Statistics and Computing (2025) 35:22
https://doi.org/10.1007/s11222-024-10551-0

ORIGINAL PAPER

A neural network-based adaptive cut-off approach to normality testing for dependent data

Minwoo Kim[1] · **Marc G. Genton**[2] · **Raphaël Huser**[2] · **Stefano Castruccio**[3]

Received: 26 July 2024 / Accepted: 10 December 2024

Abstract

There is a wide availability of methods for testing normality under the assumption of independent and identically distributed data. When data are dependent in space and/or time, however, assessing and testing the marginal behavior is considerably more challenging, as the marginal behavior is impacted by the degree of dependence, which typically leads to an inflation in Type I error rates. We propose a new approach to assess normality for dependent data by non-linearly incorporating existing statistics from normality tests as well as sample moments such as skewness and kurtosis through a neural network with adaptive cut-offs by which the Type I error inflation issue is fixed. We calibrate (deep) neural networks by simulated normal and non-normal data with a wide range of dependence structures and we determine the probability of rejecting the null hypothesis. We compare several approaches for normality tests and demonstrate the superiority of our method in terms of statistical power through an extensive simulation study. A real world application to global temperature data further demonstrates how the degree of spatio-temporal aggregation affects the marginal normality in the data.

Keywords Adaptive cut-off · Aggregation of test statistics · Neural network · Normality test · Spatio-temporal statistics

1 Introduction

One of the fundamental tasks for both model design and validation is to identify a marginal distribution for the data (or the residuals according to some trend), and to test whether it can be ascribed to a known parametric model. Arguably one, if not the most, important case is that of the normal distribution. In this case, in addition to informal methods such as quantile-quantile plots and histograms, there is a wide variety of normality tests under the assumption of independent and identically distributed (*i.i.d.*) data; see, e.g., Anderson and Darling (1952); Shapiro and Wilk (1965); Lilliefors (1967), and Jarque and Bera (1980). Normality tests are based on statistics such as skewness and kurtosis, which summarize some properties of the distribution and compare them to the statistic expected from a normal distribution. The tests may not provide unanimous results if, for instance, the data resemble a normal distribution with respect to one statistic but not with respect to others; see Thode (2002).

When the data are not *i.i.d.*, with dependence informed possibly (but not necessarily) by space and/or time, testing the marginal behavior is considerably more challenging. Indeed, while it is methodologically convenient to assume a Gaussian process, i.e., a random function with marginal Gaussian distribution, the dependence leads to excessive rejections in normality tests intended for *i.i.d.* data. As an extreme example, one may consider a Gaussian process with perfect correlation: every realization will comprise of a vector of identical values, hence leading to the impossibility of assessing the marginal behavior. Therefore, standard tests intended for *i.i.d.* data are bound to exhibit inflated Type I error rates on dependent data, even if the process is in fact

✉ Stefano Castruccio
scastruc@nd.edu

Minwoo Kim
mwkim@pusan.ac.kr

Marc G. Genton
marc.genton@kaust.edu.sa

Raphaël Huser
raphael.huser@kaust.edu.sa

[1] Department of Statistics, Pusan National University, Busan 46241, South Korea

[2] Statistics Program, King Abdullah University of Science and Technology, 23955-6900 Thuwal, Saudi Arabia

[3] Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556, USA

Published online: 30 December 2024

Gaussian. It is hence necessary to develop tests that account for dependence, and which would adjust the decision criterion accordingly. However, only a limited number of previous studies have explored testing methods for dependent data. The recent work of Horváth et al. (2020) proposed a modification of the Jacque-Bera normality test (Jarque and Bera 1980) by estimating the spatial structure. In their review on multivariate normality tests, Chen and Genton (2023) also extended the test of Horváth et al. (2020) to the multivariate setting.

While a test adjustment may provide a partial solution, relying on only a single test with dependent data is limiting, as the null distribution of the test statistic strongly depends on the correlation structure. For instance, the null distribution of a single test statistic such as the Shapiro-Wilk normality test (Shapiro and Wilk 1965) will differ depending on the strength of the spatial dependence. In order to enhance the test power, a solution is to combine different tests so as to use multiple statistics at the same time.

One simple approach is the Bonferroni correction, which predicates rejection of H_0 if at least one of the m tests is rejected at level α/m; see, e.g., Haynes (2013). The Bonferroni correction guarantees the appropriate Type I error rate but is overly conservative and has an optimal power only if the test statistics are independent. Another approach to combine m tests is to use Fisher's method, which combines information from the p-values of all tests. If the tests are all independent, then $-2\sum_{i=1}^{m} \ln p_i$ follows a χ^2_{2m} distribution (Fisher 1992; Kost and McDermott 2002). A linear combinations of p-values has also been suggested in Edgington (1972). Winkler et al. (2016) reviewed fifteen methods for combining p-values.

Neural networks-based approaches with descriptive statistics as inputs for *i.i.d.* data have been introduced to test for normality and compared with standard tests (Wilson and Engel 1990). Sigut et al. (2006) assessed univariate normality using trained neural networks with input features including sample skewness, sample kurtosis, test statistics in Shapiro and Wilk (1965), the Fisher transform of the Pearson correlation coefficient, and the family of test statistics proposed by Vasicek (1976). More recently, Simić (2021) extended previous approaches by adding summary statistics such as minimum, maximum, and sample size to the representative input set. All the past studies showed that neural network approaches can often outperform typical statistical tests by combining information in a non-linear fashion.

In this work, we propose a more general neural network-based test for normality aimed at dependent data (in space, time, space/time, or simply multivariate) with a novel adaptive cut-off technique. Since the strength of the underlying dependence results in Type I error inflation, we assume that the cut-off is a function of the dependence parameter. We employ a Matérn covariance model, which is most commonly used in spatial analysis, but our adaptive cut-off technique can be flexibly applied to any other statistical model as well. It will be shown that the adaptive cut-offs suitably control the nominal Type I error rate. Also, in terms of power, our neural network-based test outperforms currently available methods for testing normality when the independence assumption is violated.

The paper proceeds as follows. In Sect. 2, we present the general framework of combining multiple tests (Sect. 2.2), introduce our neural network methodology (Sect. 2.3), and describe our novel adaptive cut-off technique (Sect. 2.4). In Sect. 3, we conduct a simulation study for testing the assumption of normality on a spatial grid and we show the improvement against currently available methods. In Sect. 4, we apply the proposed method to spatially distributed data from a global climate model simulation in order to test normality at different levels of spatial aggregation. In Sect. 5, we discuss conclusions and directions for future research.

2 Methodology for normality testing

Let $\boldsymbol{Y} = (Y(\mathbf{s}_1), \ldots, Y(\mathbf{s}_M))^\top \in \mathbb{R}^M$ be a random vector on a manifold. This manifold can represent a spatial domain such as a Euclidean space or a sphere for a spatial random vector, the positive real line for time series or a Cartesian product of the two in the case of space-time domain. Let H_0 be any model property that $Y(\cdot)$ may satisfy (in our case the marginal distribution being Gaussian). We aim to create a classifier $C : \boldsymbol{Y} \mapsto \{0, 1\}$ (where 0 corresponds to normality) which is more powerful than any other classifier whose test statistic is a function of the test statistics from a selection of well-known classifiers, for a fixed Type I error rate α. Formally, we have $P(C(\boldsymbol{Y}) = 1 \mid H_0 \text{ true}) = \alpha$ and for any other available classifier $\tilde{C}$ at the same Type I error rate we have that $P(C(\boldsymbol{Y}) = 1 \mid H_0 \text{ false}) \geq P(\tilde{C}(\boldsymbol{Y}) = 1 \mid H_0 \text{ false})$. In our work, H_0 being false means that the data are not normal, which implies an overly general class of models. As such, we only focus on some particular classes of non-Gaussianity, as will be specified later.

2.1 Individual normality tests

For simplicity of notation, we denote with $Y_i = Y(\mathbf{s}_i)$, $i = 1, \ldots, M$, $\boldsymbol{Y} = (Y_1, \ldots, Y_M)^\top$ the data for which one wants to assess normality. We focus on four tests that are used as inputs for our neural network: Shapiro and Wilk (1965); Lilliefors (1967); Jarque and Bera (1980), and Anderson and Darling (1952).

The Shapiro–Wilk test relies on calculating the order statistics and comparing the observed versus expected values $W = (\sum_{i=1}^{M} a_i Y_{(i)})^2 / \sum_{i=1}^{M} (Y_i - \bar{Y})^2$, where $Y_{(i)}$ is the i^{th} order statistic, $\bar{Y}$ is the sample mean, and a_i is a weight calcu-

lated from the expected means and covariances of the order statistics under the null hypothesis of *i.i.d.* data. Despite its popularity, the Shapiro–Wilk test relies on the availability of appropriate values of a_i which have no closed form, so the values are determined through Monte Carlo simulation, and for large sample sizes M, it is more difficult to obtain accurate a_i estimates (Das and Imon 2016). Indeed, in all the code implementation we used throughout this work, the size of M is limited to a few thousand points.

The Lilliefors test is an adaptation of the Kolmogorov–Smirnov test for Gaussian data. It measures the maximum deviation of the empirical and theoretical cumulative distribution functions (CDFs), denoted with F_M and F, respectively: $D_M = \sup_y |F_M(y) - F(y)|$. Then D_M is compared to the expected distribution under the null hypothesis, and a p-value is calculated.

The Anderson–Darling test statistic is also based on deviation from the theoretical CDF: $A^2 = M \int_{-\infty}^{\infty} \frac{\{F_M(y)-F(y)\}^2}{F(y)\{1-F(y)\}} \mathrm{d}F(y)$. Rather than measuring the maximum deviation between the empirical and theoretical CDFs, Anderson–Darling weighs deviations in the tails more heavily.

