# Improved validation framework and R-package for artificial neural network models

**Article** *in* Environmental Modelling & Software · June 2017

DOI: 10.1016/j.envsoft.2017.01.023

CITATIONS
57

READS
991

**7 authors**, including:

Greer B. Humphrey
South Australian Health and Medical Research Institute
**28** PUBLICATIONS **964** CITATIONS
SEE PROFILE

Holger Robert Maier
University of Adelaide
**410** PUBLICATIONS **24,209** CITATIONS
SEE PROFILE

Wenyan Wu
University of Adelaide
**91** PUBLICATIONS **2,181** CITATIONS
SEE PROFILE

Graeme Clyde Dandy
University of Adelaide
**232** PUBLICATIONS **15,318** CITATIONS
SEE PROFILE

Humphrey G.B., Maier H.R., Wu W., Mount N.J., Dandy G.C., Abrahart R.J. and Dawson C.W. (2017) Improved validation framework and R-package for artificial neural network models, Environmental Modelling and Software, 92, 82-106, DOI: 10.1016/j.envsoft.2017.01.023.

# Improved Validation Framework and R-Package for Artificial Neural Network Models

Greer B. Humphrey[a], Holger R. Maier[a], Wenyan Wu[a], Nick J. Mount[b], Graeme C. Dandy[a], Robert J. Abrahart[b], Christian W. Dawson[c]

[a]*School of Civil, Environmental, and Mining Engineering, University of Adelaide, SA 5005, Australia*
[b]*School of Geography, University of Nottingham, Nottingham, NG7 2RD, UK*
[c]*Department of Computer Science, Loughborough University, Loughborough, LE11 3TU, UK*

**Abstract**

Validation is a critical component of any modelling process. In artificial neural network (ANN) modelling, validation generally consists of the assessment of model predictive performance on an independent validation set (predictive validity). However, this ignores other aspects of model validation considered to be good practice in other areas of environmental modelling, such as residual analysis (replicative validity) and checking the plausibility of the model in relation to *a priori* system understanding (structural validity). In order to address this shortcoming, a validation framework for ANNs is introduced in this paper that covers all of the above aspects of validation. In addition, the **validann** R-package is introduced that enables these validation methods to be implemented in a user-friendly and consistent fashion. The benefits of the framework and R-package are demonstrated for two environmental modelling case studies, highlighting the importance of considering replicative and structural validity in addition to predictive validity.

*Keywords:*

Artificial neural networks, multi-layer perceptron, R-package, structural validation, replicative validation, predictive validation

## Software availability

Name of software: validann
Developers: Greer B. Humphrey
Year first available: 2016
Software required: R version 3.1.0 or higher
Availability: http://CRAN.R-project.org/package=validann

## 1. Introduction

Validation has long been considered an important step in the development of environmental models (Jakeman et al., 2006). While there are some inconsistencies in terminology for this step of the model development process (e.g., see Oreskes et al., 1994; Rykiel Jr, 1996; Matott et al., 2009; Biondi et al., 2012), there is broad conceptual agreement that the purpose of model validation is to evaluate how useful a model is for a given purpose, thereby increasing confidence in model outputs (e.g. Power, 1993; Rykiel Jr, 1996; Biondi et al., 2012). Validation is also an important step in the development of artificial neural network (ANN) models, which have been used increasingly for environmental modelling over that past two decades (Maier and Dandy, 2000; Dawson and Wilby, 2001; Maier et al., 2010; Abrahart et al., 2012; Wu et al., 2014). However, the validation process for ANN models is generally restricted to assessing the predictive performance of calibrated models on an

independent validation set (Maier et al., 2010; Wu et al., 2014), which has been referred to as predictive (Power, 1993), operational (Rykiel Jr, 1996) or performance validation (Biondi et al., 2012). This is in contrast to practices in the wider environmental modelling community, where it has been recognized that model validation should also consider (i) how well a model has captured the underlying relationship in the calibration data, which has been referred to as replicative validation (Gass, 1983; Power, 1993) and (ii) how well a model is able to represent the underlying physical processes being modelled (Thomann and Mueller, 1987), which has been referred to as structural (Power, 1993), conceptual (Rykiel Jr, 1996) or scientific validation (Biondi et al., 2012).

While some aspects of replicative validation are generally considered in ANN modelling, such as the use of model goodness-of-fit statistics on the calibration (training) data, examination of the properties of model residuals, which is among the most commonly used model evaluation methods for other model types (Bennett et al., 2013), is generally not considered (Wu et al., 2014). Of even greater concern is that structural validity is generally omitted altogether (Kingston et al., 2005b; Wu et al., 2014; Mount et al., 2016). This might at least in part be due to the fact that ANNs do not represent physical processes explicitly and that the calibrated parameters (e.g. connection weights) of ANNs do not have a direct physical meaning, making the assessment of conceptual validity more difficult. However, there are now a number of approaches that provide insight into the nature of the input-output relationship that has been captured by trained ANNs (e.g. Dimopoulos et al., 1995; Lek et al., 1995; Olden and Jackson, 2002; Wilby et al., 2003; Jain

et al., 2004; Sudheer and Jain, 2004; Sudheer, 2005; Kingston et al., 2006b; Jain and Kumar, 2009; Mount et al., 2013; Dawson et al., 2014), giving an indication of whether an ANN model is able to simulate system behaviour that can be explained in a scientifically acceptable manner. Consequently, methods for assessing the structural validity of ANNs do exist and their consistent application would not only increase confidence in model outputs, but also increase the credibility of ANN models.

In order to address the shortcomings associated with the commonly adopted approach to the validation of ANN models outlined above, the objectives of this paper are:

1. To introduce a comprehensive validation framework for ANN models that includes replicative, predictive and structural validation. As pointed out by Biondi et al. (2012), there is significant benefit in the development of validation protocols, as they facilitate more objective model inter-comparison and are likely to result in the development of superior models. Furthermore, as discussed in van Voorn et al. (2016), the uptake and use of information provided by models may be improved when a user's model quality expectations are properly addressed by modellers. Such protocols help to create awareness among modellers as to what these expectations are. In acknowledging the need for more objective and consistent protocols in ANN model development, Abrahart et al. (2012) identified the development and use of an agreed set of standard diagnostics for rigorous inter-model evaluation, and the adoption of more advanced diagnostics that could be used to trade-off goodness-of-fit against stable, consistent model behaviour and physical

rationality as two main research directions for meeting this agenda and Wu et al. (2014) introduced a protocol for ANN model development, including model validation, on which the ANN validation framework introduced in this paper builds.

2. To introduce an R-package to facilitate implementation of the proposed validation framework. One potential reason for the lack of consideration of replicative and structural validity in the ANN modelling literature is the inability to implement the required analysis approaches in a convenient and user-friendly manner, as has been done for the predictive validation of ANNs (Dawson et al., 2007, 2010) and for other aspects of environmental modelling (e.g. Andrews et al. (2011); Pianosi et al. (2015); Stokes et al. (2015); Guo et al. (2016)). This R-package will not only enable ANN modellers to implement advanced validation methods in a user-friendly and efficient manner, but will also increase consistency between modelling studies, increasing confidence in the results presented and our ability to compare results in an objective manner (Galelli et al., 2014; Maier et al., 2010).
3. To demonstrate the importance of the consideration of all three types of validity (i.e. replicative, structural and predictive), as well as the application of the ANN model validation R-package, on two environmental modelling case studies, including (i) salinity forecasting in the River Murray, Australia and (ii) surface water turbidity prediction at a number of locations in southern Australia.

It should be noted that the proposed validation framework and toolbox are applicable to multi-layer perceptron (MLP) ANNs, as these are by far

the most widely used ANN model architecture used in practice (Maier et al., 2010; Wu et al., 2014). Furthermore, the current focus is on ANN models that perform regression rather than classification and, as such, the proposed methods are more suited to regression problems. However, the framework and corresponding R-package may be extended in future to also include validation methods for classification models.

The remainder of this paper is organized as follows. In Sections 2 and 3, the proposed validation framework and toolbox are introduced, respectively, followed by their application to the two case studies in Section 4. The results are presented and discussed in Section 5 and a summary and conclusions are provided in Section 6.

## 2. Proposed Validation Framework

### *2.1. Overview*

The overall aim of model validation is to ensure that a trained ANN model does not contain known or detectable flaws so that it can be used for its intended purpose with confidence. In order to achieve this, the proposed validation framework includes the assessment of three aspects of model validity, including replicative validity, predictive validity and structural validity (Gass, 1983) (Fig. 1). The purpose of replicative validation is to ensure the model has captured the underlying relationship in the training data, the purpose of predictive validation is to ensure the model can generalize over the range of training data, and the purpose of structural validation is to ensure model behaviour is plausible when compared with *a priori* knowledge of the system being modelled. Although all of these three aspects of validation

should be considered, which are most important depends on the intended purpose of the model. For example, if the primary purpose of the model is prediction and forecasting, the replicative and predictive validity are most important, although structural validity should also be considered. In contrast, if the primary purpose of a model is to gain system understanding, then structural validity is most important, although replicative and predictive validity should also be considered. Further details of each of these steps are given in the subsequent sections.

### *2.2. Replicative Validation*

#### *2.2.1. Underlying philosophy*

A model is replicatively valid if it has captured the underlying relationship in the data used for model calibration (training) (Fig. 1). ANNs work on the premise that there is a real function underlying a system that relates a set of independent predictor variables to one or more dependent variables of interest. Therefore, if $y$ is the target variable and $\mathbf{x}$ is a vector of input or predictor variables, it is assumed that:

$$y_i = f\left(\mathbf{x}_i, \theta\right) + \epsilon_i, \quad i = 1, \ldots, N \tag{1}$$

where $f\left(\cdot\right)$ is the model function, $\theta$ is a vector of "true" model parameters (e.g. connection weights) and $\epsilon$ is a random error or disturbance that accounts for the natural uncertainty inherent in the process, together with any measurement errors associated with $y$. The aim of ANN calibration, or training, is to find estimates of the model parameters $\hat{\theta}$, such that the deterministic component of $y$ (i.e. $f\left(\mathbf{x}, \theta\right)$) is appropriately captured. Here, the

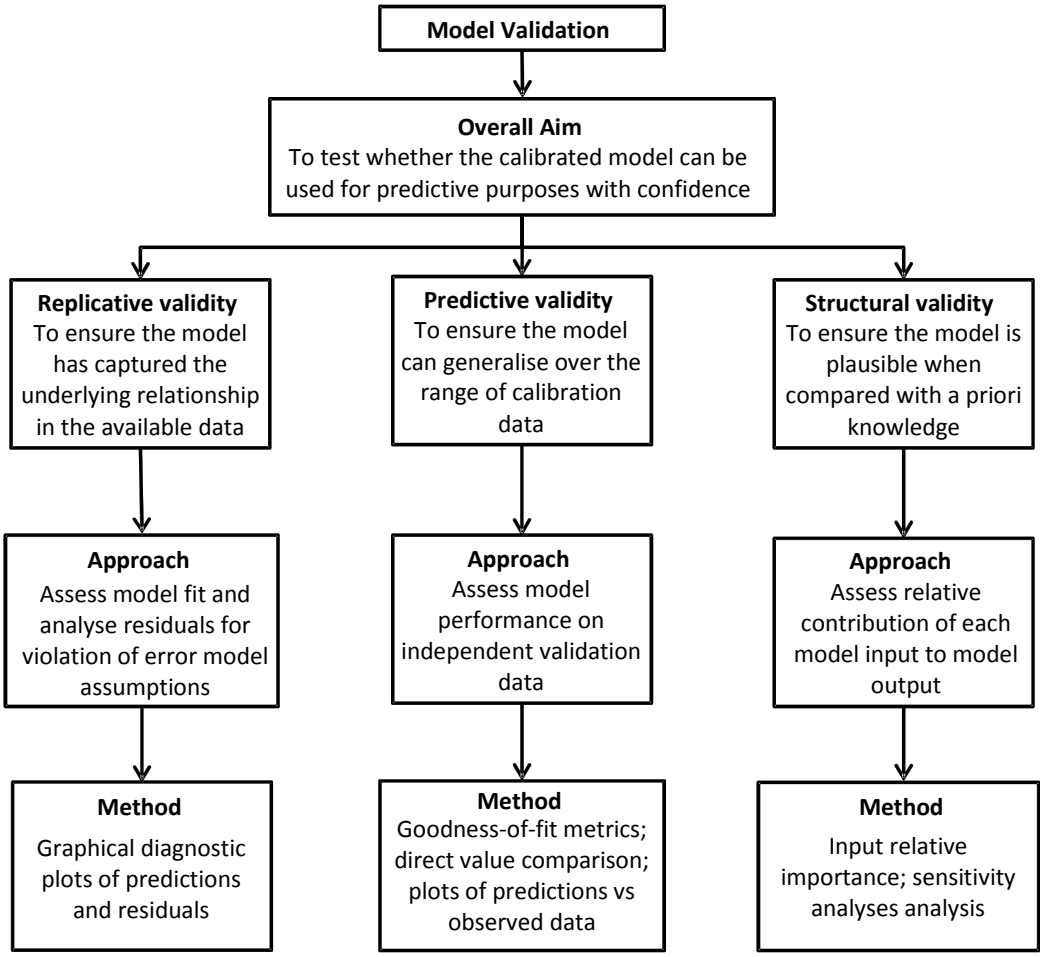


Figure 1: Proposed validation framework for multilayer perceptron ANNs.

accuracy of $\hat{\theta}$ depends not only on the estimated parameter values, but also on the size of the vector $\hat{\theta}$ (i.e. the selected dimensionality of the model).

Typically, calibration of ANNs is based on standard least squares (LS) methods, whereby parameters are sought to minimise the sum of squared (SS) residuals (or a related criterion) between the observed data and the model predictions:

$$SS\left(\hat{\theta}\right) = \sum_{i=1}^{N} \left[y_i - f\left(\mathbf{x}_i, \hat{\theta}\right)\right]^2 = \sum_{i=1}^{N} \hat{\epsilon}_i^2 \tag{2}$$

where $N$ is the number of training data points and $\hat{\epsilon}$ denotes the model residuals (the difference between the observed and predicted data, as opposed to the unobservable random component of $y$). While the SS criterion is often presumed to have general applicability, its use implies the following assumptions about the statistical distribution of $\epsilon$ (Bates and Watts, 1988):

1. $\epsilon$ has zero mean;
2. $\epsilon$ has constant variance; and
3. the $\epsilon_i$ are mutually uncorrelated.

Additionally, in order to make inferences about the model parameters, it is often assumed that $\epsilon$ follows a Gaussian distribution; however, this assumption is not a requirement for the LS parameter estimates to be valid. If an ANN model has been successful in approximating the relationship that is contained in the calibration data (i.e if the model is replicatively valid), the residuals should approximate the random error term, $\hat{\epsilon} \approx \epsilon$. As such, if the above assumptions about $\epsilon$ are reasonable, these should also hold for $\hat{\epsilon}$ (Draper and Smith, 1998).

Violation of the LS assumptions may reveal deficiencies in 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 (training). 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, or the incorrect selection of model inputs from the available data. Consequently, when there is a discernible pattern in the residuals, attempts should be made to modify the model by re-visiting previous steps in the model development process, ensuring that appropriate model-development protocols are being followed (e.g., see Abrahart et al., 2008; Wu et al., 2014). In certain situations, however, the LS assumptions may not be wholly plausible (e.g. in the case of heteroscedastic and/or autocorrelated measurement errors on $y$) and their violation may reflect the inappropriateness of the assumptions, rather than deficiencies in the model formulation (Clarke, 1973). In such cases, use of the SS criterion would result in invalid parameter estimates and inferences made about the process. Transformations, such as Box-Cox (Box and Cox, 1964), may be applied to the observed target data to correct for non-constant variance and to improve the normality of the residuals (Bates and Watts, 1988), or alternatively, an alternative error model might be assumed for the purpose of calibration, which would result in more consistent model parameter estimates $\hat{\theta}$ (Sorooshian and Dracup, 1980; Kuczera, 1983; Thyer et al., 2009; Schoups and Vrugt, 2010; Evin et al., 2013). As a result, it is suggested that diagnostic checks be performed on the model residuals to determine whether the LS assumptions have been violated, and hence, whether any modifica-

tions to the model or the error model are necessary to improve the replicative validity of the model.

### *2.2.2. Methods*

In order to check the replicative validity of ANN models, it is recommended that both the deterministic and stochastic components of Eq. (1) be analysed. The following graphical diagnostics are suggested for assessing whether the model provides a good fit to the training data and whether there is any non-random structure remaining in the model residuals. Examples of these plots are shown and discussed in Section 5.

- **Scatter plot of observed versus predicted data.** A scatter plot, where paired observations and model predictions are plotted against each other, provides a simple method for graphically assessing how well the model fits the training data. For an accurate, unbiased model, the points should plot along the 1:1 line, with scatter about this line representing the discrepancy between the observations and the model. Visual inspection of this plot may reveal systematic divergence from the 1:1 line, which indicates unmodelled behaviour. The model may be shown to under- or over-estimate in a certain range if most points lie below or above the line. As such, a scatter plot is ideal for assessing model performance at low, medium, and high magnitudes (Bennett et al., 2013).

- **Quantile-quantile (Q-Q) plot of observed versus predicted data.** Q-Q plots are powerful tools for graphically assessing goodness-of-fit and may be easier to interpret than scatter plots, especially if the num-

ber of observations is either small or very large. To construct a Q-Q plot of the model predictions against the observations, these data sets are separately ranked, which removes the pairing between them, and the sorted predictions are plotted against the sorted observations. If the modelled and observed data are similarly distributed, points should plot approximately along the 1:1 line. Unlike the scatter plot, however, there should be no scatter about this line, since quantiles are plotted rather than paired data points. As a result, deviations from the line quickly reveal any differences in the distributions of modelled and observed data (e.g. biases at low or high magnitudes) (Chang and Hanna, 2004).

- **Plot of observed and predicted data against data order.** If the data were obtained in a time or space sequence, a plot of both the observed and modelled data against the data order (spatial and/or temporal) is possibly the most powerful graphical tool for visualising model performance, providing valuable insight into any model shortcomings such as errors in timing or location, inhomogeneous performance, and failure of matching at extremes (Crout et al., 2008). Even if the data have no specific ordering, this plot may still provide insight into the accuracy of the model and how it behaves in relation to the data.

- **Plot of standardised residuals against predicted data.** This residual plot, with model output values on the x-axis and standardised residuals on the y-axis, is particularly useful for identifying non-constant variance in the residuals. Ideally, the residuals should display

no pattern, plotting more or less in a horizontal band, symmetric about zero (if the residuals are normally distributed, 95% of the standardised residuals should lie between $\pm 1.96$). Non-constant variance, or heteroscedasticity, is most commonly shown by a widening band, where there is as an increase in the variability of the residuals as the magnitude of the response increases (although it may also be shown by a narrowing band) (Bates and Watts, 1988). This plot can also be useful for identifying outliers in the data, which may be indicated by particularly large residuals.

- **Plot of standardised residuals against against order of the data.** If the spatial and/or temporal order of the data are known, this plot may be useful for identifying serial correlation in the residuals, which suggests unmodelled deterministic behaviour in the data. As above, there should ideally be no visible pattern in this residual plot and residuals should lie randomly within a horizontal band. However, if the residuals display positive serial correlation, sequences of residuals with the same sign will be present. On the other hand, negative serial correlation in the residuals may also be observed, where residuals of one sign tend to be followed by residuals of the opposite sign. If non-random structure is evident in this plot, the assumption of independent residuals and the use of the SS objective function for calibration may not be appropriate.

