Recognizing a spatial extreme dependence structure: 
A deep learning approach

Manaf Ahmed1,2 | Véronique Maume-Deschamps2 | Pierre Ribereau2

1Department of Statistics and Informatics, University of Mosul, Mosul, Iraq
2Institut Camille Jordan - ICI, Univ Lyon, Université Claude Bernard Lyon 1, CNRS UMR 5208, Villeurbanne, France

Abstract
Understanding the behavior of extreme environmental events is crucial for evaluating economic losses, assessing risks, and providing health care, among many other related aspects. In a spatial context, relevant for environmental events, the dependence structure is extremely important, influencing joint extreme events and extrapolating on them. Thus, recognizing or at least having preliminary information on the patterns of these dependence structures is a valuable knowledge for understanding extreme events. In this study, we address the question of automatic recognition of spatial asymptotic dependence versus asymptotic independence, using a convolutional neural network (CNN). We designed a CNN architecture as an efficient classifier of a dependence structure. Extremal dependence measures are used to train the CNN. We tested our methodology on simulated and real datasets: air temperature data at 2 m over Iraq and rainfall data along the east coast of Australia.

KEYWORDS
convolutional neural networks, extremal dependence, extreme spatial processes

1 INTRODUCTION

Understanding extreme environmental events such as heat waves or heavy rain remains a challenge. The dependence structure is an important element in this field. Multivariate extreme value theory (MEVT) is a good mathematical framework for modeling the dependence structure of extremes events (see for instance De Haan and Ferreira (2007) and Embrechts et al. (2013)). Max-stable processes consist of an extension of multivariate extreme value distributions to spatial processes and provide models for spatial extremes (see De Haan and Pereira (2006) and De Haan (1984)). These max-stable processes are asymptotically dependent (AD), which may not be realistic in practice. Wadsworth and Tawn (2012) introduced inverted max-stable processes that are asymptotically independent (AI). The use of AD versus AI models has important implications on the extrapolation of joint extreme events (see Bortot et al. (2000) and Coles and Pauli (2002)). Thus, recognizing the class of a dependence structure is an important task when building models for environmental data. One of the main challenges is how to recognize the dependence structure pattern of a spatial process. Despite various studies dealing with spatial extremes models, we were unable to find previous studies focusing on the question of the automatic determination of AI versus AD for a spatial process. The usual approach is to use partial maximum likelihood estimations after having chosen (from exploratory graph studies) a class of models. We propose the first deep learning approach for dealing with this question. Although many studies using deep learning for spatial and spatio-temporal processes have been developed, none are concerned with the differences between AD and AI (see Wang et al. (2020)).
Artificial intelligence techniques have demonstrated a significant efficiency in many applications, such as the environment, risk management, and image analysis. Herein, we focus on convolutional neural network (CNN), which has the ability to automatically extract the spatial features hierarchically. Such an approach has been used on the spatial dependencies from raw datasets, as an example. For instance, Zhu et al. (2018) proposed a predictive deep CNN for predicting the wind speed in a spatio-temporal context, where the spatial dependence between locations is captured. In addition, Liu et al. (2016) developed a CNN model for predicting extreme climate events, such as tropical cyclones, atmospheric rivers, and weather fronts. Lin et al. (2018) also presented an approach to forecasting the air quality (PM2.5 construction) in a spatio-temporal framework. Moreover, Zuo et al. (2015) improved the power of object recognition in images by learning the spatial dependencies of these regions using a CNN. In a spatial extreme context, Yu et al. (2016) tried to model the spatial extremes by bridging the gap between traditional statistical methods and graph methods using decision trees.

Our objective is to employ deep learning concepts to recognize patterns of spatially extreme dependence structures and distinguish between AI and AD. Upper and lower tail dependence measures \( \chi \) and \( \vartheta \) are used as a summary of the extreme dependence structure. These dependence measures were introduced by Coles et al. (1999) to quantify the pairwise dependence of extremes between two locations. The definitions and properties of these measures are provided in Section 2. The pairwise empirical versions of these measures are used as a summary dataset. The CNN will be trained to recognize the pattern of dependence structures through this summary dataset.

Owing to the influence of air temperature at 2 m above the surface when assessing climate change on all biotic processes, particularly under extreme conditions, we apply our methods to a case from the European Center for Medium-Range Weather Forecasts (ECMWF). The second case study is the rainfall amount recorded along the east coast of Australia.

The remainder of this article is organized as follows. Section 2 is devoted to the theoretical tools used in this study. An overview of the application of CNNs is provided in Section 3. Section 4 is devoted to configuring the architecture of a CNN for the classification of the dependence structures. Section 5 shows the performance of our designed CNN on simulated data. Applications to environmental data, including air temperature and rainfall events, are presented in Section 6. Finally, a discussion and the main conclusions of this study are given in Section 7.

2 | THEORETICAL TOOLS

Let us now provide a survey on spatial extreme models and tail dependence functions, see Coles et al. (1999) for more details.

2.1 | Spatial extreme models

Let \( \{X_i(s)\}_{s \in S} \), \( S \subset \mathbb{R}^d \), \( d \geq 1 \) be independent and identically distributed (i.i.d) replications of a spatial stationary process. In addition, \( a_n(s) > 0 \) and \( b_n(s) \in \mathbb{R} \), \( n \in \mathbb{N} \) be two continuous functions. If

\[
\left\{ \max_{i=1,...,n} \left( \frac{X_i(s) - b_n(s)}{a_n(s)} \right) \right\} \overset{d}{\to} \{X(s)\}_{s \in S}
\]

as \( n \to \infty \), with non-degenerated marginals, \( \{X(s)\}_{s \in S} \) is a max-stable process. Its marginals are generalized extreme value (GEV) distributions. If for all \( n \in \mathbb{N} \) and \( s \in S \), \( S \subset \mathbb{R}^d \), \( a_n(s) = 1 \) and \( b_n(s) = 0 \), then \( \{X(s)\}_{s \in S} \) is called a simple max-stable process. It has unit Fréchet marginal laws, which means \( \Pr\{X(s) \leq x\} = \exp(-1/x), \ x > 0 \) (see De Haan and Pereira (2006)). In De Haan (1984), it is proved that any simple max-stable process defined on a compact set \( S \subset \mathbb{R}^d \), \( d \geq 1 \) with a continuous sample path admits a spectral representation as follows:

