compelling case for the need to adopt state-of-the-art model development processes, which, when applied to the determination of appropriate model inputs (X), parameters (θ) and functional relationships between model inputs and outputs (f(X,θ)+ε), can assist with busting the remaining five myths (i.e. Myths 3 to 7), thereby reducing barriers to the adoption of ANNs and increasing trust and confidence in their use (Fig. 3).

Busting Myths #3 and #5 assists with reducing the barriers to adopting ANNs, as this (i) clarifies that the input data requirements of ANNs (X) can, in fact, be less than those of alternative modelling approaches, as their structure can be adapted to obtain predictions that make best use of available data in order to provide useful information, rather than requiring data for a set of pre-determined input variables (Myth #3), and (ii) highlights that the way data are used in the determination of the unknown ANN model parameters (θ) as part of the calibration (training) process is identical to the way this is done in other models, making them no more a prisoner of the data that are available for model development than is the case for other types of models (Myth #5).

Exploding Myths #4, #6 and #7 is key to increasing trust and confidence in ANN models. However, achieving this requires the adoption of state-of-the-art model development processes. For example, the elimination of any superfluous model inputs (X) via the adoption of appropriate input variable selection approaches (Myth #7) (see Section 3.3) results in smaller, more parsimonious models, increases the identifiability of model parameters and reduces the difficulty of the calibration process, which assists with ensuring that the relationship that is captured by the resulting model is in accordance with the understanding of underlying physical processes. The recognition that ANN "training" is identical to model calibration (Myth #4) opens the door to the adoption of commonly used optimisation methods for the determination of appropriate values of the unknown model parameters (θ) (see Section 3.6.2). This is likely to result in the identification of better model parameters, as well as opportunities to obtain uncertainties associated with parameter estimates. In addition, it could also remove barriers to the adoption of ANNs by de-mystifying the ANN "training" process. Checking whether ANN models (y = f(X,θ)+ε) behave in accordance with expectations based on an understanding of the underlying physics by expanding validation approaches to go beyond the assessment of model performance on an independent validation set (Myth #6) (see Section 3.7) is also key to increasing trust and confidence in ANNs, as well as breaking down potential barriers to their adoption. Details of the state-of-the-art model development process for achieving these outcomes are given in the next section.

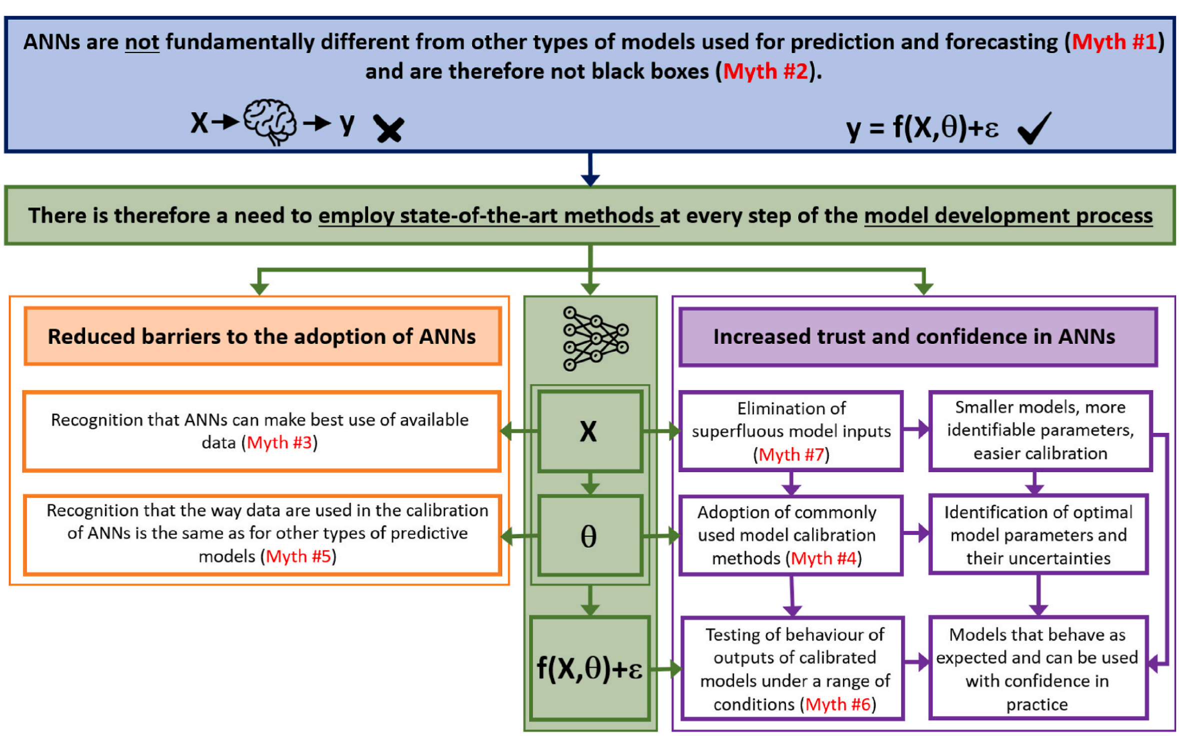

Fig. 3. Implications and positive impacts of exploding the commonly-held myths about ANNs.

3. State–of–the-art ANN model development process

3.1. Overview

As mentioned in the previous section, the adoption of state-of-the-art model development processes is key to increasing trust and confidence in ANNs and hence reducing barriers to their adoption. Consequently, the focus of this section is on the presentation of such a process, covering all elements needed for the successful development of ANNs for prediction and forecasting (see Fig. 4, adapted from Maier et al., 2010; Wu et al., 2014). It should be noted that these steps assume that the primary modelling objective (see Hunter et al., 2018) has already been established.

The first step in the process is the choice of appropriate model output (s) (i.e. the variable(s) to be predicted) and a set of *potential* model input variables. Although ANNs are data-driven models, it is up to the modeller to choose which input variables should be *considered* as part of the model development process. This can be done based on *a priori* knowledge and data availability (see Galelli et al., 2014). The resulting data set constitutes the "Selected Data (Unprocessed)". It should be noted that once the model outputs have been chosen, the number of nodes in the output layer of the ANN has also been determined (Fig. 4).

Next, the "Selected Data (Unprocessed)", which consist of measured values of the potential model input variables, as well as the model output variable(s), have to be processed so that they are in a suitable form for the subsequent steps of the model development process. Once the "Selected Data (Processed)" of potential model inputs and outputs have been assembled and the data have been transformed, the actual model can be developed. Deletion of all unimportant and redundant potential inputs from the "Selected Data (Processed)" results in the "Model Development Data", which consist of transformed values of relevant, non-redundant model inputs and corresponding model output (s). These data can then be used for the optimisation of the structure and parameter values of the selected model architecture, as well as model validation, once they have been split into appropriate subsets (Fig. 3).

After the "Optimal Model" has been determined, it needs to be validated. The objectives of model validation are to ensure the model (i) has captured the underlying relationship in the calibration data (replicative validity), (ii) can generalise over the range of the calibration data (predictive validity) and (iii) is plausible when compared with *a priori* knowledge (structural validity). Once the model has been validated, it can be deployed in an operational setting. Details of each of these steps are given in the following sub-sections.

3.2. Data processing

The primary objectives of the data processing step (Step 2, Fig. 4) are (i) to obtain a set of data that contains all potential model inputs and outputs, as distinct from the potential input and output *variables* selected previously, and (ii) to ensure that all potential input variables have equal representation during model calibration, even though different variables are likely to span different numerical ranges.

The first of these objectives is only applicable to time series problems and can be achieved by incorporating appropriate system dynamics (time structure, such as auto-regressive processes) into the model inputs and outputs by lagging the input and/or output time series. Similar concepts also apply to spatial problems. The second objective is achieved by transforming all potential input and output variables to a range that is commensurate with the limits of the transfer functions used and generally involves either re-scaling or standardisation. The "Selected Data (Processed)" resulting from the lagging and transformation processes contains the transformed values of the potential model inputs and the model output(s).

3.2.1. Lagging

In time series applications, the potential model inputs not only consist of current values of the potential input variables (e.g. flow_t, temperature_t), but also their past values (e.g., flow_{t-1}, flow_{t-2}, …, flow_{t-k}; temperature_{t-1}, temperature_{t-2}, ….., temperature_{t-k}) to represent system memory. This requires users to select the maximum time lag beyond which the input variable of interest is unlikely to have an effect on model output(s), which can be done with the aid of *a priori* knowledge about the system.

3.2.2. Data transformation

In general, model inputs consist of variables that span different ranges. For example, river flows might vary from ~100 ML/day to over 100,00 ML/day, while the corresponding river levels might vary from 0.1 m to 10 m. As weighted values of different model inputs are summed at each of the hidden neurons (Equation (2)), variables that have larger values (e.g. flow) generally have a much greater influence on the resulting activation levels I_j (Equation (3)) than variables that have smaller values (e.g. river level). To ensure that all variables have a similar chance to influence the model, they need to be transformed.

Linear transformation can take on different forms, the most common being re-scaling and standardisation. Re-scaling generally refers to the scaling of data between upper and lower bounds. In general, data are standardised to a mean of zero and standard deviation of one (Sarle, 1997). If bounded activation functions are used in the output layer, it is also necessary to scale the output data to ensure that they are within the limits of these functions. For example, if the hyperbolic tangent activation function is used in the output layer, the output of which is bounded by $[-1, 1]$, the data should be scaled to $[-0.9, 0.9]$ or $[-0.8, 0.8]$. Although it is common to use an unbounded activation function in the output layer, it is still good practice to transform the output. It should be noted that distributional transformations of the input and output data are generally not required, but may be considered in certain circumstances (e.g. when the data are highly skewed) (see Bowden et al., 2003).

3.3. Input selection

The vector of appropriate model inputs is determined during the "Input Selection" step (Step 3, Fig. 4). The difficulties associated with this step are twofold: first, the modeller is unlikely to know the functional relationship between input variables and output (i.e. f(.), Equation (1)); second, the exhaustive search of the input space is often a computationally-intractable problem (consider that there exist 2^p possible subsets of input variables, with p being the number of candidate inputs). Moreover, the cross-correlation between inputs induces redundancy and collinearity in the input pool (Maier et al., 2010). The benefits of Input Variable Selection (IVS) are therefore not limited to the pragmatic matter of identifying the smallest subset that includes all relevant input/output relationships contained in the "Model Development Data". By selecting such a subset, IVS also improves model accuracy (by removing irrelevant and redundant inputs), improves other steps of the model development process (e.g., data splitting or calibration), and results in a model that is easier to interpret (Galelli and Castelletti, 2013) (see Myth #7). For these reasons, the use of IVS algorithms is a necessary step of ANN model development (see Bowden et al., 2005a,b; Fernando et al., 2009; Galelli et al., 2014; Maier and Dandy, 1997; May et al., 2008a,b; Taormina et al., 2016).

