Supplement of Nat. Hazards Earth Syst. Sci., 26, 163–186, 2026
https://doi.org/10.5194/nhess-26-163-2026-supplement

Natural Hazards and Earth System Sciences
Open Access

Supplement of

Deciphering the drivers of direct and indirect damages to companies from an unprecedented flood event: A data-driven, multivariate probabilistic approach

Ravikumar Guntu et al.

Correspondence to: Ravikumar Guntu (ravikumar.guntu@gfz.de)

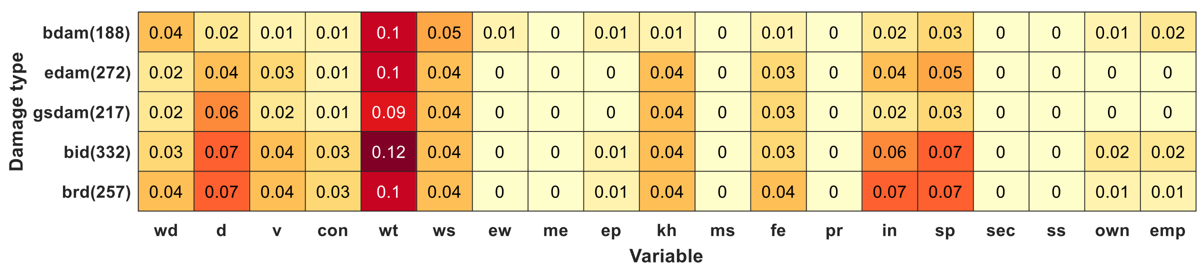

Fig. S1: Percentage of missing values per factor (x-axis) for each damage type (y-axis). The values shown in the heatmap are the percentages of missing data, where 0.1 corresponds to 10%. The value in parentheses for each damage type indicates the number of responses available out of 431. For warning time (wt), cases where no warning was received are treated as zero.

S1 Optimal hyperparameters selection

We conducted hyperparameter tuning, which randomly samples from a predefined range of values for each hyperparameter. The search was performed within a nested cross-validation framework. The framework consists of two levels of cross-validation.

Outer cross-validation: The outer loop is responsible for evaluating the overall performance of the model. It splits the dataset into training and testing subsets, ensuring that performance estimation is conducted on unseen data. We employed *RepeatedKFold* cross-validation with 10 splits and 10 repeats, resulting in a total of 100 evaluations.

Inner cross-validation: Within each outer loop iteration, the inner cross-validation loop is used to tune hyperparameters. We employed *RandomizedSearchCV*, which samples different hyperparameter combinations and evaluates them using 10-fold cross-validation within the training set.

This approach ensures that model selection (inner loop) and performance evaluation (outer loop) are conducted independently, leading to a more reliable assessment of the model's generalization ability (Cawley and Talbot, 2010). The optimal combination of hyperparameters was selected based on the lowest Mean Absolute Error (MAE) during outer cross-validation.

Example breakdown for one outer fold – bdam (total 188 samples)

Outer loop (10-fold cross-validation, repeated 10 times)

- Test set: 19 samples (1-fold, unseen during training and tuning).
- Training set: 169 samples (remaining 9 folds, used for inner-loop tuning and training).

Inner loop (10-fold cross-validation within the training set of 169 samples)

- Training fold (inner loop): 152 samples (9 folds, used for model training).
- Validation fold (inner loop): 17 samples (1-fold, used to evaluate hyperparameters tuing).

The following hyperparameters and their respective search ranges were examined. All other hyperparameters were set to their default values as provided in the relevant scikit-learn packages (Pedregosa et al., 2011). The random state/seed was kept constant across all three models.