Let \( \{\xi_i, i \geq 1\} \) be an i.i.d Poisson point process on (0, \( \infty \)), with intensity \( d\xi/\xi^2 \), and let \( \{W_i(s)\}_{i \geq 1} \) be i.i.d replicates of a positive random field \( W := \{W(s)\}_{s \in S} \), such that \( \mathbb{E}[W(s)] = 1 \). Then,

\[
X(s) := \max_{i \geq 1} \xi_i W_i(s)^+, \ s \in S, S \subset \mathbb{R}^d, d \geq 1
\]

is a simple max-stable process. The multivariate distribution function is given by

\[
\Pr\{X(s_1) \leq x_1, \ldots, X(s_d) \leq x_d\} = \exp(-V_d(x_1, \ldots, x_d)),
\]

where \( V_d \) is a spectral measure.
where $s_1, \ldots, s_d \subset S$, and $V$ is called the exponent measure, it is homogenous on the order of $-1$, so that $V_d(tx_1, \ldots, tx_d) = t^{-1}V_d(x_1, \ldots, x_d)$, $t > 0$, which can be expressed as

$$V_d(x_1, \ldots, x_d) = E\left[\max_{j=1,\ldots,d} \{W(s_j)/x_j\}\right].$$

The extremal dependence coefficient is given by $\Theta_d = V_d(1, \ldots, 1) \in [1, d]$. It was shown by Schlather and Tawn (2003) that for max-stable processes, either $\Theta_d = 1$, which means that the process is AD, or $\Theta_d = d$, which is the independent case. For max-stable processes, AI implies independence. Wadsworth and Tawn (2012) introduced inverted max-stable processes, which may be AI without being independent. Letting $\{X(s)\}_{s \in S}$ be a simple max-stable process, an inverted max-stable process $Y$ is defined as

$$Y(s) = -1/\log\{1 - \exp(-1/X(s))\}, s \in S.$$  

The inverted max-stable process $\{Y(s)\}_{s \in S}$ has unit Fréchet marginal laws and its multivariate survival function is

$$\mathbb{P}(Y(s_1) > y_1, \ldots, Y(s_d) > y_d) = \exp(-V_d(y_1, \ldots, y_d)).$$

In the definition of the max-stable processes, different models for $W$ lead to different simple and inverted max-stable models. For instance, the Brown–Resnick model is constructed using $W(s) = \{e_i(s) - \gamma(s)\}_{i \geq 1}$, where $e_i(s)$ are i.i.d replicates of a stationary Gaussian process with zero mean and variogram $\gamma(s)$ (see Brown and Resnick (1977) and Kabluchko et al. (2009)). Many other models have been introduced, such as Smith, Schlather, and Extremal-t introduced respectively by Smith (1990), Schlather (2002), and Opitz (2013).

In what follows, we consider extreme Gaussian processes whose marginals have been turned into a unit Fréchet distribution. We shall also consider max-mixture processes, which are $\max(\beta X(s), (1 - \beta)Y(s))$ with $\beta \in [0, 1]$, where $X(s)$ is a max-stable process and $Y(s)$ is an inverted max-stable process or an extreme Gaussian process.

### 2.2 Extremal dependence measures

Consider a stationary spatial process $X := \{X(s)\}_{s \in S}, S \subset \mathbb{R}^d, d \geq 2$. The upper and lower tail dependence functions have been constructed to quantify the strength of AD and AI, respectively. The upper tail dependence coefficient $\chi$ is introduced in Ledford and Tawn (1996) and defined by

$$\chi(h) = \lim_{u \to 1} \mathbb{P}(F(X(s)) > u | F(X(s+h)) > u),$$

where $F$ is the marginal distribution function of $X$. If $\chi(h) = 0$, the pair $(X(s), X(s+h))$ is AI. If $\chi(h) \neq 0$, the pair $(X(s), X(s+h))$ is AD. The process is AI (resp. AD) if $\forall h \in S$ such that $\chi(h) = 0$ (resp. $\forall h \in S$, $\chi(h) \neq 0$).

In Coles et al. (1999), the lower tail dependence coefficient $\overline{\chi}(h)$ is proposed to study the strength of dependence in AI cases. This is defined as

$$\overline{\chi}(h) = \lim_{u \to 1} \left[\frac{2 \log \mathbb{P}(F(X(s)) > u)}{\log \mathbb{P}(F(X(s+h)) > u, F(X(s+h)) > u)} - 1\right], \ 0 \leq u \leq 1.$$  

We have $-1 \leq \overline{\chi}(h) \leq 1$ and the spatial process is AD if $\exists h \in S$ such that $\overline{\chi}(h) = 1$. Otherwise, it is AI.

Of course, working on data requires empirical versions of these extreme dependence measures. We denote these respectively as $\hat{\chi}$ and $\hat{\overline{\chi}}$, which have been defined in Wadsworth and Tawn (2012), see also Bacro et al. (2016). Consider $X_i$, where $i = 1, 2, \ldots, N$ copies of a spatial process $X$, the corresponding empirical versions of $\chi(h)$ and $\overline{\chi}(h)$ are respectively

$$\hat{\chi}_d(s, t) = 2 - \frac{\log \left(N^{-1} \sum_{i=1}^{N} \mathbb{1}_{\{U_i(s) < u, U_i(t) < u\}}\right)}{\log \left(N^{-1} \sum_{i=1}^{N} \mathbb{1}_{\{U_i(s) < u\}}\right)}$$

$$\hat{\overline{\chi}}_d(s, t) = \frac{2 \log \left(N^{-1} \sum_{i=1}^{N} \mathbb{1}_{\{U_i(s) > u, U_i(t) > u\}}\right)}{\log \left(N^{-1} \sum_{i=1}^{N} \mathbb{1}_{\{U_i(s) > u\}}\right)} - 1.$$
and
\[
\hat{X}(s, t) = \frac{2 \log \left( \sum_{i=1}^{N} \mathbb{I}_{\left\{ \hat{U}_i(s) > u \right\}} \right)}{\log \left( \sum_{i=1}^{N} \mathbb{I}_{\left\{ \hat{U}_i(s) > u, \hat{U}_i(t) > u \right\}} \right)} - 1, \tag{10}
\]
where \( \hat{U}_i(\cdot) := \hat{F}(X(\cdot)) \).

3 | CONVOLUTIONAL NEURAL NETWORK

A CNN is an algorithm constructed and perfected as one of the primary branches in deep learning. We shall use this method to recognize the dependence structure in spatial patterns. This stems from two studies introduced by Hubel and Wiesel (1968) and Fukushima (1980). CNNs are used in many domains, including an image analysis. A CNN appears to be relevant for identifying the dependencies between nearby pixels (locations), and it may recognize spatial features (see Wang et al. (2020)). A CNN mainly consists of three basic layers: convolutional, pooling, and fully connected layers. The first two layers are dedicated to feature learning and the latter for classification. The CNN architectures are presented in Yamashita et al. (2018) or Caterini and Chang (2018) for a mathematical framework.

Considering spatial data, a convolutional step is necessary. Indeed, (a) the convolution operation makes the CNN invariant by translation, and (b) the convolution operation on the neighbor locations maintains the spatial hierarchy of the dependencies among the locations. In this section, we present the general scheme of a CNN, whereas Section 4 is oriented more toward our specific application.