IVS algorithms can be categorized as *model-based* (or *wrappers*) and *model-free* (or *filters*), depending on the way with which the input relevance is evaluated (Guyon and Elisseeff, 2003; Maier et al., 2010). Model-based algorithms rely on the idea of solving the input selection problem during the ANN calibration (Section 3.6). In other words, when using these algorithms, multiple ANNs with different subsets of inputs are calibrated and then the one yielding the highest prediction accuracy is selected. The search process is typically informed by global optimisation techniques (such as evolutionary algorithms) that determine the combination of inputs that maximises model performance (Bowden

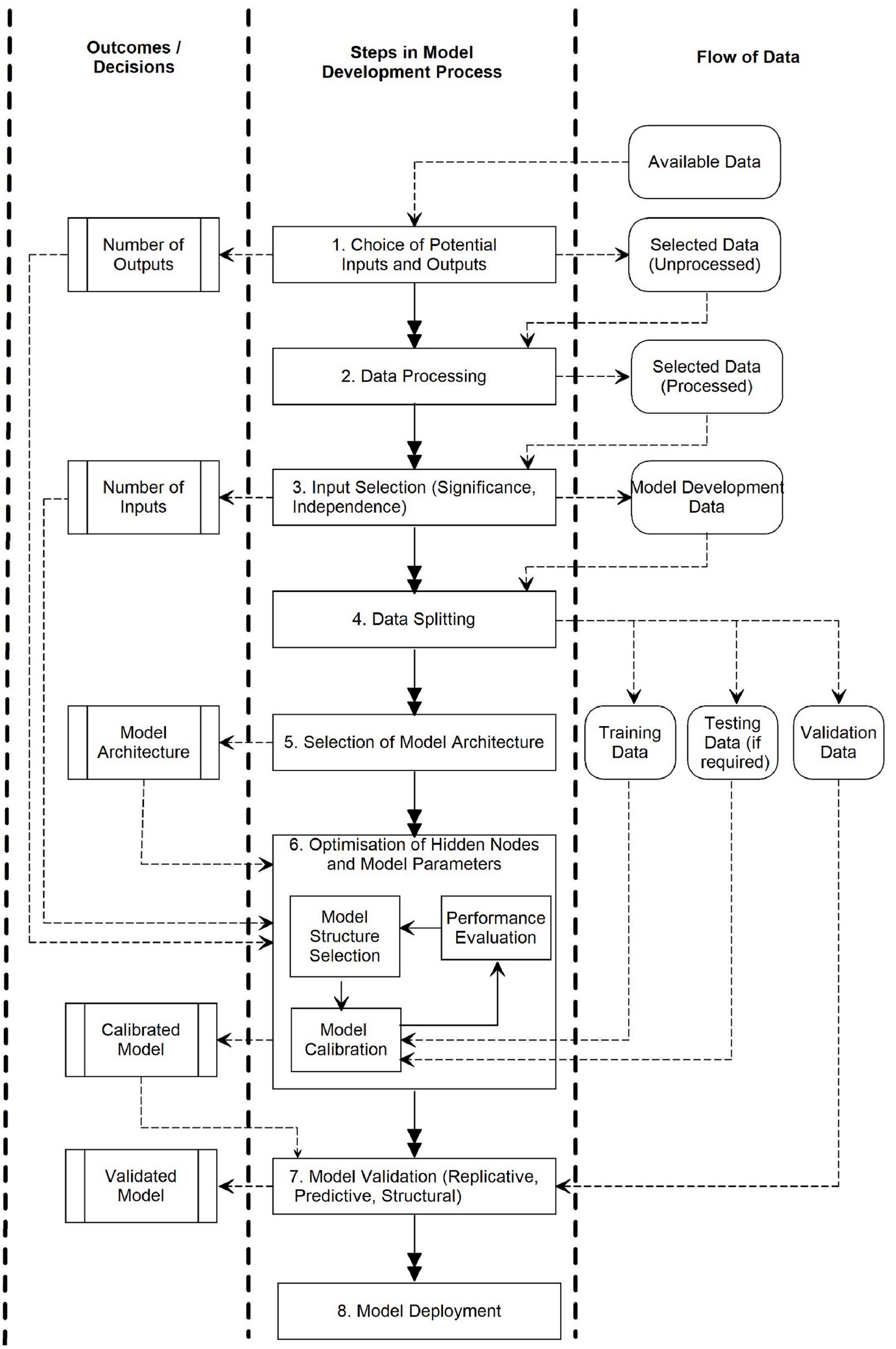

Fig. 4. Steps in ANN model development process.

et al., 2005b; Tirelli et al., 2009) (see Section 3.6). Because of this global search strategy, model-based algorithms explicitly account for interactions and dependencies between inputs—a key point when dealing with datasets characterized by strong collinearity. Their main disadvantage is their high computational requirements, as a large number of calibration runs is necessary to identify the best input subset (Chow and Huang, 2005). Thus, model-based algorithms may not be the best choice when dealing with large datasets. A second limitation is that the optimality of the selected subset is a function of the pre-defined ANN architecture and the quality of the calibration process. This can lead to misleading results (see Myth #7 and Section 3.6.2) and can restrict the application of the selected subset to the pre-determined architecture

(Maier et al., 2010). More recent approaches tackle this problem by concurrently optimizing input subset and model architecture (e.g. Taormina and Chau, 2015) or by identifying equally-informative subsets (Karakaya et al., 2016; Taormina et al., 2016).

Model-free algorithms (filters) rely on the information content of the available data, *a priori*, to measure the relevance of the inputs—and therefore do not depend on an underlying ANN. The simplest form of such a filter is represented by linear measures of correlation, which are still rather popular in environmental modelling (Maier et al., 2010; Wu et al., 2014). In this group, two common measures are the autocorrelation/partial autocorrelation (Box and Jenkins, 1976) and the Partial Correlation Input Selection (May et al., 2008a). Algorithms falling in this group are computationally efficient, but naturally tend to underperform when dealing with nonlinear input-output relationships. Such limitation has prompted many recent studies on the use of information theoretic-based dependency measures, which make no assumptions on the structure of the dependence between two variables. Measures commonly used for the development of ANNs are the Mutual Information, Partial Mutual Information, and minimum-redundancy maximum-relevance (Fernando et al., 2009; Hejazi and Cai, 2009; Li et al., 2015a, 2015b; May et al., 2008a,b; Sharma 2000). It is worth pointing out that all model-free algorithms suffer from the same limitation: both linear and nonlinear measures of dependence are typically univariate, meaning that the relevance between each potential input and output variable is considered separately. In turn, this means that the potential interactions between inputs may be ignored.

So, which IVS approach should a modeller use when identifying an ANN? The computational requirements of any IVS algorithm are a function of the number of observations and candidate inputs (Galelli et al., 2014), so it is advisable to use information on the available computational resources and characteristics of the dataset at hand to make a first decision on the IVS approach. For example, a regular laptop can easily support a model-based algorithm when working with fewer than 10–15 candidate inputs and a few thousand observations. A second consideration may be driven by the nature of the input-output relationships (e.g. linear vs nonlinear), which could, for example, be leveraged when choosing a specific measure of dependence. When these relationships are unknown, it is naturally advisable to rely on nonlinear measures, so as to fully exploit the potential of ANNs.

3.4. Data splitting

In most real-life applications, the data available for the development of predictive models are limited. Consequently, careful consideration needs to be given to the way the available data are used for model development and evaluation so as to develop the best possible model, given the available data (Maier et al., 2023). The general approach for dealing with this problem is to divide the data into model development (e.g. calibration/training and testing) and model evaluation (e.g. validation) subsets.[1] The model development subset is used to determine values of the unknown model parameters and model structure via calibration and testing, respectively, whereas the validation subset is used to evaluate whether the calibrated model can be used with confidence for its intended purpose.

In ANN modelling, the model development data are often divided into calibration (training) and testing subsets. The calibration subset is used to adjust the model parameters (e.g. connection and bias weights) and the testing subset is used to decide when to stop training in order to avoid overfitting and to decide which model structure and parameters controlling the optimisation algorithm used during model calibration are optimal. Overfitting of the calibration/training data (commonly referred to as overtraining) is a potential problem when calibrating ANN models, as the available data are often noisy, and more importantly, the degrees of freedom of the models relative to the number of training data can be large. In other words, if the number of free model parameters (i.e. connection and bias weights) is large compared with the number of training data points, and the data are noisy, there is a danger of "memorising" the data points in the training set, rather than learning the underlying relationship (Fig. 5).

By using a testing set, the generalisation ability of the model can be assessed on an independent data set (i.e. a data set that is different from that used to drive the parameter-adjustment process) during the parameter optimisation process. As ANNs "learn" the underlying relationship in the data, the errors obtained using the calibration/training and testing sets should be similar and decrease at approximately the same rate (Fig. 6, region AB). If the ANN model has sufficient degrees of freedom, the error obtained using the calibration/training data will continue to decrease as parameter optimisation progresses. However, if the calibration/training data are noisy, this reduction in error might be at the expense of generalisation ability, as illustrated in Fig. 7 (b). In the extreme case, a set of parameters can be found that enables the relationship that is represented by the ANN to pass through all data points in the calibration/training set, thus reducing model error to zero. However, this is not desirable, as this means that the model has "memorised" the specific set of calibration/training data and has lost the ability to generalise. As the testing set monitors the model's generalisation ability during calibration, overtraining, as indicated by an increase in the error obtained using the testing set while the error obtained using the training set continues to decrease, can be detected (Fig. 6, region BC). In this way, the testing set can be used to decide when to stop calibration in order to avoid overfitting (Fig. 6).

When splitting the data into the model development (calibration/training and potentially testing) and validation subsets (Step 4, Fig. 4), the objective is to ensure that the resulting subsets have similar statistical properties/contain the same "patterns" or "events" (Maier et al., 2023). In order to achieve this, data points from each of the representative regions of the multidimensional input/output space need to be included in each of the data subsets. This is because all patterns that are contained in the available data should be used for model calibration to ensure the model can perform over as wide a range of conditions as possible. At the same time, the same patterns should be incuded in the validation data so as to test the performance of the calibrated model over the same range of "patterns" or "events". If this is not the case, the evaluation of model preformance is likely to be misleading (Chen et al., 2022; May et al., 2010; Wu et al., 2013; Zheng et al., 2018). In order to achieve representative data splits, different approaches can be used, such as using formal optimisation methods to mininise the differences between the statistical properties of the different data subsets (e.g. Bowden et al., 2002) and dividing the available data into statistically representative regions (e.g. by using clustering approaches such as self-organising maps (SOMs) (Kohonen, 1995)), from which samples are drawn for each of the subsets (e.g. Chen et al., 2022; Guo et al., 2020; May et al., 2010; Wu et al., 2013; Zheng et al., 2018; Zheng et al., 2023).

3.5. Selection of model architecture

After the available data have been pre-processed, the model inputs and outputs have been determined and the available data have been split into model development (calibration and/or testing) and evaluation (validation) subsets, an appropriate model architecture has to be selected (Fig. 4). Although MLPs (Fig. 1) have traditionally been used most commonly for prediction and forecasting in environmental applications and represent the "iconic" ANN architecture, there are a number of alternatives, such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, Graph Neural Networks (GNNs), and Physics-informed Neural Networks (PINNs). While all of these network architectures convert the selected model inputs (X) into the desired

[1] In the machine learning/AI community, what we refer to here as the validation set is often referred to as the testing set and what we refer to here to as the testing set is often referred to as the validation set.