S1.1 Elastic Net (EN) Hyperparameters

Elastic Net is a linear model that combines Lasso and Ridge regularization. The following hyperparameters control its regularization strength:

- α: Controls the strength of the regularization and balances between Lasso ($\alpha = 1$) and Ridge ($\alpha = 0$) regression. α: [0.01, 0.1, 0.25, 0.5, 0.75, 0.9, 1]
- λ: Determines the balance between the Lasso and Ridge terms. λ: [0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0]

S1.2 Random Forest (RF) Hyperparameters

Random Forest is an ensemble method that constructs multiple decision trees. Key hyperparameters that influence the model's performance are:

- n_estimators: The number of trees in the forest. n_estimators = [50, 100, 200, 300]
- max_depth: The maximum depth of the individual trees. max_depth = [3, 5, 7, 10, 15, 20]

- min_samples_split: Minimum number of samples required to split an internal node. min_samples_split = [2, 5, 10]
- min_samples_leaf: Minimum number of samples required to be at a leaf node. min_samples_leaf = [1, 2, 4]
- max_features: Number of features to consider when splitting a node. max_features = ["sqrt", "log2", 0.5, 0.75, 1.0]

S1.3 XGBoost (XGB) Hyperparameters

XGBoost is a gradient boosting method that optimizes speed and performance. The following hyperparameters are crucial for controlling the boosting process:

- n_estimators: The number of boosting rounds or trees. n_estimators = [50, 100, 200, 300, 500]
- learning_rate: Step size for each iteration, which controls the contribution of each tree. learning_rate = [0.005, 0.01, 0.05, 0.1, 0.2]
- max_depth: The maximum depth of each tree. max_depth = [3, 5, 7, 10, 15, 20]
- subsample: Fraction of samples used to train each tree. subsample = [0.1, 0.3, 0.5, 0.7, 1.0]
- colsample_bytree: Fraction of features used to train each tree. colsample_bytree = [0.1, 0.3, 0.5, 0.7, 1.0]
- gamma: Regularization parameter that controls overfitting. Gamma = [0, 0.1, 0.5, 1, 5]
- lambda (L2 regularization) and alpha (L1 regularization): These terms control overfitting by penalizing the tree's complexity. $\lambda = [0, 0.1, 1, 10]$, $\alpha = [0, 0.1, 1, 10]$.

S2 Bayesian Network learning using the BDe score

The top four variables identified from the combined weighted importance scores obtained using three machine le arning models were selected to predict flood damage through Bayesian Networks (BNs). Given that the dataset i ncludes both discrete and continuous variables, continuous variables were discretized using an equal-frequency b inning approach, ensuring all variables became categorical after discretization.

S2.1 Bayesian Network Structure Learning

The Tabu search algorithm was applied to determine the Bayesian Network structure by maximizing the Bayesia n Dirichlet equivalent (BDe) score. The learning process began with a simple Naive Bayes structure, in which al l predictor variables (parent nodes) are assumed to be conditionally independent of each other given the target va riable (child node). This initial structure reflects the assumption that no dependencies exist among predictors wh en conditioned on the target.

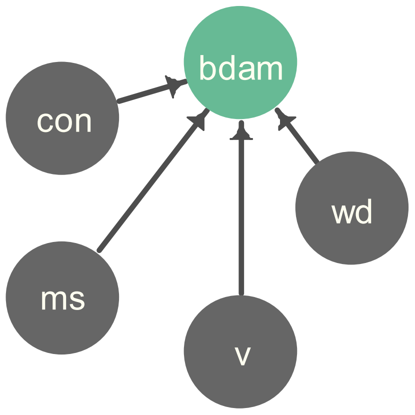

Fig. S2: Illustration of the initial Naive Bayes structure for modeling company building damage using predictor variables as parents and the damage variable as the child node.

S2.2 BDe Score Definition

The BDe score, a log-likelihood-based metric, evaluates how well a network structure (DAG) fits the data. It is computed as (Heckerman et al., 1995):

$$BDe(G|D) = \prod_{i=1}^{n}\left(\prod_{j=1}^{q_i}\left(\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})}\prod_{k=1}^{r_i}\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})}\right)\right) \tag{S1}$$

Where:

- n = number of variables
- q_i = number of parent configurations for variable X_i
- r_i = number of bins for X_i
- N_{ijk} = observed count of instances where $X_i = x_k$ given parent configuration Pa_j
- N_{ij} = total count for parent configuration j.
- α_{ijk} = Dirichlet prior parameter.
- α_{ij} = Sum of Dirichlet priors for a given parent configuration.
- $\Gamma(\cdot)$ = Gamma function.

