Predict the environmental risk of the watershed in South-Western Ohio United States

Authors: Zhiqiang Liao[1], Zhenkun Li[2],

[1] Department of Information and Service Management, School of Business, Aalto University.

[2] Department of Civil Engineering, School of Engineering, Aalto University.

1 Introduction

Ecological risk assessments (ERA) are a critical component of environmental decision-making, as the ability to make sound risk management decisions rests heavily on the information they provide [1]. In this project, we use a logistic regression model to predict the presence-absence state of environmental stressors (e.g., toxic chemicals), which has been proven highly related to fish community health [2]. Compared to the traditional ERA methods, our project provides an alternative and inexpensive way for decision-makers to environmental risk management.

1.1 Motivation

The traditional ERA process usually requires specialized models, data, and expertise, which can be cost and time-intensive. Alternatively, data-driven prediction models that use Geographic Information Systems (GIS) data are often attractive to practitioners and researchers because they offer a cost-effective way of quickly detecting areas of potential risk across broad regions; and can inform future, more targeted research activities [3]. In this project, we use the recently published StreamCat database, which includes a comprehensive list of the segment and watershed-scale metrics for use with the NHDPlus dataset within the conterminous United States [4].

This project follows the research paper of Martin et al. [3]. In this paper, the authors applied a Bayesian separate model with a latent variable. The main idea of this study is to analyze the correlation between stressors (e.g., toxic chemicals) and ecological receptors (e.g., fishes). Although we use the same dataset as the original paper, we consider a different task which is to predict the presence-absence state of the environmental stressors. To do so, we chose total phosphorus (TP) as a response variable and three measures derived from GIS database sources as predictors. This question is also interesting, given the real-time available GIS data.

In this project, we use two Bayesian models to predict the environmental risk of the watershed. We believe the proposed models would help decision-makers who are responsible for environmental risk management.

1.2 Problem

In a recent study of Ohio streams, Miltner (2010) [2] examined the effects of TP on fishes using statewide bioassessment data and suggested a seasonal average for TP at 0.1 mg/L as a feasible management target for improving fish community health and to aid future abatement and prevention of eutrophication in Ohio streams ($<$1300 km^2 in drainage area).

The research team from the Ohio Environmental Protection Agency measured 83 bioassessment sites collected over the summer of 2012. In the dataset, they collected 26 stressors (i.e., TP), in which we chose TP as the response variable. We used this management criterion to examine the effects of excess nutrients on potentially sensitive fishes and their distributions. Thus, TP concentration data (mg/L) for

each bioassessment site were converted to binary variables depending on whether values were below or above 0.1 mg/L. As the notation of the following section, our outcome variable is

$$y_n = \begin{cases} 1 & \text{if the site } n \text{ is environmental risk for fishes} \\ 0 & \text{if the site } n \text{ is environmental health for fishes} \end{cases}$$

1.3 Main modeling idea

In the original paper [3], the authors chose informative priors for all 27 responses (corresponding to records of presence-absence for fishes). The proposed separate Bayesian method cannot sufficiently fit a model including different types of responses, though the authors suggested that they could have chosen different types of priors (e.g., informative, weakly informative) for each parameter and response combination in order to fit a model including all 77 responses.

First, we propose a hierarchical model to handle this problem in this project. The separate model does not account for data heterogeneity, neither in coefficients (parameters) nor in data-generating methods. Allowing parameters to change between respondents is one technique to include heterogeneity in models.

Second, we apply the Gaussian process to fit a nonlinear model. In this model, we can model more complex regression interactions by including a more complex mean function, such as a high-order polynomial of the input covariates.

1.4 Illustrative figure

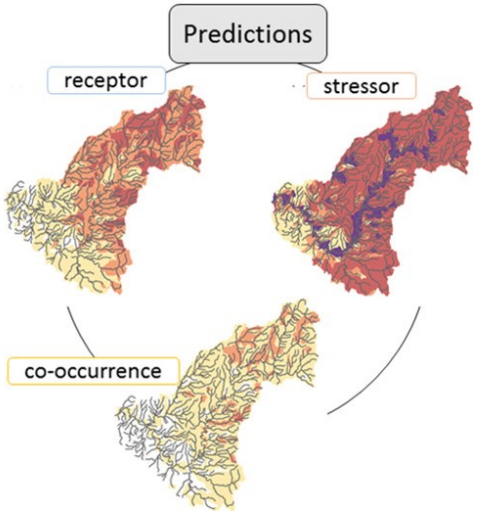


Figure 1: Illustration of the problem [3]

2 Description of the data and the analysis problem

2.1 Descriptive statistics

The response data, Y, comprised empirical data from 28 responses measured across 83 sites in the EFLMR watershed; with 27 of the responses corresponding to records of presence-absence for fishes and 1 response corresponding to records of TP exceeding the management threshold of 0.1 mg/L. The input data, X, included three predictors: drainage area, crop cover, and septic tank counts. All predictor variables were standardized prior to input into the models by subtracting the mean and dividing by 2 standard deviations.

In Table 1, we summarize the descriptive statistics of input data from the NHDPlus hydrologic spatial dataset [5] and response data from EFLMR watershed dataset by the Ohio Environmental Protection Agency's (OEPA) Statewide Biological and Water Quality Monitoring and Assessment Program. Note that we omit the data of 27 fish because of the limitation of space. The drainage area was log-transformed (natural log) for model input in order to linearize the relationship with responses. We follow the data process method to calculate the septic tank counts from septic density data.

Table 1: Data description for GIS data and TP data.

	AREA	CROP	SEP_{DENS}	TP
Min.	0.1008	0.00	0.000	0.000
1st Qu.	1.9246	30.44	4.969	0.000
Median	5.3154	58.17	14.721	1.000
Mean	133.5302	50.20	23.789	0.6988
3rd Qu.	32.7884	70.73	31.874	1.000
Max.	1293.5205	96.49	282.776	1.000

In Figure 2, we plot the correlation structure of features. Though other landscape-based predictors may have provided some explanatory power in our analysis (i.e., soil type), we chose these three predictors because they are the primary landscape-based sources of anthropogenic stress in the watershed for which data was readily available.