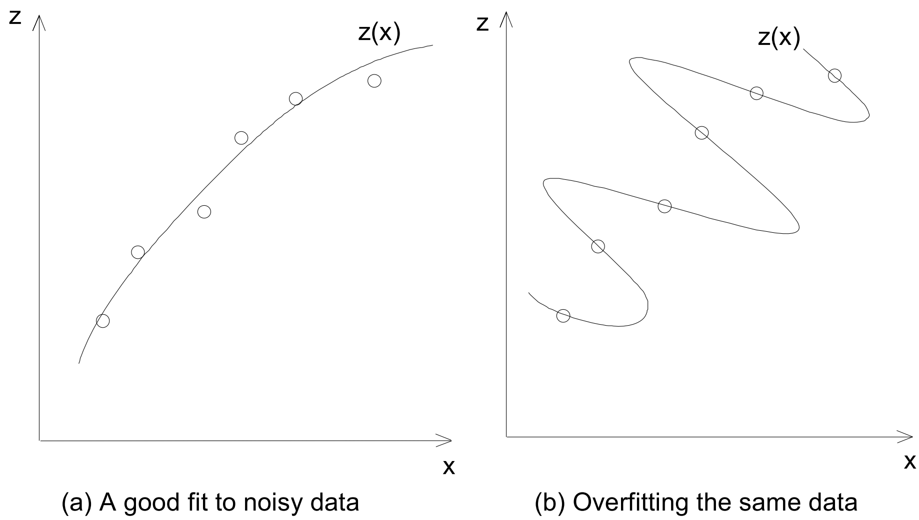

Fig. 5. Overfitting of calibration/training data.

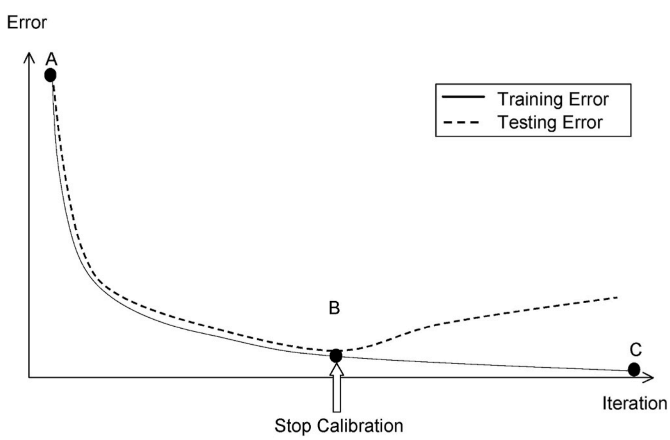

Fig. 6. The phenomenon of overtraining. Calibration/training error is estimated using the calibration/training dataset. The testing dataset is used as a proxy for estimating the validation error, during the model calibration process.

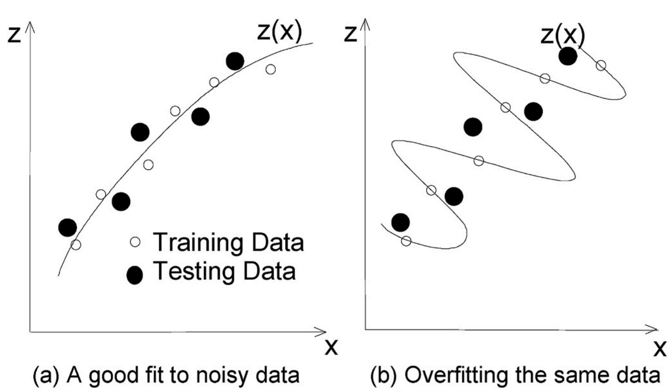

Fig. 7. Impact of overtraining on training and test set errors.

model output(s) (y), the functional form this transformation takes (f(.)) and the parameters that have to be determined by calibration (w) can be very different.

RNN architectures (Hochreiter and Schmidhuber, 1997) are particularly useful for analysing sequential data, which can be valuable when considering time-dependent processes. RNNs are designed to capture and learn patterns in sequential data by processing each input element while maintaining an internal memory. This memory allows RNNs to retain information from previous inputs, enabling them to model dependencies and context over time. By leveraging their ability to learn from historical data and capture temporal dynamics, RNNs offer powerful tools for understanding and predicting environmental phenomena.

Fig. 8 depicts the basic form of an RNN, where each step in the sequence processes an input and passes its output as input to the next

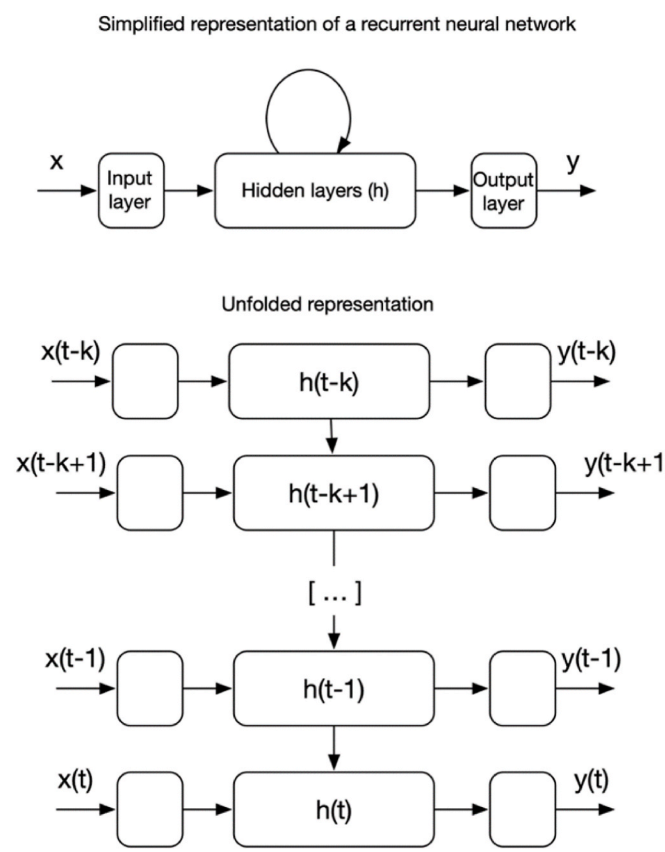

Fig. 8. Simplified representation of a RNN architecture and its unfolding over time to articulate the time dependencies within the hidden layers. The internal structure of the hidden layer can attain various levels of complexity in order to achieve the necessary features to retain the information of the previous time steps.

step. Practically, it implements the temporal lagging of inputs within the neuron structure, using temporal feedbacks. Such architectures are intended for timeseries forecasting problems, but suffer from the "vanishing gradient" problem, which hampers their ability to capture long-term temporal dependencies. Their key-advantage is that they can handle time series as inputs of varying length, rather than fixed-length vectors that are required for MLPs.

Long Short-Term Memory (LSTM) models (Yu et al., 2019) introduce a memory cell that can retain information over longer sequences. They incorporate a gating mechanism to control the flow of information, allowing it to selectively remember or forget information from previous time steps. This makes LSTM models more effective at capturing long-term dependencies (Zhou et al., 2021). Gated Recurrent Unit (GRU) (Cho, 2014) is a similar architecture to LSTM in that it also utilizes gating mechanisms. However, GRU has a simplified architecture with fewer gates compared with LSTM, reducing the complexity of the model while still capturing long-term dependencies. For a comprehensive review of the first wave of recurrent neural networks see Lipton et al. (2015).

An example application to forecasting environmental processes is provided by Huang and Kuo (2018) where a LSTM network is fed by a convolutional neural network (CNN) which extracts the relevant features from PM2.5 concentrations, cumulative wind speed and cumulative rain over the previous 24 h. Although CNNs have been traditionally applied to tasks such as character recognition and image processing (Ciresan et al., 2010), their capability to extract features from data in an unsupervised manner has been successfully exploited to feed the LSTM architecture with data that is useful to forecast PM2.5 concentrations a day ahead.

More modern ANN architectures for sequential data include encoder-decoder and transformer architectures, both of which are inspired by and have revolutionized machine translation. Encoder-Decoder architectures (Sutskever et al., 2014) are particularly suited to problems for which input and output sequences are not temporally aligned. They consist of two major components: an encoder that takes a variable-length sequence as input, and estimates a compressed (latent) representation; and a decoder that takes the compressed representation and generates a variable length sequence as an output, one step at a time. For instance, Kumar et al. (2021) used an encoder-decoder architecture to predict soil moisture, humidity, solar radiation and other micro-climatic variables from sensor data for agricultural applications. The encoder layer is composed of parallel blocks of ANNs (mostly CNNs) that focus on different time scales. An attention mechanism (see below) selects the encoder output to feed the decoder, which is composed of an LSTM and a number of fully connected MLPs. The encoder returns the micro-climate prediction of interest. The advantage of the encoder is the ability to compact the information of the various time series in the most efficient way, capturing their interdependency, so that the decoder layer can produce forceasts of higher quality when compared with those obtained using standard ANN architectures.

Graph Neural Networks (GNNs) (Sanchez-Lengeling et al., 2021) represent an alternative model architecture that has been developed in the research area of Geometric Deep Learning, which attempts to generalise deep neural models to non-Euclidean domains, in this specific case to graphs. The basic idea of a GNN is to integrate the graph structure in the weight matrix so that nodes are no longer treated as independent observations, as they would be in a fully connected neural network. GNNs exploit information about the spatial relationship of data sources in order to capitalise on the information content of the dependency among data implied by the graph structure. For example, when predicting water supply quantities in a water distribution network, the value in a specific stem of the network depends on its predecessors. This information can be efficiently used in a GNN, as shown by Zanfei et al. (2022), who used a graph convolutional recurrent neural network to predict a time series of water demand related to a number of water supply systems belonging to the same geographical area. GNNs have also been used to increase the accuracy of crop yield forecasts (Fan et al., 2022) and to predict the spread of the West Nile virus (Tonks et al., 2022).

Generative models, such as Generative Adversarial Networks (GANs, Goodfellow et al., 2014) are currently receiving siginficant attention because of their good performance in the production of artificial images and synthetic texts, but are also starting to be employed in the environmental modelling domain. For example, DeepMind published a paper (Ravuri et al., 2021) where deep generative models have been used to predict rainfall at a very high resolution (nowcasting), demonstrating a notable increase in accuracy. In the words of the authors "… as generative models are fundamentally probabilistic, they have the ability to simulate many samples from the conditional distribution of future radar given historical radar, generating a collection of forecasts similar to ensemble methods.".

Another successful type of generative models are transformer architectures that employ an "*attention mechanism*" (Vaswani et al., 2017) that learns a local context for learning "sequence to sequence" problems. This mechanism has been instrumental to the success of recent large language models, including BERT and GPT, and they are now being used in the environmental domain. A recent paper by Grigsby et al. (2023) also shows how transformer models can be used for long-range spatio-temporal forecasting. The authors claim that the transform architecture can overcome some limitations of GNNs, in particular the requirement of an adjacency matrix to explicitely represent the spatial relationship and the requirement of performing separate temporal and spatial updates. The proposed approach is based on "spatio-temporal embeddings", that is the encoding of spatio-temporal time series in vector representations (embeddings) in a way roughly similar to the word embeddings of NLP. Research in these applications of transformer models in the environmental modelling domain is just starting.

Physics-informed Neural Networks (PINNs) (Raissi et al., 2019) are an interesting option when we already possess a basic understanding of the physical processes underlying the data generation process. If we can formalise our understanding of the causal relationships among the system's variables using ordinary differential equations (ODEs) or partial differential equations (PDEs), then PINNs allow us to incorporate such knowledge in the network structure, thus reducing the amount of data needed for model development. Typically ODEs and PDEs are incorporated seamlessly in the loss function used to develop ANNs, enforcing the physical feasibility of the solution and reducing the breadth of the search space (see Fig. 9 for a schematic representation). An introductory review is presented by Karniadakis et al. (2021) and applications to climate and weather predictions seem to be very promising: Kashinath et al. (2021) present a review of various approaches to apply pre-existing knowledge on the physics of natural phenomena to machine learning problems, such as the use of symmetry and invariancy relationships, stabilty requirements, and spatio-temporal coherence. Among other examples, they report the work of Manepalli et al. (2019), as part of which a conditional GAN (generative adversarial network) is used to emulate a physiscs-based model of the spatial distribution of mountain snowpack. The physics is incorporated by remarking the areas of higher elevation usually covered with more snow and the GAN is penalized for large errors made in these areas.