Finally, the Jarque–Bera test calculates the test statistic $JB = \frac{M}{6}\{S^2 + (K-3)^2/4\}$, where S and K are the sample skewness and kurtosis, respectively. Informally, the Jarque-Bera test checks whether the sample's skewness and kurtosis match those of a normal distribution. The asymptotic expected values of the empirical skewness and kurtosis are 0 and 3, and the asymptotic variance of the empirical skewness and kurtosis are $6/M$ and $24/M$. Thus, the Jarque–Bera statistic is a squared sum of two asymptotically independent standardized normal distributions, and thus distributed as a χ^2 random variable.

These four tests focus on different yet related methods to compare the data with the Gaussian distribution: the order statistics (Shapiro-Wilks), the unweighted or weighted CDF distance (Lilliefors and Anderson-Darlin, respectively) and skewness and kurtosis (Jarque-Brera). It is therefore of interest to devise an approach to find a test which merges the information available from the individual tests.

2.2 Combining tests

Let $C_1, C_2, \ldots, C_m$ be m classifiers with Type I error α. Insofar as they are distinct classifiers, they assess at least partly different properties implied by H_0. For example, to test H_0: $\boldsymbol{Y}(\mathbf{s})$ is normally distributed, C_1 may be testing whether the skewness is zero, while C_2 may be testing whether the excess kurtosis is zero. Both are appropriate level-α tests of H_0 and their performance, measured by statistical power, will vary depending on how the departure of the alternative model hypothesis H_1 to H_0 affects the properties assessed by each classifier.

Ideally, we would like to combine the m classifiers into a single level-α classifier C that is more powerful. In our case, combining the classifiers is complicated because of two main issues. First, since each individual classifier is testing different but related properties of H_0, the m classifiers are expected to be dependent; the Bonferroni correction is overly conservative because the effective number of tests is less than m due to this dependence and Fisher's method's asymptotic distribution is no longer valid. In the following subsection, we explain how to combine m different tests. We consider both linear and non-linear combinations using a logit transformation and a neural network, respectively. Parameters in both combinations are optimized by simulated samples from H_0 and H_1.

2.3 Combining tests through neural networks

If $T_1, T_2, \ldots, T_m$ are test statistics for classifiers $C_1, C_2, \ldots, C_m$, the simplest approach to combine them is through a classifier comprising of a linear combination and a logit transformation: $\mathrm{logit}\{P(C(\boldsymbol{Y}) = 1)\} = \gamma_0 + \gamma_1 T_1 + \cdots + \gamma_m T_m$. While this approach allows to combine information across tests, its functional form limits its flexibility. In this work, we propose a more flexible approach which relies on a (deep) neural network, i.e., we filter the test statistics through a combination of multiple non-linear functions (Chapter 6 in Goodfellow et al. (2016)). More specifically, we consider the following:

$$\begin{aligned} F(\boldsymbol{Y}) &= P(C(\boldsymbol{Y}) = 1) \\ &= S\{W_L \sigma_{L-1}(W_{L-1}\sigma_{L-2} \cdots W_4 \sigma_3(W_2 \sigma_1(W_1 \boldsymbol{T})))\}, \quad (1) \end{aligned}$$

which is a composition of:

1. The m-dimensional vector of all the test statistics considered $\boldsymbol{T} = (T_1, \ldots, T_m)^\top$. If no classifiers are available, one may also consider $\boldsymbol{T}$ to be the identity function so that the vector of the observed data $\boldsymbol{Y}$ itself is the desired input. For simplicity of notation in the next points, we set $n_0 = m$.
2. L matrices representing linear transformations $W_i : \mathbb{R}^{n_{i-1}} \mapsto \mathbb{R}^{n_i}$, $i = 1, \ldots, L$. The parameter n_i is the *width* of layer i, while L is the *depth* of the neural network.
3. L fixed non-linear transformations σ_i that are applied component-wise. In this paper, we use the common rectified linear unit (ReLU, Chapter 6 in Goodfellow et al. (2016)) activation function defined by $\sigma(z) = \max(0, z)$.
4. A sigmoid function $S(z) = (1+e^{-z})^{-1}$, which guarantees an output in $[0, 1]$ that we can interpret as $P(C(\boldsymbol{Y}) = 1)$.

Inference (i.e., learning) can be performed by simulating the representative samples $\boldsymbol{Y}_1^{H_0}, \ldots, \boldsymbol{Y}_{N_0}^{H_0} \in \mathbb{R}^M$ satisfying

H_0 and $\boldsymbol{Y}_1^{H_1}, \ldots, \boldsymbol{Y}_{N_1}^{H_1}$ satisfying H_1. The matrix entries of W_i in (1) are then learned by optimizing the binary cross-entropy (or log loss), which penalizes overly-confident incorrect predictions. Indeed, if we denote by $p_i^{H_0} = P\left(C\left(\boldsymbol{Y}_i^{H_0}\right) = 1\right)$ and $p_i^{H_1} = P\left(C\left(\boldsymbol{Y}_i^{H_1}\right) = 1\right)$, both terms depend on the neural network parameters in (1). Therefore, the logloss

$$\text{logloss} = \sum_{i=1}^{N_0} \log\left(1 - p_i^{H_0}\right) + \sum_{i=1}^{N_1} \log p_i^{H_1} \tag{2}$$

is also a function of the same parameters and can be minimized with respect to them.

In this work, we use the stochastic gradient descent-based optimization algorithm Adam (Kingma and Ba 2015). Since the neural network outputs a probability, instead of setting an arbitrary cut-off of 0.5, we set it such that the method has a pre-specified Type I error rate α. Formally, this cut-off q_α is defined using a neural network calibrated with the training data and the outputs (1) as:

$$q_\alpha = \inf_{q \in [0,1]} \left[\frac{1}{N_0} \sum_{i=1}^{N_0} \mathbb{I}\{F(\boldsymbol{Y}_i^{H_0}) > q\} \leq \alpha \right], \tag{3}$$

where F is the probability of rejection of H_0 as defined in (1).

2.4 Adaptive cut-off

In this section we assume for simplicity that the Gaussian training data are spatially dependent and generated from a Matérn covariance model (Stein 1999) with varying degrees of spatial dependence. The proposed adaptive cut-off approach can however be easily generalized to other practical dependence structures including but not limited to spatial, temporal, and spatio-temporal models. For any two observations $Y(\mathbf{s}_i), Y(\mathbf{s}_j)$ at two generic locations $\mathbf{s}_i, \mathbf{s}_j \in \mathbb{R}^2$, the covariance in the Matérn model is:

$$\begin{aligned} &\text{cov}\{Y(\mathbf{s}_i), Y(\mathbf{s}_j)\} \\ &= \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)} \left(\frac{\|\boldsymbol{s}_i - \boldsymbol{s}_j\|}{\beta} \right)^{\nu} \mathcal{K}_\nu \left(\frac{\|\boldsymbol{s}_i - \boldsymbol{s}_j\|}{\beta} \right), \end{aligned} \tag{4}$$

where $\mathcal{K}_\nu$ is the modified Bessel function of the second kind of order $\nu > 0$, and $\|\boldsymbol{s}_i - \boldsymbol{s}_j\|$ is the Euclidean distance. The parameter σ^2 specifies the marginal variance and $\beta > 0$ controls the range of the spatial dependence: when we consider a distance $\sqrt{8\nu}/\beta$, the spatial correlation is near 0.1 for all ν (Stein 1999). Finally, ν specifies the regularity/smoothness of the process, i.e., the degree of mean square differentiability. Regardless of the number of locations (dimension of the data), the three parameters, σ^2, β, and ν, fully characterize a Matérn covariance model.

Since we simulate the training data by varying the spatial range β, a single cut-off value independent of this parameter would inevitably result in incorrect Type I error rates. In this work, we propose a more flexible cut-off q_α in (3) as a function of β. Specifically, let $n_{\beta;\text{train}}$ be the number of range parameters for the training set such that $\beta_1, \ldots, \beta_{n_{\beta;\text{train}}}$ are the parameters used to generate $\boldsymbol{Y}_1^{H_0}, \ldots, \boldsymbol{Y}_{N_0}^{H_0}$. For each β_g and its corresponding observations, a cut-off value is elicited as in (3) denoted by $q_\alpha(\beta_g)$ for $g = 1, \ldots, n_{\beta;\text{train}}$. We employ non-parametric kernel regression to estimate the cut-off function based on $n_{\beta;\text{train}}$ pairs $(\beta_1, q_\alpha(\beta_1))^\top, \ldots, (\beta_{n_{\beta;\text{train}}}, q_\alpha(\beta_{n_{\beta;\text{train}}}))^\top$. We use a Gaussian kernel and assume that the estimated cut-off at a new testing value β is:

$$\hat{q}_\alpha(\beta) = \frac{\sum_{g=1}^{n_{\beta;\text{train}}} K_h(\beta - \beta_g) q_\alpha(\beta_g)}{\sum_{g=1}^{n_{\beta;\text{train}}} K_h(\beta - \beta_g)}, \tag{5}$$

where $K_h(\beta - \beta_g) = h^{-1} K\left(h^{-1}(\beta - \beta_g)\right)$, $K(z) = \exp\left(-z^2/2\right)/\sqrt{2\pi}$ for any $z \in \mathbb{R}$, and h is a selected bandwidth. We implement this kernel regression using the R package `np` (Li and Racine 2003; Li et al. 2013). The figures in Sect. 3.3 show examples of estimated cut-off functions and demonstrate that the cut-off functions effectively address the inflated Type I error rates.

2.5 Algorithms with adaptive cut-offs

We describe steps for calibrating a neural network and obtaining a cut-off curve:

Algorithm 1: how to obtain adaptive cut-offs

Step 0. Determine the structure of a neural network specifying L and $(n_1, \ldots, n_L)^\top$. In the sensitivity analysis later on, we will show the robustness with respect to choices of neural network architectures.