- **Autocorrelation function (ACF) and partial-autocorrelation function (PACF) plots.** Similar to above, if the data are a time

series, the ACF and PACF plots (Box and Jenkins, 1976) can easily reveal if there is any autocorrelation in the residuals (such patterns may not be so easy to detect with a time series plot of the residuals). The ACF measures the autocorrelation in the residuals as a function of lag:

$$ACF = corr(\hat{\epsilon}_t, \hat{\epsilon}_{t-k}) \tag{3}$$

where $corr()$ gives the Pearson product-moment correlation coefficient and $k$ is the time lag. Autocorrelation is considered to be zero if the ACF values (at lags greater than $k = 0$) lie within the 95% confidence bands around zero, given by $\pm 1.96/\sqrt{N}$. Significantly non-zero ACF values and a non-random pattern indicate that the residuals are serially correlated. The PACF measures the autocorrelation at lag $k$ that is not accounted for by autocorrelations at shorter lags. While the PACF plot is not necessary for validating the model, if the ACF plot indicates correlated residuals, a time series model may be a more appropriate model for $\epsilon$ (e.g. $\epsilon_t = \phi\epsilon_{t-1} + z_t$ where $z \sim N(0, \sigma^2)$) and the PACF plot can be useful for identifying the order of this model.

- **Normal probability plot of residuals.** A normal probability plot, also known as a normal Q-Q plot, can be used to check whether the residuals are consistent with a Gaussian distribution (i.e. whether the normality assumption is reasonable). This plot is constructed by plotting sorted values of the standardised residuals against the corresponding theoretical values from the standard normal distribution. If the residuals are normally distributed, they will plot along, or close to, a straight line. Departure from this straight line indicates that the

residuals are probably not consistent with the Gaussian distribution. Additionally, the normal probability plot may indicate how the distribution differs from normal: significant deviations at the end of the line may indicate the presence of outliers, while curvature can indicate skewness or long tails (Heiberger and Holland, 2004).

- **Histogram of residuals.** A histogram of the residuals also allows for the normality of the residuals to be graphically checked. However, it is helpful to view such a plot in addition to the normal probability plot, as a histogram gives a clearer picture of the shape of the residual distribution, providing a graphical summary of the shape, scale, location and symmetry (or lack thereof) of the residuals. The normal probability plot, on the other hand, allows for easier detection of deviations from the normal distribution.

### *2.3. Predictive Validation*

#### *2.3.1. Underlying philosophy*

After the trained ANN model has passed the tests for replicative validity, all that is known is that the model provides a good fit to a single data set - the calibration data (Chapra, 1997). However, good performance of the model over the calibration data set does not guarantee correct predictive behaviour of the model (Power, 1993). This is because the calibration data might not be representative of the available data or the model might have been overfitted to the calibration data, thereby "learning" the specific patterns in the calibration data, rather than the general underlying relationship. Consequently, the purpose of predictive validation is to check whether the

model can generalize over the range of the data used for calibration (Fig. 1). In order to achieve this, the predictive performance of the model is checked on a dataset that was not used during calibration or any other part of the model development process (Maier et al., 2010). It should be noted that this independent dataset has been referred to either as test or validation set in literature. To clarify, in this paper, the validation data set is referred to as the dataset that is used to assess the performance of an ANN model once developed, and not to tune the model structure or prevent overfitting during the model development process. Care needs to be taken that the validation data are representative of the data used for calibration, which can be achieved using a range of data splitting methods (May et al., 2010; Wu et al., 2013).

### *2.3.2. Methods*

Predictive validity can be assessed by applying the trained ANN to an independent set of validation data and evaluating its performance. However, appropriate performance evaluation of a trained ANN model depends on the specific objectives of the model. Consequently, many different performance evaluation measures have been developed for indicating particular areas of model deficiency that are most important under differing viewpoints (e.g. accurate prediction of extremes may be considered more important than overall predictive accuracy or vice versa). In order to gain some consistency in the evaluation metrics used and reported in hydrological modelling studies, Dawson et al. (2007) developed HydroTest (`www.hydrotest.org.uk`), a free web resource that supports the statistical analysis of hydrological modelling output. This website provides a suite of quantitative metrics aimed primarily at

assessing hydrological model time series forecasts and, being designed to support numerous models being evaluated at the same time, it is ideally suited to data-driven modelling. While some of the HydroTest metrics will be irrelevant in certain environmental modelling studies (e.g. when the data are not a time series), the majority of these metrics are also included in the position paper by Bennett et al. (2013), who review methods and measures for evaluating the performance of environmental models in general. Therefore, in order to support and extend the use of consistent performance evaluation metrics in environmental modelling studies, it is suggested that all metrics from HydroTest be computed, allowing modellers to then select from these the appropriate measures that are most relevant to the particular requirements of the models being evaluated. If a single measure of performance is desired (allowing straightforward model inter-comparisons), integration of multiple HydroTest metrics into a single measure can be achieved using some variant of the ideal point error (IPE) as discussed in Dawson et al. (2012). The HydroTest metrics are listed in Table A.1, along with a brief description. For a more detailed explanation of these metrics readers are referred to Dawson et al. (2007, 2010); Bennett et al. (2013).

In addition to the metrics given in Table A.1, it is suggested that summary statistics of the observed and predicted datasets, including the mean, minimum, maximum, variance, standard deviation, skewness and kurtosis, be compared (these statistics are also returned by HydroTest). A comparison of such statistics between the observed and predicted data sets allows a 'direct value comparison', whereby the characteristics of the predicted and observed data sets are compared as a whole, rather than on a point-by-point

basis (Bennett et al., 2013). Ideally, the summary statistics computed for the model predictions should be very close in value to those computed based on the observations; however, a direct value comparison can be particularly useful for quickly identifying how the predictions might differ from the observations, which will not be obvious from the goodness-of-fit metrics given in Table A.1. Furthermore, the metrics in Table A.1 return a single value for the whole dataset, which can disguise significant divergent behaviour over time or space (Bennett et al., 2013). As such, it is also recommended that the first three plots described in Section 2.2.2 (scatter plot, Q-Q plot and plot of observed and predicted data versus data order) be constructed for the validation data, since these plots may provide valuable insights about the way a model performs that will not be evident from an assessment of such single-value metrics.

### *2.4. Structural Validation*

#### *2.4.1. Underlying philosophy*

As the data used to develop ANNs contain important information about the physical process being modelled, it is generally implied that a trained and (predictively) validated model represents the physical process of the system (Sudheer, 2005). However, ANN models that are both replicatively and predictively valid are not guaranteed to result in models that represent plausible physical relationships. This is most likely due to problems with equifinality (Beven and Freer, 2001), where different combinations of model parameters (e.g. connection weights) result in similar predictive performance (see Kingston et al., 2005b). Consequently, the purpose of structural validation is to check whether the input-output relationship captured by the model is

plausible in accordance with *a priori* system understanding (Fig. 1). While this approach does not determine whether the correct underlying relationship has been identified, it is helpful for identifying models that are *not* plausible from a physical perspective.

### *2.4.2. Methods*

Given the interconnected nature of MLP nodes and the nonlinear transfers applied within them, MLP connection weights are typically much less interpretable than the parameters of more traditional statistical models and, as such, provide little insight into the internal behaviour of the ANN model. In environmental modelling studies, efforts to extract the 'knowledge' embedded within a trained ANN have typically been aimed at quantifying the strength of the relationships between individual inputs and the output or at understanding the relationships represented by the hidden nodes. The latter approach is based on the idea that different physical sub-processes may be represented by individual hidden nodes (e.g., see Wilby et al., 2003; Jain et al., 2004; Sudheer and Jain, 2004; See et al., 2008; Jain and Kumar, 2009). However, due to the distributed nature of ANNs, individual hidden nodes generally do not correspond well with features in the problem domain. Rather, these physical components are likely to be encoded across a number of hidden nodes, and similarly, each hidden node may partially represent a number of different system components (Craven and Shavlik, 1997). Consequently, it may be difficult, in general, to structurally validate ANN models using these methods. The former approach includes different sensitivity analysis (SA) methods, whereby the effects of variation of the inputs on the output are assessed (Maier and Dandy, 1997; Maier et al., 1998;

Abrahart et al., 2001; Shahin et al., 2005; Sudheer, 2005; Park et al., 2007; Mount et al., 2013; Dawson et al., 2014), as well as methods based on the examination of the connection weights themselves (Olden and Jackson, 2002; Gevrey et al., 2003; Olden et al., 2004; Kingston et al., 2005b, 2006b; Jain et al., 2008).

While a number of authors have reviewed and compared the abilities of different methods to accurately quantify the relative importance (RI) of ANN inputs (Gevrey et al., 2003; Olden and Jackson, 2002; Olden et al., 2004; Kingston et al., 2010; de Oña and Garrido, 2014; Giam and Olden, 2015), the results of these comparisons have demonstrated that there is no approach for quantifying input importance that is consistently accurate. Rather, these methods are inherently unstable, being highly dependent on the network structure selected and the 'optimal' weights found during training. In addition, the results of previous comparison studies differed, and may have potentially been biased towards particular methods, as a result of the data used (i.e. certain methods may appear to be more accurate than others depending on the complexity - nonlinearity, monotonicity, variable interdependency and interactions, etc. - of the comparison data), making it difficult to reach a consensus on which method, if any, is the best for quantifying input RI. Sarle (2000) presents a useful discussion on the limitations of various methods for quantifying input RI and how some methods may be more accurate in certain situations than others. Based on this discussion, together with the results of the aforementioned comparison studies, five methods, namely Garson's, the Connection Weight (CW), modified CW (MCW), Profile and Partial derivatives (PaD) methods, are suggested for assessing the structural validity of

calibrated ANN models as part of the proposed validation framework. The first three methods directly use the connection weights to compute input RI, while the last two methods are SA approaches that examine the change in the model output as a result of input variation. These methods are described briefly below while further details, including the advantages and limitations of the methods, are provided in Appendix B.

1. **Garson's method:** Garson's algorithm (Garson, 1991), or the 'Weights' method as it was called in the comparison carried out by Gevrey et al. (2003), was one of the earliest methods proposed for quantifying the RI of ANN inputs based on the connection weights and has been used in numerous environmental modelling studies for extracting information from trained ANNs (Brosse et al., 1999; Abdul-Wahab and Al-Alawi, 2002; Mi et al., 2005; Jain et al., 2008; Langella et al., 2010; Sreekanth and Datta, 2010; Phukoetphim et al., 2014; Kumar, 2014; Coad et al., 2014; Beck et al., 2014). Using this method, input RI is calculated by partitioning the hidden-output layer connection weights into components associated with each input node using absolute values of the connection weights. Since absolute values of the weights are used, it is only possible to estimate the magnitude but not the direction of the input contributions (i.e. whether an input has a positive or negative effect on the output).
2. **CW method:** The CW approach of Olden and Jackson (2002) was found to provide the best overall methodology for quantifying ANN input RI in the comparison conducted by Olden et al. (2004) and has since been used to quantify input RI in a number of environmental

modelling studies (Joy and Death, 2004; Zanden et al., 2004; Kingston et al., 2005b, 2006b; Kemp et al., 2007; Shu and Ouarda, 2007; Watts and Worner, 2008; Watts et al., 2011; Beck et al., 2013; Sun, 2013). Using this approach, RI is computed based on an 'overall connection weight' between each input and the output, which in turn, is based on products of input-hidden and hidden-output connection weights for each input summed across all hidden nodes. In this approach, raw rather than absolute values of the weights are used, making it possible to estimate both the magnitude and direction of the input contributions.

3. **MCW method:** Kingston et al. (2006a, 2010) introduced a modified CW method, where input RI is computed in the same fashion as the CW approach; however, the raw input-hidden node weights are "squashed" using the hidden layer activation functions. This method is only suitable for use with hidden layer activation functions that are symmetric about the origin (i.e. $f(-x) = -f(x)$), such as the hyperbolic tangent (tanh) function. In comparison to the CW approach, this method has been shown to provide improved estimates of input RI in certain situations (Kingston et al., 2010).

4. **Profile method:** The Profile SA method, first described in Lek et al. (1995, 1996), involves successively varying each input variable across its range while keeping all others constant at their minimum, first quartile, median, third quartile, and maximum values; thus, producing five output profiles displaying variation in the output over the range of the input variable of interest. The median predicted responses across the

five output profiles is also calculated, from which it is possible to assess the median behaviour of the model, given a range of different input values. In addition, the RI of each input is calculated based on the magnitude of the range of median output values produced by varying each input. Being relatively quick and easy to apply, SA methods have been popular for investigating input contributions in ANNs used for environmental modelling applications (e.g., see Maier et al., 1998; Özesmi and Özesmi, 1999; Liong et al., 2000; Shahin et al., 2005; Young et al., 2011).

5. **PaD method:** The PaD method (Dimopoulos et al., 1995, 1999) is another type of SA approach that involves computing partial derivatives of the model output with respect to each input variable in order to define the local rate of change of the output with respect to the corresponding input, while holding all other inputs fixed. This method was found to be the most useful for quantifying input importance in the comparison carried out by Gevrey et al. (2003) and was also shown to perform well in the comparison presented by Olden et al. (2004). It has since been used successfully in a number of environmental modelling studies to quantify ANN input variable contributions Park and Chung (2006); Park et al. (2007); Tison et al. (2007); Vasilakos et al. (2008); Laffaille et al. (2009); Olaya-Marín et al. (2012); Kumar (2012); Mount et al. (2013); Dawson et al. (2014). Similar to the Profile method, this approach returns a profile of partial derivatives for each ANN input, which can be interpreted in a similar way to the coefficients in linear models, as well as a measure of input RI for each input. The sensitivity

(partial derivatives) profiles enable model behaviour to be interpreted with respect to process rationality (see Mount et al. (2013)).

## 3. R-Package for Implementing Proposed Validation Framework

A toolbox for implementing the proposed validation framework is available in the **validann** package, which has been developed for the R software environment (R Core Team, 2015) and is available from the Comprehensive R Archive Network (CRAN) at http://CRAN.R-project.org/package=validann. The R environment was chosen as the development platform for this toolbox for a number of reasons. Firstly, it is free, open source and runs on all major platforms. Secondly, its package system allows for the simple distribution, use and maintenance of third-party code. Finally, a user's ability to add functions and write scripts in R facilitates the extension and adaptation of the functionality provided by the standard R environment and its many add-in packages. As such, the **validann** R package should not only enable researchers to readily access the proposed ANN validation methods, but also to manipulate and adapt these methods as required in order to integrate them into their own work; thus encouraging their maximum uptake and use. While there are already methods and packages available within the R environment that can be used to perform many of the validation tests recommended within the proposed validation framework (e.g. **hydroGOF** (Zambrano-Bigiarini, 2014) for computing and plotting goodness-of-fit measures between observed and simulated values, **NeuralNetTools** (Beck, 2015) for performing sensitivity analyses and computing ANN input importance measures, and indeed many of the other statistical and plotting methods available in the pre-installed

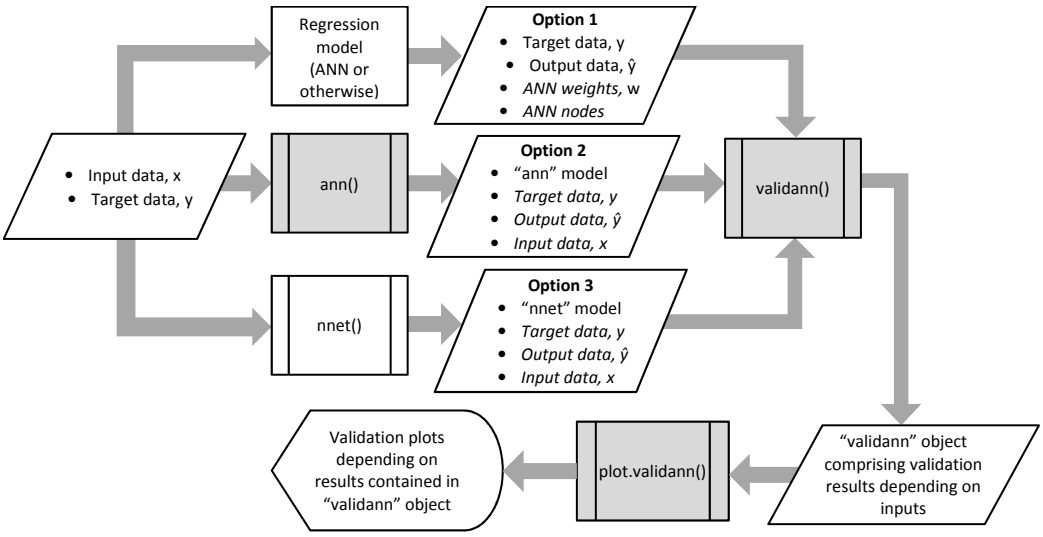


Figure 2: Structure and core functions (shaded grey) of the **validann** R-package. Italics are used to denote optional inputs to the functions.

R base packages), the **validann** package expands upon these methods and combines them into a single validation package that can be easily applied for consistent and comprehensive validation of ANN models developed both within and outside of the R environment.

As shown in Fig. 2, the **validann** package has three core functions. The `validann()` and `plot.validann()` functions have been designed to achieve the primary objective of the **validann** package, which is to compute the replicative, predictive and structural validation results associated with the proposed validation framework, as outlined in Section 2, and to present these results in a user-friendly and efficient manner. In addition, the package includes the `ann()` function for constructing ANN models. These functions are described in further detail below.

The `ann()` function is a method for training single hidden layer MLPs

with a specified model structure (i.e. number of hidden layer nodes, hidden and output layer activation functions). This function is similar to the available `nnet()` function from package **nnet** (Venables and Ripley, 2002); however, it gives greater flexiblity by providing a choice between four alternative activation functions for the hidden and output layer nodes, including the hyperbolic tangent (tanh), logistic sigmoid, linear (or identity) and exponential functions, as well as allowing a user-defined error or objective function. More importantly, in the context of ANN validation, this function returns partial derivatives of the hidden and output node outputs with respect to their inputs, enabling computation of absolute and relative input sensitivities using the PaD structural validation method described in Section 2.4.2 and Appendix B. As a result, the `ann()` function is more compatible with the proposed ANN validation framework than other available ANN fitting functions that do not provide this output.

Essential arguments to the `ann()` function are the input ($\mathbf{x}$) and target ($y$) training data and the number of hidden layer nodes. By default, the method uses a tanh activation function for the hidden layer nodes and a linear activation at the output layer. The default objective function is the sum of squared residuals as defined by Eq. 2 and training is performed using the built-in `optim()` R function with the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, a quasi-Newton gradient-based optimisation method, as a default (although any of the `optim()` methods may be selected if appropriate). Once a fitted ANN model has been obtained using `ann()`, other standard R methods are provided to work with the '`ann`' objects returned. These include `predict()` to predict model outputs using a trained ANN

and new input data, as well as `fitted()`, `observed()` and `residuals()` to extract the training outputs, targets and model residuals, respectively.