3.1 | Convolutional layers

Convolutional layers use a convolution operation instead of a matrix product, similar to classical layers. The convolution operation may be written as follows:
\[
O(s, t) = \Phi \left( \sum_k \sum_{\ell} I(s + k, t + \ell) K(k, \ell) \right),
\]
where \( I \) is the input data or output of the previous layer, and \( O \) is the output of the current layer. In addition, \( K \) is called a kernel and is learned during the training process. Moreover, \( \Phi \) is the activation function, which is chosen as ReLU: \( \Phi(x) = \max(0, x) \).

3.2 | Pooling layers

Applying the convolutional layer many times in a network yields an increase in the number of parameters. Pooling layers are used to reduce the size of the output layers when applying the max operator.

The max-pooling operation takes the max of the input values (of the current layer) \( I(s, t) \) over \( s = s_0, \ldots, s_0 + F - 1, t = t_0, \ldots, t_0 + F - 1 \) and repeats this operation for \( s_0 + S \) and \( t_0 + S \) until all input values have been treated. A common choice is \( F = 2, S = 2 \).

3.3 | Fully connected layers

Fully connected layers are used after the convolution and polling layers, to obtain the final classification. The output of the last pooling layer is transformed into a vector. Then, the output of one fully connected layer is
\[
O(j) = \Phi \left( \sum_i w_{ji} I(i) + b_j \right),
\]
where \( W = (w_{ji}) \) is a weight matrix; \( b = (b_i) \) is called a bias vector; and \( \Phi \) is the activation function, which in this case is an ReLU except for the last fully connected layer, where \( \Phi \) is chosen as a Softmax function:

\[
\text{Softmax}(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \in [0, 1].
\]

The Softmax function gives the probability of belonging to a class. Both \( W \) and \( b \) are learned during the training process.

### 3.4 Training CNN

Training is the process of adjusting values of the kernels \( K \) in the convolutional layers and weights \( W, b \) in fully connected layers using previously classified datasets. The process has two steps, the first of which is the forward propagation and the second is the backpropagation. In forward propagation, the network performance is evaluated by a loss function according to the kernels and weights updated in the previous step. From the value of the loss, the kernel and weights are updated using a gradient descent optimization algorithm. If the difference between the true and predicted class of the dataset is acceptable, the training stops. Thus, selecting the suitable loss function and a gradient descent optimization algorithm is decisive in terms of the quality of the constructed network performance.

Determining the loss or objective function should be conducted according to the network task. Because our goal consists of multi-category classification, we use multi-class cross-entropy loss as an objective function for minimization. Let \( y_a, a = 1, \ldots, A \) be the true class or label of the dataset, and let \( \rho_a \) be the estimated probability of the \( a \)th class, where \( A \) is the number of target classes. The cross-entropy loss function can be formulated as

\[
\mathcal{L} = -\sum_{a=1}^{A} y_a \log(\rho_a). \tag{11}
\]

Minimizing the loss means updating the parameters, that is, updating the kernels and weights until the CNN predicts the correct class. This update is achieved using the gradient descent optimization algorithm during the backpropagation step. Although many gradient algorithms have been proposed in the literature, a CNN, stochastic gradient descent, and Adam algorithm Kingma and Ba (2015) are typically used. The choice of optimization algorithm and its parameters are used as hyper parameters of the network. The learning process is divided into three steps: training, validation, and testing. During the training step, the parameters of the CNN are estimated, whereas the validation step is devoted to an optimization of the hyper-parameters. The testing step allows an evaluation of the final model. To conduct these three steps, the dataset is divided into three sub-datasets. This split is done randomly. The test sub-dataset is not used during the learning process and provides the loss and accuracy of the final model. The accuracy is defined as the ratio between the number of correct classes predicted by the CNN to the total number of predictions.

### 4 CONFIGURATION OF CNN FOR CLASSIFYING THE TYPES OF DEPENDENCE STRUCTURES

Our goal is to provide tools to recognize spatial dependence structures, namely to distinguish between asymptotic independence and asymptotic dependence. Environmental datasets come from measures at different locations, and thus we shall design a CNN depending on the locations where the measures are applied (see Chattopadhyay et al. (2020)).

#### 4.1 Extracting the dependence structure from data

Using the raw spatial data directly to recognize the dependence structure as the input of a CNN may not be a good idea because of the data size, which would make the computation cost extremely important. In addition, spatial data carry substantial amounts of informations, and thus, a CNN may mislead the target in recognizing the spatial dependence structure. For this reason, we propose the use of dependence measures as summary data. This summary contains the dependence structure information and reduces the size of the input. Furthermore, pairwise dependence measures are
used to construct raster data on a regular grid. A CNN is extremely effective with this type of data (see Uselis et al. (2020)). Well known measures able to capture the dependence structures include upper and lower dependence measures, $\chi$ and $\tilde{\chi}$, defined by Equations (7) and (8), respectively. The asymptotic independence is achieved if $\chi = 0$, and thus this measure is able to capture the dependence structure for asymptotic dependence models and fails to provide information in the case of asymptotic independence (see Simpson et al. (2020)). By contrast, the lower tail measure $\tilde{\chi}$ provides the dependence strength of the asymptotic independent models.

We propose the use of $\chi$ and $\tilde{\chi}$ as a learning dataset for a CNN, because the asymptotic structure is summarized through these two measures. Their pairwise empirical counterparts $\hat{\chi}(s, t)$ and $\hat{\tilde{\chi}}(s, t)$, $(s, t) \in S, S \subset \mathbb{R}^d$ defined in Equations (9) and (10) respectively, are computed above a threshold $u$ and constitute a regular raster dataset with a symmetric array in two tensors. The choice of $u = 0.975$ is a common value in extreme value theory, and we also apply this value. Figure 1 shows an array constructed from Brown–Resnick and inverse Brown–Resnick spatial models over 30 spatial locations.

4.2 Building CNN architecture

It is essential in a CNN design to consider the type of data and task to be applied, that is, classification, representations, or anomaly detection. For instance, if the application consists of extracting the temporal dependence features of a site, the data should be considered as a sequence (time series), and 1D CNN or a recurrent neural network (RNN) can be used. We are concerned with extracting the spatial dependencies for all sites simultaneously, and thus, building a 2D CNN is quite reasonable for this scenario. For extracting the spatio-temporal dependencies, one can use a hybrid model combining an RNN and a CNN, which is extremely effective (see Wang et al. (2020)).

In practical terms, the design of a CNN for the classification of complex patterns remains a challenge. First, one has to determine the number of convolutional and fully connected layers that should be used. Second, the tuning of a high number of parameters (kernel, weights) is required. Many articles are devoted to building and improving CNN architectures to achieve a good performance, that is, LeCun et al. (1990), Krizhevsky et al. (2012), He et al. (2016), and Xie et al. (2017). Although these CNN architectures are appropriate for image classification, they are not adequate for our goal because we need to maintain the dependence structure.