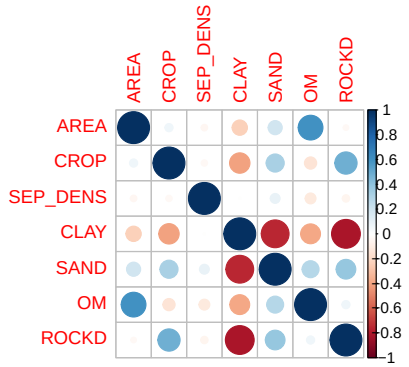


Figure 2: Correlation structure

In summary, we choose landscape-based data as input data for our models. The data were quantified at the stream segment scale (1:100,000) defined by the vector-based, segment-level catchments within the NHDPlus hydrologic spatial dataset. Using this dataset has the advantage of being easy to obtain. With the development of space technology, it is also very convenient to collect these data. We have uploaded all datasets and codes to our GitHub repository.[1]

2.2 R setup

The full R code can be found in `MVP.R`:

```
set.seed(9527)
#import packages to be used
library(gridExtra)
library(qgraph)
library(coda)
```

1. https://github.com/ZhenkunLi/BDA_Project

```r
library(rstan)
library(ggplot2)
library(loo)
library(posterior)

#set stan options for parallel processing of HMC chains
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
```

2.3 Data processing

We standardize the three predictors, including drainage area ($\beta_{\log(\text{AREA})}$), crop cover ($\beta_{\text{CROP}}$), and density of septic systems(β_{SEP}).

```r
#Function to scale and center predictors based on Gelman et al. 2006
MyStd <- function(x){ (x-mean(x))/(sd(x)*2)} #apply(x,2,MyStd)

#Apply same standardization to variables used to predict to unsampled catchments
x_predict <-  as.matrix(cbind(MyStd(log(streamCat[,4])),
                              MyStd(streamCat[,16]),
                              MyStd(streamCat[,19])
))
```

For the response variable, we use the dataset from EFLMR, which can be found in `OEPA_WATER_2012.csv`, which includes locations of OEPA bioassessment sites ($N = 83$) across the EFLMR watershed.

```r
#convert fish abundance to presence-absence (binary; 0 or 1)
y <- as.matrix(ifelse(modelData[,c(6:82,146)]>0,1,0))
```

3 Description of models

In this section, we will introduce two models used in this report. They are (1) Hierarchical model, and (2) Gaussian process model.

3.1 Hierarchical model

Our hierarchical model is shown in Eqs. (1)-(3), and its illustration can be found in Figure 3. As introduced in Section 2.1, our objective is to predict the occurrence of total phosphorus (TP). The occurrence of TP can be determined by several factors. In this report, we have selected drainage area (AREA, β_{AREA}), crop cover (CROP, β_{CROP}), and density of septic systems (SEP, β_{SEP}) as our input features. The intercept is represented by β_0.

$$P(y_n = 1) = \text{logit}^{-1}(\beta_0 + \beta_n X_n), \forall n = 1, \cdots, N \tag{1}$$

$$\beta_0 \sim N(\mu_0, \sigma_0) \tag{2}$$

$$\beta_k \sim N(\mu_1, \sigma_1) \tag{3}$$

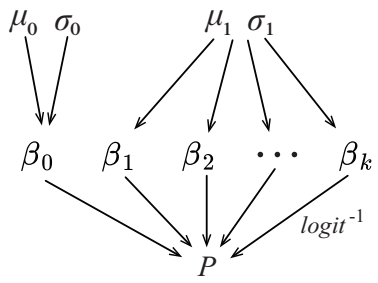


Figure 3: Illustration of hierarchical model

In the hierarchical model, the hyperparameter of β_0 is μ_0 and σ_0. The prior of μ_0 is a normal distribution with a mean of 0 and a standard deviation of 1, and σ_0 is subjected to a gamma distribution with $\alpha = 2, \beta = 2$. Like Gelman and Hill [6], for the weights of the Bayesian linear regression model β_k. The weights also follow a normal distribution $N(\mu_1, \sigma_1)$, where μ_1 is the hyperparameter for the mean values, and σ_1 is the hyperparameter for the standard deviation. μ_1 is subjected to a normal distribution following a mean value of 0 and a standard deviation of 3. σ_1 is subjected to Gamma(3,3). We set different priors for the intercept and slopes because we believe that they are not subjected to a similar distribution. X_n is the input matrix, including all factors that can influence the occurrence of TP.

3.2 Gaussian Process model

In this subsection, we consider presenting the linear model as a Gaussian process. Specifically, we extend the proposed hierarchical linear models to include nonlinear effects and implicit interactions. We describe the basic model as follows

$$P(y_n = 1) = \text{Bernoulli}(\text{logit}(f(X_n)), \forall n = 1, \cdots, N. \tag{4}$$

$$f(X_n) \sim \text{GP}(0, \boldsymbol{K}) \tag{5}$$

where the response y of TP has N sites by a receptor, representing the response of the n-th site. The response data is binary, which takes on the values 1 if the receptor is present and 0 if absent. The predictor variable matrix X has N sites as data size and K features or predictors. Therefore, in this project, we consider a multivariate regression problem in Gaussian process.

Note that we use the exponentiated quadratic kernel to define the covariance between $f(X_i)$ and $f(X_i)$ where $f : \mathbb{R}^K \to \mathbb{R}$ is a function that we aim to model:

$$\text{cov}(f(X_i), f(X_i)) = k(X_i, X_j) = \alpha^2 \exp\left(-\frac{1}{2\rho^2} \sum_{k=1}^{K} (X_{i,k} - X_{j,k})^2 \right) \tag{6}$$

where α and ρ are constrained to be positive. We use one variant of the exponentiated quadratic covariance function in Stan. Thus, the Gaussian process in Eq. (5) is based on a covariance matrix, $\boldsymbol{K} \in \mathbb{R}^{K \times K}$, where $\boldsymbol{K}_{i,j} = k(X_i, X_j)$, which is necessarily symmetric and positive semidefinite by construction. For simplicity, we use just one lengthscale in the exponentiated quadratic kernel for the three-dimensional model in Eq. (4).

In Eq. (4), we predict the ecological risk of water using logistic regression instead of analyzing the correlation between fishes and TP with the linear regression model. In reference [3], the author used Z

as a latent variable, in which the rows follow D-dimensional multivariate normal distributions, which are governed by a common correlation matrix Ω. In our GP model, we do not consider the co-occurrence of stressors (i.e., TP) and receptors (fishes).

4 Weakly informative priors their choices

4.1 Priors of the hierarchical model

In this report, for the hierarchical model, we utilized weakly informative priors. The justification for our choices is introduced below.