Function `validann()` is the foundation of the **validann** package. This generic function computes all of the validation metrics and statistics discussed in Section 2 according to the class of ANN model (if supplied) and the data provided. There are three main options for using this function, as shown in Fig. 2, where italics are used to denote optional inputs to the functions. The first option (Option 1 in Fig. 2) takes observed target data and simulated model outputs as inputs and returns goodness-of-fit metrics, model residuals and statistics related to the distribution of the residuals and the observed and simulated data. Additionally, if the weights of a trained ANN are supplied together with the numbers of nodes in each layer, input relative importance measures computed using Garson's method and the CW method will be returned. However, since this option only allows for limited information regarding the internal dynamics of the model to be provided, additional structural validation metrics cannot be computed. As such, this option is the least preferred, as it only allows for limited structural validation of the model. However, it is also the most general option and may be useful in cases where the ANN model has been built outside of the R environment and/or is not of class '`ann`' or '`nnet`' (or indeed is not even an ANN). It may also be useful for predictive validation, once replicative and structural validation metrics have already been computed using either Options 2 or 3 in Fig. 2, as discussed below.

The second `validann()` option (Option 2 in Fig. 2) is the most preferred, where the ANN model is built using function `ann()`. This allows for the

most comprehensive validation of the ANN model, as the MCW and PaD structural validation results are only produced if the ANN model is of class '`ann`' as returned by the `ann()` function. Additionally, both the Profile and PaD methods will only be carried out if the input data used for training are supplied, while the MCW input RI values will only be computed if the default tanh hidden layer activation function is employed when building the ANN model (since this is the only activation function available within function `ann()` that has a squashing effect on the weights and is symmetric about the origin. See Section 2.4.2). Output and target data are only optional inputs using this option, since if they are not supplied, the output and target data stored in the '`ann`' object will be used for computing goodness-of-fit metrics, residuals and data summary statistics. This may be sufficient for replicative validation; however, for predictive validation, observed and simulated data for an independent validation set must be supplied.

The third option for calling the `validann()` function (Option 3 in Fig. 2) allows for validation of ANN models of class '`nnet`' built using the `nnet()` function from package **nnet**. Given the same inputs, this option will return similar results to Option 2, with the exception of the MCW and PaD structural validation results. This is because the default logistic sigmoid hidden layer activation function adopted within the `nnet()` function is not symmetric about the origin as required by the MCW input RI method, while the hidden and output node partial derivatives required by the PaD method are not returned by the `nnet()` function. As with Option 2, the output and target data are optional inputs (since corresponding data stored in the '`nnet`' object may be used); however, for predictive validation, these data must be

supplied.

It is important to note that, regardless of which option is chosen, the `validann()` function must be called twice in order to produce results for predictive and replicative validation: once with the training data, and ideally the ANN weights and model structure, as inputs (replicative and structural validation) and then again using the independent validation data (predictive validation). All three of the options return a list object of class '`validann`' which includes components according to the inputs supplied when calling the `validann()` function. At most (i.e. when the ANN model is of class '`ann`' with tanh hidden layer activation function and input data are included in the function call), a '`validann`' object will be comprised of the components given in Table 1.

Finally, the `plot.validann()` function is a plot method for objects of class '`validann`' that produces a series of plots according to the components of the validann object supplied. By default, the plots produced are grouped into goodness-of-fit, residual analysis and sensitivity analysis plots, with multiple plots to a page, as follows:

- Goodness-of-fit plots (predictive, replicative validation): scatter and Q-Q plots of observed versus predicted data and observed and predicted data against data order.

- Residual analysis plots (replicative validation): histogram and normal probability plot of residuals; residual autocorrelation and partial autocorrelation plots; standardised residuals against predicted data and standardised residuals against against order of the data.

Table 1: Components of a `validann` object.

| Component name | Description |
|---|---|
| `metrics` | Values of the metrics given in Table A.1 computed based on the observed ($y$) and predicted ($\hat{y}$) data supplied or stored in the supplied ANN model. |
| `residuals` | A series of residuals ($y - \hat{y}$) computed based on the observed and predicted data supplied or stored in the ANN model. |
| `obs_stats`, `sim_stats`, `resid_stats` | Mean, minimum, maximum, variance, standard deviation, skewness and kurtosis values computed based on the observed and predicted data and on the model residuals. |
| `ri` | Relative importance values for each input computed according to the five methods described in Section 2.4.2 and Appendix B. |
| `y_hat` | Model response values indicating the local sensitivity of the model to each input, calculated using the Profile method, as described in Section 2.4.2 and Appendix B. |
| `as` | *Absolute* sensitivity values for each input calculated according to the PaD method described in Section 2.4.2 and Appendix B. |
| `rs` | *Relative* sensitivity values for each input calculated according to the PaD method described in Section 2.4.2 and Appendix B. |

- Sensitivity analysis plots (structural validation): Profile sensitivity plots: for each input, plots of predicted response versus percentile of input; PaD sensitivity plots: for each input, plots of relative and absolute sensitivity versus observed response.

The '`plot.validann`' function has as optional inputs the logical arguments '`gof`', '`resid`' and '`sa`', which control whether or not the goodness-of-fit, residual analysis and sensitivity analysis plots, respectively, will be produced and, by default, are all set to true. It is possible to 'turn off' a group of plots by setting the corresponding argument to false when calling the '`plot.validann`' function. For example, if arguments '`resid`' and '`sa`' are set to false, no residual analysis or sensitivity analysis plots will be output. This may be useful when the '`validann`' object has been computed based on independent validation data, since the goodness-of-fit plots are of primary interest for predictive validation. Additionally, plots will not be produced if the required components of the '`validann`' object are empty (e.g. no sensitivity analysis plots will be produced if components '`y_hat`', '`rs`' and '`as`' have not been populated). If the plot device is interactive (i.e. the screen), the user is prompted to view the next plot or group of plots. However, if another graphics device is specified (e.g. jpeg, postscript, pdf), all plots will be displayed in a single file. The style and format of the plots produced by the `plot.validann()` function are not easily manipulated; however, all validation results used in the creation of the plots are stored in the '`validann`' object returned by function `validann()`, giving users the ability to create their own validation plots as desired.

## 4. Case Studies

The proposed ANN validation framework was applied to two real environmental modelling case studies in order to demonstrate the benefits of considering replicative and structural validity in addition to predictive validity. Since not all of the proposed framework methods are suited to all types of problems, the case studies were selected to demonstrate the framework when applied to two problems that are fundamentally different in nature: (i) a forecasting problem with strong temporal dependencies and highly correlated inputs and (ii) a prediction problem with no temporal component and relatively independent inputs. The results of these case studies, presented in Section 5, also demonstrate the types of outputs generated by the core functions of the R-package **validann**.

### *4.1. Background and Data*

#### *4.1.1. River Murray (Australia) salinity forecasting*

The River Murray salinity (RMS) dataset has been studied extensively in the context of ANN development, where the aim has generally been to forecast salinity concentrations in the River Murray at Murray Bridge, South Australia, 14 days in advance (e.g., Maier and Dandy, 1996, 2000; Bowden et al., 2002, 2005; Kingston et al., 2005a, 2008; Fernando et al., 2009; Wu et al., 2013; Li et al., 2014). The available dataset includes 4140 daily observations of 16 variables, including streamflow, water level and salinity at several locations along the River Murray upstream of Murray Bridge, for the period from December 1986 to April 1998. Previous studies used approximately half of the available data (December 1986 - June 1992) for ANN development,

while the remaining data (July 1992 - April 1998) were reserved to simulate a real-time forecasting situation using the ANN models developed (Bowden et al., 2005; Kingston et al., 2005b, 2008; Fernando et al., 2009). To determine the important inputs for forecasting Murray Bridge salinity 14 days in advance, Fernando et al. (2009) used a partial mutual information (PMI) approach to select from a total of 1304 candidate inputs (including lags of up to 113 days for each of the 16 candidate input variables). They found three inputs to be significant: Waikerie salinity (WAS), Mannum salinity (MAS) and flow at Lock 7 (L7F), each a time lag of one day ($t-1$).

In line with previous studies, variables $\mathrm{WAS}_{t-1}$, $\mathrm{MAS}_{t-1}$ and $\mathrm{L7F}_{t-1}$ were used as inputs for forecasting Murray Bridge salinity 14 days in advance ($\mathrm{MBS}_{t+13}$), with data between December 1986 and June 1992 used for training and data from July 1992 to April 1998 used for independent validation. A time series plot of the target $\mathrm{MBS}_{t+13}$ data is shown in Fig. 3 (a), where data to the left of the red dashed line are the training targets, while those to the right of the line are the validation targets. In Fig. 3 (b), a histogram of the $\mathrm{MBS}_{t+13}$ data shows that the distribution of these data is reasonably normal. In Table 2, it can be seen that the upstream salinity and flow inputs for this forecasting problem are moderately to highly correlated with one another and with the target salinity concentration at Murray Bridge, and each input and the output are highly autocorrelated.

### *4.1.2. Surface water turbidity prediction, Australia*

The southern Australian turbidity (SAT) dataset has previously been studied by van Leeuwen et al. (1999) and Maier et al. (2004) who developed ANN models to assist treatment plant operators with determining optimal

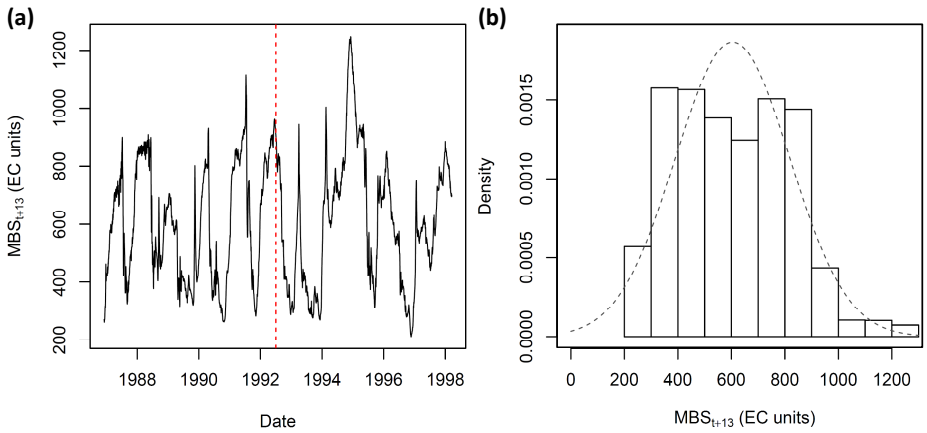


Figure 3: (a) Time series of $\text{MBS}_{t+13}$ data. The red dashed line denotes the split between training and validation data; training data are to the left and validation data to the right. (b) Histogram of the $\text{MBS}_{t+13}$ data. The grey dashed line denotes the Gaussian distribution.

Table 2: River Murray salinity data cross- and autocorrelation coefficients

| | $\text{MAS}_{t-1}$ | $\text{WAS}_{t-1}$ | $\text{L7F}_{t-1}$ | $\text{MBS}_{t+13}$ |
|---|---|---|---|---|
| *Cross-correlation* | | | | |
| $\text{MAS}_{t-1}$ | 1.00 | 0.86 | -0.66 | 0.91 |
| $\text{WAS}_{t-1}$ | | 1.00 | -0.74 | 0.94 |
| $\text{L7F}_{t-1}$ | | | 1.00 | -0.72 |
| *Autocorrelation* | | | | |
| Lag-1 | 0.996 | 0.996 | 0.999 | 0.996 |

alum doses for water treatment plants in southern Australia. In addition, the dataset has subsequently been used by Wu et al. (2013) for comparing the performance of different data splitting methods used in the development of ANN models.

The SAT dataset, as discussed in Maier et al. (2004), comprises 202 measurements of raw and treated water quality parameters including turbidity, pH, colour, ultraviolet absorbance at a wavelength of 254 nm (UVA-254), alkalinity and dissolved organic carbon (DOC), together with the corresponding alum doses. Raw water parameters were collated from 29 raw water samples collected from 14 different surface water sources located in southern Australia. The corresponding treated water quality parameters were measured from jar tests, where each of the raw water samples was dosed with a number of different alum concentrations and the resulting water quality parameters were recorded. Wu et al. (2013) used a PMI approach to select the relevant inputs for predicting treated water turbidity (TwTurbidity) from the six raw water quality parameters (RwTurbidity, RwPh, RwColour, RwUvAbs254, RwAlkalinity and RwDOC) and the alum dose, finding RwTurbidity, RwPh, RwColour, RwUvAbs254 and the alum dose to be significant. They then used four data splitting methods to divide the available data into training (60%), testing (20%) and validation (20%) datasets.

In this study, the data split obtained by Wu et al. (2013) using the DUPLEX data splitting method (Snee, 1977) was used for training and validating the ANNs developed. However, for the purposes of the current study, where optimal model selection and cross-validation during training were not applied, a testing dataset was not needed; thus, the training and testing data

Table 3: SAT dataset cross-correlation coefficients

| | RwTurbidity | RwPh | RwColour | RwUvAbs254 | Alum Dose | TwTurbidity |
|---|---|---|---|---|---|---|
| RwTurbidity | 1.00 | -0.05 | 0.14 | -0.21 | 0.10 | 0.40 |
| RwPh | | 1.00 | -0.15 | 0.08 | 0.20 | -0.01 |
| RwColour | | | 1.00 | 0.76 | 0.32 | 0.14 |
| RwUvAbs254 | | | | 1.00 | 0.39 | 0.00 |
| AlumDose | | | | | 1.00 | -0.22 |

were combined. As a result, 162 data samples (80%) were used for training and the remaining 40 samples (20%) were reserved for validation of the models. The inputs used for predicting TwTurbidity were also those selected by Wu et al. (2013) using the PMI approach (RwTurbidity, RwPh, RwColour, RwUvAbs254 and alum dose). In comparison to the River Murray salinity case study, with the exception of inputs RwUvAbs254 and RwColour, the SAT inputs are relatively uncorrelated either with each other or with the target TwTurbidity data, as can be seen in Table 3. Furthermore, unlike the RMS dataset, there is no time component to the SAT data. A plot of the TwTurbidity samples, together with a histogram of these data, is shown in Fig. 4, where it can be seen that the distribution of the TwTurbidity data is significantly non-Gaussian (positively skewed), with the majority of TwTurbidity values lying close to 0 NTU.

### *4.2. ANN Model Development and Validation*

For each case study, 15 different ANN structures were considered with the number of hidden nodes increasing from 1 to 15. Additionally, for each of the 15 network structures, the connection and bias weights were initialised five times with different random starting values between -0.1 and 0.1, re-

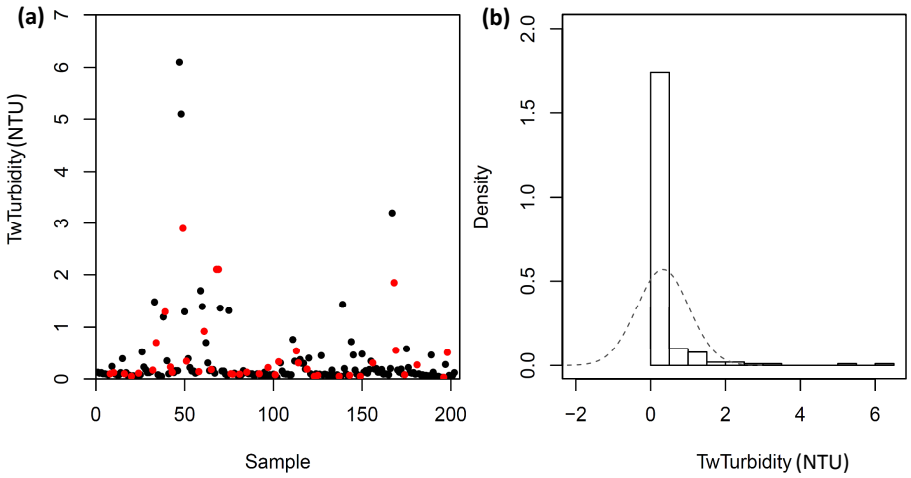


Figure 4: (a) SAT target TwTurbidity data. Black dots denote the training data; red dots denote the validation data. (b) Histogram of the TwTurbidity data. The grey dashed line denotes the Gaussian distribution.

sulting in a total of 75 ANN models being developed for each case study. All ANNs were single hidden layer networks with hyperbolic tangent (tanh) hidden layer activations and a linear activation at the output. All input data were standardised to have a mean of zero and standard deviation of one, while the target data were linearly rescaled between 0 and 1. The models were built in R (3.2.2) using the `ann()` function from the **validann** package discussed in Section 3, with the default BFGS optimisation algorithm used for training. All models were trained without cross-validation or early stopping for a maximum of 500 iterations using the default sum of squared residuals as an objective function.

To validate the models, the `validann()` function from the **validann** package was applied twice to each model: the first time using the (unscaled) training data to obtain replicative and structural validation results, and the

second time using the (unscaled) independent validation dataset to obtain predictive validation results. Three of the best performing models, in terms of predictive validity, were selected for each case study and used to compare and contrast the corresponding replicative and structural results.

## 5. Results and Discussion

### *5.1. River Murray salinity forecasting*

Predictive validation results for the RMS dataset are presented in Table 4. The three models fitted to this dataset and selected for comparison have been named RMS1, RMS2 and RMS3 and details of these models in terms of their size (number of hidden nodes and weights) and the random seed used to initialise the weights are also given in this table. Four summary statistics, namely the mean, standard deviation (SD), skewness and kurtosis, are presented in Table 4 to compare the overall distributions of the model outputs with that of the observed data. Additionally, five performance evaluation metrics, namely the RMSE, AIC, MARE, RSqr and CE, have been selected from Table A.1 to summarise the fit between the model outputs and the validation data. These performance metrics were selected as they are widely used in environmental modelling studies and provide a good summary of how well the model fits the data over a range of different magnitudes (low, average and high), as well as a comparison between the model fit and model complexity. Moreover, they are applicable to data with or without a time component and, consequently, are also suitable for assessing the performance of the turbidity case study models. As can be seen in Table 4, all three models give a good fit to the validation data (CE$\geq$ 0.9),

with relatively little difference in their predictive performance, particularly considering the large variation in the size of the three models. As can also be seen, there is no definitive "best" model in terms of the performance metrics or summary statistics presented. Rather, model RMS1 with 13 hidden nodes appears to give the best overall fit to the data, while model RMS3 with three hidden nodes is the most parsimonious, providing a comparable fit to the data with significantly fewer weights (free parameters). Model RMS2 sits between these other models, achieving a slightly better fit to the data than RMS3, but still with many fewer weights than RMS1.

Table 4: River Murray salinity predictive validation results. Best results are highlighted in bold text.

| | RMS1 | RMS2 | RMS3 | Observed |
|---|---|---|---|---|
| Hidden nodes | 13 | 5 | 3 | - |
| # of weights | 66 | 26 | 16 | - |
| Random seed | 3 | 3 | 1 | - |
| RMSE | **66.7** | 67.1 | 67.6 | - |
| AIC | 8897 | 8831 | **8824** | - |
| MARE | **7.35** | 7.95 | 7.41 | - |
| RSqr | 0.929 | 0.935 | **0.937** | - |
| CE | **0.915** | 0.914 | 0.913 | - |
| Mean | **584.8** | 582.3 | 578.9 | 608.1 |
| SD | **206.6** | 199.6 | 200.2 | 228.5 |
| Skewness | 0.35 | **0.38** | 0.47 | 0.41 |
| Kurtosis | 2.40 | 2.34 | **2.62** | 2.69 |

Model performance results for models RMS1, RMS2 and RMS3 when applied to the training data (replicative validity) are given in Table 5. These results are similar to the predictive validation results presented in Table 4, in that an improved fit to the data is achieved as the number of parameters is in-

creased. This is not surprising, since no early stopping to prevent overfitting was applied. However, when applied to the training data, the best (smallest) AIC value was also obtained using the largest model (RMS1), suggesting the extra complexity of this model is warranted given the superior fit achieved.