**FIGURE 1** Arrays representing dependence structures summarizing two datasets: (a) Brown–Resnick asymptotic dependent model; (b) inverted Brown–Resnick asymptotic independent model. According to these arrays, the training, validation, and testing of a CNN will be applied. The two models are generated from isotropic semivariogram $\gamma(s, t) = (\|s - t\|/\sigma)^\delta$, $(s, t) = 1, \ldots, 30$ with scale and smoothness parameters of $\sigma = 0.4$ and $\delta = 0.7$, respectively. The two tensors of each array are the empirical upper and lower tail dependence measures $\hat{\chi}_{0.975}(s, t)$ and $\hat{\tilde{\chi}}_{0.975}(s, t)$, respectively. The contour lines in the tensors represent the hierarchical aspect of the dependence strength among the different locations.
We will design a CNN to recognize the asymptotic dependence structure. The first layer (input layer) in a CNN is configured according to the dimensions of the array of the summarized dataset: $x_1 = [n_1, n_1, \ell_2]$, where $(n_1 \times n_1)$ represents the number of pairwise locations and the subscript of 2 indicates the two measures $\hat{r}_{0.975}(s,t)$ and $\hat{\gamma}_{0.975}(s,t)$, $(s,t) = 1, \ldots, n_1$. Now, we have a classification task, that is, recognizing asymptotic dependence versus an asymptotic independence structure (two classes). We may add a third class to detect if a spatial process is neither AD nor AI. This third class is considered an unknown dependence structure type.

There are no rules for determining the number of hidden layers in a network. Therefore, adjusting the number of hidden layers in the model is achieved through trial and error. An overly simple model may yield an underfitting, whereas an excessively complex model could lead to an overfitting. We found that we require no less than 17 million parameters. For the air temperature at 2 m above land in Iraq and rainfall along the east coast of Australia, as discussed in Section 6, the number of locations used is $n_1 = 30$ and $n_1 = 40$, respectively. This means we have two different input arrays for the CNN: The first array $x_1 = [30, 30, 2]$ represents the dependence structure of the air temperature event, and the second array $x_1 = [40, 40, 2]$ is that for the rainfall data. In this last case, the number of weights reaches 45 million parameters.

To fix these ideas, we present the architecture and compilation framework of a 2-class CNN with an input layer of $x_1 = [30, 30, 2]$. Table 1 shows the number and type of layers configured in this case.

We use the cross-entropy loss function and Adam optimization algorithm, which is known to be extremely efficient, with a CNN (see Kingma and Ba (2015)). The package Keras in R interface is used for learning the CNN. We validate the model using the procedure described in Section 3.4. Figure 2 shows the general framework of the designed CNN architecture, illustrating the array transition through the network.

### Table 1 Architecture of 2-class CNN with input layer $x_1 = [30, 30, 2]$ to investigate the dependence structure type

| Order and type of layers | Number of output feature maps | Size of kernels $(\Delta x \times \Delta x)$ | Stride size $(\Delta x \times \Delta x)$ | Padding | Activation |
|--------------------------|-------------------------------|------------------------------------------|---------------------------------------|---------|------------|
| Input                    | -                             | -                                        | -                                     | -       | -          |
| $v = 1$, 2D-convolutional| 64                            | $(p_1 \times q_1) = 3 \times 3$          | $2 \times 2$                          | Valid   | ReLU       |
| $v = 2$, 2D-max pooling  | -                             | $(r_2 \times r_2) = 2 \times 2$          | $1 \times 1$                          | Valid   | -          |
| $v = 3$, 2D-convolutional| 128                           | $(p_3 \times q_3) = 3 \times 3$          | $1 \times 1$                          | Valid   | ReLU       |
| $v = 4$, 2D-convolutional| 256                           | $(p_4 \times q_4) = 3 \times 3$          | $1 \times 1$                          | Valid   | ReLU       |
| $v = 5$, 2D-max pooling  | -                             | $(r_5 \times r_5) = 2 \times 2$          | $1 \times 1$                          | Valid   | -          |
| $v = 6$, Fully connected | 1024                          | -                                        | -                                     | -       | ReLU       |
| $v = 7$, Fully connected | 512                           | -                                        | -                                     | -       | ReLU       |
| $v = 8$, Output          | 2                             | -                                        | -                                     | -       | Softmax    |

**Note:** The same architecture for a 3-class CNN is also proposed. In this case, the last fully connected layer has three units rather than two.
5 | PERFORMANCE EVALUATION OF A CNN THROUGH A SIMULATION

To evaluate the performance of the CNN described in the previous section, three scenarios were applied. For each scenario, the 2- and 3-class networks were trained on the AD and AI processes, and for the 3-class networks, the max-mixture processes (see the definition in Section 2.1) are added to the training data. Our training data consist of the following:

- Max-stable processes (defined in Equation (2)) with 1000 observations at locations \( s_j, j = 1, \ldots, 30 \) in 60,000 datasets are generated from four spatial extreme models, that is, Smith, Schather, Brown–Resnick, and Extremal-t, with scale and smoothness parameters \( \sigma \) and \( \delta \), respectively. These parameters are either chosen at random or in regular sequences.
- Inverse max-stable processes, as defined in (5) using the same parameters as above and 5000 datasets, are generated from extreme Gaussian processes to achieve a greater variety of AI models.

In total, for the two dependence structure types, 125,000 datasets are generated and divided into three parts, 64% for training, 16% for validation, and 20% for testing. The empirical dependence measures \( \hat{\chi}_{0.975}(s, t) \) and \( \hat{\chi}_{0.975}(s, t) \) with \( (s, t) = (s_i, t_i) \in [0, 1]^2 \times [0, 1]^2, i = 1, \ldots, 30 \) are used to summarize the datasets and are the inputs for training the CNN. For the 3-class network, we added 12,000 datasets with neither an AD nor an AI dependence structure, through the max-mixture processes. We applied several scenarios:

- Under the first scenario, for each dataset, the locations \( s \in [0, 1]^2 \) are uniformly and randomly chosen. Moreover, the scale and smoothness parameters of the models are also uniformly and randomly selected: \( \sigma \sim U(0, 1) \) and \( \delta \sim U(1, 1.9) \). In the 3-class network, the mixing parameter \( \beta \) is also uniformly and randomly selected: \( \beta \sim U(0, 1) \). The AD and AI models in the max-mixture are also chosen at random in the different classes.
- Under the second scenario, the locations were fixed for all datasets, and the parameters were chosen at random.
- Under the third scenario, the locations are fixed for all datasets and the parameters run through regular sequences, \( \sigma \in [0.1, 1] \) and \( \delta \in [0.1, 1.9] \), with steps of 0.2, and the mixing parameter \( \beta \in [0.3, 0.7] \) with steps of 0.1.