- For the intercept β_0, its hyperparameters are selected as $\mu_0 \sim N(0, 1)$. Since we have no idea of how will the mean value of the intercept will be. However, we can understand that it can be around zero if the weights are good. Thus, we select a normal distribution $N(0, 1)$ as the prior for the mean value of the intercept. For the standard deviation, as it should be greater than zero, and Gamma$(2, 2)$ is employed.
- For a weight of a factor, we assume that it is subjected to a normal distribution $N(\mu_1, \sigma_1)$. The prior of the mean values μ_1 are set to be subjected to a normal distribution $N(0, 3)$. Because there are several weights that are selected, we have increased the standard derivation from 1 to 3 compared to the intercept's mean value. For the standard derivation of the normal distribution σ_1, Gamma$(3, 3)$ is selected as it should be greater than 0. The Gamma distribution can be found in Figure 4, where k $= \alpha$, and $\theta = 1/\beta$ [7].

All in all, our priors for the hierarchical model are:

$$\mu_0 \sim N(0, 1), \sigma_0 \sim \text{Gamma}(2, 2), \mu_1 \sim N(0, 3), \sigma_1 \sim \text{Gamma}(3, 3) \tag{7}$$

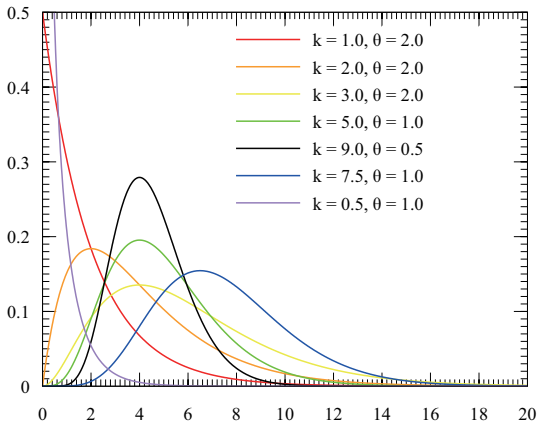


Figure 4: Gamma distribution

4.2 Priors of the Gaussian Process model

In our Gaussian process model, the hyperparameter comprises α and ρ in Eq. (6), for which we can construct appropriate priors. The justification for our choices is introduced below.

- For the Gaussian process in Eq. (5), it has prior with zero mean and exponentiated quadratic covariance function. In this project, we set zero mean for the Gaussian process. It is, on the one hand, for an easier interpretation of the model, on the other hand, for simplicity.

- In the exponentiated quadratic covariance function: Eq. (6), the hyperparameters controlling the covariance function of a Gaussian process can be fitted by assigning them priors. In our second model, the priors on the parameters should be defined based on prior knowledge of the scale of the output values (α) and the scale at which distances are measured among inputs (ρ). First, we choose the weakly informative prior as follows

$$\begin{aligned} \alpha &\sim normal(0,1) \\ \rho &\sim normal(0,1) \end{aligned} \tag{8}$$

5 Stan codes

Model 1: Hierarchical model

We illustrate how to write Stan code that computes and stores the likelihood using the model example from Section 3. We save the .stan code of model 1 in the file `Hier_model_TP_new.stan`.

```
data {
int<lower=1> K;  //features dimension
int<lower=1> D;
int<lower=1> N1; //samples for training
int<lower=1> N2; //samples for test
int<lower=1> NN; //samples for prediction
int idx1[N1]; //indices for training observations
int idx2[N2]; //indices for test observations
int<lower=1> M; //samples for marginal effects est
int<lower=0,upper=1> y[N1+N2, D];
int<lower=0,upper=1> Y[N1+N2];
int trials[N1+N2];
matrix[N1+N2,K]  x;
vector[K] X[NN];
matrix[NN,K] x_predict;
matrix[M,K] x_m_1; //marginal effect AREA
matrix[M,K] x_m_2; //marginal effect CROP
matrix[M,K] x_m_3; //marginal effect SEP
}
parameters {
real beta0;
vector[K] betak; //beta k
real mu_beta0;
real <lower=0> sigma_beta0;
real mu_betak;
real <lower=0> sigma_betak;
}
transformed parameters{
vector[N1+N2] f;
f = betak[1] * x[,1] + betak[2] * x[,2] + betak[3] * x[,3] + beta0;
}
model {
// priors
mu_beta0 ~ normal(0, 1);              // mu_beta0 ~ normal(0, 10); //in Section
    10
```

```
sigma_beta0 ~ gamma(2, 2);              // sigma_beta0 ~ gamma(1, 0.5); //in
    Section 10
beta0 ~ normal(mu_beta0, sigma_beta0);
mu_betak ~ normal(0, 3);                // mu_betak ~ normal(0, 30); //in Section
     10
sigma_betak ~ gamma(3, 3);              // sigma_betak ~ gamma(3, 0.5); //in
    Section 10

for (i in 1:K)
betak[i] ~ normal(mu_betak, sigma_betak);
//likelihood
Y[idx1] ~ binomial_logit(trials[idx1], f[idx1]);
}
generated quantities{
  vector[N1+N2] y_predict;
  vector[N1+N2] f_invlogit;
  vector[N1+N2] log_lik;
  vector[N2] log_y_new;

for(i in 1:(N1+N2)){
y_predict[i] = binomial_rng(1, inv_logit(f[i]));
f_invlogit[i] = inv_logit(f[i]);
log_lik[i] = bernoulli_logit_lpmf(Y[i] | inv_logit(f[i]));
}

for(i in 1:N2){
log_y_new[i] = binomial_logit_lpmf(Y[idx2[i]] | trials[idx2[i]], f[idx2[i]]);
}
}
```