Step 1. Generate representative Gaussian samples, $\boldsymbol{Y}_1^{H_0}, \ldots, \boldsymbol{Y}_{N_0}^{H_0}$, and non-Gaussian samples, $\boldsymbol{Y}_1^{H_1}, \ldots, \boldsymbol{Y}_{N_1}^{H_1}$. Here, different values of dependence parameters are used to generate the samples. We focus on the most commonly used Matérn covariance model, but other dependence model can also be applied (see supplementary material).

Step 2. Using the generated samples in **Step 1**, optimize W_i by minimizing the loss (2). Here we use the popular Adam optimizer, though other methods are also possible.

Step 3. Let β represent a general dependence parameter for a statistical model. For each β, using corresponding samples, compute the adaptive cut-offs in (3) denoted by $q_\alpha(\beta)$.

Step 4. Employing non-parametric kernel regression in (5), obtain a smooth cut-off curve.

Note that although we outlined here the steps needed to derive a one-dimensional cut-off curve, our adaptive cut-off approach can be straightforwardly generalized to hypersurfaces when β is a vector of dependence parameters. We also note that, after calibrating a neural network in **Step 2**, the adaptive cut-offs $q_\alpha(\beta)$ for varying values of α can be computed without recalibrating a neural network.

Next, we explain the step-by-step process of conducting our normality test for a given data vector using the Matérn covariance function as an example. Let $\mathbf{Y}$ denote a realization of a random vector from a spatial domain. Then, the steps to determine the normality of $\mathbf{Y}$ are as follows:

Algorithm 2: how to conduct normality test
Step 1. Estimate β and ν in (4) given $\mathbf{Y}$ and let the estimators be denoted by $\hat{\beta}$ and $\hat{\nu}$, respectively. In this paper, we use the software `ExaGeoStatR` (Abdulah et al. 2023) to estimate the parameters.
Step 2. Using the estimated value, $\hat{\nu}$, generate representative Gaussian and non-Gaussian samples as in the **Step 1** of the **Algorithm 1**.
Step 3. Calibrate a neural network following the **Algorithm 1** and obtain a cut-off curve.
Step 4. Determine the normality of $\mathbf{Y}$ comparing the output of the calibrated neural network and the cut-off value corresponding to $\hat{\beta}$ from the obtained cut-off curve. Here, the inputs of the calibrated neural network are individual test statistics computed using $\mathbf{Y}$.

According to the steps outlined in **Algorithm 2**, representative data must be generated and a new neural network must be calibrated whenever a new data vector is encountered. However, this approach is inefficient when dealing with a large number of data points. We fix the value of ν for the simulation experiments in Sect. 3 and, for a real-world example in Sect. 4, we describe how to practically apply the algorithms. Specifically, we use six representative values of ν and six corresponding neural networks are calibrated. Then, each data point is assigned to one of the six neural networks based on its estimated ν value. This approach is computationally convenient as it only uses a small number of pre-trained networks.

2.6 An existing test for dependent normal data

Horváth et al. (2020) introduced a test to determine whether some dependent data on a regular grid can be regarded as a realization of a Gaussian process. We show here the main idea behind their approach, and we refer to their manuscript for a comprehensive derivation of the test statistic and relevant estimators. Their method involves modeling a process that accounts for the spatial correlation and computing two statistics related to sample skewness and kurtosis. The test can be performed since Horváth et al. (2020) demonstrated that the sum of squares of the two statistics asymptotically follows a chi-square distribution with two degrees of freedom. Specifically, the data $\{Y(\mathbf{s}_1), \ldots, Y(\mathbf{s}_M)\}$, where $\{\mathbf{s}_1, \ldots, \mathbf{s}_M\} \in \mathbb{Z}^d$ are locations in a d-dimensional spatial domain, are assumed to follow the moving average model $Y(\mathbf{s}) = \mu + \sum_{\mathbf{s}' \in \mathbb{Z}^d} a(\mathbf{s}')\epsilon(\mathbf{s} - \mathbf{s}')$, $\mathbf{s} \in \mathbb{Z}^d$, where μ is the process mean and $\epsilon(\mathbf{s}), \mathbf{s} \in \mathbb{Z}^d$ are independent, standard normal innovations. We denote sample skewness and kurtosis with the standardized data by $\mathcal{S}_M$ and $\mathcal{K}_M$ respectively, and by $\phi^2_{\mathcal{S}}$ and $\phi^2_{\mathcal{K}}$ their asymptotic variances (which depend on μ and $a(\mathbf{s}')$). The test statistic is defined as $\mathcal{S}^2_M/\hat{\phi}^2_{\mathcal{S}} + \mathcal{K}^2_M/\hat{\phi}^2_{\mathcal{K}}$, where $\hat{\phi}^2_{\mathcal{S}}$ and $\hat{\phi}^2_{\mathcal{K}}$ are kernel estimators whose detailed explanation and comprehensive derivations are given in their paper. In Sect. 3 of this work, we use this test as a benchmark to compare the performance of our proposed method. We use a truncated kernel with bandwidth $h = \lfloor 4(M/100)^{0.2} \rfloor$, as noted for its stability in Section 3 of Horváth et al. (2020). The truncated kernel K_{TR} is defined as $K_{TR}(t) = \mathbb{I}(|t| \leq 1)$.

Horváth et al. (2020)'s method exhibits lower Type I error rates compared to classical tests which assume independence, however, it still incurs unstable Type I error rates as the degree of the dependence varies (see figures in Sect. 3.3).

3 Simulation study

3.1 Simulation design

We simulate a zero mean, isotropic Gaussian random field with Matérn covariance function in (4) on a two dimensional unit square regular grid of size 60×60 such that the spatial dimension 3600 is sufficiently large to be advantageous for the asymptotic properties discussed in Horváth et al. (2020). We assume $\nu \in \{0.5, 1.0\}$ where the former value simplifies the covariance function to $\sigma^2 \exp(-\|s_i - s_j\|/\beta)$. We present results for $\nu = 0.5$ in this section, while the results for $\nu = 1.0$ are deferred to the supplement Section A. We choose $n_{\beta;\text{train}} = 30$ equally spaced values of β between 0 and $\beta_{\max} = 0.234$ (including both endpoints) in the training set, spanning from zero to strong dependence on a unit square. The range parameter bound $\beta_{\max}$ is chosen so that the *effective range*, i.e., the distance at which the correlation between two locations reaches 0.05, is 0.7. This bound is valid only for the unit square, so it requires a rescaling in the application, and also depends on ν. In the test set we choose $n_{\beta;\text{test}} = 50$ equally spaced values of β from 0 to $\beta_{\max}$, to demonstrate that the neural network is capable of interpolating between different choices of range parameters.

The sets of βs in the training set and testing set are denoted by $\mathcal{B}_{\text{train}}$ and $\mathcal{B}_{\text{test}}$, respectively, such that $|\mathcal{B}_{\text{train}}| = n_{\beta;\text{train}}$ and $|\mathcal{B}_{\text{test}}| = n_{\beta;\text{test}}$. Non-normal distributions in the training and testing set were created by applying a signed power transformation to the baseline Matérn Gaussian random field. Specifically, for an exponent parameter p, a value z was transformed to $f(z; p) = |z|^p \text{sign}(z)$, for values of p in the set $\mathcal{P}_{\text{train}} = \{1.2, 1.4, 1.6, 1.8\}$ in the training set, and in the set $\mathcal{P}_{\text{test}} = \{1.1, 1.2, \ldots, 2.0\}$ in the testing set, to demonstrate the neural network's ability to interpolate and (modestly) extrapolate. We denote by $|\mathcal{P}_{\text{train}}| = n_{p;\text{train}}$ and $|\mathcal{P}_{\text{test}}| = n_{p;\text{test}}$, and we generate $n_{\text{sample}} = 200$ sample points for each combination of (β, p) in the case of non-normal data. Therefore, the training set contains $n_{\beta;\text{train}} \times n_{p;\text{train}} \times n_{\text{sample}} = 24,000$ (non-normal) data points, while the testing set contains $n_{\beta;\text{test}} \times n_{p;\text{test}} \times n_{\text{sample}} = 100,000$ (non-normal) data points. For the null hypothesis, i.e., normal data with $p = 1$, we generate an equivalent number of samples, i.e, the training set contains 48,000 points, while the testing set contains 200,000 points using the same sets $\mathcal{B}_{\text{train}}$ and $\mathcal{B}_{\text{test}}$, respectively.

Type I errors for individual normality tests introduced in Sect. 2.1 are presented in Sect. 3.2. Results in terms of Type I error and power for our neural network, the linear classifiers and Horváth et al. (2020)'s method are shown in Sect. 3.3.

The hypotheses H_0 and H_1 can be specified in terms of the aforementioned values and set notations. For the training set, we have:

$$H_0 : p = 1 \text{ and } \beta \in \mathcal{B}_{\text{train}}, \quad H_1 : p \in \{1.2, 1.4, 1.6, 1.8\} \text{ and } \beta \in \mathcal{B}_{\text{train}}.$$

In the testing set, the hypotheses are:

$$H_0 : p = 1 \text{ and } \beta \in \mathcal{B}_{\text{test}}, \quad H_1 : p \in \{1.1, 1.2, \ldots, 2.0\} \text{ and } \beta \in \mathcal{B}_{\text{test}}.$$

3.2 Classical tests

The Type I errors for the classical normality tests increase as the range of dependence increases in the simulation data, as is apparent in Fig. 1. These tests are therefore not appropriate given their assumption of independence. Given their uncalibrated Type I error, we do not calculate the power of these tests and do not compare them with the other methods shown in the following sections.