A higher BDe score indicates a better-fitting network.

Example calculation of the BDe score

Consider a simplified BN with two variables: wd (predictor: water depth) and $bdam$ (target: relative damage to building). Both are discretized into bins as shown below.

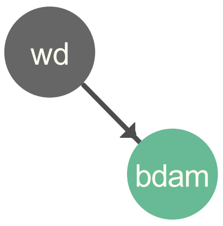

Fig. S3: A simple Bayesian network representing the relationship between water depth (wd) and building damage ($bdam$).

S2.3 Frequency Counts and Total Counts

Observed frequencies (N_{ijk}) from the sample data are presented in Table S1.

Table S1: Observed frequency of $bdam$ conditional on wd.

wd	$bdam$	Count (N_{ijk})
$\leq 1\,m$	[0.00025,0.120)	30
$\leq 1\,m$	[0.12000,0.300)	19
$\leq 1\,m$	[0.30000,0.528)	7
$\leq 1\,m$	[0.52761,0.960)	4
$\leq 1\,m$	[0.96000,1.000]	5
$> 1\,m\,\&\, \leq 2\,m$	[0.00025,0.120)	3
$> 1\,m\,\&\, \leq 2\,m$	[0.12000,0.300)	18
$> 1\,m\,\&\, \leq 2\,m$	[0.30000,0.528)	16
$> 1\,m\,\&\, \leq 2\,m$	[0.52761,0.960)	20
$> 1\,m\,\&\, \leq 2\,m$	[0.96000,1.000]	16
$> 2\,m$	[0.00025,0.120)	5
$> 2\,m$	[0.12000,0.300)	4
$> 2\,m$	[0.30000,0.528)	18
$> 2\,m$	[0.52761,0.960)	7
$> 2\,m$	[0.96000,1.000]	16

Table S2: Total counts (N_{ij}) for each parent configuration (wd bins).

$\boldsymbol{wd}$	**Total** $\boldsymbol{N_{ij}}$
$\leq\ 1\,m$	65
$>\ 1\,m\,\&\ \leq\ 2\,m$	73
$>\ 2\,m$	50

S2.4 Dirichlet Prior

Assuming a uniform Dirichlet prior with α = 15:

$$\alpha_{ijk} = \frac{\alpha}{r \times q} = \frac{15}{5 \times 3} = 1$$

$$\alpha_{ij} = \sum_{k=1}^{r} \alpha_{ijk} = 5$$

S2.5 BDe Score Computation

the BDe score is calculated for each parent configuration.

For $wd\ (j = 1)$: $\frac{\Gamma(5)}{\Gamma(5+65)} \times \frac{\Gamma(1+30)}{\Gamma(1)} \times \frac{\Gamma(1+19)}{\Gamma(1)} \times \frac{\Gamma(1+7)}{\Gamma(1)} \times \frac{\Gamma(1+4)}{\Gamma(1)} \times \frac{\Gamma(1+5)}{\Gamma(1)}$

For $wd\ (j = 2)$: $\frac{\Gamma(5)}{\Gamma(5+73)} \times \frac{\Gamma(1+3)}{\Gamma(1)} \times \frac{\Gamma(1+18)}{\Gamma(1)} \times \frac{\Gamma(1+16)}{\Gamma(1)} \times \frac{\Gamma(1+20)}{\Gamma(1)} \times \frac{\Gamma(1+16)}{\Gamma(1)}$

For $wd\ (j = 3)$: $\frac{\Gamma(5)}{\Gamma(5+50)} \times \frac{\Gamma(1+5)}{\Gamma(1)} \times \frac{\Gamma(1+4)}{\Gamma(1)} \times \frac{\Gamma(1+18)}{\Gamma(1)} \times \frac{\Gamma(1+7)}{\Gamma(1)} \times \frac{\Gamma(1+16)}{\Gamma(1)}$

$BDe\left(\frac{G}{D}\right) = wd\ (j = 1) \times wd\ (j = 2) \times wd\ (j = 3) \approx 8.98 \times 10^{-125}$

The Tabu Search algorithm iteratively explores modifications to the DAG (adding, deleting, or reversing edges) while avoiding cycles. Each candidate structure is evaluated using the BDe score, and the highest-scoring network is selected.