Model 2: Gaussian process

The .stan code of model 2 is in the file `GP.stan`.

```
functions {
  //GP covariance function
  vector gp(vector[] x, real sdgp, real lscale, vector zgp) {
    matrix[size(x), size(x)] cov;
    cov = cov_exp_quad(x, sdgp, lscale);
    for (n in 1:size(x)) {
      cov[n, n] = cov[n, n] + 1e-12;
    }
    return cholesky_decompose(cov) * zgp;
  }
}
data {
 int<lower=1> K;  //features dimension
 int<lower=1> D;
 int<lower=1> N1; //samples for training
 int<lower=1> N2; //samples for test
 int<lower=1> NN; //samples for prediction
```

```
 int idx1[N1]; //indices for training observations
 int idx2[N2]; //indices for test observations
 int<lower=1> M; //samples for marginal effects est
 int<lower=0,upper=1> y[N1+N2, D];
 int<lower=0,upper=1> Y[N1+N2];
 int trials[N1+N2];
 vector[K] x[N1+N2];
 vector[K] X[NN];
 matrix[NN,K] x_predict;
 matrix[M,K] x_m_1; //marginal effect AREA
 matrix[M,K] x_m_2; //marginal effect CROP
 matrix[M,K] x_m_3; //marginal effect SEP
 }
parameters {
 real<lower=0> alpha[K]; // lengthscale of f
 real<lower=0> rho[K];       // scale of f
 matrix[N1+N2,K] eta;
 }
transformed parameters{
  vector[N1+N2] f;

  f = gp(x, alpha[1], rho[1], eta[,1]);

}
model {
  // priors
 alpha ~ normal(0, 1);
 rho ~ normal(0, 1);
 to_vector(eta) ~ normal(0,1);
 Y[idx1] ~ binomial_logit(trials[idx1], f[idx1]);
 }
generated quantities{
  vector[N1+N2] y_predict;
  vector[N1+N2] f_invlogit;
  vector[N1+N2] log_lik;
  // vector[NN] log_y_predict;
  // vector[N] y_new;
  vector[N2] log_y_new;

  for(i in 1:(N1+N2)){
    y_predict[i] = binomial_rng(1, inv_logit(f[i]));
    f_invlogit[i] = inv_logit(f[i]);  //bernoulli_logit_rng(f[i])
    log_lik[i] = bernoulli_logit_lpmf(Y[i] | inv_logit(f[i]));
  }

  for(i in 1:N2){
    log_y_new[i] = binomial_logit_lpmf(Y[idx2[i]] | trials[idx2[i]], f[idx2[i]])
    ;
  }
}
```

6 How the Stan model was run

Model 1: Hierarchical model

In the following line, we used Rstan to compile our model. The working environment is:

- Operate system: Ubuntu 22.04.1 LTS
- Processor: i7-9750 H CPU @ 2.60 GHz × 8
- Memory: 7.7 GiB

```
HierfitDSO <- stan_model(file = "models/Hier_model_TP_new.stan") # Here Rstan
    used to compile our model
```

Then, the used data are put into a list that is convenient to be used for later fit.

```
HierdataList <- list(
    y=y,
    Y=Y,
    x=x,
    X=x_predict,
    x_m_1=as.matrix(cbind(seq(min(x[,1]),max(x[,1]),length.out=M),rep(0, M), rep
    (0, M))),
    x_m_2=as.matrix(cbind(rep(0, M), seq(min(x[,2]),max(x[,2]),length.out=M),
    rep(0, M))),
    x_m_3=as.matrix(cbind(rep(0, M), rep(0, M), seq(min(x[,3]),max(x[,3]),length
    .out=M))),
    K=ncol(x),
    D=ncol(y),
    N1=N1,
    N2=N2,
    NN=NN,
    M=M,
    idx1=idx1,
    idx2=idx2,
    trials=trials,
    x_data=x,
    x_predict=x_predict) # predictive data
```

Finally, we fitted the model using the above data. The optional are introduced below.

- The number of chains: 4
- The number of iterations: 2000
- The number of cores: 4
- The period for saving samples: 1
- Initial values specification: 0
- max_treedepth: 15, as there are warnings indicating that 10 is not enough for fitting this model.

```
Hierfit <- sampling(object=HierfitDSO,
    data=HierdataList,
    chains=4, # the number of chains we used
    iter=2000, # the number of iterations we used
    cores=4, # the number of cores we used
    thin=1, # the period for saving samples is set to 1
    init=0, # Initial values specification
    control = list( # we added several options to improve sampling
```

```
    adapt_delta=0.98, # adapt_delta default=0.8
    max_treedepth =15) # max_treedepth default= 10
```

Model 2: GP model

In the following line, we used Rstan to compile our model. The working environment is:

- Operate system: Windows 10
- Processor: i5-1135G7 CPU @ 2.40 GHz × 6
- Memory: 16 GiB

```
GPfitDSO <- stan_model(file = "models/GP.stan") # Here Rstan used to compile our
    model
```

Then, the used data are put into a list that is convenient to be used for the fit of the GP model.

```
GPdataList <- list(y=y,
    Y=Y,
    x=x,
    X=x_predict,
    x_m_1=as.matrix(cbind(seq(min(x[,1]),max(x[,1]),length.out=M),rep(0, M), rep
    (0, M))),
    x_m_2=as.matrix(cbind(rep(0, M), seq(min(x[,2]),max(x[,2]),length.out=M),
    rep(0, M))),
    x_m_3=as.matrix(cbind(rep(0, M), rep(0, M), seq(min(x[,3]),max(x[,3]),length
    .out=M))),
    K=ncol(x),
    D=ncol(y),
    N1=N1,
    N2=N2,
    NN=NN,
    M=M,
    idx1=idx1,
    idx2=idx2,
    trials=trials,
    x_data=x,
    x_predict=x_predict)#real data
```

Finally, we fitted the model using the above data. The optional are introduced below.

- The number of chains: 4
- The number of iterations: 1000
- The number of cores: 4
- The period for saving samples: 1
- Initial values specification: 0
- max_treedepth: 15, as there are warnings indicating that 10 is not enough for fitting this model.

```
GPfit <- sampling(object=GPfitDSO,
    data=GPdataList,
    chains=4,
    iter=1000,
    cores=4,
    thin=1,
    init=0,
```

```
    control = list( #add options to improve sampling (divergent transitions)
    adapt_delta=0.98, #default=0.8
    max_treedepth =15 #default= 10
        )
    )
```

7 Convergence diagnostics

Model 1: Hierarchical model

For the hierarchical model, at first, when we have only 1000 iterations, we found that the $\hat{R}$ for our hyper-parameter is a little bit high, which means that our MCMC progress may not converge well. Therefore, we improved the iterations to 2000, then all $\hat{R}$ values are smaller than 1.05, meaning that the MCMC for all posterior values converges now.

But we found that in our hierarchical model, the Bulk Effective Samples Size (ESS) value is a little low (even though no warnings), which means that the posterior means and medians may not be quite reliable. We will find more ways to solve this problem. The interesting thing is that when we did not use the hierarchical model; instead, we used the separate model, this problem did not emerge. We would like to have some discussions with the reviewers and TAs after the presentation. In Table 2, the first 8 lines of the posterior are shown, and the full lines can be found on our GitHub website [2].