3.3 Tests for dependent data

We use $m = 6$ inputs: the four test statistics of the normality tests in Sect. 2.1 along with the sample skewness and kurtosis. We rely on a neural networks with $L = 2$ hidden layers and with $n_1 = 256$ and $n_2 = 128$ nodes. To at least partly mitigate overfitting we use dropout (Srivastava et al. 2014) during training, which randomly removes a fraction of nodes

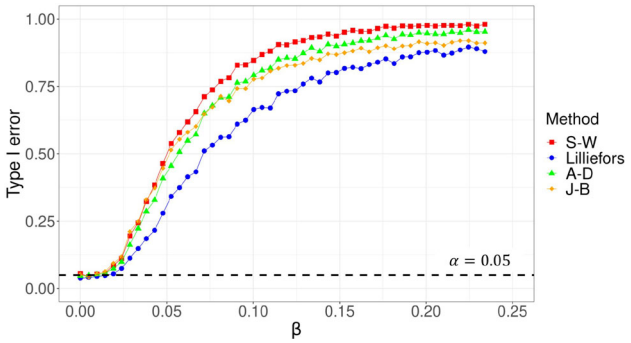


Fig. 1 y-axis: Type I errors for Shapiro–Wilk test (red), Lilliefors test (blue), Anderson–Darling test (green), and Jarque–Bera test (orange). x-axis: The dependence parameter β of a Matérn covariance function as shown in (4) when the other two parameters are fixed, $\sigma^2 = 1$ and $\nu = 0.5$. The black dashed horizontal line in the figure represents 5% of Type I error

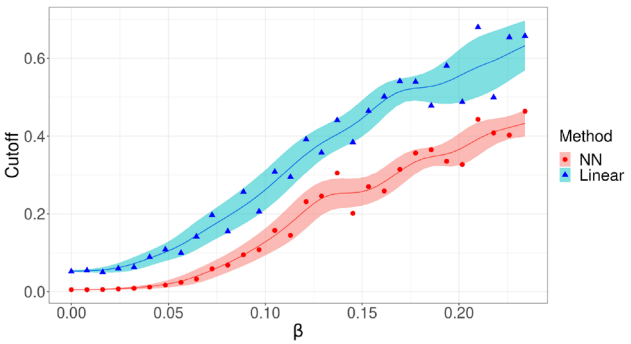


Fig. 2 Simulation study: non-parametric Gaussian kernel regressions as defined in (5) with a bandwidth $h = 0.3$ for neural network (red) and linear (blue) classifiers. On the x-axis are the range parameter β of the Matérn covariance (4), while on the y-axis the predicted cut-off and corresponding pointwise 95% confidence interval are represented by solid lines and bands, respectively. The other two parameters are fixed at $\sigma^2 = 1$ and $\nu = 0.5$

during each training step and acts as a form of regularization. In each of the L layers, 30% of nodes are randomly removed during each training step. We provide a sensitivity study in Sect. 3.3.4 to demonstrate the robustness of the results with respect to other choices of network depth, width and dropout rate. Inference is performed by minimizing the binary cross-entropy logarithmic loss (2), which is equivalent to maximizing the log-likelihood. For each $\beta \in \mathcal{B}_{\text{train}}$, we set a cut-off at the observed $1-\alpha = 95$th percentile in (3) using the associated Gaussian data in training set such that we collect $(\beta_1, q_\alpha(\beta_1))^\top, \ldots, (\beta_{n_{\beta;\text{train}}}, q_\alpha(\beta_{n_{\beta;\text{train}}}))^\top$ and obtain cut-off functions for neural network and linear classifiers from non-parametric kernel regression as shown in Fig. 2.

3.3.1 Type I error comparison

First, we compare the Type I errors for the method in Horváth et al. (2020), the linear and the neural networks

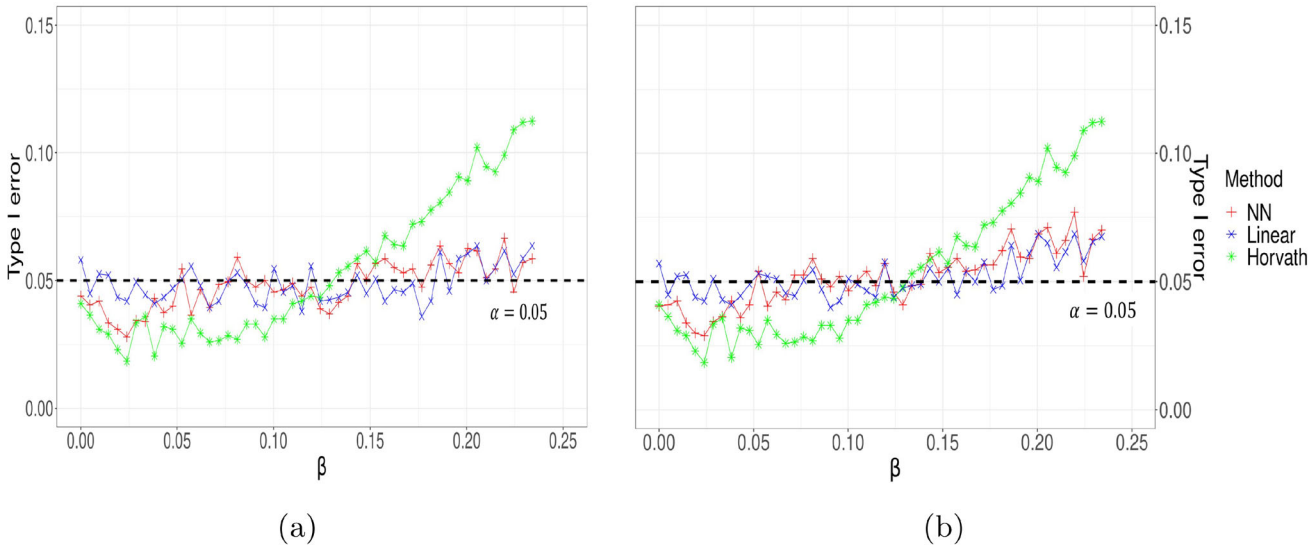


Fig. 3 Panel **a** Type I errors for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) assuming that the parameters $\beta \in \mathcal{B}_{\text{test}}$ are known where $|\mathcal{B}_{\text{test}}| = 50$ and the other two parameters of a Matérn covariance function are fixed, $\sigma^2 = 1$ and $\nu = 0.5$. Averaged standard errors for neural network, linear classifier, and Horváth et al. (2020)'s method are 0.0150, 0.0149, and 0.0152, respectively; Panel **b** Same as (a) when the parameters β on the x-axis and σ^2 are estimated by maximum likelihood estimates and the smoothness parameter is fixed to $\nu = 0.5$. The black dashed horizontal line in each panel represents the 5% Type I error

classifiers assuming that the true βs in $\mathcal{B}_{\text{test}}$ are known, in order to calibrate the testing data points with a suitable cut-off value from the pre-computed kernel regressions. In practice, the true values of β are unknown and require estimation, so in order to assess the Type I errors in a real case, we estimate β and σ^2 simultaneously with fixed $\nu = 0.5$ using the software `ExaGeoStatR` (Abdulah et al. 2023), which allows a unified, high-performance parallel system designed to optimize a covariance-based Gaussian likelihood for spatial data. With the help of advanced high performance dense linear algebra libraries, `ExaGeoStatR` offers exact solutions for calculating the inverse of the covariance matrix and its determinant, which are necessary for evaluating the Gaussian log-likelihood. The optimization step in `ExaGeoStatR` relies on the Bound Optimization BY Quadratic Approximation (BOBYQA) method, which is a numeric, global, derivative-free and bound-constrained optimization algorithm (Powell 2009), such that we can obtain faster and more accurate estimation than brute force methods. Figure 3 illustrates the resulting Type I errors for both the cases of known and unknown parameters. In the first case (known parameters), our adaptive cut-off methods have approximately nominal 5% Type I error rates for all β values (see the red and blue lines in Fig. 3) while Horváth et al. (2020)'s method has unstable Type I error rates as the dependence parameter varies (see the green lines in Fig. 3). In the second scenario (unknown parameters), the outcomes are still comparable to those of known parameters although we utilize estimated βs instead of the true values. Overall, with both non-linear and linear combinations, our adaptive cut-off approach efficiently mitigates the Type I error issue in the testing for dependent data.

We discuss the case where the parameter ν is misspecified in Section A of the supplement. Specifically, we train the linear and neural network models using the data generated with $\nu = 1$, while the actual test data are generated with $\nu = 0.5$, and vice versa. The misspecification of ν significantly worsens the size of tests because the value of $\beta_{\max}$, which controls the size of a test, is computed based on the wrong ν, so incorrect cut-off functions are derived (see Fig. 2 and Figure S1 in the supplement). In Sect. 3.4, we present simulation results when ν is varied in order to further investigate the impact of various smoothness parameters on Type I errors and powers. The results in Sect. 3.4 demonstrate that our adaptive cut-off methods achieve well-controlled Type I errors across all smoothness parameter choices, from rough to smooth, while also providing higher powers.

In real-world scenarios, ν has to be estimated along with the linear models and neural networks. In Sect. 4, we demonstrate how to practically calibrate the tests with an estimated ν.