Table 5: River Murray salinity replicative validation results. Best results are highlighted in bold text.

| | RMS1 | RMS2 | RMS3 | Observed |
|---|---|---|---|---|
| RMSE | **32.7** | 35.4 | 37.6 | - |
| AIC | **7187** | 7266 | 7367 | - |
| MARE | **4.2** | 4.5 | 4.8 | - |
| RSqr | **0.973** | 0.968 | 0.964 | - |
| CE | **0.973** | 0.968 | 0.964 | - |
| Mean | **600.7** | **600.7** | **600.7** | 600.7 |
| SD | **194.9** | 194.4 | 194.0 | 197.6 |
| Skewness | **0.11** | **0.11** | 0.12 | 0.11 |
| Kurtosis | 1.68 | **1.69** | 1.67 | 1.75 |

From the results presented in Tables 4 and 5, RMS1 may be considered the "optimal model", since this model gives the best fit to both the training and validation datasets. However, the results of the residuals analysis for this model, presented in Fig. 5, show that the residuals are strongly autocorrelated, as indicated by the ACF plot in Fig. 5(c), where the majority of lags show significant autocorrelation (ACF values outside of the 95% confidence bands). In fact, similar results were observed for all three models RMS1, RMS2 and RMS3 (although not shown here for the purpose of brevity), indicating a possible deficiency in the models, which might be due to the omission of important input information. Ideally, in such circumstances, the model development steps should be revisited, including the selection of model inputs.

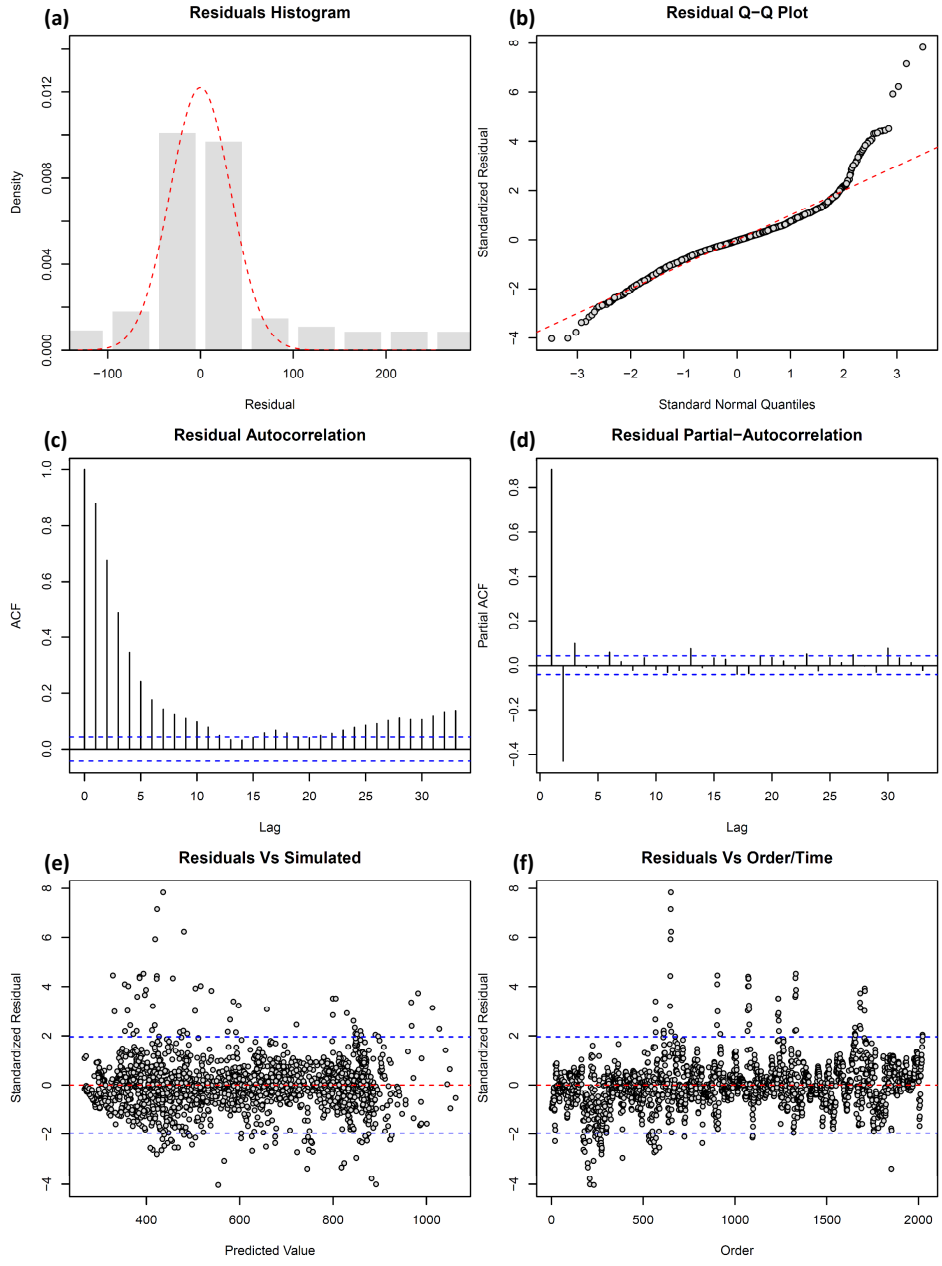


Figure 5: Residual analysis plots obtained using the `plot.validann()` function applied to model RMS1. Blue dashed lines denote the 95% confidence bands, while red dashed lines in (a) and (b) denote the Gaussian distribution and those in (e) and (f) show the zero line.

However, reselection of model inputs was beyond the scope of this paper and the following autoregressive error model with lag-2 autocorrelations (AR(2)) was instead assumed in an attempt to account for any predictable component remaining in the residuals:

$$\epsilon_t = \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + z_t; \quad z_t \sim N(0, \sigma_z^2) \tag{4}$$

The order of this error model was selected according to the number of lags displaying significant autocorrelation in the PACF plot shown in Fig. 5(d). The models were retrained using the new error model and residual analysis methods were subsequently applied to the innovations, $z$, rather than the raw residuals, in order to test the replicative validity of the three new models RMS1-AR2, RMS2-AR2 and RMS3-AR2.

As can be seen in Fig. 6, the autocorrelation was reasonably well captured by the error model given by Eq. 4 for all three models, since the ACF of the innovations, $z_t$, at lags $\geq 1$ are mostly within the 95% confidence bands around zero (as denoted by the blue dashed lines in Fig. 6). While there is some autocorrelation (predictable structure) remaining, this is minimal, particularly for models RMS2-AR2 and RMS3-AR2. In addition, with reference to the predictive validation results presented in Table 6, it can be seen that, although a slightly inferior fit to the validation data was achieved using an AR(2) error model than the standard SS residuals objective function, a good fit (CE $\geq 0.9$) to these data was still achieved by all three models. In this case, the RMS2-AR2 and RMS3-AR2 models appear to be the most predictively valid according to the metrics and statistics presented in Table 6.

Using the PMI input selection procedure, Fernando et al. (2009) found

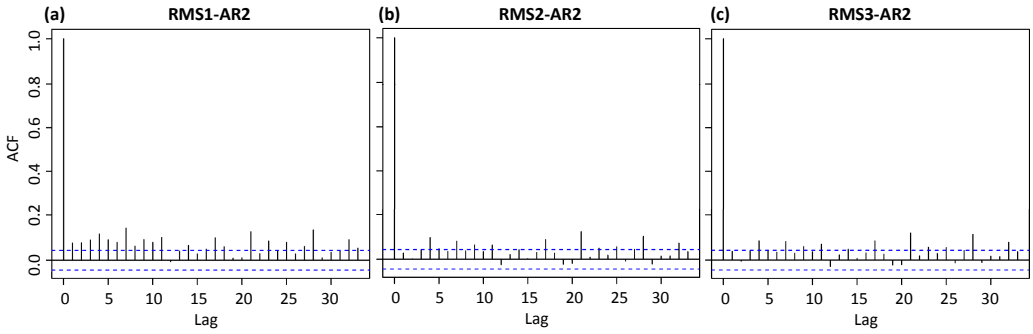


Figure 6: ACF plots obtained using models (a) RMS1-AR2, (b) RMS2-AR2 and (c) RMS3-AR2. Blue dashed lines denote the 95% confidence bands around zero.

Table 6: Predictive validation results for models RMS1-AR2, RMS2-AR2 and RMS3-AR2. Best results are highlighted in bold text.

| | RMS1-AR2 | RMS2-AR2 | RMS3-AR2 | Observed |
|---|---|---|---|---|
| RMSE | 72.4 | **71.1** | 71.3 | - |
| AIC | 9068 | 8952 | **8936** | - |
| MARE | 8.3 | **7.7** | 8.4 | - |
| RSqr | 0.927 | **0.935** | 0.932 | - |
| CE | 0.900 | **0.903** | **0.903** | - |
| Mean | **583.8** | 581.1 | 582.7 | 608.1 |
| SD | 189.7 | 190.0 | **192.1** | 228.5 |
| Skewness | 0.37 | 0.26 | **0.40** | 0.41 |
| Kurtosis | 2.44 | 2.34 | **2.58** | 2.69 |

that the order of importance of the selected RMS inputs, from most important to least, was $WAS_{t-1}$, $MAS_{t-1}$ then $L7F_{t-1}$. This finding is supported by the scatterplot of RMS model inputs versus $MBS_{t+13}$ presented in Fig. 7, where it can be seen that there is strong, positive correlation between the output $MBS_{t+13}$ and inputs $WAS_{t-1}$ and $MAS_{t-1}$, with the $WAS_{t-1}$-$MBS_{t+13}$ relationship showing slightly less scatter. It can also be seen that there is a strong inverse relationship between $MBS_{t+13}$ and $L7F_{t-1}$, particularly at the lower salinity levels (higher flows) (the correlation coefficients presented in Table 2 also support the findings of the PMI input selection; however, these coefficients only capture linear relationships). However, there are also important interactions between the inputs, given the way in which salinity transport depends on both flow rates and upstream salinity levels. The travel time between Waikerie and Murray Bridge is approximately 14 days when flow rates are around 17,000-21,000 ML/day, while the travel time between Mannum and Murray Bridge is approximately 14 days when flow is around 6500 ML/day (Maier and Dandy, 1996). As such, the importance of inputs $WAS_{t-1}$ and $MAS_{t-1}$ in predicting $MBS_{t+13}$ varies depending on the flow rate. For flows greater than 21,000 ML/day, the travel times between both upstream locations and Murray Bridge are less than 14 days and, thus, current salinity levels at Waikerie and Mannum become irrelevant to the salinity concentration at Murray Bridge 14 days in advance. This flow rate coincides with that in Fig. 7 where a significant change in the relationship between $MBS_{t+13}$ and $L7F_{t-1}$ can be seen.

The RI values for models RMS1-AR2, RMS2-AR2 and RMS3-AR2 as calculated using the five methods discussed in Section 2.4.2 are presented in

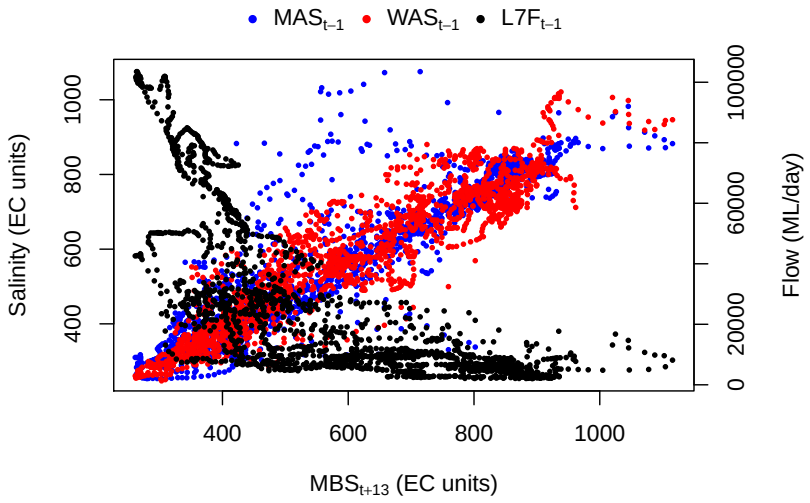


Figure 7: Scatter plot of RMS inputs versus MBS.

Table 7. As can be seen, model RMS2-AR2 is the only model for which the input RI values across all of the calculation methods correspond to the order of input importance found by Fernando et al. (2009). Additionally, this is the only model for which the signs of the input contributions are correct when calculated according to the CW and MCW methods (the only methods that indicate the sign of the contribution). While variations in the RI results may be due to deficiencies in the methods used to compute these values, the fact that all structural validity results for model RMS2-AR2 are consistent with *a priori* knowledge about the input-output relationship gives confidence that the modelled relationship is plausible.

Table 7: River Murray salinity input RI values.

| Model | $MAS_{t-1}$ | $WAS_{t-1}$ | $L7F_{t-1}$ |
|---|---|---|---|
| | *Garson* | | |
| RMS1-AR2 | 19.2 | 34.3 | 46.5 |
| RMS2-AR2 | 28.3 | 45.4 | 26.3 |
| RMS3-AR2 | 15.8 | 31.4 | 52.8 |
| | *CW* | | |
| RMS1-AR2 | 0.2 | 36.8 | -63.0 |
| RMS2-AR2 | 41.5 | 51.9 | -6.6 |
| RMS3-AR2 | 71.3 | 25.7 | 3.0 |
| | *MCW* | | |
| RMS1-AR2 | -5.6 | 49.0 | -45.4 |
| RMS2-AR2 | 36.4 | 44.8 | -18.8 |
| RMS3-AR2 | 38.2 | 20.5 | -41.3 |
| | *Profile* | | |
| RMS1-AR2 | 24.8 | 42.2 | 33.0 |
| RMS2-AR2 | 33.0 | 49.4 | 17.7 |
| RMS3-AR2 | 32.1 | 45.6 | 22.4 |
| | *PaD* | | |
| RMS1-AR2 | 33.8 | 26.7 | 39.5 |
| RMS2-AR2 | 37.6 | 26.2 | 36.2 |
| RMS3-AR2 | 38.3 | 24.6 | 37.1 |

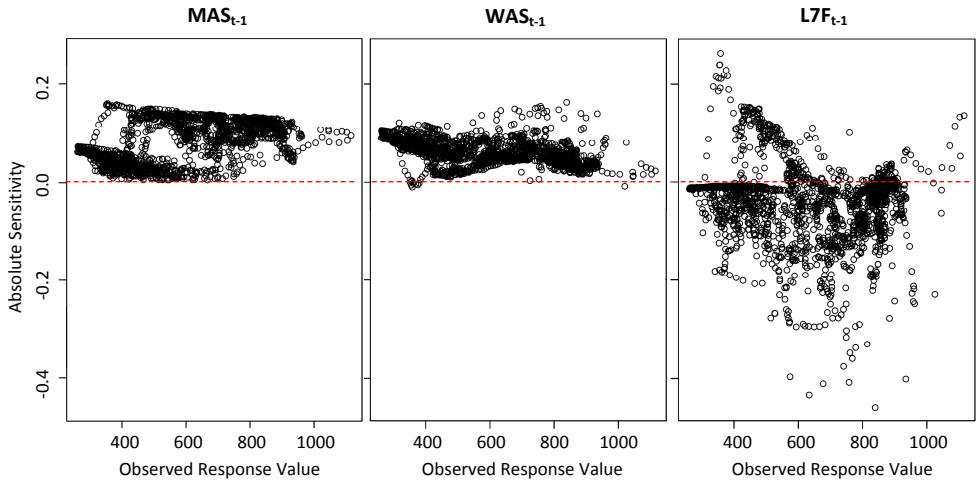


Figure 8: Absolute sensitivity plots for each RMS input obtained using the PaD method applied to model RMS2-AR2.

In addition to the single-valued input RI measures, it is important to consider the profiles of input sensitivities, which reveal detailed, local patterns of input-output sensitivity; thus, giving better insight into how the model behaves as an input is varied over its range. However, given that the inputs associated with this case study are strongly correlated with one another, the Profile method is not suitable for assessing input sensitivities, as infeasible combinations of the inputs would most likely be used in their calculation. The PaD method, on the other hand, is suitable for computing input sensitivities for this case study. The absolute sensitivity, or partial derivative, profiles obtained using the PaD method applied to model RMS2-AR2 are shown in Fig. 8.

By inspection of these profiles, the modelled relationships again appear to be consistent with knowledge about the underlying process: the partial derivatives of the output calculated with respect to inputs $\text{WAS}_{t-1}$ and

$MAS_{t-1}$ predominantly lie above the zero line (denoted by the red dashed line), indicating a positive relationship between these inputs and $MBS_{t+13}$, while those calculated with respect to input $L7F_{t-1}$ mostly lie below the zero line, indicating that there is typically an inverse relationship between $L7F_{t-1}$ and $MBS_{t+13}$. Additionally, there appear to be two separate relationships between input $MAS_{t-1}$ and $MBS_{t+13}$ (as observed from the two apparent clusters of absolute sensitivity values in the $MAS_{t-1}$ plot in Fig. 8), with $MAS_{t-1}$ displaying relatively little importance (absolute sensitivity values close to zero) when $MBS_{t+13}$ values are low (these typically correspond with relatively high flows) and greater importance when $MBS_{t+13}$ values are greater than 600 EC units (which tend to occur when flow rates are less than 20,000 ML/day). The absolute sensitivity profile for input $WAS_{t-1}$, on the other hand, suggests that this input is most important when forecasting low to mid-range Murray Bridge salinities and less important when forecasting high salinities, which typically occur when flow rates are low. These results are consistent with knowledge about the ranges of flow rates that result in travel times of around 14 days from both of the upstream locations and, hence, under which flow rates the upstream salinity inputs would contribute most to the prediction of $MBS_{t+13}$. Consequently, since the results presented in Table 7 and Fig. 8 demonstrate plausible input-output relationships have been captured by RMS2-AR2, this model can be considered structurally valid. This is in contrast to model RMS1-AR2, whose absolute sensitivity profiles are shown in Fig. 9. As can be seen in this figure, significantly more partial derivative values lie below the zero line for input $MAS_{t-1}$ and above the zero line for input $L7F_{t-1}$ when compared with the

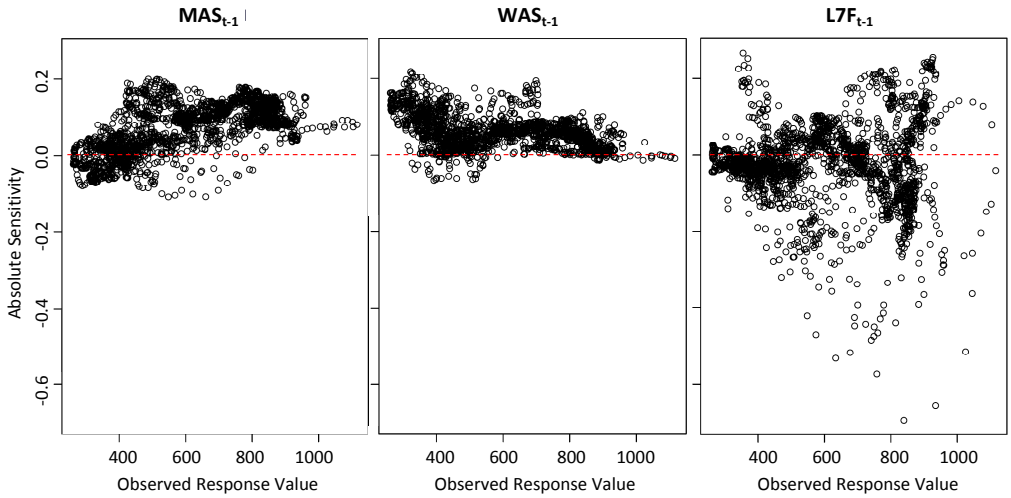


Figure 9: Absolute sensitivity plots for each RMS input obtained using the PaD method applied to model RMS1-AR2.

plots shown in Fig. 8 for model RMS2-AR2. Additionally, the magnitudes of the sensitivities for input $\text{L7F}_{t-1}$ for certain $\text{MBS}_{t+13}$ values between 600-1200 EC units (corresponding to low-mid range flows) are greater than those obtained using model RMS2-AR2 for the same range of $\text{MBS}_{t+13}$ values, suggesting that model RMS1-AR2 attributes greater importance to this variable than RMS2-AR2 over this range of values. Moreover, model RMS1-AR2 attributes significantly more importance to input $\text{L7F}_{t-1}$ than either of the upstream salinity inputs over this range (which coincides with low-mid range flow rates), which is not in agreement with the results of the PMI input selection or the scatter plots presented in Fig. 7. These results can also be seen in the RI values computed using the CW and MCW methods, with relatively little importance given to input $\text{MAS}_{t-1}$ and greater (negative) importance attributed to input $\text{L7F}_{t-1}$.