An evaluation was conducted for the three scenarios described above. The AD or AI datasets were used for the 2-class networks and we added max-mixture processes for the 3-class networks. For the random scenarios, the evaluation dataset locations \( s \) and parameters were chosen at random. For the fixed location scenario, the evaluation datasets sites were chosen differently from the training datasets. For scenario 3, the scale and smoothness parameters are chosen differently for the evaluation and training. The loss \( \mathcal{L} \) and accuracy \( A \) are computed and presented in Table 2 for the various scenarios described above.

The first three rows of Table 2 show the training, validation, and testing losses. We can conclude that both 2- and 3-class networks perform well for the three scenarios. The generalization is better with the third scenario; in addition, when the locations are fixed or the parameters are chosen sequentially, or both \( \hat{\chi} \) and \( \hat{\chi} \) have specific dependence patterns, it is easier to distinguish between the dependence structures compared with the first and second scenarios. The performance of the networks can also be examined specifically for different dependence structures. Table 2 shows that asymptotic independence structure (inverse max-stable or extreme Gaussian) is recognized almost perfectly by all networks. The performance in recognizing the asymptotic dependent structures is less satisfactory. This is because there is relatively no difference in the dependence strength levels between \( \hat{\chi} \) and \( \hat{\chi} \) for asymptotic dependence models, whereas the differences for asymptotic independence models are high. This can be seen in Figure 1, where the contour levels in for the dependence structure of the Brown–Resnick model in the two tensors seem comparable, whereas that for the inverse Brown–Resnick model are different. The best results are obtained for Scenario 3, for both 2- and 3-class networks. The mixed dependence structure may be recognized by the 3-class networks, that is, it was not distinguished under the second scenario, whereas the two other scenarios provide acceptable results. For the different location tests, the networks trained under scenario 1 overcame those trained under scenario 3. This is because the networks trained using random \( \sigma \) and \( \delta \) parameters led to different dependence structures. In other words, no specific dependence structures were learned, contrary to the third scenario. The performances were improved by training the networks with datasets whose parameters cover the parameter space. Finally, scenario 3 achieved a good performance, even for datasets with untrained scale and smoothness parameters. These observations lead us to use the third scenario in our application studies: the air temperature 2 m over Iraq and the rainfall over the eastern coast of Australia.
TABLE 2  Loss and accuracy of the designed 2- and 3-class CNN networks trained under the three scenarios

| Scenario 1 fixed locations and random parameters | Scenario 2 random locations and random parameters | Scenario 3 fixed locations and sequential parameters |
|-----------------------------------------------|-----------------------------------------------|-----------------------------------------------|
| 2-class network                               | 2-class network                               | 2-class network                               |
| 3-class network                               | 3-class network                               | 3-class network                               |
| Loss  | Accuracy  | Loss  | Accuracy  | Loss  | Accuracy  | Loss  | Accuracy  | Loss  | Accuracy  | Loss  | Accuracy  |
| Training       | 0.3255  | 0.8668 | 0.6829  | 0.7232  | 0.3649  | 0.8386  | 0.3740  | 0.8749  | 0.2985  | 0.8824  | 0.4729  | 0.8242  |
| Validation     | 0.3525  | 0.8564 | 0.7071  | 0.7108  | 0.3675  | 0.8406  | 0.4763  | 0.8411  | 0.3285  | 0.8658  | 0.5204  | 0.8018  |
| Testing        | 0.3582  | 0.8547 | 0.7080  | 0.7146  | 0.3694  | 0.8400  | 0.5165  | 0.8039  | 0.3275  | 0.8672  | 0.5169  | 0.8041  |
| Gaussian       | 0.0700  | 0.9969 | 0.0928  | 1.0000  | 0.2851  | 0.9550  | 0.3868  | 0.9540  | 0.0658  | 0.9980  | 0.0655  | 0.9980  |
| Asymptotic dependent | 0.4085  | 0.7880 | 0.7052  | 0.6870  | 0.4796  | 0.7050  | 0.6378  | 0.6660  | 0.3772  | 0.7940  | 0.5493  | 0.7480  |
| Asymptotic independent | 0.2889  | 0.9390 | 0.5124  | 0.8970  | 0.2753  | 0.9610  | 0.4621  | 0.9120  | 0.3284  | 0.9020  | 0.3317  | 0.9240  |
| Mixtures       | -       | -      | 0.7437  | 0.6510  | -       | -      | 6.0335  | 0.1370  | -       | -      | 0.8299  | 0.6633  |
| Different locations | 0.779   | 0.8060 | 0.8888  | 0.8000  | -       | -      | -       | -      | 0.9536  | 0.8010  | 1.1231  | 0.7990  |
| Different scale parameters | -       | -      | -       | -      | -       | -      | -       | -      | 0.3908  | 0.8480  | 0.5024  | 0.8100  |
| Different smooth parameters | -       | -      | -       | -      | -       | -      | -       | -      | 0.3295  | 0.8744  | 0.4361  | 0.8355  |

The general performances of the scenarios are illustrated in Figure 3 and the training progress of all networks is correct, the training and validation losses decrease, no underfitting or overfitting is observed, and the procedure is stable. The network training will stop when there is no more improvement to the validation loss. Although the two networks (2- and 3-class) perform well, the network for 2 classes slightly outperforms the network with 3 classes. The reason for this slight loss of performance is probably derived from the fact that training using the AD, AI, and max-mixture processes leads to a slight fuzziness in the learned dependence structure.

6 | APPLICATION TO ENVIRONMENTAL CASE STUDIES

Modeling the spatial extreme dependence structure on environmental data is the initial purpose of this study. We finish this article with two specific studies on the Iraqi air temperature and Australian coastal rainfall.

6.1 | Spatial dependence pattern of the air temperature at 2 m above Iraq

The temperature of the air at 2 m above the surface has a major influence on assessing climate change as well as on all biotic processes. Such data inherently indicate a spatio-temporal process (see Hooker et al. (2018)).

6.1.1 | Data

We used data produced through the meteorological reanalysis ERA5 and achieved by the European Center for Medium-Range Weather Forecasts (ECMWF). An overview and quality assessment of the data can be found at http://dx.doi.org/10.24381/cds.adbb2d47. Our objective is to study the spatial dependence structure pattern of the data recorded from a high-temperature region in Iraq. Let \( X_k(S) \) be the daily average of the air temperature process at a spatio-temporal point \( (s, k) \) computed at 2 m during peak hours of 11:00 to 17:00 for the period of 1979–2019 during the summer (June, July, and August). This collection of data results in \( \mathcal{K} = 3772 \) temporal replications and \( |S| = 1845 \) grid cells. The data have a naturally spatio-temporal nature. Nevertheless, a preprocessing detailed in Section 6.1.2 suggests they be treated as independent replications of a stationary spatial process. The left panel in
loss vs. epochs for training and validation of designed 2- and 3-class CNNs for each scenario. Each row represents the progress for scenarios 1–3, respectively. The columns represent the process of the 2- and 3-class CNNs, respectively.