3.3.2 Power comparison

In order to identify the best test, we need to assess the power under the alternative hypothesis H_1 while maintaining a predetermined Type I error rate α. We compare powers for our proposed neural network model and linear aggregation with

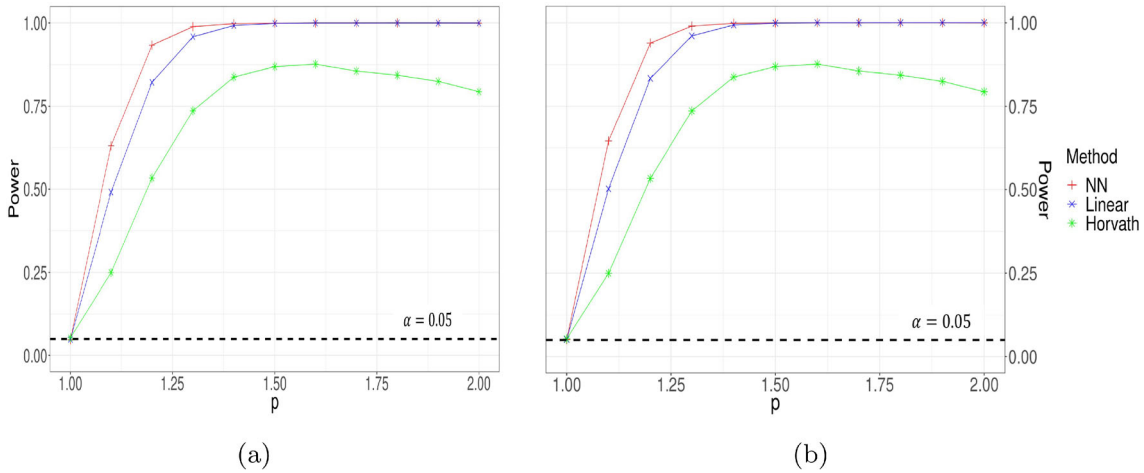


Fig. 4 Panel **a** Averaged powers across all choices of $\beta \in \mathcal{B}_{\text{test}}$ for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) given a value of exponent p (i.e., non-normality parameter) on the x-axis assuming that the parameters $\beta \in \mathcal{B}_{\text{test}}$ are known where $|\mathcal{B}_{\text{test}}| = 50$ and the other two parameters of a Matérn covariance function are fixed, $\sigma^2 = 1$ and $\nu = 0.5$. Averaged standard errors for neural network, linear classifier, and Horváth et al. (2020)'s method are 0.0044, 0.0059, and 0.0232, respectively; Panel **b** Same as (a) when the parameters (β, σ^2) are estimated by maximum likelihood estimates and the smoothness parameter is fixed, $\nu = 0.5$. The black dashed horizontal line in each panel represents the power of 5%

adaptive cut-off, along with the approach in Horváth et al. (2020). Figure 4 shows the power curves as a function of the departure from normality, measured by the exponent p. Each curve is computed as an average across all choices of dependence parameters $\beta \in \mathcal{B}_{\text{test}}$ assuming that they are known (See Panel (a)) or estimated (See Panel (b)).

It is readily apparent that the neural network classifier achieves the highest power for all choices of $p \in \{1.1, 1.2, \ldots, 2.0\}$. Also, our adaptive cut-off method has higher power as the non-normal distribution's tails become heavier (with larger p). Here, neural networks perform only slightly better than linear combinations. The use of only six inputs can be one reason for the slight improvement in this case. It is expected that the accuracy of neural networks would be enhanced if a larger number of inputs are employed.

3.3.3 Time comparison

All experiments were run on a desktop with a 12th Generation Intel(R) Core(TM) i5-12400 2.50 GHz processor and RAM 8.0GB of memory. We checked the runtime of the three methods we used. In the case of the neural network and the linear classifier, the runtime includes the time required to compute the six test statistics used as inputs and the time for learning $W_i, i = 1, \ldots, L$. In the case of Horváth et al. (2020), the runtime includes the duration needed to obtain the test statistic which asymptotically follows a χ^2-distribution with degree of freedom 2. For the 200,000 testing data points, the algorithm of Horváth et al. (2020) takes the longest time, 39.2 min, while our neural network and linear classifier take 18.5 min and 15.6 min, respectively. The algorithm of Horváth et al. (2020) is the slowest because it necessarily uses many `for` loops to compute the kernel estimators.

3.3.4 Sensitivity analysis

We perform a sensitivity analysis with respect to the choice of depth L, width $(n_1, \ldots, n_L)$, and dropout rate of the neural network. First, we consider the same drop-out rate of 0.3 but different number of layers and nodes: 1) three hidden layers with $(n_1, n_2, n_3) = (256, 128, 64)$; 2) two hidden layers with $(n_1, n_2) = (32, 16)$; and 3) one hidden layer with $n_1 = 128$. Second, we use the same number of layers and nodes as in Sect. 3.3 but different drop-out rates, 0.6 or 0.1. Hence, we have a total of six distinct network structures, including the original one, and the results are summarized in Table 1.

We also recompute the Type I error and power in Fig. 3-(a) and Fig. 4-(a) for all models. The results, shown in Fig. 5, show how all six networks display a very similar pattern.

3.4 Varying smoothness parameters

In this subsection, we present the simulation results of varying the smoothness parameter ν in the Matérn covariance in (4) for zero mean Gaussian data with the signed power transformation, $f(z; p) = |z|^p \text{sign}(z)$, for non-Gaussian data. As in the previous subsection, we generate data on a unit square and use the values of p in the sets $\mathcal{P}_{\text{train}} = \{1.2, 1.4, 1.6, 1.8\}$ and $\mathcal{P}_{\text{test}} = \{1.1, 1.2, \ldots, 2.0\}$ for the training data and the testing data, respectively. The purpose of this subsection is

Table 1 Summary of different network architectures: Model 1 is the original network we used in Sect. 3.3.

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
# of layers	2	3	2	1	2	2
# of nodes	(256, 128)	(256, 128, 64)	(32, 16)	(128)	(256, 128)	(256, 128)
Drop-out	0.3	0.3	0.3	0.3	0.6	0.1

The drop-out rate in Models 2, 3, and 4 is identical to that of the original model, however, they differ in their network structures. Models 5 and 6 have modified drop-out rates with the same structure as the original one

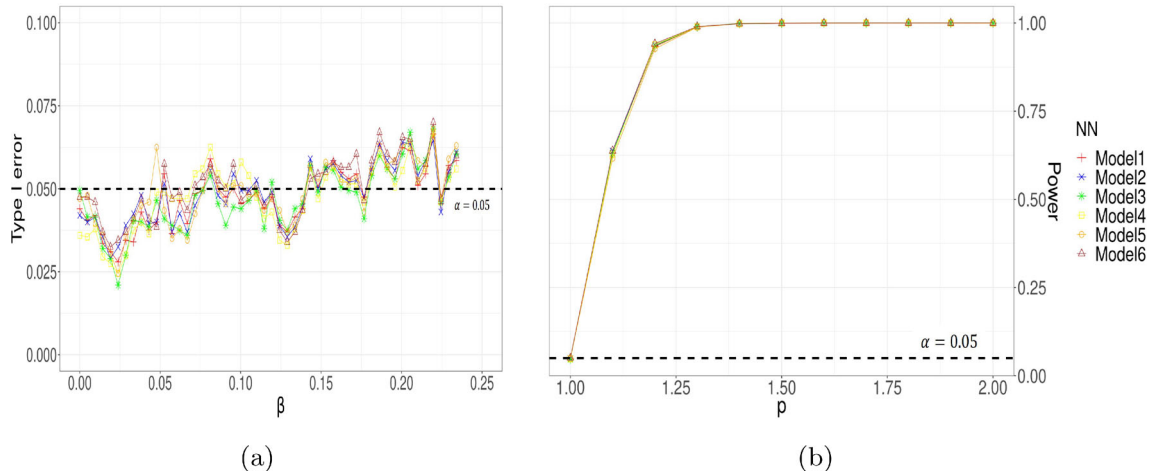


Fig. 5 Panel **a** Type I errors for various architectures of neural networks—Model 1 (red), Model 2 (blue), Model 3 (green), Model 4 (yellow), Model 5 (orange), and Model 6 (brown)—assuming that the parameters $\beta \in \mathcal{B}_{\text{test}}$ are known where $|\mathcal{B}_{\text{test}}| = 50$; Panel **b** Overall powers for various architectures of neural networks computed as an average over all values of $\beta \in \mathcal{B}_{\text{test}}$. For both panels, the other two parameters of the Matérn covariance function are fixed to $\sigma^2 = 1$ and $\nu = 0.5$ and the black dashed horizontal lines represent $y = 0.05$

to examine the impact of changes in the smoothness parameter. Therefore, we fix the range parameter at a moderate value. Results for other values of β are provided in Figure S.5 of the supplement. With a fixed $\beta = 0.1$, we denote the sets of νs for the training data and testing data by $\mathcal{V}_{\text{train}}$ and $\mathcal{V}_{\text{test}}$, respectively. In $\mathcal{V}_{\text{train}} = \{\nu_1, \ldots, \nu_{n_{\nu;\text{train}}}\}$, we choose $n_{\nu;\text{train}} = 30$ equally spaced values of ν including the minimum 0.1 and the maximum $\nu_{\max} = 3.65$, while $n_{\nu;\text{test}} = 50$ equally spaced values are chosen in $\mathcal{V}_{\text{test}}$ including the same endpoints, 0.1 and $\nu_{\max}$, to demonstrate the capability of interpolation. Here, the value of $\nu_{\max}$ is chosen to achieve the corresponding effective range of 0.7. Hence, we consider the interval from 0.1 to $\nu_{\max}$ as sufficiently covering the required smoothness levels, allowing us to concentrate solely on interpolation and omit extrapolation.

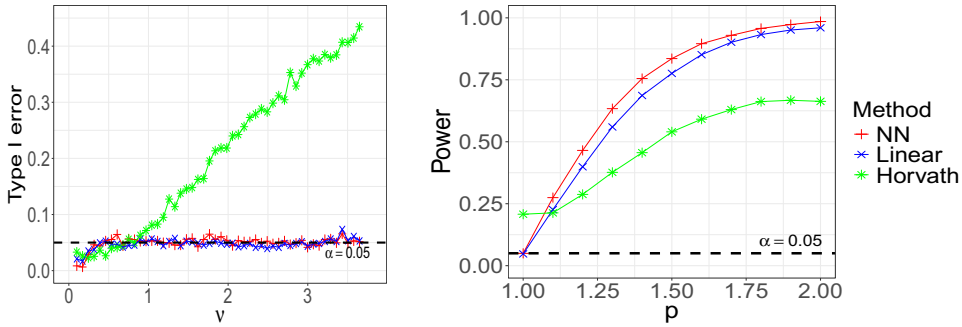
The linear and neural network classifiers are trained using the data generated with $\nu \in \mathcal{V}_{\text{train}}$. We point out that as the value of ν increases, the data exhibit greater dependence due to the higher effective range and view the adaptive cut-offs as a function of ν such that we estimate the cut-off functions using the same kernel smoothing method in (5):

$$\hat{q}_\alpha(\nu) = \frac{\sum_{g=1}^{n_{\nu;\text{train}}} K_h(\nu - \nu_g) q_\alpha(\nu_g)}{\sum_{g=1}^{n_{\nu;\text{train}}} K_h(\nu - \nu_g)}.$$

The resulting Type I errors and powers for the neural network, the linear classifier and the method in Horváth et al. (2020) are shown in Fig. 6. Our adaptive cut-off methods achieve approximately nominal 5% Type I error rates for almost all ν values (see the panel (a) of Fig. 6). In terms of powers, as shown in the panel (b) of Fig. 6, the neural network classifier has the highest performance. For more practical applications, one could extend the adaptive cut-off approach to bivariate functions of β and ν.