Overall, it has been found that model RMS2-AR2 is best suited to forecasting $MBS_{t+13}$, when taking into account the predictive, replicative and structural validity of the models considered. This is in contrast to model RMS1, which, although resulting in the best fit to both the training and validation data, was a significantly more complex model (with 66 weights compared to 26 for RMS2-AR2) and did not appropriately capture the underlying input-output relationship (there was remaining non-random structure in the residuals).

### *5.2. Surface water turbidity prediction*

The models fitted to the SAT dataset and selected for comparison were named SAT1, SAT2 and SAT3 and the predictive validity of these models was compared using the same performance metrics and data summary statistics as were used for the previous case study. These results, along with details about the size of the models and the random seeds used to initialise the network weights, are given in Table 8. As was the case for the River Murray salinity case study, the models had similar predictive performance, but a large variation in size and number of weights (model SAT3 has 91 fewer parameters than model SAT1). The majority of metrics presented in Table 8 suggest that model SAT1 with 14 hidden nodes is the most predictively valid; however, as can be seen, there were relatively large predictive errors associated with all three models (RMSEs $\geq$ 0.29 in comparison to the mean TwTurbidity value of 0.3 and MARE values $\geq$ 70%), which is consistent with the results obtained by Wu et al. (2013).

Scatter plots of the observed versus predicted TwTurbidity values obtained by applying the three models to the validation data are displayed in

Table 8: Surface water turbidity predictive validation results. Best results are highlighted in bold text.

| | SAT1 | SAT2 | SAT3 | Observed |
|---|---|---|---|---|
| Hidden nodes | 14 | 12 | 1 | - |
| # of weights | 99 | 85 | 8 | - |
| Random seed | 4 | 5 | 4 | - |
| RMSE | **0.29** | 0.31 | 0.32 | - |
| AIC | 148.6 | 123.3 | **-29.4** | - |
| MARE | 100.0 | 72.8 | **72.0** | - |
| RSqr | **0.81** | 0.80 | 0.78 | - |
| CE | **0.81** | 0.78 | 0.76 | - |
| Mean | **0.42** | 0.36 | 0.38 | 0.44 |
| SD | **0.66** | 0.65 | 0.64 | 0.67 |
| Skewness | **2.0** | 3.0 | 3.3 | 2.3 |
| Kurtosis | **7.3** | 12.8 | 13.6 | 7.4 |

Fig. 10. Here, it can be seen that while the SAT1 model predictions have the least scatter about the 1:1 line (perfect predictions), this model also displays a tendency to under-predict TwTurbidity at smaller magnitudes, with a number of unrealistic negative turbidities predicted. Likewise, model SAT2 has also predicted some negative turbidities (although fewer than model SAT1), showing a slight tendency to under-predict TwTurbidity at smaller magnitudes. Model SAT3, on the other hand, has the greatest scatter about the 1:1 line, but is the only model that predicted all TwTurbidity values to be greater than zero.

The same scatter plots obtained by applying models SAT1, SAT2 and SAT3 to the training data (indicating replicative validity) are shown in Fig. 11, where it can be seen that models SAT1 and SAT2 give an almost perfect fit to the observed TwTurbidity values, while for model SAT3, there

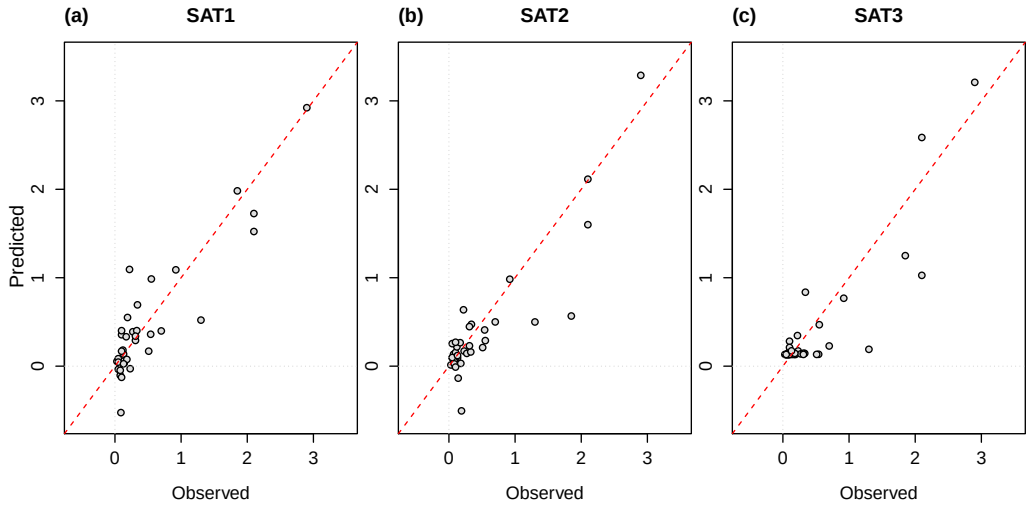


Figure 10: Scatter plots of observed versus predicted TwTurbidity (NTU) obtained by applying models (a) SAT1, (b) SAT2 and (3) SAT3 to the validation data. The red dashed line denotes a perfect fit.

is some discrepancy between the observations and the predictions. This may be due to the larger models overfitting the training data; however, AIC values of -246, -275 and -248 obtained for models SAT1, SAT2 and SAT3, respectively, suggest that the extra complexity of model SAT2 over that of model SAT3 is warranted given the improved fit to the training data.

For this case study, there is no time component (or spatial correlation) associated with the data; therefore, it is unnecessary to assess the autocorrelation structure of the residuals. However, it is still important to consider the distributions of the model residuals and whether the residuals have constant variance. Histograms of the residuals resulting from the three models when applied to the training data are shown in Fig. 12. For models SAT1 and SAT2, the residuals appear to be approximately normally distributed with

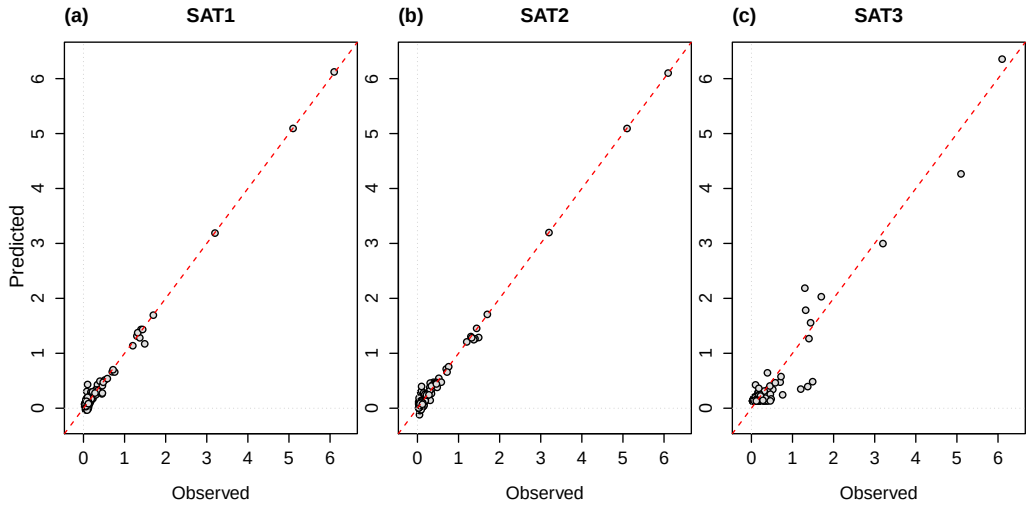


Figure 11: Scatter plots of observed versus predicted TwTurbidity (NTU) obtained by applying models (a) SAT1, (b) SAT2 and (3) SAT3 to the training data. The red dashed line denotes a perfect fit.

a mean of zero. While the residuals resulting from model SAT3 also have a mean of zero, their distribution appears to be somewhat skewed, which, given that the TwTurbidity data are also significantly skewed (see Fig. 4), is unsurprising. However, this result suggests that Box-Cox transformed target data may produce more efficient parameter estimates. From the plots of standardised residuals versus predictions shown in Fig 13, where the predicted TwTurbidity data are given in logarithmic scale for easier interpretation, it can be seen that there are no obvious patterns in the residuals from any of the models. There do, however, appear to be a number of possible outliers.

In terms of the structural validity of the models, for this case study, it is difficult to determine the "true" magnitudes of input RI or even the order of input importance for predicting TwTurbidity. This is because the inputs

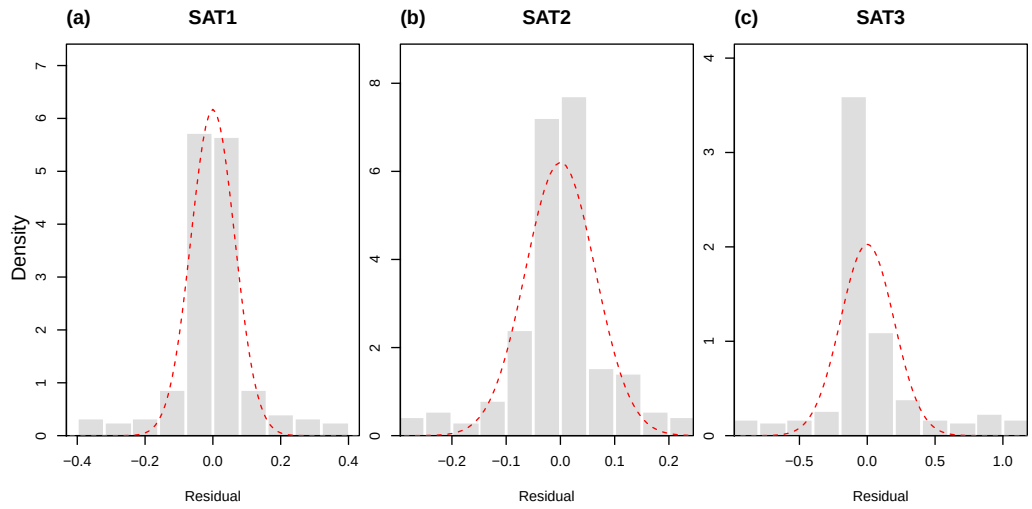


Figure 12: Histograms of model residuals obtained by applying models (a) SAT1, (b) SAT2 and (3) SAT3 to the training data. The red dashed line denotes the Normal distribution.

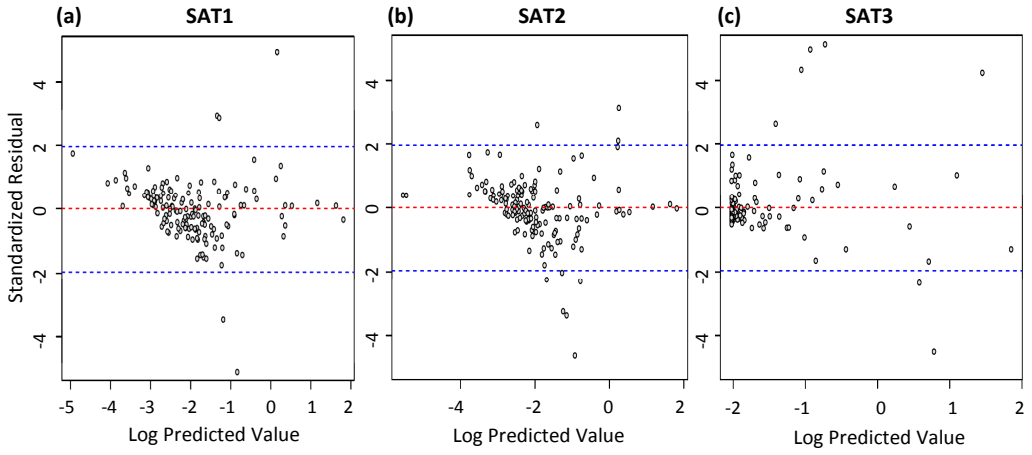


Figure 13: Standardised residuals obtained using models (a) SAT1, (b) SAT2 and (3) SAT3 versus log predicted TwTurbidity. The red dashed line shows the zero line, while blue dashed lines denote the 95% limits of the standard normal distribution.

are significantly more important in combination than they are individually. For example, when considering RwTurbidity or alum dose alone, these inputs seem to be of relatively low importance for predicting TwTurbidity (accounting for approximately 19% and 7% of the variance in TwTurbidity, respectively); however, in combination, the contribution of these inputs in predicting the output is far greater (accounting for approximately 76% of the variance in TwTurbidity). In this case, a SA based structural validation approach is more useful for assessing the plausibility of the modelled relationships than input RI values, as the output profiles provided by such methods allow for the behaviour of the models to be examined when one input is varied and the others remain fixed. The Profile SA method is more informative than the PaD approach in this case, as the way in which the model responds to changes in a given input can be viewed in relation to different levels of the fixed inputs, which is important when the influence of an input depends on the value of another input. For this case study, where the associated model inputs are relatively uncorrelated with one another (see Table 3), the Profile method is considered to be suitable for assessing input sensitivities, as it is unlikely that infeasible combinations of the inputs would be used in their calculation.

When assessing the results of the Profile method, a plausible model would be one that produces outputs roughly within the range of the observed data (TwTurbidity between $\approx 0 - 6$ NTU) and displays reasonably monotonic relationships between the variable of interest and TwTurbidity when all other explanatory variables are fixed. In addition, it would generally be expected that as the turbidity of the raw water (RwTurbidity) increases, the resulting

turbidity of the treated water (TwTurbidity) would also increase for fixed values of all other explanatory variables. Likewise, the higher the UVA-254 of the raw water (RwUvAbs254), the higher the TwTurbidity would be expected to be, since UVA-254 is used as a surrogate for dissolved natural organic matter (NOM) concentration, which negatively impacts turbidity removal (alum reacts preferentially with dissolved NOM) (White et al., 1997). Colour is also an indicator of NOM and, as such, a similar relationship might be expected. However, in the study by van Leeuwen et al. (1999), colour was found not to be significant for predicting optimum alum doses for the SAT dataset. Consequently, it could be expected that this variable would have little influence on the resulting TwTurbidity for the SAT dataset. Similarly, pH was found to be unimportant for predicting optimum alum doses in the study carried out by van Leeuwen et al. (1999). While optimum doses of alum do depend on the pH of the water, with lower doses possible when pH is maintained in the neutral range between 6-8 (Crittenden et al., 2012), the raw water pH (RwPh) range of the SAT dataset is 7.48-8.63, which when lowered through the addition of alum should generally be within the neutral range. Therefore, it would be expected that for the range of RwPh in the SAT dataset, this variable would have little influence on the resulting TwTurbidity. On the other hand, alum dose is certainly important for predicting treated water turbidities, with generally decreasing TwTurbidity expected for increasing alum dose.

Shown in Figs. 14-16 are the input sensitivity profiles for models SAT1, SAT2 and SAT3, respectively, obtained using the Profile method. As can be seen when comparing these figures, only model SAT3 could be considered

physically plausible, with both SAT1 and SAT2 producing negative values of TwTurbidity for certain input values (as was also observed in Fig. 10). In addition, the response of model SAT1 to variation in several of the key inputs is contradictory to the expected behaviour of the model (e.g. predicted TwTurbidity reduces with increasing RwTurbidity and increases with increasing alum dose). Model SAT2, on the other hand, displays input-output relationships that are more complicated than would be expected when all other variables are fixed (e.g. the non-monotonic relationships between RwUvAbs254 and RwPh and TwTurbidity). Model SAT3 appears to be the most structurally valid, displaying input-output relationships in line with physical understanding. In agreement with the findings of van Leeuwen et al. (1999), model SAT3 indicates that inputs RwPh and RwColour are relatively unimportant for predicting TwTurbidity for the SAT dataset, as indicated by the limited scale of the y-axis in Figs. 16 (b) and (c). Furthermore, for the remaining inputs, the resulting predicted TwTurbidity ranged between approximately 0-6.5, which is a plausible range for this variable given the ranges of the input variables considered. The threshold behaviour observed for model SAT3 when increasing alum dose and fixing all other inputs at their maximum values is as would be expected, as it was observed by White et al. (1997) that a threshold alum dose is often required before a sharp reduction in turbidity is achieved.

Based on these case study results, model SAT3 is considered to be the most structurally valid, while the predictive and replicative validity of models SAT1 and SAT2 appear to be the best. The results suggest that model SAT3, with one hidden node, is perhaps too simple to appropriately cap-

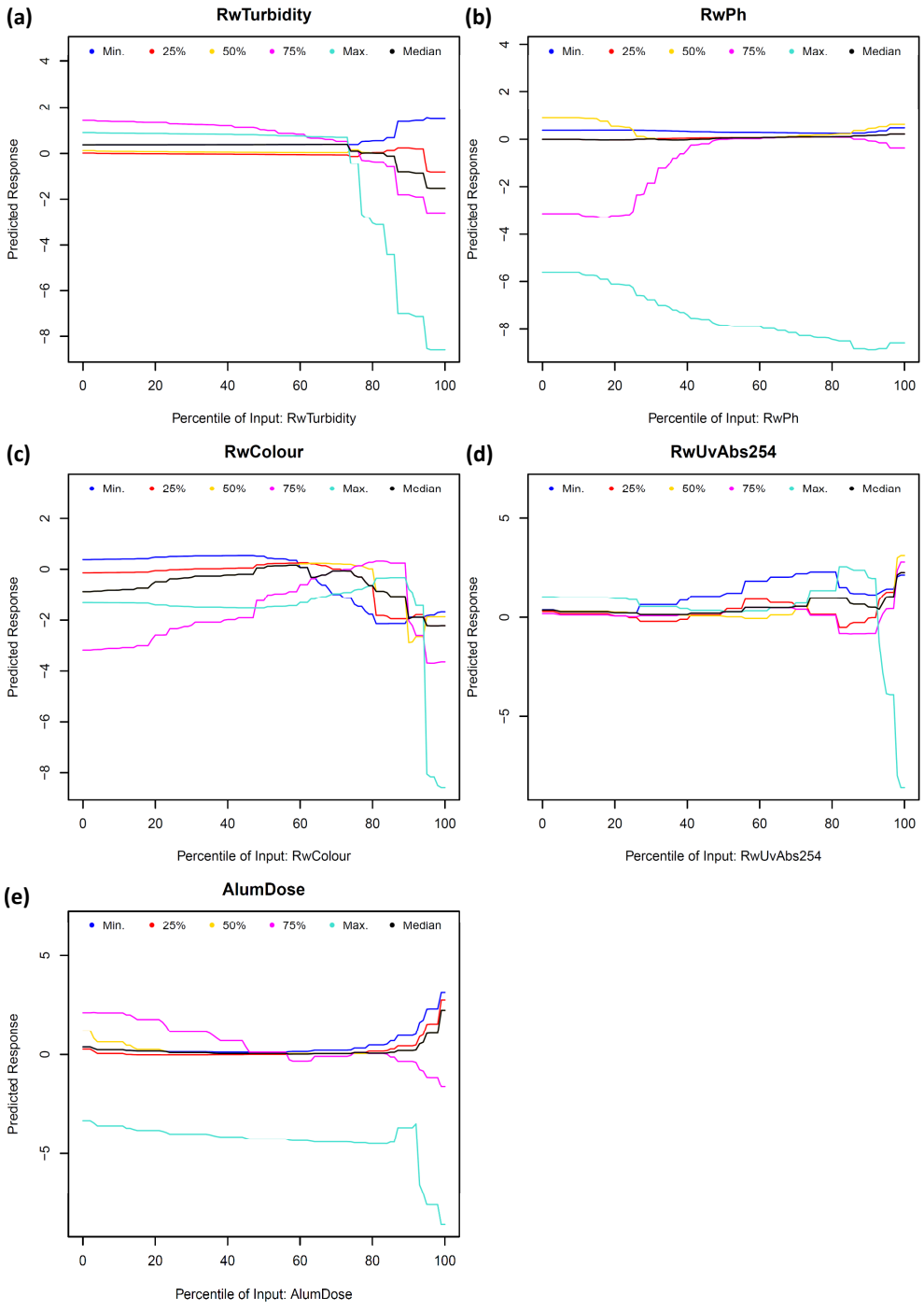


Figure 14: Input sensitivity profiles obtained using the Profile method applied to model SAT1.