3.6. Determination of Model Structure and Parameters

Selection of a particular model architecture is insufficient to define the mathematical relationship between the model output(s) and a set of model inputs (Equation (1)). This also requires determination of the model structure for the selected architecture (e.g. number of hidden layers, number of hidden layer nodes, degree of connectivity of nodes, types of transfer functions etc.), as well as determination of the values of the unknown model parameters (e.g. connection weights and biases) for the selected model architecture and structure via calibration/training. Once a model with a given structure has been calibrated, it can be used

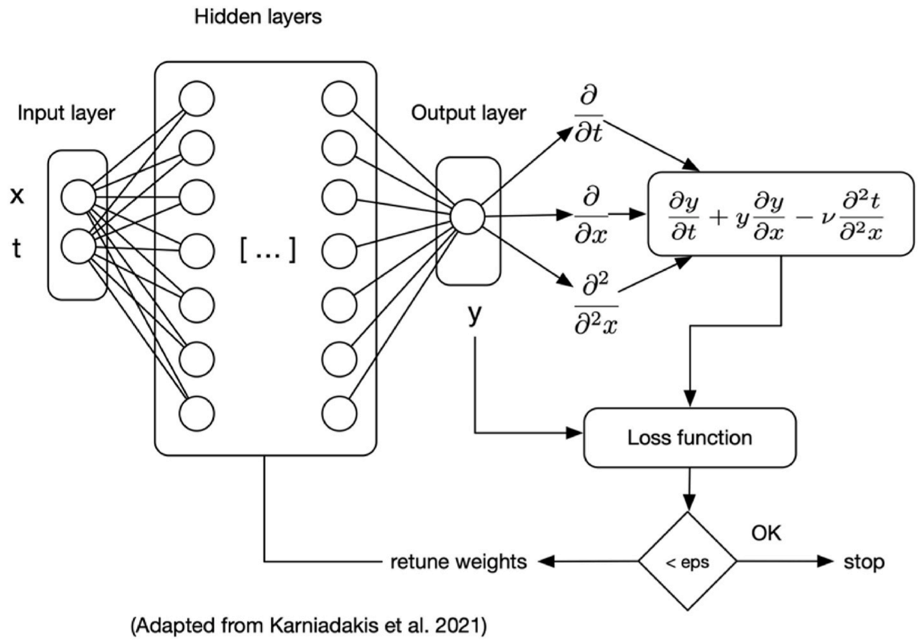

Fig. 9. Simplified representation of a Physically-informed Neural Network (PINN).

to obtain a set of model outputs, given a corresponding set of model inputs. However, as ANNs are data-driven models, the most appropriate functional form of the desired input/output relationship (i.e. the optimal model structure) cannot be determined *a priori*. Instead, there is a need to evaluate the performance of a number of models with different structures using an appropriate performance measure, necessitating repeated application of the "Model Structure Selection", "Model Calibration" and "Performance Evaluation" steps in Fig. 4, which are discussed in more detail below.

3.6.1. *Model Structure Selection*

As mentioned above, the combination of model architecture and structure defines the functional form of the relationship between model inputs and output(s) (i.e. f(.) in Equation (1)). For example, for models with a MLP architecture, determination of an appropriate model structure involves the selection of a suitable number of processing elements, how they are arranged (e.g. number of hidden layers, number of nodes per layer), how they are connected (e.g. fully connected, feedforward connections) and how they process incoming signals (e.g. type of transfer function etc.) (see Myth #2). The optimal network structure generally strikes a balance between generalisation ability and model complexity. If model complexity is too low or an inappropriate functional form is selected, the model might be unable to capture the desired relationship. However, if model complexity is too high, the model might have decreased generalisation ability and processing speed, could be more difficult to calibrate and might be less transparent (see Myth #7).

Optimal model structure is generally determined with the aid of the available data (see Table 2). However, while it would be possible to vary all factors that influence model structure (e.g. number of hidden layers, number of nodes in each hidden layer, type and degree of connectivity and type of processing at each node in the case of an MLP) in order to find the combination that is best suited to the available data, this is generally not done due to the large search space involved. Numerous techniques have been suggested to make decisions regarding network structure less arbitrary, such as growing and pruning algorithms (Reed, 1993), statistically based comparison procedures and 'rules of thumb'. However, the most used method for selecting the number of hidden layer nodes is trial-and-error, where a number of ANNs are calibrated until the best-performing model is found (Maier et al., 2010; Wu et al., 2014).

Which model structure is "best" is generally assessed with the aid of a range of performance metrics, such as root mean square error, mean average percentage error, the coefficient of correlation etc. (see Dawson et al., 2007), as is the case for any predictive model. However, given that the structure of ANNs is determined as part of the model development process and that ANNs can potentially have many parameters, the use of performance metrics that balance model complexity with predictive performance, such as the Akaike Information Criterion (AIC) (Hagiwara et al., 1993; Humphrey et al., 2017) or the Bayesian Information Criterion (BIC) is worth considering (Kingston et al., 2008; Mei and Smith, 2021).

3.6.2. *Model calibration (training)*

The aim of model calibration is to identify a set of model parameters that enables a model with a given functional form (i.e. architecture and structure) to best represent the desired input/output relationship. In almost all ANN applications, deterministic model calibration approaches are used, which try to obtain a single, optimal parameter vector. If overfitting is not considered to be a problem and the training data are representative of the modelling domain, this is achieved when ANN model performance is maximised, which generally occurs when a suitable error measure between actual and predicted training outputs is minimised. If overfitting is a possibility, optimal generalisation ability is achieved when a suitable error measure between actual and predicted outputs in the test set is minimised, provided that training and testing data are representative of the modelling domain (see Section 3.4).

Determination of the combination of model parameters that minimises the training or testing error is not a simple problem. As each combination of parameters generally results in a different model error, an error surface exists in parameter space (see Maier et al., 2019). This is illustrated for a model with a single parameter in Fig. 10, where different values of the model parameter generally result in different model errors (Kingston et al., 2005a; Maier et al., 2019; Zhu et al., 2022b). As can be seen, the degree of difficulty in finding the combination of parameters that results in the smallest model error is affected by the "ruggedness" of the error surface. Ruggedness is a measure of the number, spacing and steepness of the craters and valleys in the error surface (see Zhu et al., 2022a). If the error surface is smooth (Fig. 10(a), there are fewer local minima, and the global optimum can be found more easily. In contrast, as illustrated in Fig. 10(b), if the error surface is more rugged, it generally has more local minima, and the global optimum is more

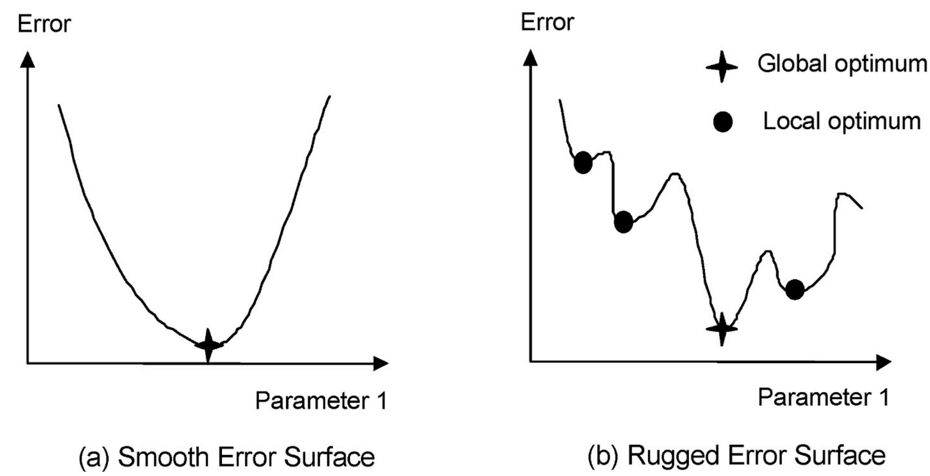

Fig. 10. Error surface with different degrees of ruggedness for a model with one parameter.

difficult to find (see Maier et al., 2019). The degree of ruggedness of an error surface is usually problem dependent and is affected by the number of model parameters, among other factors. As the number of model parameters increases, so does the size of the search space and, generally, the number of local optima and ruggedness of the error surface. Consequently, it is important to find the model with the smallest number of inputs and parameters that is able to describe the underlying relationship in the data (see Myth #7 and Section 3.3).

As deterministic model calibration involves the determination of the combination of model parameters that maximises model performance, it is an optimisation problem. In the vast majority of applications of MLPs to environmental modelling, back-propagation has been used as the optimisation algorithm for obtaining the optimal set of model parameters (Maier and Dandy, 2000a,b; Maier et al., 2010; Wu et al., 2014). However, as the calibration for any predictive model involves the same process, any optimisation approach can be used, such as gradient-based methods (e.g. conjugate-gradient methods, quasi-Newton methods, the Levenberg-Marquardt algorithm) or more global optimisation approaches, such as evolutionary algorithms (see Maier et al., 2019). The importance of recognising the independence of the selection of a particular model architecture (f(.)) and the optimisation algorithm used to obtain values of the unknown model parameters (θ) via calibration (training) cannot be overemphasised. This is exemplified by the commonly-used term "backpropagation network", which conflates the ANN model architecture (MLP) with the optimisation algorithm used to calibrate (train) it (backpropagation).

As an alternative to the deterministic calibration methods discussed above, Bayesian statistics can be employed to estimate the model parameters. Unlike deterministic calibration, Bayesian calibration involves finding a probability distribution of the model parameters, given the data available for calibration. This is usually done by assuming that the model parameters can be approximated by a normal distribution centred on the most probable parameters (those that maximise model performance) or by using a Markov chain Monte Carlo (MCMC) approach, which makes no assumptions about the distribution of the parameters (see Kingston et al., 2005b). This has a number of advantages, such as the ability to obtain confidence limits on predictions (Kingston et al., 2005b) and the ability to determine an optimal model structure (see Kingston et al., 2008). The primary disadvantage of MCMC Bayesian approaches to model calibration include their high computational cost.

3.7. Model validation

Before the calibrated model can be deployed for forecasting or predictive purposes with confidence, it needs to be validated. This requires a check of whether the model has captured the underlying relationship that is contained in the calibration data (replicative validity), whether the model is able to generalise over the range of the calibration data (predictive validity) and whether the model is physically plausible (structural validity) (Wu et al., 2014; Humphrey et al., 2017) (Fig. 4). Each of these is discussed below.

3.7.1. Replicative validity

ANNs work on the assumption that there is a real function that underlies a system that relates a set of independent predictor variables (model inputs) to one or more dependent variables (model output(s)) of interest (Equation (1)). Consequently, if the ANN model has been successful in capturing the relationship that is contained in the calibration data, the model residuals (errors) should be white noise, as the random error term (ε) in Equation (1). As a result, any observable patterns exhibited by the model residuals are due to a failure of the model to capture some parts of the deterministic components of the data. This possible failure is ubiquitous across all types of models used in environmental sciences, including physically-based models, and always requires careful attention to minimise its impact on the credibility of the results.