3.5 Other models

Besides the Matérn covariance model and signed power transformation, other options can be considered. In Section A of the supplement, we provide the results for several other combinations: three covariance models (Matérn, squared exponential, and spherical covariance) and two non-normal distributions (signed power transformation and multivariate t-distribution). For every option, our adaptive cut-off methods successfully control Type I errors and show increasing



(a) (b)

Fig. 6 Panel **a** Type I errors for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) when the other two parameters of a Matérn covariance function are fixed, $\sigma^2 = 1$ and $\beta = 0.1$; Panel **b** Averaged powers across all choices of $\nu \in \mathcal{V}_{\text{test}}$ for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) given a value of exponent p (i.e., non-normality parameter) on the x-axis when the other two parameters of a Matérn covariance function are fixed, $\sigma^2 = 1$ and $\beta = 0.1$

powers as the deviation from normality grows across scenarios of weak to strong dependence.

4 Testing normality for global climate data

4.1 Motivation

Climate change is bound to affect both natural and human systems, with varying outcomes depending on the region, economic sector, and time. The magnitude and range of future climate does not only rely on the dynamics of the Earth's system but also on scenarios of socio-economic developments (IPCC 2022). Computer models or simulators are the standard tool to understand and quantify future changes in the climate, as well as their social, political and economic effects. The high complexity, spatial and temporal resolution of modern climate models make it impossible to explore future climate for a fully exhaustive range of scenarios, as every simulation puts a considerable strain on the computational and storage resources of an institution's cyberinfrastructures (Huang et al. 2023). As such, sensitivity analysis is limited to a selected set representative of physical parametrizations and scenarios, and uncertainty quantification can be performed partially at best. Statistical surrogates, or emulators (Sacks et al. 1989; Kennedy and O'Hagan 2001) are then routinely trained on a small set of available simulations, and then used to provide a considerably faster (yet approximate) assessment of the behavior of (some variables at some spatio-temporal resolutions of) a climate model (Castruccio and Stein 2013; Castruccio et al. 2014; Castruccio and Genton 2016). A useful simplifying assumption for climate emulation is that of Gaussianity, which at some level of spatial and/or temporal aggregation is more or less explicitly assumed to be valid owing to the central limit theorem. The presence of spatial and temporal dependence within the data, however, makes it challenging to formally assess this assumption. Testing for normality in this framework is therefore of high relevance as it would provide indications as to which modeling strategy would be more appropriate: a Gaussian process emulator (Sacks et al. 1989) or more complex trans-Gaussian (Jeong et al. 2019; Tagle et al. 2020) or latent Gaussian models (Zhang et al. 2024). In this application, we make use of our adaptive cut-off method to assess normality of a widely used collection of climate simulations under different levels of aggregation.

4.2 CMIP6 data

We focus on the data from the Coupled Model Intercomparison Project Phase 6 (CMIP6, Eyring et al. (2016)), the reference collection of simulations (*ensemble*) of the Intergovenmental Panel on Climate Change Assessment Report 6 (Juckes et al. 2020) and in particular on the MIROC-ES2L model (Hajima et al. 2020) given its complete record of simulations. We consider on monthly near surface air temperature data (at 2 ms above the ground level, in Celsius) under SSP245, an intermediate scenario in terms of global mean temperature increase and degree of global socio-economic collaboration throughout the 21st century (Van Vuuren et al. 2014). The data set comprises $T = 12 \times 86 = 1032$ time points (all months in 2015–2100) on a regular $2.79° \times 2.81°$ latitude and longitude grid, for a total of $M = 64 \times 128 = 8192$ locations. We denote the temperature as $Y_t(\mathbf{s}_i)$ at location $i = 1, \ldots, M$ and time point $t = 1, \ldots, T$. Before assessing normality, we provide a model for the trend and the temporal dependence, which need to be removed before applying our proposed methodology.

4.3 Modeling trend and temporal dependence

We consider the following additive spatio-temporal autoregressive moving average (ARMA)-like model:

$$Y_t(\mathbf{s}_i) = \mu_{r(t)}(\mathbf{s}_i) + \epsilon_t(\mathbf{s}_i), \tag{6a}$$

$$\epsilon_t(\mathbf{s}_i) = \sum_{j=1}^{p} \psi_{j;i}\epsilon_{t-j}(\mathbf{s}_i) + \sum_{k=0}^{q} \theta_{k;i}\eta_{t-k}(\mathbf{s}_i). \tag{6b}$$

where $\theta_{0;i} = 1$, $\mu_{r(t)}(\mathbf{s}_i)$ is the monthly trend with indices $r(t) \in \{0, \ldots, 11\}$ representing the remainder when t is divided by 12 and $\eta_t(\mathbf{s}_i)$ is a zero-mean residual uncorrelated in time. Further, we assume that $\text{Var}\{\epsilon_t(\mathbf{s}_i)\} = \sigma^2_{r(t)}(\mathbf{s}_i)$ for $t = 1, \ldots, T$, i.e., there is a month-specific variance. For each location independently, both mean and variance are estimated in a non-parametric fashion with a moving window estimator:

$$\widehat{\mu}_{r(t)}(\mathbf{s}_i) = \frac{1}{|A_{r(t)}|} \sum_{t \in A_{r(t)}} Y_t(\mathbf{s}_i),$$

$$\widehat{\sigma}^2_{r(t)}(\mathbf{s}_i) = \frac{1}{|A_{r(t)}|} \sum_{t \in A_{r(t)}} \left\{Y_t(\mathbf{s}_i) - \widehat{\mu}_{r(t)}(\mathbf{s}_i)\right\}^2,$$

where $A_{r(t)} = \{t : t \bmod 12 = r(t)\}$. The average R^2 across all locations is 0.80 with standard deviation 0.21 and 89% values of R^2 are greater than 0.5, which is better than harmonic regression (performed in the supplementary material). We then remove the trend and variance by computing the standardized residuals as:

$$\widehat{\epsilon}_t(\mathbf{s}_i) = \frac{Y_t(\mathbf{s}_i) - \widehat{\mu}_{r(t)}(\mathbf{s}_i)}{\widehat{\sigma}^2_{r(t)}(\mathbf{s}_i)}.$$

Finally, for each location, we perform inference on the ARMA model (6b) on $\widehat{\epsilon}_t(\mathbf{s}_i)$ using the R package `forecast` (Hyndman and Khandakar 2008), with the orders p and q selected via Bayesian information criterion (BIC). Once the model orders are identified, the model parameters $\psi_{j;i}$ and $\theta_{k;i}$ are estimated by maximum likelihood inference and we use them to compute the residuals $\widehat{\eta}_t(\mathbf{s}_i)$ as estimates of our target quantity $\eta_t(\mathbf{s}_i)$.

Intuitively, the normality assumption for the air temperature data would be violated due to the occurrence of exceptional temperatures at certain locations, resulting in heavier tail probabilities compared to a Gaussian distribution. Hence, it might not be preferable to employ the normality assumption for modeling the original temperature data. In this regard, we are interested in assessing the impact of spatial aggregation on the normality of $\widehat{\eta}_t(\mathbf{s}_i)$. To simplify the notation, we will abuse the notation and use the same expression for the residuals at different levels of spatial aggregation.

4.4 Data aggregation

The emulator residuals $\hat{\eta}_t(\mathbf{s}_i)$ are likely not normal at the native grid resolution, as it is expected that some locations will have unusual temperatures with heavier-than-normal tails. However, some degree of spatial aggregation should result in more normal residuals, and we aim at formally testing this assumption with our proposed approach. We partition the pixels (locations) into smaller squares and compute the mean of the estimated residuals, $\hat{\eta}_t(\mathbf{s}_i)$, within each square. We choose the square of sizes 2×2, 4×4, 8×8, and 16×16 such that the corresponding aggregated data have the number of locations $M = 2048, 512, 128, 32$, respectively. Figure 7 shows the map of the estimated residuals in January 2015 at all four different levels of aggregation.

4.5 Calibration of classifiers

First of all, we simulate the data from a Gaussian distribution using the Matérn covariance in (4) with $\nu \in \mathcal{N}_{\text{train}} = \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$ covering rough to smooth spatial processes, $\sigma^2 = 1$, and $\beta \in \mathcal{B}_{\text{train}} = \{0, \ldots, \beta_{\max}\}$, thereby covering independence to strong dependence and $n_{\beta;\text{train}} = |\mathcal{B}_{\text{train}}| = 30$ as in Sect. 3. The range parameter bound $\beta_{\max}$ depends on the choice of ν and the spatial domain. Since in the case of a unit square we had the effective range of 0.7 corresponding to the strong dependence, for the domain here, we rescale it using the following ratio: effective range/maximum distance = $0.7/\sqrt{2}$, where the maximum distance and the effective range are 12742 km and 6307 km, respectively, in chordal distance for all levels of aggregation. The different values of $\beta_{\max}$ across different choices of the smoothness parameter ν are shown in Table S2 of the supplementary materials. Here, we emphasize that we train six pairs of neural networks and linear classifiers for each value of ν and every testing data point will be assigned to one of the six based on the estimated value of ν.