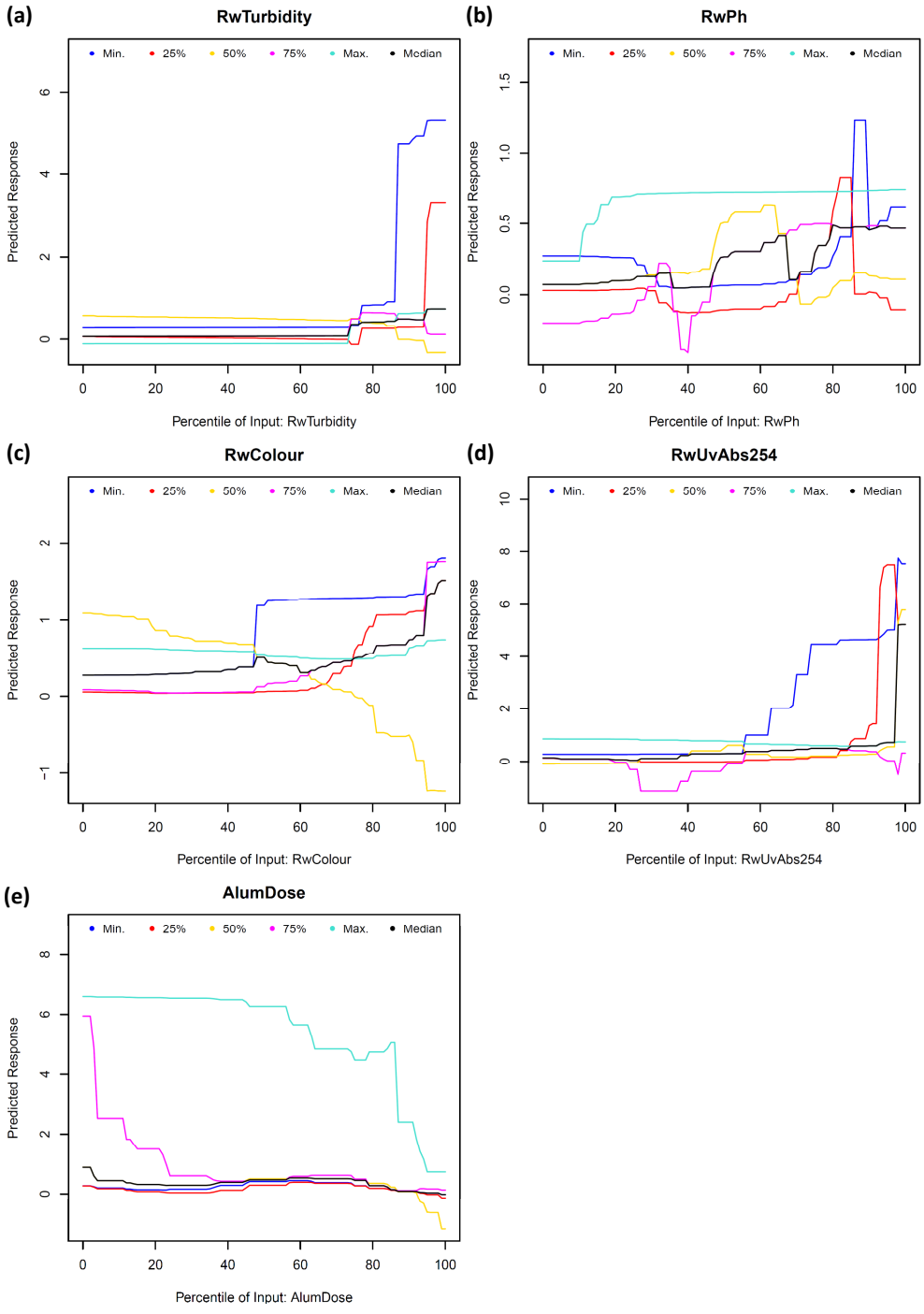


Figure 15: Input sensitivity profiles obtained using the Profile method applied to model SAT2.

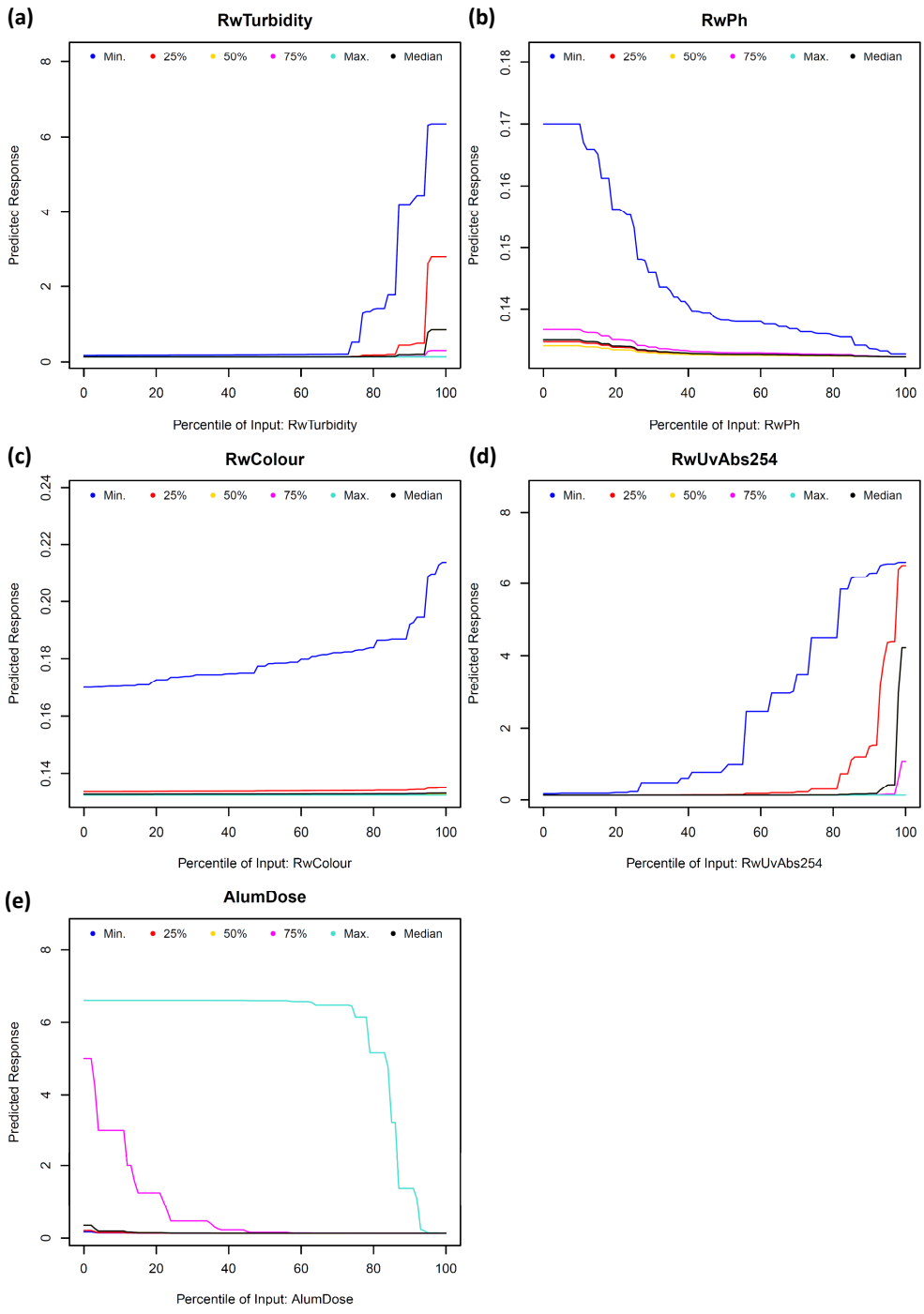


Figure 16: Input sensitivity profiles obtained using the Profile method applied to model SAT3.

ture the relationship in the data, while the much larger models SAT1 and SAT2, with 14 and 12 hidden nodes, respectively, are too complex resulting in overly complicated and unrealistic modelled relationships. In this study, little attention was paid to model training and it is possible that a model with slightly more complexity than SAT3 (e.g. a 2 hidden node ANN) could be developed, taking care to optimally train the model (e.g. applying early stopping and perhaps a different training algorithm), that is predictively, replicatively and structurally valid.

## 6. Summary and Conclusions

Validation is a critical step in any model development process and ANN models are no exception. Although validation is generally performed during the development of ANN models, this is mainly restricted to predictive validation, as part of which the predictive performance of a trained (calibrated) ANN is assessed on an independent validation set. While this is an important aspect of the model validation process, residual analysis (replicative validation) and an assessment of how plausible the input-output relationship represented by the calibrated model is (structural validation) are considered important components of validation in other areas of environmental modelling, but are generally ignored in the validation of ANNs. In order to enable these additional aspects of validation to be incorporated in the development of ANN models, a validation framework for ANNs and an R-package that enables this framework to be implemented in a user-friendly and consistent fashion are introduced and tested in this paper. Adoption of the framework not only improves the quality and credibility of the resulting ANNs, but also

makes it easier to compare the results from different studies in an objective fashion.

Results of the application of the framework and **validann** R-package to two different environmental modelling case studies highlight the importance of performing replicative and structural validation in addition to predictive validation. In each case, the results revealed that ANN models producing the best fit to the data do not necessarily result in either plausible models or models which best capture the underlying relationship in the training data. By considering the predictive, replicative and structural validity of the ANN models developed, areas of model deficiency were identified, which would not have been evident if predictive validation alone had been performed. Thus, it was seen that application of the ANN validation framework may provide important insights into how an ANN model may be improved in order to improve the overall validity of the model. The **validann** R-package has been developed such that the proposed framework can be implemented in a user-friendly and consistent fashion, while the methods provided have been designed to be flexible and adaptable, such that validation of ANNs developed using different software or tools is also supported. It is hoped that this will encourage the maximum uptake and application of the proposed validation framework, such that the comprehensive validation of ANNs in environmental modelling becomes commonplace.

## References

Abdul-Wahab, S.A., Al-Alawi, S.M., 2002. Assessment and prediction of tropospheric ozone concentration levels using artificial neural net-

works. Environmental Modelling & Software 17, 219–228. doi:10.1016/S1364-8152(01)00077-9.

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. Progress in Physical Geography 36, 480–513. doi:10.1177/0309133312444943.

Abrahart, R.J., See, L., Kneale, P.E., 2001. Investigating the role of saliency analysis with a neural network rainfall-runoff model. Computers & Geosciences 27, 921–928. doi:10.1016/S0098-3004(00)00131-X.

Abrahart, R.J., See, L.M., Dawson, C.W., 2008. Neural network hydroinformatics: maintaining scientific rigour. Springer-Verlag, Berlin Heidelberg. Chapter 3. Water Science and Technology Library, pp. 33–47.

Andrews, F.T., Croke, B.F.W., Jakeman, A.J., 2011. An open software environment for hydrological model assessment and development. Environmental Modelling & Software 26, 1171–1185. doi:10.1016/j.envsoft.2011.04.006.

Bates, D.M., Watts, D.G., 1988. Nonlinear Regression Analysis and Its Applications. John Wiley & Sons, Inc.

Beck, H.E., van Dijk, A.I.J.M., Miralles, D.G., de Jeu, R.A.M., Bruijnzeel, L.A., McVicar, T.R., Schellekens, J., 2013. Global patterns in base flow index and recession based on streamflow observations from 3394 catchments.

Water Resources Research 49, 7843–7863. doi:10.1002/2013wr013918. (Sampurno).

Beck, M., 2015. NeuralNetTools: Visualization and analysis tools for neural networks. URL: http://CRAN.R-project.org/package=NeuralNetTools.

Beck, M.W., Wilson, B.N., Vondracek, B., Hatch, L.K., 2014. Application of neural networks to quantify the utility of indices of biotic integrity for biological monitoring. Ecological Indicators 45, 195–208. doi:10.1016/j.ecolind.2014.04.002.

Bennett, N.D., Croke, B.F.W., Guariso, G., Guillaume, J.H.A., Hamilton, S.H., Jakeman, A.J., Marsili-Libelli, S., Newham, L.T.H., Norton, J.P., Perrin, C., Pierce, S.A., Robson, B., Seppelt, R., Voinov, A.A., Fath, B.D., Andreassian, V., 2013. Characterising performance of environmental models. Environmental Modelling & Software 40, 1–20. doi:10.1016/j.envsoft.2012.09.011.

Beven, K.J., Freer, J., 2001. Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems using the GLUE methodology. Journal of Hydrology 249, 11–29. doi:10.1016/S0022-1694(01)00421-8.

Biondi, D., Freni, G., Iacobellis, V., Mascaro, G., Montanari, A., 2012. Validation of hydrological models: Conceptual basis, methodological approaches and a proposal for a code of practice. Physics and Chemistry of the Earth, Parts A/B/C 4244, 70–76. doi:10.1016/j.pce.2011.07.037.

1167 Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Oxford Uni-
1168 versity Press, Oxford.

1169 Bowden, G.J., Maier, H.R., Dandy, G.C., 2002. Optimal division of data for
1170 neural network models in water resources applications. Water Resources
1171 Research 38, 1–11. doi:10.1029/2001wr000266.

1172 Bowden, G.J., Maier, H.R., Dandy, G.C., 2005. Input determination for
1173 neural network models in water resources applications. Part 2. Case study:
1174 forecasting salinity in a river. Journal of Hydrology 301, 93–107. doi:10.
1175 1016/j.jhydrol.2004.06.020.

1176 Box, G.E.P., Cox, D.R., 1964. An analysis of transformations. Journal of
1177 the Royal Statistical Society. Series B (Methodological) 26, 211–252.

1178 Box, G.E.P., Jenkins, G., 1976. Time Series Analysis: Forecasting and Con-
1179 trol. Holden-Day.

1180 Brosse, S., Guegan, J.F., Tourenq, J.N., Lek, S., 1999. The use of artificial
1181 neural networks to assess fish abundance and spatial occupancy in the
1182 littoral zone of a mesotrophic lake. Ecological Modelling 120, 299–311.
1183 doi:10.1016/S0304-3800(99)00110-6.

1184 Chang, J.C., Hanna, S.R., 2004. Air quality model performance evalua-
1185 tion. Meteorology and Atmospheric Physics 87, 167–196. doi:10.1007/
1186 s00703-003-0070-7.

1187 Chapra, S.C., 1997. Surface Water Quality Modeling. McGraw-Hill.

1188 Clarke, R.T., 1973. A review of some mathematical models used in hydrology,
1189 with observations on their calibration and use. Journal of Hydrology 19,
1190 1–20. doi:10.1016/0022-1694(73)90089-9.

1191 Coad, P., Cathers, B., Ball, J.E., Kadluczka, R., 2014. Proactive management
1192 of estuarine algal blooms using an automated monitoring buoy coupled
1193 with an artificial neural network. Environmental Modelling & Software
1194 61, 393–409. doi:10.1016/j.envsoft.2014.07.011.

1195 Craven, M.W., Shavlik, J.W., 1997. Using neural networks for data min-
1196 ing. Future Generation Computer Systems 13, 211–229. doi:10.1016/
1197 S0167-739X(97)00022-8.

1198 Crittenden, J.C., Trussell, R.R., Hand, D.W., Howe, K.J., Tchobanoglous,
1199 G., 2012. MWH's Water Treatment Principles and Design. 3rd ed., John
1200 Wiley & Sons, Hoboken, NJ, USA.

1201 Crout, N., Kokkonen, T., Jakeman, A.J., Norton, J.P., Newham, L.T.H., An-
1202 derson, R., Assaf, H., Croke, B.F.W., Gaber, N., Gibbons, J., Holzworth,
1203 D., Mysiak, J., Reichl, J., Seppelt, R., Wagener, T., Whitfield, P., 2008.
1204 Good modelling practice. Elsevier, Amsterdam. Chapter 2. Developments
1205 in Integrated Environmental Assessment, pp. 15–31.

1206 Dawson, C.W., Abrahart, R.J., See, L.M., 2007. HydroTest: A web-based
1207 toolbox of evaluation metrics for the standardised assessment of hydro-
1208 logical forecasts. Environmental Modelling & Software 22, 1034–1052.
1209 doi:10.1016/j.envsoft.2006.06.008.

Dawson, C.W., Abrahart, R.J., See, L.M., 2010. HydroTest: Further development of a web resource for the standardised assessment of hydrological models. Environmental Modelling & Software 25, 1481–1482. doi:10.1016/j.envsoft.2009.01.001.

Dawson, C.W., Mount, N.J., Abrahart, R.J., Louis, J., 2014. Sensitivity analysis for comparison, validation and physical-legitimacy of neural network-based hydrological models. Journal of Hydroinformatics 16, 1–18. doi:10.2166/hydro.2013.222.

Dawson, C.W., Mount, N.J., Abrahart, R.J., Shamseldin, A.Y., 2012. Ideal point error for model assessment in data-driven river flow forecasting. Hydrology and Earth System Sciences 16, 3049–3060. doi:10.5194/hess-16-3049-2012.

Dawson, C.W., Wilby, R.L., 2001. Hydrological modelling using artificial neural networks. Progress in Physical Geography 25, 80–108. doi:10.1177/030913330102500104.

Dimopoulos, I., Chronopoulos, J., Chronopoulou-Sereli, A., Lek, S., 1999. Neural network models to study relationships between lead concentration in grasses and permanent urban descriptors in Athens city (Greece). Ecological Modelling 120, 157–165. doi:10.1016/S0304-3800(99)00099-X.