S2.6 Conditional Probability Tables (CPTs)

With the learned structure, CPTs are computed:

$$P\big(X = x_k \big| Pa(X) = pa_j\big) = \frac{N_{ijk}}{N_{ij}} \tag{S2}$$

Table S3: Conditional Probability Table (CPT) representing $P(bdam|wd)$.

bdam	$\boldsymbol{wd\ \leq\ 1\,m}$	$\boldsymbol{wd\ >\ 1\,m\,\&\,wd\ \leq\ 2\,m}$	$\boldsymbol{wd\ >\ 2\,m}$
[0.000, 0.120)	$\frac{30}{65} = 0.46$	$\frac{3}{73} = 0.04$	$\frac{5}{50} = 0.10$
[0.120, 0.300)	$\frac{19}{65} = 0.29$	$\frac{18}{73} = 0.25$	$\frac{4}{50} = 0.08$
[0.300, 0.528)	$\frac{7}{65} = 0.11$	$\frac{16}{73} = 0.22$	$\frac{18}{50} = 0.36$
[0.528, 0.960)	$\frac{4}{65} = 0.06$	$\frac{20}{73} = 0.27$	$\frac{7}{50} = 0.14$
[0.960, 1.000]	$\frac{5}{65} = 0.08$	$\frac{16}{73} = 0.22$	$\frac{16}{50} = 0.32$

S2.7 Bayesian Inference using the Markov Blanket

With CPTs established, Bayesian inference is conducted using the Markov Blanket of the target node. The posterior is calculated as:

$$P(X_i|E) = \frac{P(E|X_i)P(X_i)}{P(E)} \tag{S3}$$

Where:

$P(E|X_i)$ = likelihood of evidence given X_i.

$P(X_i)$ = prior probability of X_i.

$P(E) = \sum_{X_i} P(E|X_i)P(X_i)$

Example: For evidence $wd > 2\,m$:

1. Compute marginal probabilities $P(bdam_k)$

Table S4: Marginal probabilities of $bdam$ bins.

bdam	**count**	$\boldsymbol{P(bdam_k)}$
[0.000, 0.120)	30+3+5=38	$\frac{38}{188} = 0.2021$
[0.120, 0.300)	19+18+4=41	$\frac{41}{188} = 0.2181$
[0.300, 0.528)	7+16+18=41	$\frac{41}{188} = 0.2181$
[0.528, 0.960)	4+20+7=31	$\frac{31}{188} = 0.1649$
[0.960, 1.000]	5+16+16=37	$\frac{37}{188} = 0.1968$
Total	188	

2. Compute conditional probabilities $P(wd = 3|bdam_k)$

Table S5: Conditional probabilities

bdam	$\boldsymbol{wd > 2\,m}$ **Count**	$\boldsymbol{P(wd > 2\,m|bdam_k)}$
[0.000, 0.120)	5	$\frac{5}{38} = 0.1316$
[0.120, 0.300)	4	$\frac{4}{41} = 0.0976$
[0.300, 0.528)	18	$\frac{18}{41} = 0.4390$
[0.528, 0.960)	7	$\frac{7}{31} = 0.2258$
[0.960, 1.000]	16	$\frac{16}{37} = 0.4324$
Total	50	

3. Compute $P(wd > \mathbf{2\,m})$

$$P(wd > \mathbf{2\,m}) = \frac{50}{188} = 0.26596$$

4. Compute posterior probabilities:

$$P(bdam_k|wd > \mathbf{2\,m}) = \frac{P(wd > \mathbf{2\,m}|bdam_k) \times P(bdam_k)}{P(wd > \mathbf{2\,m})}$$