For non-normal data, the same transformation as in Sect. 3 is used with $p \in \mathcal{P}_{\text{train}} = \{1.2, 1.4, 1.6, 1.8\}$. We draw $n_{\text{sample}} = 200$ sample points for each setup such that we have $n_{\nu;\text{train}} \times n_{\beta;\text{train}} \times n_{p;\text{train}} \times n_{\text{sample}} = 144,000$ non-normal data points and the same amount of normal data points where $n_{\nu;\text{train}} = |\mathcal{N}_{\text{train}}| = 6$. Calibration is performed with the simulated normal and non-normal data and the resulting cut-off functions for each value of $\nu \in \mathcal{N}_{\text{train}}$ are obtained using non-parametric kernel regression as illustrated in Fig. 8.

For the structure of neural networks, the number of hidden layers is $L = 2$ with $n_1 = 256$ and $n_2 = 128$ nodes and we use $m = 5$ inputs among those we used in Sect. 3. We do not use the Shapiro–Wilk test because the number of locations at the original resolution, $M = 8192$, exceeded the maximum allowed by the `R` implementation of the test (see the discussion on the methods about reli-

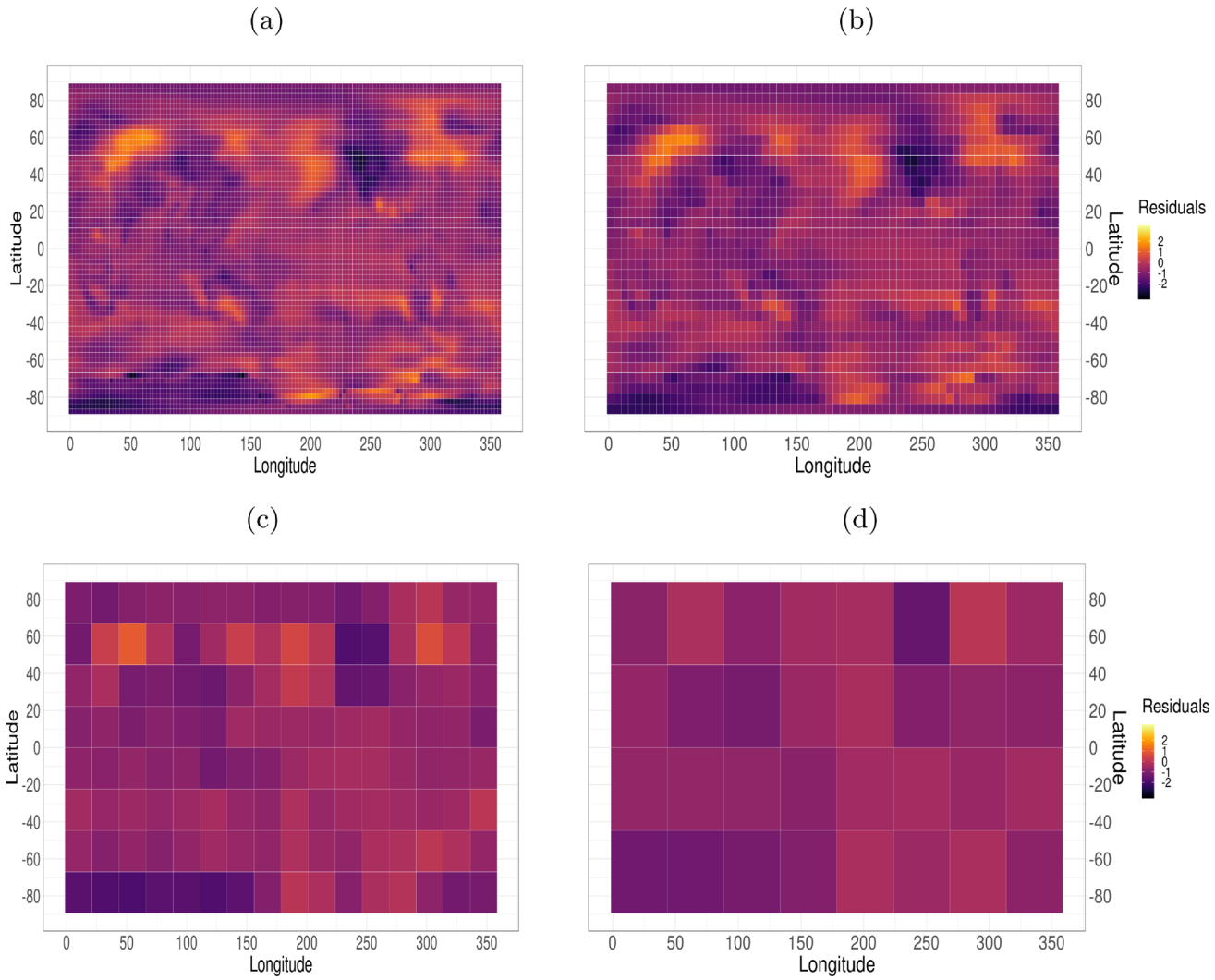


Fig. 7 Standardized emulator residuals $\hat{\eta}_t(\mathbf{s}_i)$ in January 2015 for different levels of spatial aggregation. **a** Original grid resolution; **b** 4 observations in each square of size 2×2; **c** 64 observations in each square of size 8×8; **d** 256 observations in each square of size 16×16

ability of the test for large M in Sect. 2.1). To determine suitable neural network and linear classifiers and corresponding cut-off values for each time point $t = 1, \ldots, T$, we estimate the Matérn parameters (σ^2, β, ν) simultaneously given the location information with chordal distances and the spatial residuals $\left(\hat{\eta}_t(\mathbf{s}_1), \ldots, \hat{\eta}_t(\mathbf{s}_M)\right)^\top$ using the package `ExaGeoStat` (Abdulah et al. 2018) which relies on BOBYQA optimization (Powell 2009). Then, each testing data vector is allocated to a trained neural network and a linear classifier according to the closest approximation of the estimated smoothness parameter. For example, if the estimated smoothness parameter for a data vector is $\hat{\nu} = 0.8$, we use the neural network and linear classifier calibrated with $\nu = 1$, if $\hat{\nu} = 0.3$, we use the neural network and linear classifier calibrated with $\nu = 0.5$.

4.6 Test results

For the different levels of data aggregation, we perform the calibration as detailed in Sect. 4.5 and compute the rejection rates across all time points ($T = 1032$). Here, the hypotheses H_0 and H_1 can be specified in terms of the values and set notations in Sect. 4.5. Formally, for each time point $t \in \{1, 2, \ldots, T\}$, $\nu \in \mathcal{N}_{\text{train}}$, $\beta \in \mathcal{B}_{\text{train}}$, we have:

$$H_0 : p = 1, \qquad H_1 : p \in \{1.2, 1.4, 1.6, 1.8\},$$

The results are shown in Table 2. As expected by the central limit theorem, as the spatial aggregation increases, both the neural network and the linear test highlight that the residuals become more normally distributed. Indeed, at native resolution the normality tests are rejected for more than 95% of time points for both classifiers, while higher levels of aggrega-

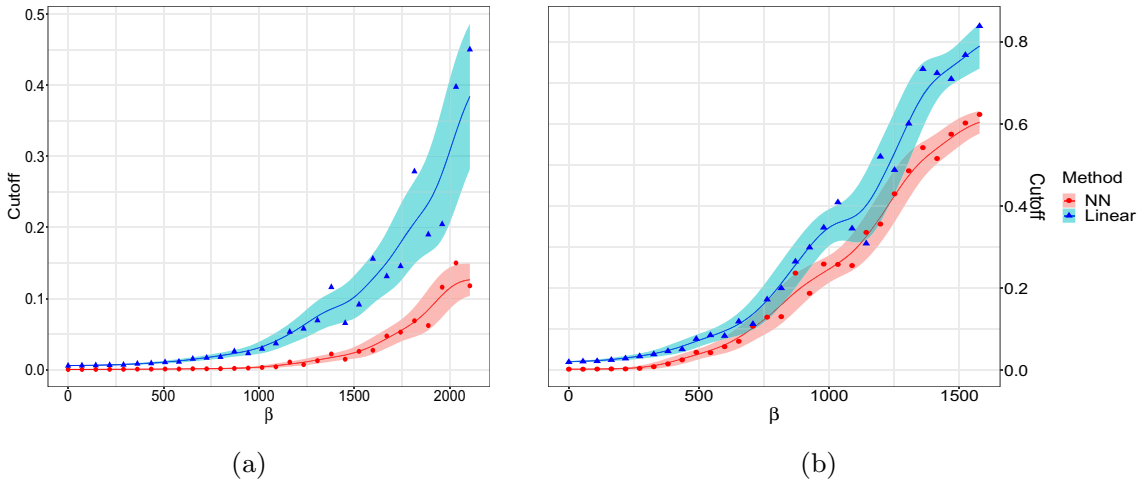


Fig. 8 Application (native grid resolution): non-parametric Gaussian kernel regressions as defined in (5) with a bandwidth $h = 0.3$ for neural network (red) and linear (blue) classifiers. On the x-axis are the range parameter β of the Matérn covariance (4), while on the y-axis the predicted cut-off and corresponding pointwise 95% confidence interval are represented by solid lines and bands, respectively. Since the residuals are normalized, we set $\sigma^2 = 1$, while we have the smoothness parameter equal to **a** $\nu = 0.5$ and **b** in $\nu = 1.0$

Table 2 Rejection rates for the estimated residuals of the emulator (6a) for the neural network and linear normality testing approach.

Rejection rate	All locations ($M = 8192$)	$M = 2048$	$M = 512$	$M = 128$	$M = 32$
NN	0.994	0.967	0.845	0.532	0.227
Linear	0.958	0.924	0.735	0.511	0.191

The results are shown across the different level of spatial aggregation

tion decrease the rejection rates down to approximately 20%. The neural network model is overall less favorable towards the normality assumption, and the discrepancy between the two approaches is slightly higher when the degree of spatial aggregation is moderate ($M = 512$). As we expected, the rejection rate is very high with the original resolution of the temperature data, and interestingly, the rejection rate is still high with the moderate level of aggregation, therefore flagging the normality assumption as generally inappropriate. This can likely be attributed to a large number of time points ($T = 1032$), which result in high power of a normality test against any alternative distribution.