Dimopoulos, Y., Bourret, P., Lek, S., 1995. Use of some sensitivity criteria for choosing networks with good generalization ability. Neural Processing Letters 2, 1–4. doi:10.1007/bf02309007.

Draper, N.R., Smith, H., 1998. Applied Regression Analysis. Wiley Series in Probability and Statistics. Texts and References Section, Wiley-Interscience, New York. Accession Number: 26118; Language: English.

Evin, G., Kavetski, D., Thyer, M., Kuczera, G., 2013. Pitfalls and improvements in the joint inference of heteroscedasticity and autocorrelation in hydrological model calibration. Water Resources Research 49, 4518–4524. doi:10.1002/wrcr.20284.

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. Journal of Hydrology 367, 165–176. doi:10.1016/j.jhydrol.2008.10.019.

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. Environmental Modelling & Software 62, 33–51. doi:10.1016/j.envsoft.2014.08.015.

Garson, G.D., 1991. Interpreting neural-network connection weights. AI Expert 6, 46–51.

Gass, S.I., 1983. Decision-aiding models: validation, assessment, and related issues for policy analysis. Operations Research 31, 603–631. doi:10.1287/opre.31.4.603.

Geary, R.C., 1970. Relative efficiency of count of sign changes for assessing residual autoregression in least squares regression. Biometrika 57, 123–127. doi:10.2307/2334942.

Gevrey, M., Dimopoulos, I., Lek, S., 2003. Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecological Modelling 160, 249–264. doi:10.1016/S0304-3800(02)00257-0.

Giam, X., Olden, J.D., 2015. A new R2-based metric to shed greater insight on variable importance in artificial neural networks. Ecological Modelling 313, 307–313. doi:10.1016/j.ecolmodel.2015.06.034.

Guo, D., Westra, S., Maier, H.R., 2016. An R package for modelling actual, potential and reference evapotranspiration. Environmental Modelling & Software 78, 216–224. doi:10.1016/j.envsoft.2015.12.019.

Hashem, S., 1992. Sensitivity analysis for feedforward artificial neural networks with differentiable activation functions, in: IJCNN., International Joint Conference on Neural Networks, 1992, IEEE. pp. 419–424. doi:10.1109/ijcnn.1992.287175.

Heiberger, R.M., Holland, B., 2004. Statistical Analysis and Data Display: An Intermediate Course with Examples in S-Plus, R, and SAS. Springer-Verlag, New York.

Jain, A., Kumar, S., 2009. Dissection of trained neural network hydrologic models for knowledge extraction. Water Resources Research 45, W07420. doi:10.1029/2008wr007194.

Jain, A., Sudheer, K.P., Srinivasulu, S., 2004. Identification of physical processes inherent in artificial neural network rainfall runoff models. Hydrological Processes 18, 571–581. doi:10.1002/hyp.5502.

Jain, S.K., Nayak, P.C., Sudheer, K.P., 2008. Models for estimating evapotranspiration using artificial neural networks, and their physical interpretation. Hydrological Processes 22, 2225–2234. doi:10.1002/hyp.6819.

Jakeman, A.J., Letcher, R.A., Norton, J.P., 2006. Ten iterative steps in development and evaluation of environmental models. Environmental Modelling & Software 21, 602–614. doi:10.1016/j.envsoft.2006.01.004.

Joy, M.K., Death, R.G., 2004. Predictive modelling and spatial mapping of freshwater fish and decapod assemblages using GIS and neural networks. Freshwater Biology 49, 1036–1052. doi:10.1111/j.1365-2427.2004.01248.x.

Kemp, S.J., Zaradic, P., Hansen, F., 2007. An approach for determining relative input parameter importance and significance in artificial neural networks. Ecological Modelling 204, 326–334. doi:10.1016/j.ecolmodel.2007.01.009.

Kingston, G., Maier, H., Lambert, M., 2010. Bayesian Artificial Neural Networks: with Applications in Water Resources Engineering. VDM Verlag.

Kingston, G.B., Lambert, M.F., Maier, H.R., 2005a. Bayesian training of artificial neural networks used for water resources modeling. Water Resources Research 41, W12409. doi:10.1029/2005WR004152.

Kingston, G.B., Maier, H.R., Lambert, M.F., 2005b. Calibration and validation of neural networks to ensure physically plausible hydrological modeling. Journal of Hydrology 314, 158–176. doi:10.1016/j.jhydrol.2005.03.013.

1301 Kingston, G.B., Maier, H.R., Lambert, M.F., 2006a. Forecasting cyanobacte-
1302 ria with Bayesian and deterministic artificial neural networks, in: IJCNN
1303 '06. International Joint Conference on Neural Networks, 2006., IEEE. pp.
1304 4870–4877. doi:10.1109/ijcnn.2006.247166.

1305 Kingston, G.B., Maier, H.R., Lambert, M.F., 2006b. A probabilistic method
1306 for assisting knowledge extraction from artificial neural networks used for
1307 hydrological prediction. Mathematical and Computer Modelling 44, 499–
1308 512. doi:10.1016/j.mcm.2006.01.008.

1309 Kingston, G.B., Maier, H.R., Lambert, M.F., 2008. Bayesian model selection
1310 applied to artificial neural networks used for water resources modeling.
1311 Water Resources Research 44, W04419. doi:10.1029/2007wr006155.

1312 Kuczera, G., 1983. Improved parameter inference in catchment models: 1.
1313 evaluating parameter uncertainty. Water Resources Research 19, 1151–
1314 1162. doi:10.1029/WR019i005p01151.

1315 Kumar, B., 2012. Neural network prediction of bed material load transport.
1316 Hydrological Sciences Journal/Journal des Sciences Hydrologiques 57, 956–
1317 966. doi:10.1080/02626667.2012.687108.

1318 Kumar, B., 2014. Flow prediction in vegetative channel using hybrid arti-
1319 ficial neural network approach. Journal of Hydroinformatics 16, 839–849.
1320 doi:10.2166/hydro.2013.255.

1321 Laffaille, P., Lasne, E., Baisez, A., 2009. Effects of improving longitudinal
1322 connectivity on colonisation and distribution of European eel in the Loire

catchment, France. Ecology of Freshwater Fish 18, 610–619. doi:10.1111/j.1600-0633.2009.00378.x.

Langella, G., Basile, A., Bonfante, A., Terribile, F., 2010. High-resolution space-time rainfall analysis using integrated ANN inference systems. Journal of Hydrology 387, 328–342. doi:10.1016/j.jhydrol.2010.04.027.

van Leeuwen, J., Chow, C.W.K., Bursill, D., Drikas, M., 1999. Empirical mathematical models and artificial neural networks for the determination of alum doses for treatment of southern Australian surface waters. Aqua 48, 115–127. doi:10.1046/j.1365-2087.1999.00135.x.

Lek, S., Belaud, A., Dimopoulos, I., Lauga, J., Moreau, J., 1995. Improved estimation, using neural networks, of the food consumption of fish populations. Marine and Freshwater Research 46, 1229–1236. doi:10.1071/MF9951229.

Lek, S., Delacoste, M., Baran, P., Dimopoulos, I., Lauga, J., Aulagnier, S., 1996. Application of neural networks to modelling nonlinear relationships in ecology. Ecological Modelling 90, 39–52. doi:10.1016/0304-3800(95)00142-5.

Li, X., Zecchin, A.C., Maier, H.R., 2014. Selection of smoothing parameter estimators for general regression neural networks applications to hydrological and water resources modelling. Environmental Modelling & Software 59, 162–186. doi:10.1016/j.envsoft.2014.05.010.

Liong, S., Lim, W., Paudyal, G., 2000. River stage forecasting in Bangladesh:

neural network approach. Journal of Computing in Civil Engineering 14, 1–8. doi:10.1061/(ASCE)0887-3801(2000)14:1(1).

Maier, H.R., Dandy, G.C., 1996. The use of artificial neural networks for the prediction of water quality parameters. Water Resources Research 32, 1013–1022. doi:10.1029/95WR03529.

Maier, H.R., Dandy, G.C., 1997. Determining inputs for neural network models of multivariate time series. Computer-Aided Civil and Infrastructure Engineering 12, 353–368. doi:10.1111/0885-9507.00069.

Maier, H.R., Dandy, G.C., 2000. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. Environmental Modelling & Software 15, 101–124. doi:10.1016/S1364-8152(99)00007-9.

Maier, H.R., Dandy, G.C., Burch, M.D., 1998. Use of artificial neural networks for modelling cyanobacteria Anabaena spp. in the River Murray, South Australia. Ecological Modelling 105, 257–272. doi:10.1016/S0304-3800(97)00161-0.

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. Environmental Modelling & Software 25, 891–909. doi:10.1016/j.envsoft.2010.02.003.

Maier, H.R., Morgan, N., Chow, C.W.K., 2004. Use of artificial neural networks for predicting optimal alum doses and treated water quality param-

eters. Environmental Modelling & Software 19, 485–494. doi:10.1016/S1364-8152(03)00163-4.

Matott, L.S., Babendreier, J.E., Purucker, S.T., 2009. Evaluating uncertainty in integrated environmental models: A review of concepts and tools. Water Resources Research 45, W06421. doi:10.1029/2008wr007301.

May, R.J., Maier, H.R., Dandy, G.C., 2010. Data splitting for artificial neural networks using SOM-based stratified sampling. Neural Networks 23, 283–294. doi:10.1016/j.neunet.2009.11.009.

McCuen, R.H., 1973. The role of sensitivity analysis in hydrologic modeling. Journal of Hydrology 18, 37–53. doi:10.1016/0022-1694(73)90024-3.

Mi, X., Zou, Y., Wei, W., Ma, K., 2005. Testing the generalization of artificial neural networks with cross-validation and independent-validation in modelling rice tillering dynamics. Ecological Modelling 181, 493–508. doi:10.1016/j.ecolmodel.2004.06.035.

Mount, N.J., Dawson, C.W., Abrahart, R.J., 2013. Legitimising data-driven models: exemplification of a new data-driven mechanistic modelling framework. Hydrology and Earth System Sciences 17, 2827–2843. doi:10.5194/hess-17-2827-2013.

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. Hydrological Sciences Journal 61, 1192–1208. doi:10.1080/02626667.2016.1159683.

de Oña, J., Garrido, C., 2014. Extracting the contribution of independent variables in neural network models: a new approach to handle instability. Neural Computing & Applications 25, 859–869. doi:10.1007/s00521-014-1573-5.

Olaya-Marín, E.J., Martínez-Capel, F., Soares Costa, R.M., Alcaraz-Hernández, J.D., 2012. Modelling native fish richness to evaluate the effects of hydromorphological changes and river restoration (Júcar River Basin, Spain). Science of The Total Environment 440, 95–105. doi:10.1016/j.scitotenv.2012.07.093.

Olden, J.D., Jackson, D.A., 2002. Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. Ecological Modelling 154, 135–150. doi:10.1016/S0304-3800(02)00064-9.

Olden, J.D., Joy, M.K., Death, R.G., 2004. An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. Ecological Modelling 178, 389–397. doi:10.1016/j.ecolmodel.2004.03.013.

Oreskes, N., Shrader-Frechette, K., Belitz, K., 1994. Verification, validation, and confirmation of numerical models in the earth sciences. Science 263, 641–646. doi:10.1126/science.263.5147.641.

Özesmi, S.L., Özesmi, U., 1999. An artificial neural network approach to spatial habitat modelling with interspecific interaction. Ecological Modelling 116, 15–31. doi:10.1016/S0304-3800(98)00149-5.

Park, Y.S., Chung, Y.J., 2006. Hazard rating of pine trees from a forest insect pest using artificial neural networks. Forest Ecology and Management 222, 222–233. doi:10.1016/j.foreco.2005.10.009.

Park, Y.S., Rabinovich, J., Lek, S., 2007. Sensitivity analysis and stability patterns of two-species pest models using artificial neural networks. Ecological Modelling 204, 427–438. doi:10.1016/j.ecolmodel.2007.01.021.

Phukoetphim, P., Shamseldin, A., Melville, B., 2014. Knowledge extraction from artificial neural networks for rainfall-runoff model combination systems. Journal of Hydrologic Engineering 19, 1422–1429. doi:10.1061/(ASCE)HE.1943-5584.0000941.

Pianosi, F., Sarrazin, F., Wagener, T., 2015. A Matlab toolbox for Global Sensitivity Analysis. Environmental Modelling & Software 70, 80–85. doi:10.1016/j.envsoft.2015.04.009.

Power, M., 1993. The predictive validation of ecological and environmental models. Ecological Modelling 68, 33–50. doi:10.1016/0304-3800(93)90106-3.

R Core Team, 2015. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

Rykiel Jr, E.J., 1996. Testing ecological models: the meaning of validation. Ecological Modelling 90, 229–244. doi:10.1016/0304-3800(95)00152-2.

Sarle, W.S., 2000. How to measure the importance of inputs? URL: ftp://ftp.sas.com/pub/neural/importance.html.

1436 Schoups, G., Vrugt, J.A., 2010. A formal likelihood function for param-
1437 eter and predictive inference of hydrologic models with correlated, het-
1438 eroscedastic, and non-Gaussian errors. Water Resources Research 46,
1439 W10531. doi:10.1029/2009wr008933.

1440 See, L.M., Jain, A., Dawson, C.W., Abrahart, R.J., 2008. Visualisation of
1441 Hidden Neuron Behaviour in a Neural Network Rainfall-Runoff Model.
1442 Springer, Berlin Heidelberg. Volume 68 of *Water Science and Technology*
1443 *Library*. Chapter 7. pp. 87–99.

1444 Shahin, M.A., Maier, H.R., Jaksa, M.B., 2005. Investigation into the ro-
1445 bustness of artificial neural networks for a case study in civil engineering,
1446 in: Argent, A.Z., M., R. (Eds.), MODSIM 2005 International Congress on
1447 Modelling and Simulation, Modelling and Simulation Society of Australia
1448 and New Zealand. pp. 79–83.

1449 Shu, C., Ouarda, T.B.M.J., 2007. Flood frequency analysis at ungauged
1450 sites using artificial neural networks in canonical correlation analysis phys-
1451 iographic space. Water Resources Research 43, W07438. doi:10.1029/
1452 2006wr005142.

1453 Snee, R.D., 1977. Validation of regression models: methods and examples.
1454 Technometrics 19, 415–428. doi:10.2307/1267881.

1455 Sorooshian, S., Dracup, J.A., 1980. Stochastic parameter estimation proce-
1456 dures for hydrologic rainfall-runoff models: Correlated and heteroscedas-
1457 tic error cases. Water Resources Research 16, 430–442. doi:10.1029/
1458 WR016i002p00430.

Sreekanth, J., Datta, B., 2010. Multi-objective management of saltwater intrusion in coastal aquifers using genetic programming and modular neural network based surrogate models. Journal of Hydrology 393, 245–256. doi:10.1016/j.jhydrol.2010.08.023.

Stokes, C.S., Simpson, A.R., Maier, H.R., 2015. A computational software tool for the minimization of costs and greenhouse gas emissions associated with water distribution systems. Environmental Modelling & Software 69, 452–467. doi:10.1016/j.envsoft.2014.11.004.

Sudheer, K.P., 2005. Knowledge extraction from trained neural network river flow models. Journal of Hydrologic Engineering 10, 264–269. doi:10.1061/(ASCE)1084-0699(2005)10:4(264).

Sudheer, K.P., Jain, A., 2004. Explaining the internal behaviour of artificial neural network river flow models. Hydrological Processes 18, 833–844. doi:10.1002/hyp.5517.

Sun, A.Y., 2013. Predicting groundwater level changes using GRACE data. Water Resources Research 49, 5900–5912. doi:10.1002/wrcr.20421.

Thomann, R.V., Mueller, J.A., 1987. Principles of Surface Water Quality Modeling and Control. Harper & Row, New York.

Thyer, M., Renard, B., Kavetski, D., Kuczera, G., Franks, S.W., Srikanthan, S., 2009. Critical evaluation of parameter consistency and predictive uncertainty in hydrological modeling: A case study using Bayesian total error analysis. Water Resources Research 45, W00B14. doi:10.1029/2008wr006825.

Tison, J., Park, Y.S., Coste, M., Wasson, J.G., Rimet, F., Ector, L., Delmas, F., 2007. Predicting diatom reference communities at the French hydrosystem scale: A first step towards the definition of the good ecological status. Ecological Modelling 203, 99–108. doi:10.1016/j.ecolmodel.2006.02.047.

Vasilakos, C., Kalabokidis, K., Hatzopoulos, J., Matsinos, I., 2008. Identifying wildland fire ignition factors through sensitivity analysis of a neural network. Natural Hazards 50, 125–143. doi:10.1007/s11069-008-9326-3.

Venables, W.N., Ripley, B.D., 2002. Modern Applied Statistics with S. 4th ed., Springer, New York.

van Voorn, G.A.K., Verburg, R.W., Kunseler, E.M., Vader, J., Janssen, P.H.M., 2016. A checklist for model credibility, salience, and legitimacy to improve information transfer in environmental policy assessments. Environmental Modelling & Software 83, 224–236. doi:10.1016/j.envsoft.2016.06.003.

Watts, M.J., Li, Y., Russell, B.D., Mellin, C., Connell, S.D., Fordham, D.A., 2011. A novel method for mapping reefs and subtidal rocky habitats using artificial neural networks. Ecological Modelling 222, 2606–2614. doi:10.1016/j.ecolmodel.2011.04.024.

Watts, M.J., Worner, S.P., 2008. Using artificial neural networks to determine the relative contribution of abiotic factors influencing the establishment of insect pest species. Ecological Informatics 3, 64–74. doi:10.1016/j.ecoinf.2007.06.004.

White, M.C., Thompson, J.D., Harrington, G.W., Singer, P.C., 1997. Evaluating criteria for enhanced coagulation compliance. Journal AWWA 89, 64–77.

Wilby, R.L., Abrahart, R.J., Dawson, C.W., 2003. Detection of conceptual model rainfall-runoff processes inside an artificial neural network. Hydrological Sciences Journal 48, 163–181. doi:10.1623/hysj.48.2.163.44699.

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. Environmental Modelling & Software 54, 108–127. doi: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 Resources Research 49, 7598–7614. doi:10.1002/2012wr012713.

Young, W.A., Millie, D.F., Weckman, G.R., Anderson, J.S., Klarer, D.M., Fahnenstiel, G.L., 2011. Modeling net ecosystem metabolism with an artificial neural network and Bayesian belief network. Environmental Modelling & Software 26, 1199–1210. doi:10.1016/j.envsoft.2011.04.004.

Zambrano-Bigiarini, M., 2014. hydroGOF: Goodness-of-fit functions for comparison of simulated and observed hydrological time series. URL: http://CRAN.R-project.org/package=hydroGOF.

Zanden, M.J.V., Olden, J.D., Thorne, J.H., Mandrak, N.E., 2004. Predicting occurrences and impacts of smallmouth bass introductions in north temperate lakes. Ecological Applications 14, 132–148. doi:10.1890/02-5036.