Figure 3 shows losses during training and validation of the designed 2- and 3-class CNNs for each scenario. Each row represents the progress for scenarios 1–3, respectively. The columns represent the process of the 2- and 3-class CNNs, respectively.

Figure 4 shows process $X$ as time series for three locations located in the northern, central, and southern areas of Iraq (white triangles on the right panel). The right panel shows the temporal mean.

Regarding the time series in the left panel, the data from the three locations may be considered as stationary in time. To remove the spatial non-stationarity, we apply a simple moving average, as used by Huser (2021), see Section 6.1.2.

### 6.1.2 Preprocessing of 2-m air temperature data

As mentioned above, the 2-m summer air temperature data in Iraq appear to be spatially non-stationary. We propose following the approach by Huser (2021) to remove the non-stationarity. We shall decompose the spatial process \( \{X(s)\}_{s \in S} \) into two terms, the average part \( \mu(s) \) and the residual part \( R(s) \), such that

\[
X(s) = \mu(s) + R(s). \tag{12}
\]

Smoothing the empirical estimation of \( \mu_k(s) \) by a moving average of over 10 days leads to

\[
\hat{R}_k(s) = X_k(s) - \hat{\mu}_k(s), s \in S, k \in K.
\]

Figure 5 shows the spatial variability for August 15, 2019. The non-stationarity of \( (X_k(s))_{s \in S} \) can be seen, whereas the residuals \( \hat{R}_k(s) \) seem stationary (right panel).

In model (12), the residual process carries a dependence structure, the isotropy of which is described below. Figure 6 shows the estimated tail dependence functions with respect to certain directions (where 0 is the northern direction). From this graphical study, we may retain the isotropic hypothesis.

We estimated the extremal dependence structure using the block maxima approach. This approach is more appropriate than considering the hourly observations as i.i.d. For more details see Naveau et al. (2009) and Ferreira and
FIGURE 4  (Left panel) Gray lines represent the time series of the daily average of the 2-m air temperature for the period of 1979–2019 during the summer months (June, July, and August). The red lines represent the simple 10-day moving average. The smoothing temporal data are shown in blue line. The contour plot in the right panel shows the gradient level in the mean of $X$ for the entire period above the Iraq land

FIGURE 5  Air temperature $X_k(s)$ 2 m over Iraq on August 15, 2019 is shown in the left panel, whereas the estimated residual process $\hat{R}_k(s)$ is shown in the right panel. The black dots are the locations chosen to construct the air temperature dependence structure

De Haan (2015). Let $m \in \mathcal{K}$ and

$$B_{m,k}(s) = \{(s,k^c) : (k - m) \leq k^c \leq (k + m)\} \cap (S \times \mathcal{K})$$

be a temporal neighborhood set of $k$ for each grid cell $s$, the spatial block maxima process is defined as

$$Y_k(s) = \max_{(s,k^c) \in B_{m,k}(s)} \hat{R}_{k^c}(s).$$
Empirical tail dependence measures $\hat{\chi}_{0.975}(h)$ and $\hat{\chi}_{0.975}(h)$, for each direction. The red line is for the direction $(-\pi/8, \pi/8]$, blue indicates $(\pi/8, 3\pi/8]$, green represents $(3\pi/8, 5\pi/8]$, and black indicates $(5\pi/8, 7\pi/8]$, where $h = \|s - t\|, s, t \in S$. The gray dots represent the pairwise $\hat{\chi}_{0.975}(s, t)$ and $\hat{\chi}_{0.975}(s, t)$. The blue indicates $(\pi/8, 3\pi/8]$ , green represents $(3\pi/8, 5\pi/8]$ and black indicates $(5\pi/8, 7\pi/8]$ where $h = \|s - t\|, s, t \in S$.

Loss of training and validation recorded for each of the 14 epochs during the training progress.

Then, according to $Y_k(s)$, the dependence structure of the air temperature is estimated, using $\hat{\chi}_{0.975}(s, t)$ and $\hat{\chi}_{0.975}(s, t)$, $s, t = 1, \ldots, 30, (s, t) \in S$. To transform the margin into the unit Fréchet, we consider the rank transformation applied to $Y$,

$$\hat{Y}_k(s) = \begin{cases} -1/\log(\text{Rank}(Y_k(s))/(|B_{m,k}(s)| + 1)) & \text{if } \text{Rank}(Y_k(s)) > 0 \\ 0 & \text{if } \text{Rank}(Y_k(s)) = 0, \end{cases}$$

where $k = 1, \ldots, |B_{m,k}(s)|, s = 1, \ldots, 30$, and $| \cdot |$ is the cardinality. We therefore obtain a $[30 \times 30 \times 2]$ table consisting of CNN inputs.

### 6.1.3 Training designed CNN

We shall now use the CNN procedure described in Sections 4 and 5, and we consider the locations of the data according to scenario 3. We resize the data into $[0, 1]^2$. The training datasets are generated as shown in scenario 3. For the parameters, we use regular sequences with steps of 0.1. Figure 7, shows the loss of the training progress for the designed 2-class CNN network. The performance of the 3-class network is comparable.
As we mentioned previously, the CNN will stop the training when the validation loss reaches the minimum. At epoch 14, the training and validation losses recorded were 0.2282 and 0.2427, respectively, with an accuracy of 0.9333 and 0.9321, respectively. For the testing data, the loss was 0.2512 and the accuracy was 0.9260. This shows that the training process worked well.

6.1.4 Predicting the dependence structure class for summer air temperature at 2 m

The pattern prediction of the air temperature dependence structure is applied using 7 sizes of block maxima, $m = 192, 30, 15, 7, 5, 3, 1$, which respectively give 41, 125, 251, 538, 754, 1257 blocks from $K = 3772$ measurements, and thus we consider the influence of the block size on the predicted class. Table 3 shows the predicted pattern of the dependence structure of the air temperature at 2 m corresponding to the size of each block maxima proposed. For all block sizes, the predicted pattern showed asymptotic dependence, with no significant effect of the block size on the probability of prediction for both 2- and 3-class CNNs. Thus, we can conclude that the air summer temperature at 2 m has an asymptotic dependent spatial structure.

6.2 Rainfall dataset: Case study in Australia

Another dependence structure investigated in this article is the daily rainfall data recorded at 40 monitoring stations located in eastern Australia, as illustrated in Figure 8 by the red dots.

This dataset has been studied by several authors, that is, Bacro et al. (2016); Ahmed et al. (2017); and Abu-Awwad et al. (2020).