Table S6: Posterior probabilities

bdam bin	$P(wd > 2\,m \mid bdam_k)$	$P(bdam_k)$	$P(wd > 2\,m)$	$P(bdam_k \mid d > 2\,m)$
[0.000, 0.120)	$\frac{5}{38} = 0.1316$	$\frac{38}{188} = 0.2021$	0.26596	0.10
[0.120, 0.300)	$\frac{4}{41} = 0.0976$	$\frac{41}{188} = 0.2181$	0.26596	0.08
[0.300, 0.528)	$\frac{18}{41} = 0.4390$	$\frac{41}{188} = 0.2181$	0.26596	0.36
[0.528, 0.960)	$\frac{7}{31} = 0.2258$	$\frac{31}{188} = 0.1649$	0.26596	0.14
[0.960, 1.000]	$\frac{16}{37} = 0.4324$	$\frac{37}{188} = 0.1968$	0.26596	0.32

$$P(bdam_k \mid wd > 2\,m) = [0.10, 0.08, 0.36, 0.14, 0.32]$$

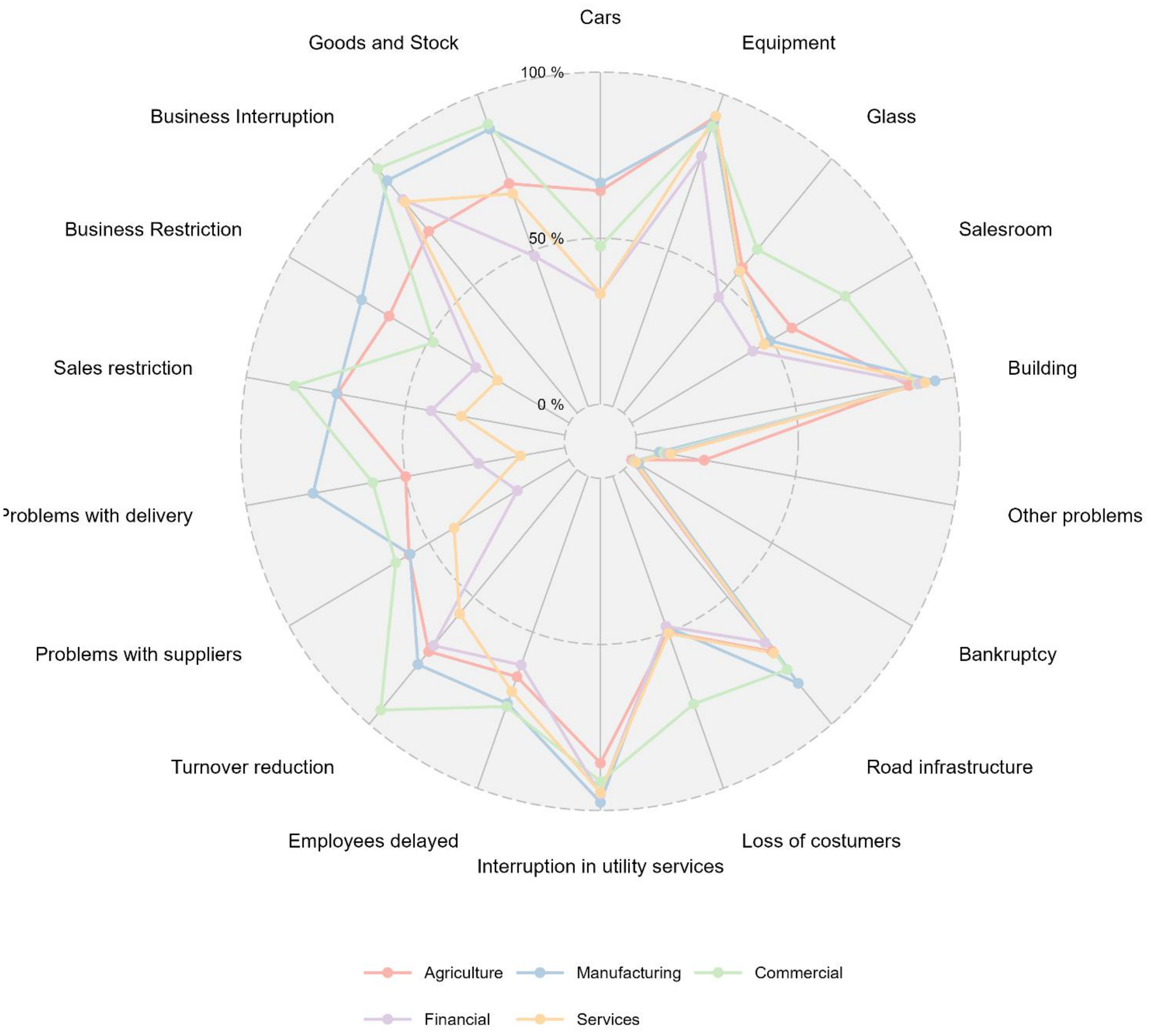

Fig. S4: Spider chart illustrating the percentage of companies experiencing different types of flood impacts, categorized by the sector.