# Appendix A

Table A.1: Performance evaluation metrics included in HydroTest (Dawson et al., 2007, 2010).

| Statistic | Description |
|---|---|
| | *Absolute Metrics* |
| Absolute Maximum Error (AME) | Magnitude of the maximum (positive or negative) residual. Useful for establishing whether a maximum permissible error has been exceeded. Range = $[0, \infty)$; ideal value = 0. |
| Peak Difference (PDIFF) | Difference between maximum predicted and observed values. Useful for indicating whether the range of the predicted data is similar to the observed data. Range = $(-\infty, \infty)$; ideal value = 0. |
| Mean Error (ME) | Mean of the residuals. Residuals of opposite sign cancel each other out; thus, a low score may not indicate an accurate model. Range = $(-\infty, \infty)$; ideal value = 0. |
| Mean Absolute Error (MAE) | Mean of the absolute residuals (which are unaffected by cancellation). Useful for assessing overall fit with no bias towards larger or smaller values since all residuals are weighted equally. Range = $[0, \infty)$; ideal value = 0. |
| Root Mean Squared Error (RMSE) | Calculates mean of the squared residuals (which are unaffected by cancellation). Taking the square root then returns values in real units. Squaring the residuals causes bias towards the largest events; thus, this metric may be useful for assessing performance when it is more important to accurately model large values. Range = $[0, \infty)$; ideal value = 0. |
| Fourth Root of the Mean Quadrupled Error (R4MS4E) | Similar to RMSE but using the fourth power. Gives greater weighting to larger residuals than RMSE, further biasing the evaluation in favour of higher magnitude records. Range = $[0, \infty)$; ideal value = 0. |

Table A.1: Performance evaluation metrics included in HydroTest (continued).

| Statistic | Description |
|---|---|
| Mean Squared Logarithmic Error (MSLE) | Mean squared difference between logged values of observed and predicted records. Taking the logarithm of the data biases the evaluation towards smaller events. Range = $[0, \infty)$; ideal value = 0. |
| Mean Squared Derivative Error (MSDE) | Mean squared difference between the residuals at two successive time steps. Penalises noisy time series and series with timing errors. Useful for indicating the fit to the hydrograph shape in hydrological models. Not appropriate for data sets that are not in or have no temporal order. Range = $[0, \infty)$; ideal value = 0. |
| Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) | Goodness-of-fit measures, modified to penalise model complexity. In this instance, the logarithm of the RMSE is increased according to the number of free parameters in the model and the number of data points used for calibration. BIC typically penalises complexity more than AIC. Useful for selecting the minimal model that best explains the observed data. |
| Number of Sign Changes (NSC) | Number of times the sequence of residuals changes sign. Useful for identifying systematic behaviour in the residuals. Range = $[1, N-1]$, where $N$ is the number of data points. For random residuals (ideal model), the frequency of NSC sign changes should be the binomial coefficient with the number of trials equal to $N-1$, where $N$ is the number of data points (Geary, 1970). |
| | *Relative Metrics* |
| Relative Absolute Error (RAE) | Sum of the absolute residuals relative to the sum of absolute differences between the observed data and the mean of the observed data. Useful for determining whether the performance of the model is better than that of the average forecasting approach. Range = $[0, \infty)$; ideal value = 0. |
| Inertia Root Mean Squared Error (IRMSE) | Weighted RMSE, where the weight is the standard deviation of calculated increments in the observed data. Thus the original RMSE is adjusted according to the fit between the observed data and the previous observed value. Useful for determining whether the performance of the model is better than that of the naive forecasting approach. Not appropriate for data sets that are not in or have no temporal order. Range = $[0, \infty)$; values $< 80\%$ are considered satisfactory, while values $< 70\%$ are regarded as good. |

Table A.1: Performance evaluation metrics included in HydroTest (continued).

| Statistic | Description |
|---|---|
| Percent Error in Peak (PEP) | Difference between maximum predicted and observed values relative to the maximum observed value. For a perfect model, the result would be zero. Useful for indicating the mismatch in peak values for single event time series data. Range = $(-\infty, \infty)$; ideal value = 0. |
| Mean Absolute Relative Error (MARE) | Mean of absolute residual relative to the observed value. Useful for assessing performance when it is more important to accurately model lower magnitude events. Range = $[0, \infty)$; ideal value = 0. |
| Median Absolute Percentage Error (MdAPE) | Median of absolute residual relative to the observed value. Similar to MARE, but being based on the median relative residual rather than mean, this metric is less affected by skewed error distributions and outliers. Range = $[0, \infty)$; ideal value = 0. |
| Mean Relative Error (MRE) | Mean of residual relative to the observed value. Relative residuals of opposite sign cancel each other out; thus a low score may not indicate an accurate model. MARE and MdAPE are generally preferred. Range = $(-\infty, \infty)$; ideal value = 0. |
| Mean Squared Relative Error (MSRE) | Mean of squared residual relative to the observed value. Similar to MARE, but squaring the relative residual makes this metric more sensitive to the larger relative errors that occur at lower magnitudes. Range = $[0, \infty)$; ideal value = 0. |
| Relative Volume Error (RVE) | Sum of the residuals relative to the sum of the observed data. Useful for indicating the overall water balance of the model and is recommended for evaluating continuous hydrographs. Range = $(-\infty, \infty)$; ideal value = 0. |
| | *Dimensionless Metrics* |
| Coefficient of Determination (Rsq) | Square of the "Pearson product-moment correlation coefficient", describing the linear correlation between the observed and predicted data. Useful for comparisons of model performance between studies since this metric is independent of the scale of data used. This metric is insensitive to additive and proportional differences between the observed and predicted datasets; thus a high value may not indicate a good fit. Range = $[0, 1]$; ideal value = 1. |

Table A.1: Performance evaluation metrics included in HydroTest (continued).

| Statistic | Description |
|---|---|
| Coefficient of Efficiency (CE) | Also known as Nash-Sutcliffe coefficient. Compares the sum of squared residuals to the sum of squared differences between the observed data and the mean of the observed data. This metric represents an improvement over Rsq, as it is more sensitive to differences in the observed and modelled means and variances. Squared residuals may add bias to large magnitude events. Use of the observed mean as a baseline may lead to overestimation of model skill for highly seasonal variables. Range = $(-\infty, 1]$; ideal value = 1. |
| Index of Agreement measure (IoAd) | Compares the sum of squared residuals to the potential error. This metric is similar to Rsq, but is better able to handle differences in modelled and observed means and variances. Squared residuals may add bias to large magnitude events. Range = $[0, 1]$; ideal value = 1. |
| Persistence Index (PI) | Compares the sum of squared residuals to the sum of squared differences between the observed data and the previous observed value. Represents an improvement over CE when data are seasonal due to the use of previous observed value as a baseline model. Squared residuals may add bias to large magnitude events. Not appropriate for data sets that are not in or have no temporal order. Range = $(-\infty, 1]$; ideal value = 1. |
| Volumetric Efficiency (VE) | Compares the sum of absolute residuals relative to the sum of the observed data. Represents the fraction of water delivered at the proper time. Range = $(-\infty, 1]$; ideal value = 1. |
| Kling-Gupta efficiency (KGE) | CE decomposed into linear correlation, bias and variability components. Represents an improvement over CE. Range = $[0, 1]$; ideal value = 1. |

# Appendix B

## *B.1 Garson's method*

Using Garson's method, the RI of the $i$th input in predicting the output is calculated by:

$$RI_{Garson,i} = \frac{\sum_{j=1}^{J}\left(\frac{|w_{i,j}|}{\sum_{k=1}^{K}|w_{k,j}|}\times|w_{j,O}|\right)}{\sum_{l=1}^{K}\left[\sum_{j=1}^{J}\left(\frac{|w_{l,j}|}{\sum_{k=1}^{K}|w_{k,j}|}\times|w_{j,O}|\right)\right]}\times 100\% \tag{5}$$

where $w_{ij}$ is the connection weight between the $i$th input and the $j$th hidden node, $w_{j,O}$ is the connection weight between the $j$th hidden node and the output, $K$ is the number of inputs and $J$ is the number of hidden nodes in the network. Rewriting Eq. 5 as:

$$RI_{Garson,i} = \sum_{j=1}^{J}\left[\frac{|w_{i,j}|}{\sum_{k=1}^{K}|w_{k,j}|}\times\frac{|w_{j,O}|}{\sum_{j=1}^{J}|w_{j,O}|}\right]\times 100\% \tag{6}$$

it can be seen that Garson's measure of RI is the sum of products of normalised weights.

The main limitation of this method is that, because it uses absolute values of the weights, the signs of the input contributions are not taken into account, which can result in misleading RI values. For example, if an input has a positive impact on the output through one hidden node and an inhibitory effect on the output through another hidden node, the overall impact of the input should be somewhere in between (i.e. the overall contribution of an input is diminished if it has counteracting impacts through individual hidden nodes). However, as Garson's measure only accounts for the magnitude of the impacts through different hidden nodes, and not the direction, counteracting impacts are added together to strengthen the overall contribution.

*B.2 Connection Weight (CW) method*

This method is based on the sum of the products of input-hidden and hidden-output connection weights, or 'overall connection weight' (OCW) (Olden and Jackson, 2002). The OCW of the $i$th input can be calculated by:

$$OCW_i = \sum_{j=1}^{J} w_{i,j} \times w_{j,O} \tag{7}$$

The OCW values are subsequently used to compute RI values for each input as follows:

$$RI_{CW,i} = \frac{OCW_i}{\sum_{k=1}^{K} |OCW_k|} \times 100\% \tag{8}$$

The main limitation of the CW method is that it does not account for the "squashing" effect of the typically sigmoidal hidden layer activation functions (Sarle, 2000). The amount of squashing increases with the magnitude of the summed input to a hidden node; thus, if the summed input to a hidden node is large, the computed $RI$ measures are unlikely to accurately describe the modelled input-output relationships. The effect of squashing is unlikely to be a problem when modelling linear relationships, since the weights and biases feeding into a sigmoidal hidden node are generally very small, such that the summed input to the node lies on the linear part of the sigmoidal curve near the origin (Bishop, 1995). On the other hand, nonlinear relationships, such as those typical of environmental processes, rely on the nonlinear portion of the sigmoidal curve to accurately capture the input-output relationship; thus, the impact of squashing on the $RI$ values computed using the CW method is likely to be more significant.

While squashing of the input-to-hidden node weights may also affect Garson's measure, normalisation of these weights (see Eq. 6) reduces the effect

of squashing to some extent, as the excessive influence of large weights is diminished (Sarle, 2000).

### *B.3 Modified Connection Weight (MCW) method*

If the activation function used on the hidden layer nodes is symmetric about the origin (e.g. the hyperbolic tangent function), the MCW approach may be used to account for the effect of squashing on computed RI values to some extent by using this activation function to "squash" the input-hidden node weights as follows:

$$MCW_i = \sum_{j=1}^{J} g\left(w_{i,j}\right) \times w_{j,O} \tag{9}$$

where $g\left(\cdot\right)$ is the activation function used on the hidden layer nodes. If the input data are standardised, large weights feeding into the hidden nodes would be the primary cause, overall, for large summed inputs into the nodes, and hence, significant amounts of squashing. Therefore, by squashing the input-hidden node weights using the hidden layer activation functions, the influence of excessively large weights is removed. The MCW values calculated using Eq. 9 are used to compute RI values for each input using Eq. 8, substituting MCW for OCW. It is important that this method is only used when the hidden layer activation function is symmetric about the origin (i.e. $f(-x) = -f(x)$) so that the magnitude of large positive or negative weights is reduced without the sign of the weights being affected.

A limitation of this method is that the magnitudes of the input-hidden node weights are not considered in relation to those of the other weights feeding into the same hidden node (including the bias), or the values of the inputs

themselves, which all influence the degree of squashing. Consequently, the resulting $RI$ values computed using "squashed" input-hidden node weights may not give an accurate representation of the actual relative contributions of the various ANN inputs. This may be a particular issue when large bias weights saturate the activation function of a hidden node, requiring large input-hidden node weights to offset the large bias, such that the associated hidden node does not simply behave as a bias node itself. In such circumstances, squashing the input-hidden node weights in the computation of input $RI$ values may not be appropriate.

### *B.4 Profile method*

The sensitivity of an input variable describes the degree to which the output is affected by variations of that input - the more 'sensitive' the input, the greater its influence on the model output. The Profile method, like other one-at-a-time SA methods, involves successively varying each input variable over its range while keeping all others constant at arbitrary values. However, as these arbitrary values may significantly influence the results, all variables except for the variable of interest are fixed initially at their minimum values, then successively at their first quartile, median, third quartile, and maximum values. As a result, five output profiles corresponding to the five summary statistics are produced for each input variable of interest. The median of these five output profiles is then calculated to represent the median output variation over the range of the input variable of interest. Using the Profile method, the RI of each input can be calculated based on the magnitude of the range of output values produced by varying each input (Gevrey et al., 2003; Olden et al., 2004). To express this range in a similar manner to the RI

values given by Eqs. 6 and 8, the following equation is used in the **validann** implementation:

$$RI_{Profile,i} = \frac{\max(\hat{\mathbf{y}}_i) - \min(\hat{\mathbf{y}}_i)}{\sum_{k=1}^{K} [\max(\hat{\mathbf{y}}_k) - \min(\hat{\mathbf{y}}_k)]} \times 100\% \tag{10}$$

where $\hat{\mathbf{y}}_i$ is the vector of 101 median output values obtained by varying the $i$th input over its range.

In the **validann** implementation of this algorithm, each input is increased in turn from its minimum value to its maximum value in increments of 1%, producing five output values for each of the 101 input values considered. While the $RI_{Profile,i}$ values are calculated based on median output values, six profiles of output variation are returned for each input: the five profiles corresponding to the five summary statistics, together with the median of these profiles.

### *B.5 Partial Derivatives (PaD) method*

A similar, but more direct and computationally efficient, method for evaluating the sensitivities of model inputs involves computing the partial derivative of the model output with respect to each input variable of interest. By definition, each partial derivative defines the local rate of change of the output with respect to the corresponding input, while holding all other inputs fixed (Sarle, 2000). Using a simple backward chaining partial differentiation rule, the partial derivative of an ANN output $O$ with respect to its $i$th input $I_i$ is calculated according to (Hashem, 1992):

$$\frac{\partial O}{\partial I_i} = \sum_{j=1}^{J} \frac{\partial O}{\partial h_j} \frac{\partial h_j}{\partial Z_j} \frac{\partial Z_j}{\partial I_i} \tag{11}$$

where $h_j$ is the output from the $j$th hidden node, $Z_j$ is the input to the $j$th hidden node, and $J$ is the number of hidden nodes in the network. The original PaD approach of Dimopoulos et al. (1995, 1999) was based on the assumption of logistic sigmoid activation functions, giving:

$$\frac{\partial O_n}{\partial I_{i,n}} = O_n(1 - O_n) \sum_{j=1}^{J} w_{jO} h_{j,n}(1 - h_{j,n}) w_{ij} \tag{12}$$

which returns a partial derivative value for every $n = 1, \ldots, N$ observation in a given dataset, where $N$ is the total number of observations. Consequently, the PaD approach returns a profile of partial derivatives for each ANN input, where the partial derivative values can be interpreted in a similar way to the coefficients in linear models: a positive partial derivative indicates that the model output will increase with an increase in the input variable, while a negative partial derivative indicates a reduction in the output value will occur (Gevrey et al., 2003). An important advantage of the PaD approach over the Profile method is that the input sensitivities are calculated based on observed data rather than on synthetic input data that often include infeasible combinations of input values.

A limitation of the original PaD approach is due to the assumption of logistic sigmoid activation functions (to the authors' knowledge, the PaD approach has not been applied to ANNs with different activation functions). In a recent paper, Coad et al. (2014) stated that their reason for choosing Garson's method over the PaD approach for quantifying ANN input importance was that logistic sigmoid activation functions had not been used in their model. However, the PaD approach is easily extended to include other commonly used differentiable activation functions. As such, a more general form

of Eq. 12 is used in the **validann** implementation of this method, which can be used to compute partial derivatives for ANNs with arbitrary differentiable activation functions:

$$\frac{\partial O_n}{\partial I_{i,n}} = \sum_{j=1}^{J} w_{jO} \frac{\partial O_n}{\partial Z_{O,n}} \cdot w_{ij} \frac{\partial h_{j,n}}{\partial Z_{j,n}} \tag{13}$$

where $Z_{O,n}$ is the summed input to the output node $O$. For commonly used activation functions, including the identity, logistic sigmoid, hyperbolic tangent and exponential functions, $\partial O_n/\partial Z_{O,n}$ and $\partial h_{j,n}/\partial Z_{j,n}$ in Eq. 13 may be substituted by Eqs. 14-17, respectively:

$$\frac{\partial y}{\partial x} = 1 \tag{14}$$

$$\frac{\partial y}{\partial x} = y(1-y) \tag{15}$$

$$\frac{\partial y}{\partial x} = \frac{1}{cosh^2(x)} \tag{16}$$

$$\frac{\partial y}{\partial x} = \exp(x) \tag{17}$$

Another potential disadvantage of the original PaD approach is that the input sensitivities returned by Eqs. 12 and 13 are in *absolute* form, meaning they are not invariant to the magnitudes of either $O$ or $I_i$ (McCuen, 1973). For example, a large absolute partial derivative, $\partial O_n/\partial I_{i,n}$, indicates the model output $O$ is particularly sensitive to input $I_i$ about its $n$th value. However, if the magnitude of $I_{i,n}$ itself was particularly small, a 'small' variation in $I_{i,n}$ (i.e. $\partial I_{i,n}$) may in fact not be so small relative to its size and, thus, the relative influence of $I_{i,n}$ on the output would be less than that computed using absolute partial derivatives. To overcome this, Mount et al. (2013); Dawson et al. (2014) computed the *relative* sensitivity (RS) of each input by

normalising the partial derivatives given by Eq. 12, as follows:

$$RS_{i,n} = \frac{\partial O_n / O_n}{\partial I_{i,n} / I_{i,n}} = \frac{\partial O_n}{\partial I_{i,n}} \cdot \frac{I_{i,n}}{O_n} \tag{18}$$

Unlike absolute sensitivity, RS values allow the assessment of an input's relative influence on the output, taking into account the magnitudes of the input and output values at which sensitivity is calculated. Consequently, the **validann** implementation of the PaD method returns both absolute and relative sensitivity profiles, as defined by Eqs. 13 and 18, respectively. However, for ANNs, whose inputs and outputs are usually standardised in some way, care must be taken when interpreting the RS values, since the way in which the data are standardised may significantly affect the resulting RS values (e.g. a value of $O = 0$ results in an undefined value of RS). As such, when using this method, it is recommended that the input and output data be rescaled such that all values are greater than zero (e.g. $0.1 < I_i < 0.9$ and $0.1 < O < 0.9$).

In order to reduce the large number of sample partial derivatives returned by the PaD method into a single measure of importance for each input, the sum of square partial derivatives (SSD) over the observed dataset has been used (Dimopoulos et al., 1999; Gevrey et al., 2003):

$$SSD_i = \sum_{n=1}^{N} \left( \frac{\partial O_n}{\partial I_{i,n}} \right)^2 \tag{19}$$

This measure may be suitable for ranking input importance in individual studies; however, since Eq. 19 deals with squared sensitivities and is not normalised, a more comparable measure of RI is calculated in the **validann** implementation of the PaD method by normalising the root mean squared

partial derivatives (RMSD) as follows:

$$RI_{PaD,i} = \frac{RMSD_i}{\sum_{k=1}^{K} RMSD_k} \times 100\% \tag{20}$$

where

$$RMSD_i = \sqrt{\sum_{n=1}^{N} \left( \frac{\partial O_n}{\partial I_{i,n}} \right)^2 / N} \tag{21}$$

**Highlights**

- A comprehensive validation framework for ANNs is proposed.
- The ‘validann’ R-package for implementing the validation framework is introduced.
- Application of the framework and R-package is demonstrated on two real case studies.
- Results reveal that predictively valid ANN models may not be credible.
- Adoption of the framework leads to improvements in overall ANN validity.