Discussion about divergences and tree depth: Initially, we have a small number of divergences. So we improve the adapt_delta value to 0.98 and tree depth to 15 (shown in Section 6). For the GP model, the same parameters are employed.

Inference for Stan model: `Hier_model_TP_new.stan`

4 chains, each with iter=2000; warmup=1000; thin=1;

post-warmup draws per chain=1000, total post-warmup draws=4000.

Table 2: Results of fitting: first 8 lines

	mean	se_mean	sd	2.50%	25%	50%	75%	97.50%	n_eff	Rhat
beta0	1.03	0.01	0.31	0.45	0.82	1.03	1.24	1.65	1671	1
betak[1]	1.25	0.01	0.59	0.13	0.86	1.25	1.63	2.43	2265	1
betak[2]	1.51	0.01	0.54	0.5	1.13	1.48	1.84	2.66	2200	1
betak[3]	1.19	0.01	0.63	-0.02	0.77	1.18	1.6	2.46	2155	1
mu_beta0	0.61	0.02	0.7	-0.97	0.19	0.67	1.07	1.88	2043	1
sigma_beta0	0.91	0.01	0.63	0.12	0.45	0.77	1.21	2.44	2249	1
mu_betak	1.29	0.01	0.63	0.02	0.9	1.3	1.7	2.49	1776	1
sigma_betak	0.73	0.01	0.42	0.17	0.43	0.65	0.94	1.78	1540	1

7.1 Model 2: GP model

For the GP model, we have tested 1000 iterations, and good results have been obtained. All $\hat{R}$ values are very close to 1.0 and smaller than 1.05. Therefore, we did not increase the number of iterations for the GP model because within 500 drawings after warmup; the MCMC chain converged nicely. The results are shown below in Table 3, and the full lines can be found on our GitHub website [3]

2. https://github.com/ZhenkunLi/BDA_Project/blob/main/report/Hier_model_TP_new.txt
3. https://github.com/ZhenkunLi/BDA_Project/blob/main/report/GP_EFLMR_cooc_Final.txt

Inference for Stan model: `GP`.

4 chains, each with iter=1000; warmup=500; thin=1;

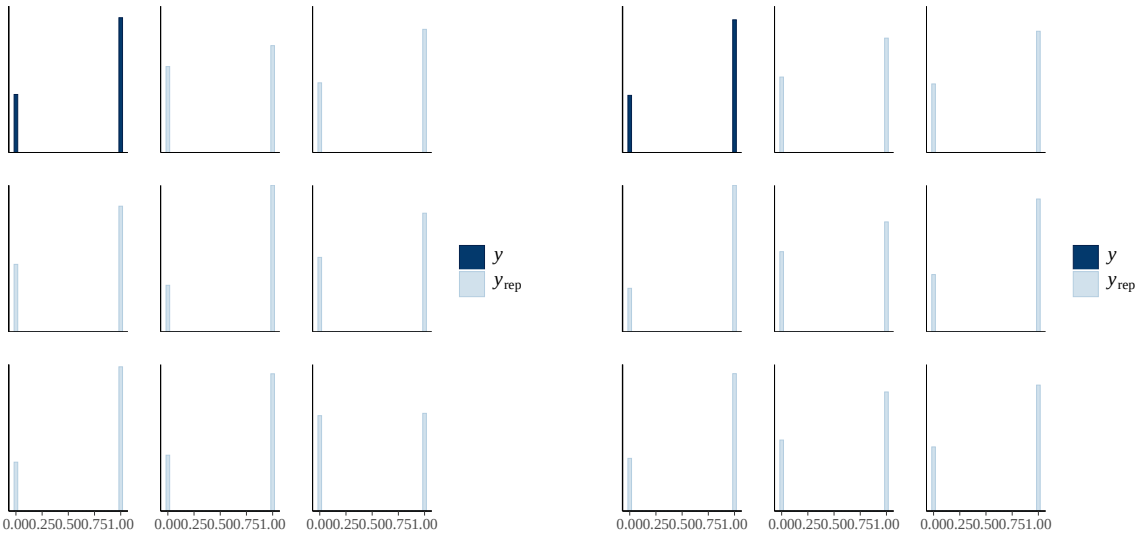
post-warmup draws per chain=500, total post-warmup draws=2000.

Table 3: Results of fitting: first 10 lines

	mean	se_mean	sd	2.50%	25%	50%	75%	97.50%	n_eff	Rhat
alpha[1]	1.25	0.01	0.48	0.43	0.91	1.2	1.55	2.31	1379	1
alpha[2]	0.8	0.01	0.62	0.03	0.31	0.66	1.14	2.29	4055	1
alpha[3]	0.8	0.01	0.59	0.03	0.32	0.71	1.13	2.24	2414	1
rho[1]	1.06	0.01	0.48	0.41	0.71	0.96	1.33	2.21	1024	1
rho[2]	0.78	0.01	0.61	0.03	0.3	0.64	1.15	2.29	2674	1
rho[3]	0.81	0.01	0.58	0.04	0.34	0.69	1.16	2.18	2042	1
sigma	4.09	0.06	3.18	0.13	1.53	3.46	5.87	11.67	3055	1
eta[1,1]	-0.2	0.01	0.55	-1.28	-0.54	-0.2	0.12	0.91	1390	1
eta[1,2]	-0.01	0.02	0.99	-1.88	-0.67	-0.02	0.68	1.93	3173	1
eta[1,3]	0	0.02	0.97	-1.96	-0.64	0.01	0.65	1.92	3117	1

8 Posterior predictive checks and what was done to improve the model

To rationalize our modeling, the basic technique for checking the fit of a model to data is to draw simulated values from the joint posterior predictive distribution of replicated data and compare these samples to the observed data. First, we compare the data to the posterior predictions of the model using the histogram of y, and histograms of several predicted y_{pre}. In Figure 5, we show that the predicted distribution has nearly the same shape as the observed distribution.