5 Discussion and conclusion

We proposed a new test for dependent data to test Gaussianity by merging the test statistic of individual normality tests (which may or may not assume dependence) via neural networks. By means of a simulation study, we have shown how the proposed approach results in higher power than individual tests as well as a linear aggregation of the tests. Our application for temperature data highlighted how increasing the level of spatial aggregation results in more normal data, as could be expected from the central limit theorem.

The proposed approach has been applied to normality test for dependence data, but its extent is far more general. In fact, other marginal distributions can be tested: a generalized extreme value distribution can be assessed for maxima at different levels of temporal aggregation, or skew-normality for high resolution weather data. Such approach could also be generalized to multivariate data to test marginal univariate properties.

While the proposed approach represents a significant step forward in assessing Gaussianity under dependence, it comes with several caveats that a practitioner must be aware of. Firstly, the method must assume a given structure of spatial dependence, so the reliability of the results are inextricably linked with the assumptions associated with it, most noticeably isotropy and stationarity. While these assumptions may be hard to defend for the original data, the focus on residuals would at least partially justify the spatial structure. Additionally, the proposed method depends on a prespecified type of alternative hypothesis, in this case a non-Gaussian power transformation, and this may or may not be a good alternative hypothesis depending on the application.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11222-024-10551-0.

Acknowledgements We would like to thank Brian Greco for insightful discussions. This publication is based upon work supported by King Abdullah University of Science and Technology Research Funding (KRF) under Award No. ORFS-2022-CRG11-5069.

Author Contributions M.K. performed the analysis. S.C. and M.G. conceptualized the method. All authors wrote the manuscript.

Data Availability The code for this work is available at https://github.com/stat-kim/adaptive-cutoff. The data that support the findings of this study are openly available as part of the Large Ensemble project at the National Center for Atmospheric Research at www.earthsystemgrid.org.

Declarations

Conflict of interest The authors declare no competing interests.

References

Abdulah, S., Ltaief, H., Sun, Y., Genton, M.G., Keyes, D.E.: Exageostat: a high performance unified software for geostatistics on manycore systems. IEEE Trans. Parallel Distrib. Syst. **29**(12), 2771–2784 (2018)

Abdulah, S., Li, Y., Cao, J., Ltaief, H., Keyes, D.E., Genton, M.G., Sun, Y.: Large-scale environmental data science with ExaGeoStatR. Environmetrics **34**(1), 2770–28 (2023)

Anderson, T.W., Darling, D.A.: Asymptotic theory of certain goodness of fit criteria based on stochastic processes. Ann. Math. Stat. **23**(2), 193–212 (1952). https://doi.org/10.1214/aoms/1177729437

Castruccio, S., Genton, M.G.: Compressing an ensemble with statistical models: An algorithm for global 3D spatio-temporal temperature. Technometrics **58**(3), 319–328 (2016)

Castruccio, S., Stein, M.L.: Global space-time models for climate ensembles. Ann. Appl. Stat. **7**(3), 1593–1611 (2013)

Castruccio, S., McInerney, D.J., Stein, M.L., Liu Crouch, F., Jacob, R.L., Moyer, E.J.: Statistical emulation of climate model projections based on precomputed GCM runs. J. Clim. **27**(5), 1829–1844 (2014)

Chen, W., Genton, M.G.: Are you all normal? It depends! Int. Stat. Rev. **91**, 114–139 (2023)

Das, K.R., Imon, A.: A brief review of tests for normality. Am. J. Theor. Appl. Stat. **5**(1), 5–12 (2016)

Edgington, E.S.: An additive method for combining probability values from independent experiments. J. Psychol. **80**(2), 351–363 (1972). https://doi.org/10.1080/00223980.1972.9924813

Eyring, V., Bony, S., Meehl, G.A., Senior, C.A., Stevens, B., Stouffer, R.J., Taylor, K.E.: Overview of the coupled model intercomparison project phase 6 (CMIP6) experimental design and organization. Geosci. Model Dev. **9**(5), 1937–1958 (2016)

Fisher, R.A.: Statistical Methods for Research Workers. Springer, Heidelberg (1992)

Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge, MA (2016)

Hajima, T., Watanabe, M., Yamamoto, A., Tatebe, H., Noguchi, M.A., Abe, M., Ohgaito, R., Ito, A., Yamazaki, D., Okajima, H.: Development of the miroc-es2l earth system model and the evaluation of biogeochemical processes and feedbacks. Geosci. Model Dev. **13**(5), 2197–2244 (2020)

Haynes, W.: Bonferroni correction. Encycl. Syst. Biol., 154–154 (2013)

Horváth, L., Kokoszka, P., Wang, S.: Testing normality of data on a multivariate grid. J. Multivar. Anal. **179**, 104640 (2020)

Huang, H., Castruccio, S., Baker, A.H., Genton, M.G.: Saving storage in climate ensembles: a model-based stochastic approach (with discussion). J. Agric. Biol. Environ. Stat. **28**, 324–344 (2023)

Hyndman, R.J., Khandakar, Y.: Automatic time series forecasting: the forecast package for r. J. Stat. Softw. **27**, 1–22 (2008)

IPCC: IPCC, 2022: Climate Change 2022: impacts, adaptation, and vulnerability. Contribution of working group II to the sixth assessment report of the intergovernmental panel on Cclimate change vol. 9. Cambridge University Press, Cambridge, UK (2022). H.-O. Pörtner, D.C. Roberts, M. Tignor, E.S. Poloczanska, K. Mintenbeck, A. Alegría, M. Craig, S. Langsdorf, S. Löschke, V. Möller, A. Okem, B. Rama (eds.)

Jarque, C.M., Bera, A.K.: Efficient tests for normality, homoscedasticity and serial independence of regression residuals. Econ. Lett. **6**(3), 255–259 (1980)

Jeong, J., Yan, Y., Castruccio, S., Genton, M.G.: A stochastic generator of global monthly wind energy with Tukey g-and-h autoregressive processes. Stat. Sin. **29**, 1105–1126 (2019)

Juckes, M., Taylor, K.E., Durack, P.J., Lawrence, B., Mizielinski, M.S., Pamment, A., Peterschmitt, J.-Y., Rixen, M., Sénési, S.: The cmip6 data request (dreq, version 01.00. 31). Geosci. Model Dev. **13**(1), 201–224 (2020)

Kennedy, M.C., O'Hagan, A.: Bayesian calibration of computer models. J. R. Stat. Soc.: Ser. B (Statistical Methodology) **63**(3), 425–464 (2001)

Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)

Kost, J.T., McDermott, M.P.: Combining dependent p-values. Stat. Probab. Lett. **60**(2), 183–190 (2002)

Li, Q., Racine, J.: Nonparametric estimation of distributions with categorical and continuous data. J. Multivar. Anal. **86**(2), 266–292 (2003)

Li, Q., Lin, J., Racine, J.S.: Optimal bandwidth selection for nonparametric conditional distribution and quantile functions. J. Bus. Econ. Stat. **31**(1), 57–65 (2013)

Lilliefors, H.W.: On the kolmogorov-smirnov test for normality with mean and variance unknown. J. Am. Stat. Assoc. **62**(318), 399–402 (1967)

Powell, M.J.: The bobyqa algorithm for bound constrained optimization without derivatives. Cambridge NA Report NA2009/06, University of Cambridge, Cambridge **26** (2009)

Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. Stat. Sci. **4**(4), 409–423 (1989)

Shapiro, S.S., Wilk, M.B.: An analysis of variance test for normality (complete samples). Biometrika **52**(3–4), 591–611 (1965)

Sigut, J., Piñeiro, J., Estévez, J., Toledo, P.: A neural network approach to normality testing. Intell. Data Anal. **10**(6), 509–519 (2006)

Simić, M.: Testing for normality with neural networks. Neural Comput. Appl. **33**(23), 16279–16313 (2021)

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(56), 1929–1958 (2014)
Stein, M.L.: Interpolation of Spatial Data: Some Theory for Kriging. Springer, Heidelberg (1999)
Tagle, F., Genton, M.G., Yip, A., Mostamandi, S., Stenchikov, G., Castruccio, S.: A high-resolution bilevel skew-t stochastic generator for assessing Saudi Arabia's wind energy resources (with discussion). Environmetrics **31**(7), 2628 (2020)
Thode, H.C.: Testing for Normality, vol. 164. CRC Press, Boca Raton (2002)
Van Vuuren, D.P., Kriegler, E., O'Neill, B.C., Ebi, K.L., Riahi, K., Carter, T.R., Edmonds, J., Hallegatte, S., Kram, T., Mathur, R.: A new scenario framework for climate change research: scenario matrix architecture. Clim. Change **122**, 373–386 (2014)
Vasicek, O.: A test for normality based on sample entropy. J. Roy. Stat. Soc. B **38**(1), 54–59 (1976)
Wilson, P.R., Engel, A.B.: Testing for normality using neural networks. In: Proceedings. First international symposium on uncertainty modeling and analysis, pp. 700–704 (1990). https://doi.org/10.1109/ISUMA.1990.151340
Winkler, A.M., Webster, M.A., Brooks, J.C., Tracey, I., Smith, S.M., Nichols, T.E.: Non-parametric combination and related permutation tests for neuroimaging. Hum. Brain Mapp. **37**(4), 1486–1511 (2016). https://doi.org/10.1002/hbm.23115
Zhang, J., Crippa, P., Genton, M.G., Castruccio, S.: Sensitivity analysis of wind energy resources with Bayesian non-Gaussian and non-stationary functional ANOVA. Ann. Appl. Stat. **18**, 23–41 (2024)