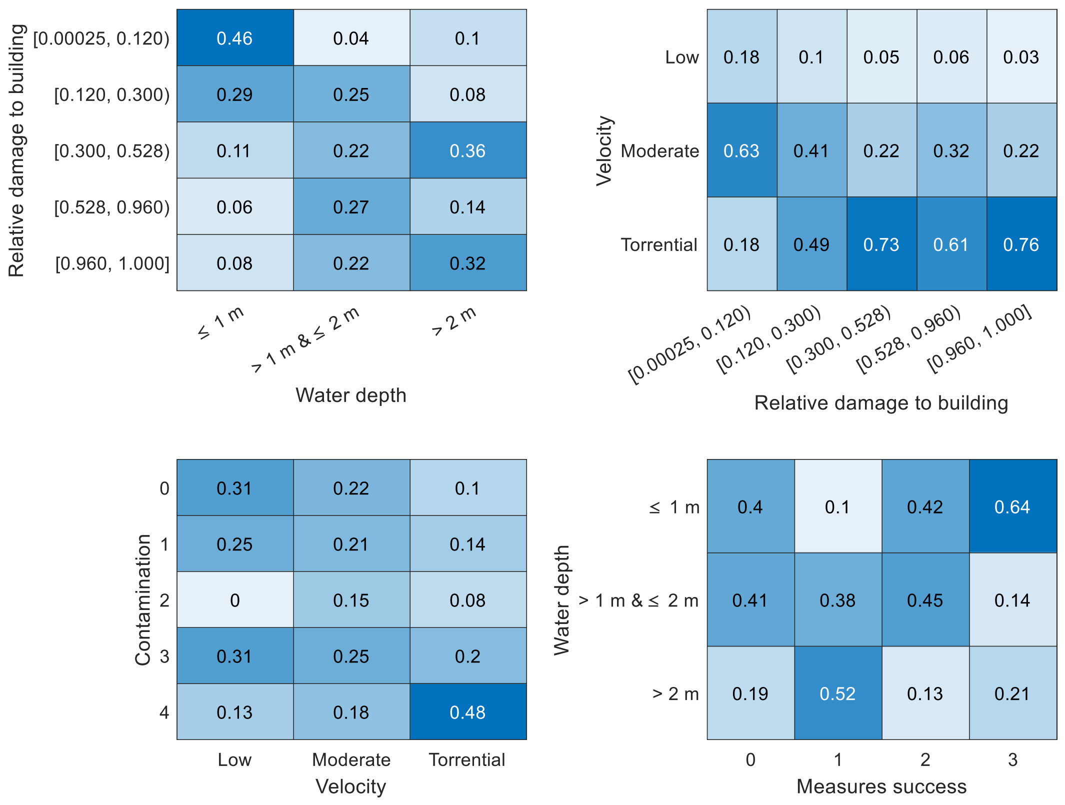

Fig. S5: Conditional probability table of companies building Bayesian network (Fig. 7a). X-axis denotes the parent node, while Y-axis denotes the children node within the network.

<u>References:</u>

Cawley, G. C. and Talbot, N. L. C.: On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation, Journal of Machine Learning Research, 11, 2079–2107, 2010.

Heckerman, D., Geiger, D., and Chickering, D. M.: Learning Bayesian networks: The combination of knowledge and statistical data, Mach Learn, 20, 197–243, https://doi.org/10.1007/BF00994016, 1995.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É.: Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 12, 2825–2830, 2011.