If the model residuals are white noise, then (i) their expected value should be zero, (ii) their variance should be constant and (iii) they should be statistically independent of each other. In order to determine whether the model residuals approximate white noise, a number of diagnostic tests need to be performed, such as visual inspection of histograms of the residuals to see if their distribution is centred on zero.

If the model residuals exhibit an autocorrelation structure (i.e. the error is not white noise), the underlying relationship to be modelled has not been fully captured by the model. This could be due to an inappropriate model structure, such as insufficient model complexity, or the failure to find near-global optima in the error surface during calibration. Alternatively, the inability to approximate the desired relationship could be due to the absence of data on potential model inputs that have a significant impact on the model outputs. In cases where an inappropriate model structure or difficulties with model calibration are responsible for the deterministic relationship in the model residuals, the "Determination of Model Structure and Parameters" step may need to be repeated. Another option is to develop a model that captures the relationships that remain in the residuals and to add the output of the error model to that of the primary model (e.g. Forouhar et al., 2022).

It should be noted that if the residuals do approximate white noise, this only indicates that the relationship that is contained in the calibration data has been approximated adequately, and not necessarily that the underlying relationship of the system to be modelled has been captured, which also depends on how representative the available data are of this process.

3.7.2. Predictive validity

At the completion of the model calibration process, all that is known is that the model provides a good fit to a single data set – the calibration data (Chapra, 2008). Consequently, to test the generalisation ability of the model over the range of the data used for calibration, the performance of the model is assessed on an independent data set – the validation set (see Section 3.4). It is important that the validation data are used for the first time at this stage of the model development process. In other words, the validation data should not be used to determine optimal model parameters during calibration/training or the optimal model structure or appropriate parameters controlling the behaviour of the optimisation algorithm used during calibration.

As mentioned in Section 3.6.1, model performance on the validation data is generally assessed using a range of performance metrics. While it is important to evaluate model performance on the validation set in absolute terms, it is also important to assess the validation performance relative to the calibration performance. If there are significant differences in the model performance on the calibration and validation sets, this is either an indication of (i) overfitting, as discussed, or (ii) that the statistical properties and the information content of the calibration and validation data are dissimilar (see Section 3.4). Consequently, predictive validation issues can be overcome by ensuring overfitting does not occur and that each of the data subsets contains data that are representative of all of the “patterns”/“events” that are contained in the available data by using appropriate data splitting methods.

3.7.3. Structural validity

If an ANN model is to be used for predictive purposes, it is vital that its behaviour does not violate any *a priori* knowledge about the relationship being modelled. Because ANNs are data-driven, models that have residuals that approximate white noise and perform well on independent validation data have not necessarily captured the underlying physical relationship to be modelled. As discussed previously, this could be due to the presence of rugged error surfaces with local optima, where different combinations of calibrated model parameters result in the same predictive model performance (Kingston et al., 2005a; Zhu et al., 2022a, 2022b). Some of these may be contradictory to *a priori* knowledge about the system under consideration and the model might therefore not be physically plausible. This has important implications in relation to the general applicability and credibility of the model.

As the parameters of ANNs do not have a physical meaning, it is not possible to assess structural validity by examining the calibrated values of model parameters in isolation, as is the case for more physically-based models. However, a number of approaches can be used to check whether the calibrated ANN model has not violated any *a priori* understanding of the relationship to be modelled, such as methods based on the overall connection weight or methods based on sensitivity analysis (see Humphrey et al., 2017). For example, an overall connection weight between rainfall and runoff that is negative (i.e. that increased rainfall results in decreased runoff) suggests that the input-output relationship determined by the calibrated ANN model violates *a priori* understanding of the underlying physical processes and is therefore not suitable for deployment. While such checks do not guarantee that the “correct” relationship has been captured, it ensures that the relationship is not “incorrect”.

If a model is found to be structurally invalid, efforts can be made to better understand if the error surface is likely to contain many local optima (see Zhu et al., 2022a,b), to ensure good solutions have been found during the calibration process and to attempt to reduce model complexity by using appropriate IVS methods and reducing the number of hidden nodes, where possible. Another option includes constraining the overall connection weight during calibration to ensure the calibrated model is physically plausible. For example, the overall connection weight between a rainfall input and the corresponding runoff output can be constrained to be positive and the overall connection weight between an evaporation input and the corresponding runoff output can be constrained to be negative (see Kingston et al., 2005a).

3.8. Model deployment

Given that the data that are available for model development are limited, it is likely that, once deployed, a validated model will be exposed to “patterns”/“events” that were not used during the model development (calibration/training and testing) and evaluation (validation) phases. Such extrapolation could result in a significant reduction in model performance over time, as the model is exposed to more and more non-representative input/output samples. Consequently, the primary objective of the deployment phase is to ensure that the performance of the model does not deteriorate significantly when exposed to new input/output data.

The simplest way to achieve this is to re-calibrate the model as new data become available. To do this in a computationally efficient manner, Bowden et al. (2012) introduced an approach for checking any new data so see if these data contain any “patterns”/“events” that were not used during model development with the aid of cluster analysis using a self-organising map (SOM) (Kohonen, 1995). If this is not the case, there is no need to re-calibrate. However, if the new input data are significantly different from those in the calibration data, then they should be added to the calibration data, and the model recalibrated.

3.9. Summary

In many applications of ANNs in environmental modelling, the primary focus is on the choice of model architecture, structure and “training” (calibration) (Fig. 11), at the expense of the consideration of the other steps in the model development process shown in Fig. 4 (Maier et al., 2010; Wu et al., 2014). However, as discussed above, this is likely to have significant negative consequences on the performance of the resulting model. This is because, for a given model architecture, structure and optimisation (training) algorithm, which model is ultimately developed (i.e. which set of model parameters is selected as part of the calibration process) is a function of the characteristics of the error surface (in turn dependent on how the data are processed), the selected inputs, and how the available data are split into their calibration, testing and validation subsets. These points are summarised in Fig. 11 and discussed below.

As can be seen in Fig. 11, determination of appropriate model parameters (e.g. connection weights and biases) involves the interaction between an optimisation algorithm (e.g. back-propagation) and the error surface, which represents how the calibration (training) error varies with changes in model parameters, as discussed previously. The optimisation algorithm navigates through the error surface in order to identify the combination of parameters that corresponds to the lowest error. However, how successful the optimisation algorithm is in achieving this goal depends on the searching behaviour of the algorithm (e.g. gradient descent or global search) and the characteristics of the error surface (e.g. smooth with one optimum or rough with many local optima). While significant attention is paid to the optimisation algorithm side of this equation, the characteristics of the error surface are often ignored (e.g. Zhu et al., 2022a, b). However, consideration of the characteristics of the error surface is vitally important, as these characteristics not only influence the ability of the optimisation algorithm to find a near-globally optimal solution in this surface, but also what parameter combinations these correspond to and how well these are defined. For example, if there are many local optima with similar errors, it is unknown which of these parameter combinations results in the “best” input-output relationship. In fact, having multiple local optima is likely to confuse and slow down the optimisation algorithm, and more importantly, make it almost impossible to identify a model that represents the desired input-output relationship.

So, how do we maximise our chances that the error surface is as “nicely behaved” as possible – that it contains a single, clearly-defined

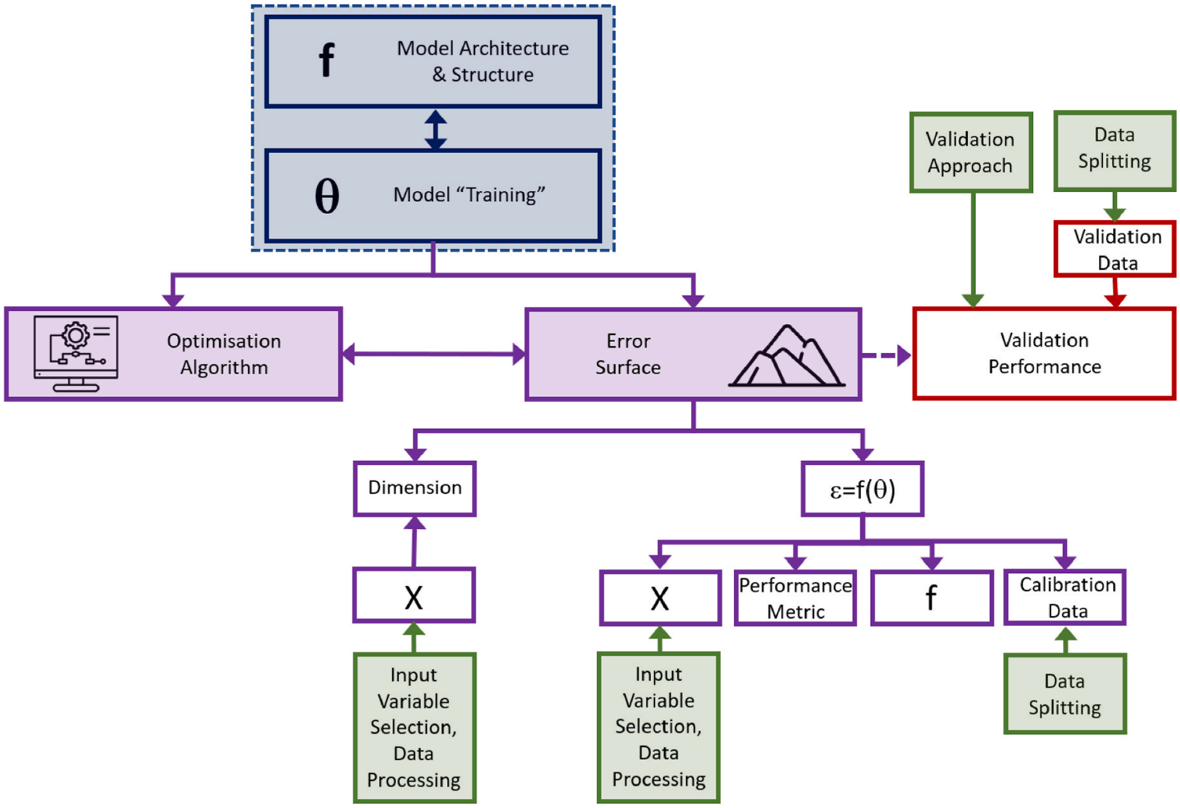

Fig. 11. Summary of the importance of data processing, input variable selection, data splitting and validation approaches in increasing trust and confidence in ANNs and reducing barriers to their adoption.

optimum and that this optimum is easy-to-find by the selected optimisation algorithm? In order to answer this question, we need to understand what factors affect the characteristics of the error surface. As the error surface shows how calibration errors vary with changes in model parameters, the dimension of the space the optimisation algorithm has to search is equal to the number model parameters (i.e. the number of axes in the error surface is equal to the number of model parameters), which, for an MLP, is a function of the number of inputs and hidden nodes (assuming the nodes are fully connected), as well as the bounds on the values the model parameters can take (i.e. their maximum and minimum values). Consequently, if an MLP has 15 weights, the values of which need to be determined by calibration (training), the optimisation algorithm has to search through a 15-dimensional space. Consequently, there is a significant incentive to reduce the number of model parameters, which can be achieved by using state-of-the-art input variable selection techniques to ensure only relevant, non-redundant model inputs are included.