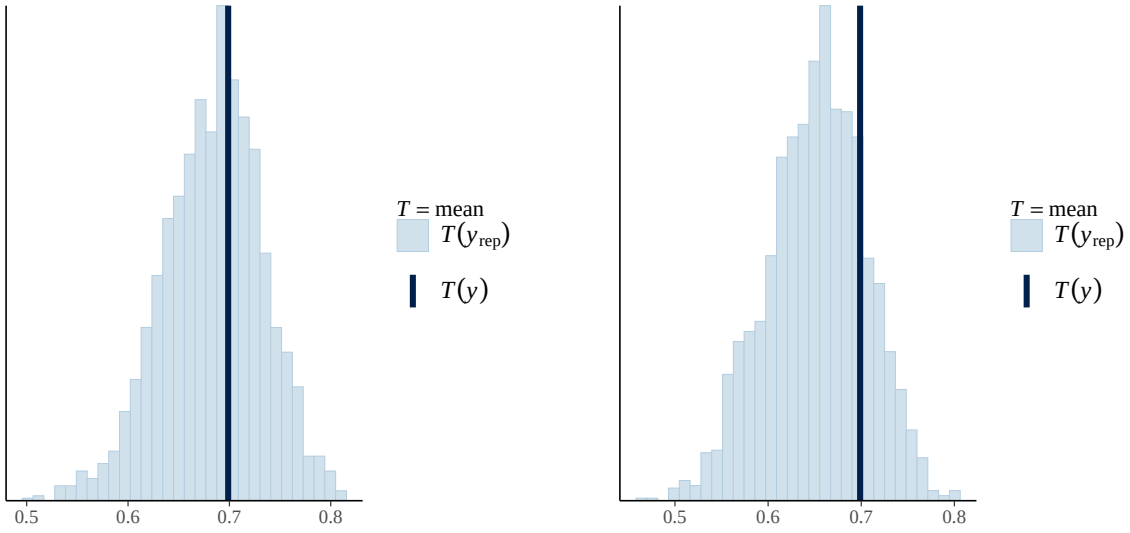


(a) Hierarchical model

(b) Gaussian process model

Figure 5: Histograms of posterior predictive checking

Second, we compute test statistics for the mean of y and test statistics y_{pre} for many replicated datasets. Figure 6 shows the histograms of posterior predictive checking. It can be seen that the two models perform well. The predictive mean values are close to the observed mean.

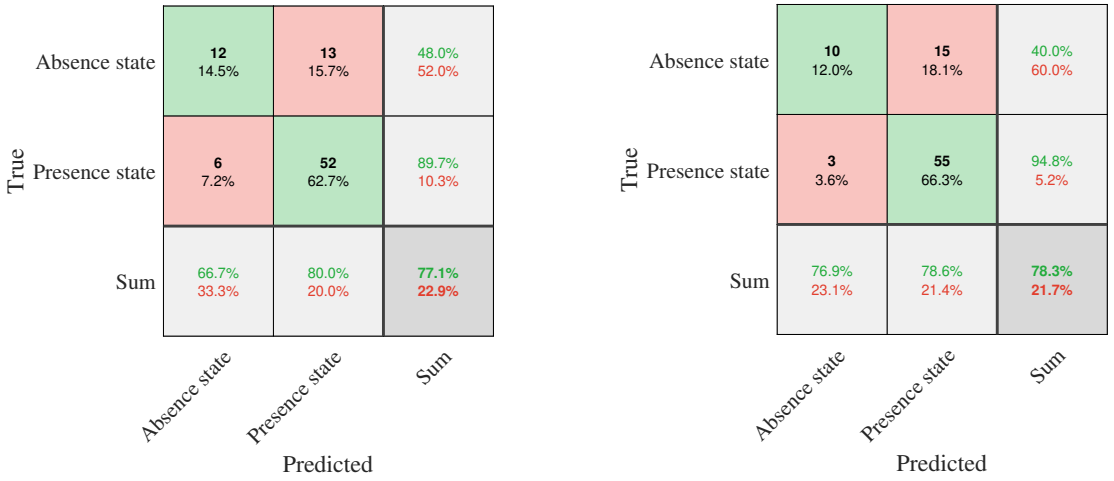


(a) Hierarchical process model (b) Gaussian process model

Figure 6: Histograms of posterior predictive checking

9 Predictive performance assessment

To represent the accuracy of prediction, we plotted the confusion matrix for our prediction, as shown in Figure 7. It can be seen that both two models have an accuracy of more than 77%. The accuracy value can denote the presence of TP. Thus, we can understand that by using the proposed two models, the probability of accurate prediction is more than 77%.



(a) Hierarchical model (b) Gaussian process model

Figure 7: Confusion matrix for classification accuracy

10 Sensitivity analysis with respect to prior choices

In this section, we will show different results using different priors. Even though in the previous sections, we select some weakly informative priors based on experiences, the selection of prior may cause different

results of posterior. Hopefully, the posterior will not be greatly influenced by the selection of priors, which means the proposed model is more robust to the selection of priors. The following two sections will discuss about the sensitivity of prior sections of hierarchical and GP models proposed in this report.

10.1 Prior sensitivity of the hierarchical model

For the hierarchical model, in Section 4.1, we selected the following priors: $\mu_0 \sim N(0,1)$, $\sigma_0 \sim \text{Gamma}(2,2)$, $\mu_1 \sim N(0,3)$, $\sigma_1 \sim \text{Gamma}(3,3)$ After fitting the model, we can get the posterior means of the intercept, and slopes are shown in Eqs. (9)-(12) for the TP values.

$$\mu(\beta_0) = 1.03 \tag{9}$$
$$\mu(\beta_{\text{AREA}}) = 1.25 \tag{10}$$
$$\mu(\beta_{\text{CROP}}) = 1.51 \tag{11}$$
$$\mu(\beta_{\text{SEP}}) = 1.19 \tag{12}$$

In this section, we modified those priors for all hyperparameters, which is shown in Eq. (13).

$$\mu_0 \sim N(0,10), \sigma_0 \sim \text{Gamma}(1,0.5), \mu_1 \sim N(0,30), \sigma_1 \sim \text{Gamma}(3,0.5) \tag{13}$$

The updated stan codes can be found here [4] in comments for the priors. Then, employing the updated priors, the fitting results are shown in Table 4. Full-length results can be found here [5]. The posterior means of the intercept and slopes are shown in Eqs. (14)-(17).