6.2.1 Data

For each location, the cumulative rainfall amount (in millimeters) over a 24-h period is recorded for the years 1972–2019 during the extended rainfall season (April-September). This results in $|\mathcal{K}| = 8784$ observations. To ensure the spatial stationarity, many monitoring locations were selected, as it may be seen on Figure 8 with the elevations maintained above the mean sea level at between 2 to 540 m. The data are available for free on the website of the Australian Meteorology Bureau http://www.bom.gov.au. The spatial stationarity and isotropic properties of these data have been investigated in many different papers; for instance, see Bacro et al. (2016), Ahmed et al. (2017), and Abu-Awwad et al. (2020). We consider the data to be stationary and isotropic. This leads to a corresponding dependence structure constructed directly from the data themselves without having to estimate the residuals, as described in the previous section. Let $\{X_k(s)\}_{s \in \mathcal{S}, k \in \mathcal{K}}, s = 1, \ldots, 40, k = 1, \ldots, 8784$ be the spatial process representing the rainfall along the eastern coast of Australia. Adopting

| TABLE 3 Predicted class and its probability for the air temperature data at 2 m for each block maxima size proposed |
|---------------------------------------------------------------|
| Block maxima size | 2-class CNN | 3-class CNN |  |
|                  | Probability of AD | Probability of AI | Probability of AD | Probability of AI | Probability of AD/AI mixture |
| $m = 92$ days    | 1.000 | 0.000 | 1.000 | 0.000 | 0.000 |
| $m = 30$ days    | 1.000 | 0.000 | 1.000 | 0.000 | 0.000 |
| $m = 15$ days    | 0.990 | 0.001 | 0.645 | 0.355 | 0.000 |
| $m = 7$ days     | 0.860 | 0.140 | 0.791 | 0.199 | 0.010 |
| $m = 5$ days     | 0.929 | 0.071 | 0.686 | 0.013 | 0.301 |
| $m = 3$ days     | 0.864 | 0.136 | 0.702 | 0.085 | 0.213 |
| $m = 1$ day      | 0.995 | 0.005 | 0.950 | 0.037 | 0.013 |

Note: The dependence structure is validated using the two CNNs. AD and AI refer to the asymptotic dependence and asymptotic independence, respectively.
the block maxima size as described in the previous section, we consider the following extreme process

\[ Y_k(s) = \max_{(s,k^*) \in \mathcal{B}_u(s)} X_{k^*}(s), \]

and transform \( Y \) into a unit Fréchet marginal process. The dependence structure of these data are summarized in an array with a size of \( 40 \times 40 \times 2 \). The first and second tensors are \( \hat{Y}_{0.975}(s,t) \) and \( \hat{Y}_{0.975}(s,t) \), \( s, t = 1, \ldots, 40 \), respectively, with a threshold of \( u = 0.975 \).

6.2.2 Predicting the dependence structure pattern of rainfall amount in eastern Austria.

We use the same designed CNN as described in the previous section. The training and validation are shown in Figure 9. The validation loss reached the minimum at epoch 16, and thus the final performance of the CNN will be calculated at this epoch. The recorded training and validation losses are 0.307 and 0.336, respectively. The accuracies are 0.889 and 0.875. The loss and accuracy of the tested data were 0.342 and 0.870, respectively. Table 4 shows the predicted class for each proposed block maxima size.
TABLE 4  Predicted class of rainfall data for each block maxima size proposed

| Block maxima size | 2-class CNN |                     |                      |                      |                     |
|-------------------|-------------|---------------------|----------------------|----------------------|---------------------|
|                   | Probability of AD | Probability of AI | Probability of AD | Probability of AI | Probability of AD/AI mixture |
| $m = 183$ days    | 0.020       | 0.980               | 0.020               | 0.980               | 0.000                |
| $m = 30$ days     | 0.020       | 0.980               | 0.020               | 0.980               | 0.000                |
| $m = 15$ days     | 0.060       | 0.940               | 0.063               | 0.937               | 0.000                |
| $m = 10$ days     | 0.271       | 0.729               | 0.143               | 0.808               | 0.049                |
| $m = 5$ days      | 0.580       | 0.420               | 0.411               | 0.368               | 0.221                |
| $m = 3$ days      | 0.746       | 0.254               | 0.000               | 0.009               | 0.991                |
| $m = 1$ day       | 0.946       | 0.054               | 0.000               | 0.001               | 0.999                |

Note: The dependence structure is classified using the two trained CNNs. AD and AI refer to the asymptotic dependence and asymptotic independence, respectively.

The classification procedure shows that the asymptotic independence structure is more suitable for block maxima sizes of up to 10 days. This is in accord with Bacro et al. (2016), where the rainfall amount for different locations within the same region was studied. The authors concluded that it is unsuitable to choose asymptotic dependence models for modeling the seasonal maxima. In addition, for a block maxima of size $m = 5$, the prediction is not decisive for the 2-class CNN. For 3 days and the daily block maxima, the classifier with 2 classes gives a high probability for an asymptotic dependence model. However, a CNN with 3 classes gives different predictions, in which a high probability of a mixture between AD and AI should be chosen. Furthermore, an investigation using the same data was conducted in previous studies using different block maxima sizes; see Bacro et al. (2016), Ahmed et al. (2017), and Abu-Awwad et al. (2020). The authors found that the max-mixture models are suitable. This was confirmed through the prediction of a CNN with 3 classes.

7 | DISCUSSION AND CONCLUSIONS

Because the type of dependence structure may have an influence on the nature of joint extreme events, it is important to study this matter. Most such studies have dealt with modeling extreme events directly through parametrical statistics methods, usually without a preliminary investigation into which pattern of dependence structure would be the most suitable. Moreover, the block maxima size has an influence on the dependence structure. In this article, differing significantly from classical methods, we proposed exploiting the power of a CNN to investigate the pattern of the dependence structure of extreme events. Two environmental data (air temperature at 2 m over Iraq and rainfall over the eastern coast of Australia) were studied to classify the patterns of their dependence structures. The input of the designed CNN are the empirical upper and lower tail dependence measures $\hat{\chi}_{0.975}(s,t)$ and $\hat{\chi}_{0.025}(s,t)$. The training process was applied on data generated from the max-stable models and inverse max-stable and extreme Gaussian processes to obtain asymptotic independent models. The data were generated according to fixed coordinates rescaled within $[0,1]^2$. The ability of this model to recognize the pattern of a dependence structure was emphasized in terms of training, validation, testing loss, and accuracy.

It is worth mentioning that the sensitivity of the dependence structure class when considering the size of the block maxima should be taken into account in the models. When adopting this classification procedure, it may be advisable to choose a reasonable sized block maxima allowing the data to achieve a good representation. For instance, for the air temperature event, whatever the block maxima size that is chosen, the temperature dataset exhibited asymptotic dependence rather than independence. For rainfall data, however, the dependence structure class changed across the block size.