Apart from dimensionality, the characteristics of the error surface are also a function of the error values themselves, which are a function of the selected input variables and how they are processed (X), the selected performance (error) metric, the selected model architecture and structure (f) and which data are used to calculate the error (i.e. the calibration data) (Fig. 11). Consequently, the characteristics of the error surface are directly related to the Data Processing, Input Selection, Data Splitting and Performance Metric Selection steps in Fig. 4. For example, if there is redundancy in the selected inputs, different parameter combinations are likely to result in the same calibration error, introducing local optima into the error surface, potentially resulting in different models that have the same calibration and validation error, but have captured different underlying relationships, some of which might not align with an *a priori* understanding of the underlying physics. However, whether this is the case or not can only be ascertained by going beyond the use of an independent validation set to also include structural validation methods (Fig. 11). Consequently, there is a need to adopt state-of-the-art approaches at each step of the model development process.

4. Summary and conclusions

ANNs were considered a "novelty item" by industry 20–30 years ago, primarily confining their application to the research domain. However, this is certainly not the case anymore. Today, many businesses are familiar with ANNs and their capability, and often employ machine learning experts. While the potential of ANNs is clear, they are still surrounded by an air of mystery and intrigue, leading to a lack of understanding of their inner workings. This has led to the perpetuation of a number of myths, resulting in the misconception that the application of ANNs primarily involves "throwing" a large amount of data at a "black-box" software package. While this is certainly a convenient way to sidestep the principles and level of rigor applied to the development of many other types of environmental models, this comes at significant cost in terms of the validity and usefulness of the resulting models, the ability to assess and compare research findings and the ability to progress research efforts. In order to address these issues, this paper explodes a number of the common myths surrounding the use of ANNs for the prediction and forecasting of environmental systems and outlines state-of-the-art approaches to developing ANN models that enable them to be applied with confidence in practice.

In terms of exploding the myths surrounding the use of ANNs for prediction and forecasting, this paper clarifies that:

1. Like all other types of models, ANNs convert a set of model inputs into a model output via a (complex) mathematical relationship. They are therefore not a "special case" that is exempt from the need to apply good model development practices.

2. The mathematical relationship between model inputs and outputs is well-defined and known for ANN models. Consequently, ANNs are not "black boxes" from a numerical point of view.
3. Data requirements of ANNs are typically more flexible than those of other types of models. Consequently, they generally do not require a large number of inputs for their development. In addition, unlike many other models, ANNs can handle incomplete or missing data and still generate effective input-output relationships.
4. The vast majority of models used for prediction and forecasting "learn from examples" as their unknown model parameters are determined with the aid of previous examples of the input-output relationship of interest as part of model calibration. Consequently, the "training" of ANN models is equivalent to the "calibration" of any other predictive model and not a special attribute that sets ANNs apart.
5. Most types of models perform best when they interpolate between the data used for their development (e.g. as part of model calibration) and are therefore a "prisoner of data". This is not a particular disadvantage of ANNs relative to other types of models. In fact, recent developments have shown that ANNs have significant capacity for generalization.
6. While the assessment of the performance of calibrated/trained ANNs on an independent validation data set is essential, good performance on this data set is insufficient evidence to enable ANNs to be used with confidence in practice. There is also a need to check whether the input-output relationships ANN models have captured are in accord with known system understanding. This ensures ANNs are able to generalise and therefore able to be used with confidence in practice.
7. The inputs to ANNs must be selected carefully to ensure good model performance. The inclusion of inputs that provide irrelevant and/or redundant information makes it more difficult to calibrate/train models and to obtain models that provide meaningful and trustworthy results.

While the application of ANNs is synonymous with different types of architectures ("types of ANNs") and calibration ("training") approaches, all steps in the model development need to be considered carefully. These include:

1. Selecting potential model inputs and outputs based on the problem being addressed and any available system understanding.
2. Processing of the available data to ensure any temporal dynamics are accounted for explicitly via lagging (unless ANNs with recurrent architectures are used) and to ensure input variables spanning different ranges have the opportunity to have an influence on model outputs by being scaled appropriately.
3. Selecting the subset of potential model inputs so that the selected inputs have a substantial influence on the model output(s) and are independent of each other (i.e. do not provide redundant information) (see Galelli et al., 2014). This is crucial in terms of ensuring that the most parsimonious model is developed, which is vital to ensuring the model parameters (e.g. connection weights) are as well-defined as possible and the resulting model is able to capture and generalise over the underlying physical relationships that are contained in the available data.
4. Splitting the available data into model development (calibration (and testing)) and evaluation (validation) subsets using approaches that ensure the statistical properties of the data in each of these subsets are as close to each other as possible (see Chen et al., 2022). This is critical to maximise generalisation ability (i.e. that the model is calibrated/trained over all types of events/patterns that are contained in the available data) and the degree to which this is assessed independently (i.e. that the independent performance of the model is assessed over all types of events/patterns that are contained in the available data).
5. Selecting a model architecture (e.g. MLP, RNN, GNN, PINN) that is most appropriate for the problem under consideration.
6. Optimizing the model structure (e.g. number of hidden layers and nodes) and model parameters (e.g. connection weights and biases) so as to maximise model performance on the calibration or testing set, while trying to minimise the number of model parameters that need to be determined via calibration.
7. Checking the replicative, predictive and structural validity of the best-performing calibrated model by checking (i) the model residuals, (ii) the predictive performance of the model on the independent validation set and (iii) whether the relationship captured by the model conforms with *a priori* knowledge (see Humphrey et al., 2017).
8. Deploying the model in a way that ensures performance does not deteriorate over time by re-calibrating the model using any newly collected data that are significantly different from the data used for initial model development (see Bowden et al., 2012).

Given the universal function approximation ability of ANNs, they present a "one stop shop" for developing prediction and forecasting models for environmental problems. However, the lack of understanding of how ANNs work and the misconception that they represent a "unique" modelling approach and are therefore exempt from modelling development methods that are considered best-practice for other types of environmental models presents a significant threat to the usefulness and credibility of ANN models. We hope this paper provides a useful resource for those starting, as well as those continuing, on their ANN modelling journey by shedding light on common myths and misconceptions and providing guidance on best-practice model development approaches.

Declaration of competing interest

Members of Editorial Borad for this journal: Holger Maier, Stefano Galelli, Andrea Castelletti, Saman Razavi, Marco Acutis, Wenyan Wu.

Data availability

No data was used for the research described in the article.

Acknowledgements

The authors would like to thank Barbara Robson for her comments on preliminary versions of this manuscript, as well as the two anonymous reviewers of this paper, whose comments have helped to improve its quality significantly. Wenyan Wu acknowledges support from the Australian Research Council via the Discovery Early Career Researcher Award (DE210100117).

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., 2016. TensorFlow: a system for large-scale machine learning. https://doi.org/10.48550/ARXIV.1605.08695.

Abrahart, R.J., Anctil, F., Coulibaly, P., Dawson, C.W., Mount, N.J., See, L.M., Shamseldin, A.Y., Solomatine, D.P., Toth, E., Wilby, R.L., 2012. Two decades of anarchy? Emerging themes and outstanding challenges for neural network river forecasting. Prog. Phys. Geogr. Earth Environ. 36, 480–513. https://doi.org/10.1177/0309133312444943.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000a. Artificial neural networks in hydrology. I: preliminary concepts. J. Hydrol. Eng. 5, 115–123. https://doi.org/10.1061/(ASCE)1084-0699(2000)5:2(115).

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000b. Artificial neural networks in hydrology. II: hydrologic applications. J. Hydrol. Eng. 5, 124–137. https://doi.org/10.1061/(ASCE)1084-0699(2000)5:2(124).

Binder, A., Montavon, G., Lapuschkin, S., Müller, K.-R., Samek, W., 2016. Layer-wise relevance propagation for neural networks with local renormalization layers. In: Villa, A.E.P., Masulli, P., Pons Rivero, A.J. (Eds.), Artificial Neural Networks and