Table 4: Results of fitting: first 8 lines

	mean	se_mean	sd	2.50%	25%	50%	75%	97.50%	n_eff	Rhat
beta0	1.13	0.01	0.35	0.47	0.89	1.11	1.36	1.85	2273	1
betak[1]	1.33	0.02	0.77	-0.08	0.81	1.29	1.8	2.97	2560	1
betak[2]	1.65	0.01	0.6	0.53	1.24	1.63	2.05	2.86	2933	1
betak[3]	1.83	0.02	0.97	0.12	1.14	1.76	2.45	3.92	2164	1
mu_beta0	1.07	0.08	2.49	-4.69	0.19	1.09	1.96	6.6	899	1
sigma_beta0	2.05	0.06	1.87	0.15	0.69	1.49	2.78	7.04	1118	1
mu_betak	1.54	0.06	2.05	-2.8	0.63	1.54	2.52	5.57	1008	1
sigma_betak	2.85	0.06	2.12	0.45	1.35	2.28	3.76	8.67	1351	1

$$\hat{\mu}(\beta_0) = 1.13 \tag{14}$$
$$\hat{\mu}(\beta_{\text{AREA}}) = 1.33 \tag{15}$$
$$\hat{\mu}(\beta_{\text{CROP}}) = 1.65 \tag{16}$$
$$\hat{\mu}(\beta_{\text{SEP}}) = 1.83 \tag{17}$$

From the above results, we can see that even though the priors have been modified greatly, the posterior distribution can be stably captured. It can be seen that the intercept mean is close to the results as before. For the slopes β_{AREA} and β_{CROP}, the posterior means do not change very much. However, for the slope of β_{SEP}, we can see $\mu(\beta_{\text{SEP}}) = 1.19$, and $\hat{\mu}(\beta_{\text{SEP}}) = 1.83$ (53% increase). Its mean value has

4. https://github.com/ZhenkunLi/BDA_Project/blob/main/models/Hier_model_TP_new.stan
5. https://github.com/ZhenkunLi/BDA_Project/blob/main/report/Hier_model_TP_new_updated.txt

relatively changed much. Based on the above analysis, we can conclude that the hierarchical model is robust to the selection of priors, but some slopes may vary to some degree.

10.2 Prior sensitivity of the GP model

For the GP model, in Section 4.2, the priors $\alpha \sim normal(0, 1)$ and $\rho \sim normal(0, 1)$ are selected. In this section, we will discuss the sensitivity of the selection of priors for the GP models. For more robust results of our model, we also set ρ as another commonly used inverse gamma distribution `inv_gamma_lpdf` in Stan's language.

Next, we assess the predictive relevance of priors via sensitivity analysis of the posterior predictive distribution. In Table 5, we observe that the estimates $elpd\_diff$ (expected log predictive density) do not change in a large range. Especially, when we set $\sigma \geq 5$, the $elpd\_diff$ values become close to -47.7. Thus, we state that our model is insensitive to priors α. Although we have done the same experiment on other priors, we omit the result here because of the limitation of space.

Table 5: Prior sensitivity of α for the GP model

σ	elpd_diff	se_diff
1	-48.7	2.6
5	-47.8	2.6
10	-47.7	2.6

Further, we change the prior of ρ to an inverse gamma distribution (with parameters α and β, we used $\alpha = \beta$) as follows

$$\begin{aligned} \alpha &\sim normal(0, 1) \\ \rho &\sim InvGamma(\alpha, \beta) \end{aligned} \tag{18}$$

Here we hold other priors the same as Eq. (8). In Table 6, we observe that the estimates $elpd\_diff$ change in a very small range. In addition, the results show that though using a different prior (i.e., inverse gamma distribution), the results do not change a lot. We contend that the GP model is insensitive to priors.

Table 6: Prior sensitivity of ρ for the GP model

α/β	elpd_diff	se_diff
2	-48.8	2.6
3	-48.9	2.6
5	-48.8	2.6
10	-48.7	2.6

11 Model comparison

In this section, we utilized the leave-one-out validation (LOO) method to compare the proposed two models: hierarchical and GP models. The computed pointwise log-likelihood required for using the LOO package has been added to the `stan` codes that have been introduced in Section 5. For the principle of the loo method, please refer to the reference [8]. The R codes are shown below, in which we printed the results of PSIS-LOO and plotted all k-values.

```r
loo1 <- loo(Hierfit, save_psis = TRUE)
loo2 <- loo(GPfit, save_psis = TRUE)
plot(loo1,diagnostic = c("k", "n_eff"),main = "Hierarchical model")
plot(loo2,diagnostic = c("k", "n_eff"),main = "GP model")
```

From the LOO results, we can see that the estimated $\hat{elpd}_{loo}$ (expected log predictive density) are -38.9 and -39.2, respectively. $\hat{p}_{loo}$ (effective number of parameters) are 0.1 and 0.2. The last line in Figure 8 shows the reliability of the LOO approximation. We can see that all k-values are smaller than 0.5, which means that in this case, all of the estimates for k are fine for both of hierarchical and GP models. Then, we can understand that the distribution of raw importance ratios has finite variance and the central limit theorem holds [9].

```
Computed from 4000 by 68 log-likelihood matrix

         Estimate  SE
elpd_loo    -38.9 2.5
p_loo         0.1 0.0
looic        77.8 5.1
------
Monte Carlo SE of elpd_loo is 0.0.

All Pareto k estimates are good (k < 0.5).
See help('pareto-k-diagnostic') for details.
```

(a) LOO results of the hierarchical model

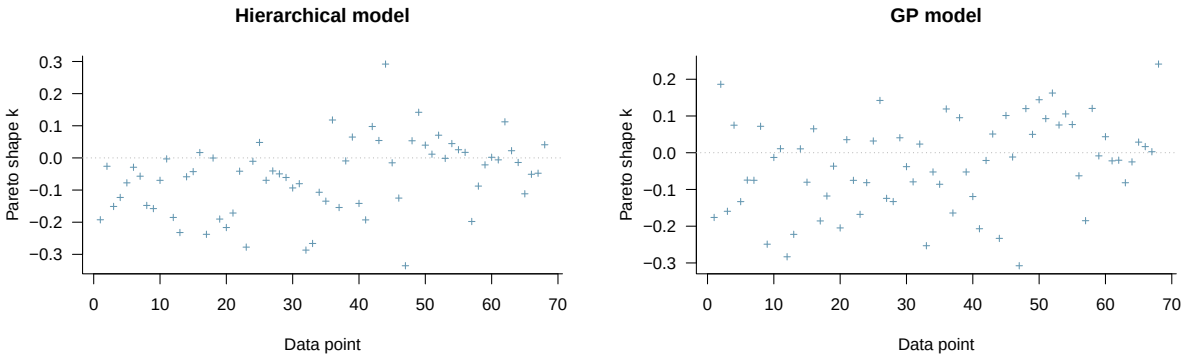
```
Computed from 2000 by 68 log-likelihood matrix

         Estimate  SE
elpd_loo    -39.2 2.4
p_loo         0.2 0.0
looic        78.4 4.8
------
Monte Carlo SE of elpd_loo is 0.0.

All Pareto k estimates are good (k < 0.5).
See help('pareto-k-diagnostic') for details.
```