ACKNOWLEDGMENT

This work was supported by PAUSE which is operated by Collège de France, and the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program “Investissements d’Avenir” (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).
REFERENCES

Abu-Awwad, A., Maume-Deschamps, V., & Ribereau, P. (2020). Fitting spatial max-mixture processes with unknown extremal dependence class: An exploratory analysis tool. Test, 29, 479–522.

Ahmed, M., Maume-Deschamps, V., Ribereau, P., & Vial, C. (2017). A semi-parametric estimation for max-mixture spatial processes. arXiv preprint arXiv:1710.08120.

Bacro, J., Gaetan, C., & Toulemonde, G. (2016). A flexible dependence model for spatial extremes. Journal of Statistical Planning and Inference, 172, 36–52.

Bortot, P., Coles, S., & Tawn, J. (2000). The multivariate gaussian tail model: An application to oceanographic data. Journal of the Royal Statistical Society: Series C (Applied Statistics), 49(1), 31–49.

Brown, B. M., & Resnick, S. I. (1977). Extreme values of independent stochastic processes. Journal of Applied Probability, 14(4), 732–739.

Caterini, A. L., & Chang, D. E. (2018). Deep neural networks in a mathematical framework. Springer.

Chattopadhyay, A., Hassanazadeh, P., & Pasha, S. (2020). Predicting clustered weather patterns: A test case for applications of convolutional neural networks to spatio-temporal climate data. Scientific Reports, 10(1), 1–13.

Coles, S., & Heffernan, J., & Tawn, J. (1999). Dependence measures for extreme value analyses. Extremes, 2(4), 339–365.

Coles, S., & Pauli, F. (2002). Models and inference for uncertainty in extremal dependence. Biometrika, 89(1), 183–196.

De Haan, L. (1984). A spectral representation for max-stable processes. The Annals of Probability, 12(4), 1194–1204.

De Haan, L., & Ferreira, A. (2007). Extreme value theory: An introduction. Springer Science & Business Media.

De Haan, L., & Pereira, T. T. (2006). Spatial extremes: Models for the stationary case. The Annals of Statistics, 34(1), 146–168.

Embrevets, P., Klüppelberg, C., & Mikosch, T. (2013). Modelling extremal events: For insurance and finance (Vol. 33). Springer Science & Business Media.

Ferreira, A., & De Haan, L. (2015). On the block maxima method in extreme value theory: PWM estimators. Annals of Statistics, 43(1), 276–298.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological cybernetics, 36(4), 193–202.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770–778). Las Vegas, NV.

Hooker, J., Duveiller, G., & Cescatti, A. (2018). A global dataset of air temperature derived from satellite remote sensing and weather stations. Scientific Data, 5(1), 1–11.

Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. The Journal of Physiology, 195(1), 215–243.

Huser, R. (2021). EVA 2019 data competition on spatio-temporal prediction of red sea surface temperature extremes. Extremes, 24, 91–104.

Kabuchi, Z., Schlather, M., & De Haan, L. (2009). Stationary max-stable fields associated to negative definite functions. The Annals of Probability, 37(5), 2042–2065.

Kingma, D. & Ba, J. (2015). Adam: A method for stochastic optimization. Proceedings of the International Conference on Learning Representations (ICLR). San Diego.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 1097–1105.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., & Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In: M. I. Jordan, Y. LeCun, & S. A. Solla (Eds.), Advances in neural information processing systems (pp. 396–404).

Ledford, A., & Tawn, J. A. (1996). Statistics for near independence in multivariate extreme values. Biometrika, 83(1), 169–187.

Lin, Y., Mago, N., Gao, Y., Li, Y., Chiang, Y., Shahabi, C., & Ambite, J. L. (2018). Exploiting spatiotemporal patterns for accurate air quality forecasting using deep learning. Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (pp. 359–368). New York, NY.

Liu, Y., Racah, E., Correa, J., Khosrowshahi, A., Lavers, D., Kunkel, K., Wehner, M., & Collins, W. (2016). Application of deep convolutional neural networks for detecting extreme weather in climate datasets. arXiv preprint arXiv:1605.01156.

Naveau, P., Guillou, A., Cooley, D., & Diebolt, J. (2009). Modelling pairwise dependence of maxima in space. Biometrika, 96(1), 1–17.

Optz, T. (2013). Extremal T processes: Elliptical domain of attraction and a spectral representation. Journal of Multivariate Analysis, 122, 409–413.

Schlather, M. (2002). Models for stationary max-stable random fields. Extremes, 5(1), 33–44.

Schlather, M., & Tawn, J. A. (2003). A dependence measure for multivariate and spatial extreme values: Properties and inference. Biometrika, 90(1), 139–156.

Simpson, E. S., Wadsworth, J. L., & Tawn, J. A. (2020). Determining the dependence structure of multivariate extremes. Biometrika, 107(3), 513–532.

Smith, R. L. (1990). Max-stable processes and spatial extremes [Unpublished manuscript].

Uselis, A., Lukosevicius, M., & Stasysit, L. (2020). Localized convolutional neural networks for geospatial wind forecasting. Energies, 13(13), 3440.

Wadsworth, J. L., & Tawn, J. A. (2012). Dependence modelling for spatial extremes. Biometrika, 99(2), 253–272.

Wang, S., Cao, J., & Yu, P. S. (2020). Deep learning for spatio-temporal data mining: A survey. IEEE Transactions on Knowledge and Data Engineering, https://doi.org/10.1109/TKDE.2020.3025580.

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1492-1500). IEEE, Honolulu, HI, USA.
Yamashita, R., Nishio, M., Do, R., & Togashi, K. (2018). Convolutional neural networks: An overview and application in radiology. *Insights into Imaging, 9*(4), 611–629.

Yu, H., Uy, W. I. T., & Dauwels, J. (2016). Modeling spatial extremes via ensemble-of-trees of pairwise copulas. *IEEE Transactions on Signal Processing, 65*(3), 571–586.

Zhu, Q., Chen, J., Zhu, L., Duan, X., & Liu, Y. (2018). Wind speed prediction with spatio–temporal correlation: A deep learning approach. *Energies, 11*(4), 705.

Zuo, Z., Shuai, B., Wang, G., Liu, X., Wang, X., Wang, B., & Chen, Y. (2015). Convolutional recurrent neural networks: Learning spatial dependencies for image representation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 18-26). IEEE, Boston, MA, USA.

**How to cite this article:** Ahmed, M., Maume-Deschamps, V., & Ribereau, P. (2022). Recognizing a spatial extreme dependence structure: A deep learning approach. *Environmetrics, 33*(4), e2714. [https://doi.org/10.1002/env.2714](https://doi.org/10.1002/env.2714)