Machine Learning – ICANN 2016, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 63–71. https://doi.org/10.1007/978-3-319-44781-0_8.
Bowden, G.J., Maier, H.R., Dandy, G.C., 2002. Optimal division of data for neural network models in water resources applications: division of data for neural network models. Water Resour. Res. 38 https://doi.org/10.1029/2001WR000266 Bowden et al, 2-1-2–11, 2002.
Bowden, G.J., Dandy, G.C., Maier, H.R., 2003. Data transformation for neural network models in water resources applications. J. Hydroinf. 5, 245–258. https://doi.org/10.2166/hydro.2003.0021.
Bowden, G.J., Dandy, G.C., Maier, H.R., 2005a. Input determination for neural network models in water resources applications. Part 1—background and methodology. J. Hydrol. 301, 75–92. https://doi.org/10.1016/j.jhydrol.2004.06.021.
Bowden, G.J., Maier, H.R., Dandy, G.C., 2005b. Input determination for neural network models in water resources applications. Part 2. Case study: forecasting salinity in a river. J. Hydrol. 301, 93–107. https://doi.org/10.1016/j.jhydrol.2004.06.020.
Bowden, G.J., Maier, H.R., Dandy, G.C., 2012. Real-time deployment of artificial neural network forecasting models: understanding the range of applicability. Water Resour. Res. 48, 2012WR011984 https://doi.org/10.1029/2012WR011984.
Box, G.E.P., Jenkins, G.M., 1976. Time Series Analysis: Forecasting and Control, Revised Edition. Holden Day, San Francisco.
Broad, D.R., Dandy, G.C., Maier, H.R., 2015. A systematic approach to determining metamodel scope for risk-based optimization and its application to water distribution system design. Environ. Model. Software 69, 382–395. https://doi.org/10.1016/j.envsoft.2014.11.015.
Cabaneros, S.M., Calautit, J.K., Hughes, B.R., 2019. A review of artificial neural network models for ambient air pollution prediction. Environ. Model. Software 119, 285–304. https://doi.org/10.1016/j.envsoft.2019.06.014.
Castelletti, A., Galelli, S., Ratto, M., Soncini-Sessa, R., Young, P.C., 2012. A general framework for Dynamic Emulation Modelling in environmental problems. Environ. Model. Software 34, 5–18. https://doi.org/10.1016/j.envsoft.2012.01.0002.
Chapra, S.C., 2008. Surface Water-Quality Modeling, Reissued. Waveland Press, Long Grove, Ill.
Chen, J., Zheng, F., May, R., Guo, D., Gupta, H., Maier, H.R., 2022. Improved data splitting methods for data-driven hydrological model development based on a large number of catchment samples. J. Hydrol. 613, 128340 https://doi.org/10.1016/j.jhydrol.2022.128340.
Cho, et al., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734.
Chow, T.W.S., Huang, D., 2005. Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information. IEEE Trans. Neural Network. 16, 213–224. https://doi.org/10.1109/TNN.2004.841414.
Ciresan, D., Meier, U., Gambardella, L., Schmidhuber, J., 2010. Deep big simple neural nets for handwritten digit recognition. Neural Comput. 22 (12), 3207–3220.
Dawson, C.W., Abrahart, R.J., See, L.M., 2007. HydroTest: a web-based toolbox of evaluation metrics for the standardised assessment of hydrological forecasts. Environ. Model. Software 22, 1034–1052. https://doi.org/10.1016/j.envsoft.2006.06.008.
Dawson, C.W., Wilby, R.L., 2001. Hydrological modelling using artificial neural networks. Prog. Phys. Geogr. Earth Environ. 25, 80–108. https://doi.org/10.1177/030913330102500104.
Fan, J., Bai, J., Li, Z., Ortiz-Bobea, A., Gomes, C.P., 2022. A GNN-RNN approach for harnessing geospatial and temporal information: application to crop yield prediction. http://arxiv.org/abs/2111.08900.
Fernando, T.M.K.G., Maier, H.R., Dandy, G.C., 2009. Selection of input variables for data driven models: an average shifted histogram partial mutual information estimator approach. J. Hydrol. 367, 165–176. https://doi.org/10.1016/j.jhydrol.2008.10.019.
Forouhar, L., Wu, W., Wang, Q.J., Hakala, K., 2022. A hybrid framework for short-term irrigation demand forecasting. Agric. Water Manag. 273, 107861 https://doi.org/10.1016/j.agwat.2022.107861.
Galelli, S., Castelletti, A., 2013. Tree-based iterative input variable selection for hydrological modeling: tree-Based Input Selection. Water Resour. Res. 49, 4295–4310. https://doi.org/10.1002/wrcr.20339.
Galelli, S., Humphrey, G.B., Maier, H.R., Castelletti, A., Dandy, G.C., Gibbs, M.S., 2014. An evaluation framework for input variable selection algorithms for environmental data-driven models. Environ. Model. Software 62, 33–51. https://doi.org/10.1016/j.envsoft.2014.08.015.
Gardner, M.W., Dorling, S.R., 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmos. Environ. 32, 2627–2636. https://doi.org/10.1016/S1352-2310(97)00447-0.
Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., Yacalis, G., 2018. Could machine learning break the convection parameterization deadlock? Geophys. Res. Lett. 45, 5742–5751. https://doi.org/10.1029/2018GL078202.
Ghalandari, M., Forootan Fard, H., Komeili Birjandi, A., Mahariq, I., 2021. Energy-related carbon dioxide emission forecasting of four European countries by employing data-driven methods. J. Therm. Anal. Calorim. 144, 1999–2008. https://doi.org/10.1007/s10973-020-10400-y.
Granata, F., Di Nunno, F., 2021. Forecasting evapotranspiration in different climates using ensembles of recurrent neural networks. Agric. Water Manag. 255, 107040 https://doi.org/10.1016/j.agwat.2021.107040.
Grigsby, J., Wang, Z., Nguyen, N., Qi, Y., 2023. Long-range transformers for dynamic spatiotemporal forecasting. http://arxiv.org/abs/2109.12218.
Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets (PDF). Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014) 2672–2680.
Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning, Adaptive Computation and Machine Learning. The MIT Press, Cambridge, Massachusetts.
Guillaume, J.H.A., Jakeman, J.D., Marsili-Libelli, S., Asher, M., Brunner, P., Croke, B., Hill, M.C., Jakeman, A.J., Keesman, K.J., Razavi, S., Stigter, J.D., 2019. Introductory overview of identifiability analysis: a guide to evaluating whether you have the right type of data for your modeling purpose. Environ. Model. Software 119, 418–432. https://doi.org/10.1016/j.envsoft.2019.07.007.
Guo, D., Zheng, F., Gupta, H., Maier, H.R., 2020. On the robustness of conceptual rainfall-runoff models to calibration and evaluation data set splits selection: a large sample investigation. Water Resour. Res. 56 https://doi.org/10.1029/2019WR026752.
Gupta, H.V., Perrin, C., Kumar, R., Blöschl, G., Montanari, A., Kumar, R., Clark, M., Andreassian, V., 2014. Large-sample hydrology: a need to balance depth with breadth. Hydrol. Earth Syst. Sci. 18, 1–15.
Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. J. Mach. Learn. Res. 3 (Mar), 1157–1182.
Hagiwara, K., Toda, N., Usui, S., 1993. On the problem of applying AIC to determine the structure of a layered feedforward neural network. In: Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan). Presented at the 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan). IEEE, Nagoya, Japan, pp. 2263–2266. https://doi.org/10.1109/IJCNN.1993.714176.
Hejazi, M.I., Cai, X., 2009. Input variable selection for water resources systems using a modified minimum redundancy maximum relevance (mMRMR) algorithm. Adv. Water Resour. 32, 582–593. https://doi.org/10.1016/j.advwatres.2009.01.009.
Heydari, A., Garcia, D.A., Keynia, F., Bisegna, F., Santoli, L.D., 2019. Renewable energies generation and carbon dioxide emission forecasting in microgrids and national grids using GRNN-GWO methodology. Energy Proc. 159, 154–159. https://doi.org/10.1016/j.egypro.2018.12.044.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.
Huang, C.-J., Kuo, P.H., 2018. A deep CNN-LSTM model for particulate matter (PM2.5) forecasting in smart cities. Sensors 18 (7), 2220. https://doi.org/10.3390/s18072220.
Huber, J., Lohmann, K., Schmidt, M., Weinhardt, C., 2021. Carbon efficient smart charging using forecasts of marginal emission factors. J. Clean. Prod. 284, 124766 https://doi.org/10.1016/j.jclepro.2020.124766.
Humphrey, G.B., Maier, H.R., Wu, W., Mount, N.J., Dandy, G.C., Abrahart, R.J., Dawson, C.W., 2017. Improved validation framework and R-package for artificial neural network models. Environ. Model. Software 92, 82–106. https://doi.org/10.1016/j.envsoft.2017.01.023.
Hunter, J.M., Maier, H.R., Gibbs, M.S., Foale, E.R., Grosvenor, N.A., Harders, N.P., Kikuchi-Miller, T.C., 2018. Framework for developing hybrid process-driven, artificial neural network and regression models for salinity prediction in river systems. Hydrol. Earth Syst. Sci. 22, 2987–3006. https://doi.org/10.5194/hess-22-2987-2018.
Kalogirou, S.A., 2001. Artificial neural networks in renewable energy systems applications: a review. Renew. Sustain. Energy Rev. 5, 373–401. https://doi.org/10.1016/S1364-0321(01)00006-5.
Karakaya, G., Galelli, S., Ahipasaoglu, S.D., Taormina, R., 2016. Identifying (quasi) equally informative subsets in feature selection problems for classification: a max-relevance min-redundancy approach. IEEE Trans. Cybern. 46, 1424–1437. https://doi.org/10.1109/TCYB.2015.2444435.
Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L., 2021. Physics-informed machine learning. Nat Rev Phys 3, 422–440. https://doi.org/10.1038/s42254-021-00314-5.
Karpatne, A., Atluri, G., Faghmous, J.H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N., Kumar, V., 2017. Theory-guided data science: a new paradigm for scientific Discovery from data. IEEE Trans. Knowl. Data Eng. 29, 2318–2331. https://doi.org/10.1109/TKDE.2017.2720168 (PLEASE CHECK!).
Kashinath, K., Mustafa, M., Albert, A., Wu, J.-L., Jiang, C., Esmaeilzadeh, S., Azizzadenesheli, K., Wang, R., Chattopadhyay, A., Singh, A., Manepalli, A., Chirila, D., Yu, R., Walters, R., White, B., Xiao, H., Tchelepi, H.A., Marcus, P., Anandkumar, A., Hassanzadeh, P., Prabhat, 2021. Physics-informed machine learning: case studies for weather and climate modelling. Phil. Trans. R. Soc. A. 379, 20200093 https://doi.org/10.1098/rsta.2020.0093.
Kingston, G.B., Maier, H.R., Lambert, M.F., 2005a. Calibration and validation of neural networks to ensure physically plausible hydrological modeling. J. Hydrol. 314, 158–176. https://doi.org/10.1016/j.jhydrol.2005.03.013.
Kingston, G.B., Lambert, M.F., Maier, H.R., 2005b. Bayesian training of artificial neural networks used for water resources modeling: bayesian training of anns in water resources modeling. Water Resour. Res. 41 https://doi.org/10.1029/2005WR004152.
Kingston, G.B., Maier, H.R., Lambert, M.F., 2008. Bayesian model selection applied to artificial neural networks used for water resources modeling: bms of anns in water resources modeling. Water Resour. Res. 44 https://doi.org/10.1029/2007WR006155.
Kohonen, T., 1995. Self-Organizing Maps, Springer Series in Information Sciences. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-97610-0.
Kuhn, M., 2008. Building predictive models in *R* using the **caret** package. J. Stat. Software 28. https://doi.org/10.18637/jss.v028.i05.
Kumar, P., Chandra, R., Bansal, C., Kalyanaraman, S., Ganu, T., Grant, M., 2021. Micro-climate prediction - multi scale encoder-decoder based deep learning framework. In:

Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, vol. 2021. Virtual Event Singapore: ACM. https://doi.org/10.1145/3447548.3467173, 3128–38.
Li, X., Maier, H.R., Zecchin, A.C., 2015a. Improved PMI-based input variable selection approach for artificial neural network and other data driven environmental and water resource models. Environ. Model. Software 65, 15–29. https://doi.org/10.1016/j.envsoft.2014.11.028.
Li, X., Zecchin, A.C., Maier, H.R., 2015b. Improving partial mutual information-based input variable selection by consideration of boundary issues associated with bandwidth estimation. Environ. Model. Software 71, 78–96. https://doi.org/10.1016/j.envsoft.2015.05.013.
Li, L., Lambert, M.F., Maier, H.R., Partington, D., Simmons, C.T., 2015c. Assessment of the internal dynamics of the Australian Water Balance Model under different calibration regimes. Environ. Model. Software 66, 57–68. https://doi.org/10.1016/j.envsoft.2014.12.015.
Lipton, Z.C., 2018. The Mythos of Model Interpretability: in machine learning, the concept of interpretability is both important and slippery. Queue 16, 31–57. https://doi.org/10.1145/3236386.3241340.
Lipton, Z.C., Berkowitz, J., Elkan, C., 2015. A Critical Review of Recurrent Neural Networks for Sequence Learning arXiv:1506.00019.
Maier, H.R., Dandy, G.C., 1997. Determining inputs for neural network models of multivariate time series. Comp-aided Civil Eng 12, 353–368. https://doi.org/10.1111/0885-9507.00069.
Maier, H.R., Dandy, G.C., 2000a. Application of artificial neural networks to forecasting of surface water quality variables: issues, applications and challenges. In: Govindaraju, R.S., Rao, A.R. (Eds.), Artificial Neural Networks in Hydrology. Kluwer, Dordrecht, The Netherlands, pp. 287–309.
Maier, H.R., Dandy, G.C., 2000b. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. Environ. Model. Software 15, 101–124. https://doi.org/10.1016/S1364-8152(99)00007-9.
Maier, H.R., Jain, A., Dandy, G.C., Sudheer, K.P., 2010. Methods used for the development of neural networks for the prediction of water resource variables in river systems: current status and future directions. Environ. Model. Software 25, 891–909. https://doi.org/10.1016/j.envsoft.2010.02.003.
Maier, H.R., Razavi, S., Kapelan, Z., Matott, L.S., Kasprzyk, J., Tolson, B.A., 2019. Introductory overview: optimization using evolutionary algorithms and other metaheuristics. Environ. Model. Software 114, 195–213. https://doi.org/10.1016/j.envsoft.2018.11.018.
Maier, H.R., Zheng, F., Gupta, H., Chen, J., Mai, J., Savic, D., Loritz, R., Wu, W., Guo, G., Bennett, A., Zhao, J., 2023. On How Data Are Used in Model Development: the Elephant in the Room. Environmental Modelling & Software submitted for publication. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4477173.
Manepalli, A., Albert, A., Rhoades, A., Feldman, D., 2019. Emulating numeric hydroclimate models with physics-informed cGANs. In: Climate Change AI Workshop at the 33rd Conf. On Neural Information Processing Systems.
May, R.J., Dandy, G.C., Maier, H.R., Nixon, J.B., 2008b. Application of partial mutual information variable selection to ANN forecasting of water quality in water distribution systems. Environ. Model. Software 23, 1289–1299. https://doi.org/10.1016/j.envsoft.2008.03.008.
May, R.J., Maier, H.R., Dandy, G.C., 2010. Data splitting for artificial neural netowrks using SOM-based stratified sampling. Neural Network. 23 (2), 283–294. https://doi.org/10.1016/j.neunet.2009.11.009.
May, R.J., Maier, H.R., Dandy, G.C., Fernando, T.M.K.G., 2008a. Non-linear variable selection for artificial neural networks using partial mutual information. Environ. Model. Software 23, 1312–1326. https://doi.org/10.1016/j.envsoft.2008.03.007.
Mei, X., Smith, P.K., 2021. A comparison of in-sample and out-of-sample model selection approaches for artificial neural network (ANN) daily streamflow simulation. Water 13, 2525. https://doi.org/10.3390/w13182525.
Mount, N.J., Maier, H.R., Toth, E., Elshorbagy, A., Solomatine, D., Chang, F.-J., Abrahart, R.J., 2016. Data-driven modelling approaches for socio-hydrology: opportunities and challenges within the Panta Rhei Science Plan. Hydrol. Sci. J. 1–17. https://doi.org/10.1080/02626667.2016.1159683.
Noshad, M., Choi, J., Sun, Y., Hero III, A., Dinov, I.D., 2021. A data value metric for quantifying information content and utility. Journal of Big Data 8, 82. https://doi.org/10.1186/s40537-021-00446-6.
Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. PyTorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. Curran Associates, Inc.
Paudel, D., de Wit, A., Boogaard, H., Marcos, D., Osinga, S., Athanasiadis, I.N., 2023. Interpretability of deep learning models for crop yield forecasting. Comput. Electron. Agric. 206, 107663 https://doi.org/10.1016/j.compag.2023.107663.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É., 2011. Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.
Pylianidis, C., Athanasiadis, I.N., 2022. Learning latent representations for operational nitrogen response rate prediction. AI for Earth Sciences Workshop at ICLR2022 arXiv. http://arxiv.org/abs/2205.09025.
Pylianidis, C., Snow, V., Overweg, H., Osinga, S., Kean, J., Athanasiadis, I.N., 2022. Simulation-assisted machine learning for operational digital twins. Environ. Model. Software 148, 105274. https://doi.org/10.1016/j.envsoft.2021.105274.
Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045.
Ravuri, S., Lenc, K., Willson, M., Kangin, D., Lam, R., Mirowski, P., Fitzsimons, M., Athanassiadou, M., Kashem, S., Madge, S., Prudden, R., Mandhane, A., Clark, A., Brock, A., Simonyan, K., Hadsell, R., Robinson, N., Clancy, E., Arribas, A., Mohamed, S., 2021. Skilful precipitation nowcasting using deep generative models of radar. Nature 597, 672–677. https://doi.org/10.1038/s41586-021-03854-z.
Razavi, S., Araghinejad, S., 2009. Reservoir inflow modeling using temporal neural networks with forgetting factor approach. Water Resour. Manag. 23, 39–55.
Razavi, S., 2021. Deep learning, explained: fundamentals, explainability, and bridgeability to process-based modelling. Environ. Model. Software 144, 105159. https://doi.org/10.1016/j.envsoft.2021.105159.
Razavi, S., Hannah, D.M., Elshorbagy, A., Kumar, S., Marshall, L., Solomatine, D.P., Dezfuli, A., Sadegh, M., Famiglietti, J., 2022. Coevolution of machine learning and process-based modelling to revolutionize Earth and environmental sciences: a perspective. Hydrol. Process. 36 https://doi.org/10.1002/hyp.14596.
Razavi, S., Tolson, B.A., Burn, D.H., 2012. Review of surrogate modeling in water resources: review. Water Resour. Res. 48 https://doi.org/10.1029/2011WR011527.
Reed, R., 1993. Pruning algorithms-a survey. IEEE Trans. Neural Network. 4, 740–747. https://doi.org/10.1109/72.248452.
Reis, S., Seto, E., Northcross, A., Quinn, N.W.T., Convertino, M., Jones, R.L., Maier, H.R., Schlink, U., Steinle, S., Vieno, M., Wimberly, M.C., 2015. Integrating modelling and smart sensors for environmental and human health. Environ. Model. Software 74, 238–246. https://doi.org/10.1016/j.envsoft.2015.06.003.
Samadianfard, S., Hashemi, S., Kargar, K., Izadyar, M., Mostafaeipour, A., Mosavi, A., Nabipour, N., Shamshirband, S., 2020. Wind speed prediction using a hybrid model of the multi-layer perceptron and whale optimization algorithm. Energy Rep. 6, 1147–1159. https://doi.org/10.1016/j.egyr.2020.05.001.
Sanchez-Lengeling, B., Reif, E., Pearce, A., Wiltschko, A., 2021. A gentle introduction to graph neural networks. Distill 6. https://doi.org/10.23915/distill.00033, 10.23915/distill.00033.
Sarle, W.S., 1997. Neural Network FAQ, Part 1 of 7: Introduction. Periodic Posting to the Usenet Newsgroup comp.ai.Neural-Nets, vol. 1997.
Schillaci, C., Perego, A., Valkama, E., Märker, M., Saia, S., Veronesi, F., Lipani, A., Lombardo, L., Tadiello, T., Gamper, H.A., Tedone, L., Moss, C., Pareja-Serrano, E., Amato, G., Kühl, K., Damatirca, C., Cogato, A., Mzid, N., Eeswaran, R., Rebelo, M., Sperandio, G., Bosino, A., Bufalini, M., Tunçay, T., Ding, J., Fiorentini, M., Tiscornia, G., Conradt, S., Botta, M., Acutis, M., 2021. New pedotransfer approaches to predict soil bulk density using WoSIS soil data and environmentalcovariates in Mediterranean agro-ecosystems. Sci. Total Environ. 146609 https://doi.org/10.1016/j.scitotenv.2021.146609.
Sharma, A., 2000. Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 — a strategy for system predictor identification. J. Hydrol. 239, 232–239. https://doi.org/10.1016/S0022-1694(00)00346-2.
Singh, S.K., Bárdossy, A., 2012. Calibration of hydrological models on hydrologically unusual events. Adv. Water Resour. 38, 81–91.
Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112. https://arxiv.org/abs/1409.3215.
Taormina, R., Chau, K., Sethi, R., 2012. Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon. Eng. Appl. Artif. Intell. 25, 1670–1676. https://doi.org/10.1016/j.engappai.2012.02.009.
Taormina, R., Chau, K., 2015. Data-driven input variable selection for rainfall–runoff modeling using binary-coded particle swarm optimization and Extreme Learning Machines. J. Hydrol. 529, 1617–1632. https://doi.org/10.1016/j.jhydrol.2015.08.022.
Taormina, R., Galelli, S., Karakaya, G., Ahipasaoglu, S.D., 2016. An information theoretic approach to select alternate subsets of predictors for data-driven hydrological models. J. Hydrol. 542, 18–34. https://doi.org/10.1016/j.jhydrol.2016.07.045.
Tirelli, T., Pozzi, L., Pessani, D., 2009. Use of different approaches to model presence/absence of Salmo marmoratus in Piedmont (Northwestern Italy). Ecol. Inf. 4, 234–242. https://doi.org/10.1016/j.ecoinf.2009.07.003.
Tonks, A., Harris, T., Li, B., Brown, W., Smith, R., 2022. Forecasting West nile virus with graph neural networks: harnessing spatial dependence in irregularly sampled geospatial data. http://arxiv.org/abs/2212.11367.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. http://arxiv.org/abs/1706.03762.
Werbos, P.J., 1975. Ph.D. Thesis. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, vol. 1974. Harvard University, Cambridge.
Wu, W., Dandy, G.C., Maier, H.R., 2014. Protocol for developing ANN models and its application to the assessment of the quality of the ANN model development process in drinking water quality modelling. Environ. Model. Software 54, 108–127. https://doi.org/10.1016/j.envsoft.2013.12.016.
Wu, W., May, R.J., Maier, H.R., Dandy, G.C., 2013. A benchmarking approach for comparing data splitting methods for modeling water resources parameters using artificial neural networks. Water Resour. Res. 49 (11), 7598–7614. https://doi.org/10.1002/2012WR012713.
Yu, Y., Si, X., Hu, C., Zhang, J., 2019. A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. 31, 1235–1270. https://doi.org/10.1162/neco_a_01199.
Zanfei, A., Brentan, B.M., Menapace, A., Righetti, M., Herrera, M., 2022. Graph convolutional recurrent neural networks for water demand forecasting. Water Resour. Res. 58 https://doi.org/10.1029/2022WR032299.
Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.), Computer Vision – ECCV 2014,

Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 818–833. https://doi.org/10.1007/978-3-319-10590-1_53.
Zhang, B., Zhang, H., Zhao, G., Lian, J., 2020. Constructing a PM2.5 concentration prediction model by combining auto-encoder with Bi-LSTM neural networks. Environ. Model. Software 124, 104600. https://doi.org/10.1016/j.envsoft.2019.104600.
Zheng, F., Chen, J., Ma, Y., Chen, Q., Maier, H.R., Gupta, H., 2023. A robust strategy to account for data sampling variability in the development of hydrological models. Water Resour. Res. 59, e2022WR033703 https://doi.org/10.1029/2022WR033703.
Zheng, F., Chen, J., Maier, H.R., Gupta, H., 2022. Achieving robust and transferable performance for conservation-based models of dynamical physical systems. Water Resour. Res. 58 https://doi.org/10.1029/2021WR031818.
Zheng, F., Maier, H.R., Wu, W., Dandy, G.C., Gupta, H.V., Zhang, T., 2018. On lack of robustness in hydrological model development due to absence of guidelines for selecting calibration and evaluation data: demonstration for data-driven models. Water Resour. Res. 54, 1013–1030. https://doi.org/10.1002/2017WR021470.
Zhou, Y., Wu, W., Nathan, R., Wang, Q.J., 2021. A rapid flood inundation modelling framework using deep learning with spatial reduction and reconstruction. Environ. Model. Software 143, 105112. https://doi.org/10.1016/j.envsoft.2021.105112.
Zhu, S., Maier, H.R., Zecchin, A.C., 2022b. Identification of metrics suitable for determining the features of real-world optimisation problems. Environ. Model. Software 148, 105281. https://doi.org/10.1016/j.envsoft.2021.105281.
Zhu, S., Zecchin, A.C., Maier, H.R., 2022a. Use of exploratory fitness landscape metrics to better understand the impact of model structure on the difficulty of calibrating artificial neural network models. J. Hydrol. 612, 128093 https://doi.org/10.1016/j.jhydrol.2022.128093.
Zou, S., Zhang, L., Huang, X., Osei, F.B., Ou, G., 2022. Early ecological security warning of cultivated lands using RF-MLP integration model: a case study on China's main grain-producing areas. Ecol. Indicat. 141, 109059 https://doi.org/10.1016/j.ecolind.2022.109059.