(b) LOO results of the GP model

Figure 8: LOO results of the proposed two models



(a) k-values of the hierarchical model

(b) k-values of the GP model

Figure 9: k-values of the proposed two models

```r
comp = loo_compare(list("Hier_model" = loo1, "GP_model" = loo2))
```

Table 7: Model comparison

	elpd_diff	se_diff
Hier_model	0.0	0.0
GP_pool	-0.3	0.3

Table 7 shows the comparison results between the hierarchical and GP models. The first column shows the difference in ELPD relative to the model with the largest ELPD. We can see that the difference between the two models is -0.3, and the standard error difference is 0.3. It means that there is no large

difference between the hierarchical and GP models for the utilized data. But if we have to select one, the hierarchical model is a better choice for the prediction of TP occurrence in this report.

12 Discussion of issues and potential improvements

In this section, we will discuss about issues and improvements related to the hierarchical and GP models.

12.1 Possible issues

(1) When fitting the hierarchical model, sometimes there are warning messages.

Warning messages: 1: There were 37 divergent transitions after warmup.

Warning messages: 2: Examine the pairs() plot to diagnose sampling problems

Thus, we can see that we got a small number of divergences. One effective way is to increase the value of adapt_delta. However, increasing adapt_delta > 0.99 doesn't usually make sense [10]. Thus, we are supposed to investigate the identifiability or parameter transformations. We will do this later after the presentation and may ask for help from the teacher or TAs.

(2) Since the priors we select are weakly informative, informative priors will be beneficial to obtain the posterior. For the problem in this report, we may get more data from the corresponding author of the original paper to determine better priors for both the hierarchical and GP models.

12.2 Potential improvements

(1) In this report, we only considered three factors that can influence environmental risk. However, in practice, there are more factors that can impact TP values and the occurrence of different types of fish. In the future development of our proposed model, we are expected to consider more factors and their slopes.

(2) A variety of designs (such as temporal and spatial), data types (such as dichotomous and continuous), and sampling techniques can be used with the joint modeling framework (e.g., estimating detection probability) in our future work.

13 Conclusion

In this report, we employed the Bayesian hierarchical (linear) and GP (nonlinear) models to predict the environmental risk of the watershed. Different from the original paper, where we get the open access data, we consider a different problem. That is to predict the presence-absence state of the environmental stressors (TP, in this report). The main conclusions are drawn below.

(1) Both the hierarchical and GP models are robust to the selection of priors. When different priors are selected for hyperparameters, most posterior means of intercept and slopes do not change very much for the hierarchical model. For the GP model, results show that though using a different prior (i.e., inverse gamma distribution), the results do not change much. We contend that the GP model is insensitive to priors.

(2) The performance of the hierarchical and GP models is similar, with the hierarchical model showing better capability. All k-values of the two used models are smaller than 0.5, meaning that the LOO validation is reliable.

14 Self-reflection of what the group learned while making the project

We appreciate the project work assigned in the CS-E5710 - Bayesian Data Analysis D (BDA) course. The project help students to better understand what they have learned from the course. The project work is more practical compared with 9 assignments in which the prepared data are provided. During the process of finishing this project work, we encountered some problems. By solving these, we learned more general ways of Bayesian models and concepts. Here we have listed some important aspects that we want to share with our peers.

- Apparently, we understand the difference between separate, pooling, and hierarchical models. Even though in this report, only the hierarchical model is utilized, we actually have several different models and different priors. To save the length of this report, we did not cover those here. In the assignments, we understand that because the hierarchical model has hyperparameters that can adjust prior to distributions. However, for our dataset, the Bayesian linear regression model can sometimes perform better than the hierarchical model, whose prior is awful.
- Since we do not have assignments related to the GP process, the project work provides us with a very good opportunity to explore new models. By studying the teacher's demo code, stan documentation, and google answers from others, we learned very much about how to program R and stan codes. These will benefit our careers in the long future.
- By finishing the report, we understand how to compare different models using the PSOS-LOO validation method. Compared to the k-folder cross-validation method, it is a more efficient way to compare different models.
- A more useful skill we learned in the project work is communication. Because the project work is finished in a group rather than by an individual, we must learn to represent our own thoughts to others. We have to consider the best ways to share ideas and results with others. With the development of techniques around us, cooperation is becoming more and more significant. The authors of this report would like to thank the teacher's arrangement for the project.
- The project is finished in Overleaf, and during the project work, we shared our codes in a repo in GitHub. Both of us learned a lot of git commands as we had to exchange codes with each other. In future work, when we have to cooperate with others, using GitHub to commit and share codes is very convenient.
- Because the members of our group are from different departments, we also understand interdisciplinary topics between different schools and departments. The thoughts of different students are based on their understanding of a certain problem. So having a different perspective on a certain problem really helps with understanding it.

References

[1] L. W. Barnthouse, W. R. Munns Jr, M. T. Sorensen, Population-level ecological risk assessment, CRC Press, 2007.

[2] R. J. Miltner, A method and rationale for deriving nutrient criteria for small rivers and streams in ohio, Environmental management 45 (4) (2010) 842–855.

[3] R. W. Martin, E. R. Waits, C. T. Nietch, Empirically-based modeling and mapping to consider the co-occurrence of ecological receptors and stressors, Science of The Total Environment 613 (2018) 1228–1239.

[4] R. A. Hill, M. H. Weber, S. G. Leibowitz, A. R. Olsen, D. J. Thornbrugh, The stream-catchment (streamcat) dataset: A database of watershed metrics for the conterminous united states, JAWRA Journal of the American Water Resources Association 52 (1) (2016) 120–128.

[5] R. B. Moore, L. D. McKay, A. H. Rea, T. R. Bondelid, C. V. Price, T. G. Dewald, C. M. Johnston, et al., User's guide for the national hydrography dataset plus (nhdplus) high resolution., Open-File Report-US Geological Survey (2019-1096).

[6] A. Gelman, J. Hill, Data analysis using regression and multilevel/hierarchical models, Cambridge university press, 2006.

[7] Wikipedia, Gamma distribution, https://en.wikipedia.org/wiki/Gamma_distribution, Last accessed on 2022-11-30 (2022).

[8] A. Vehtari, A. Gelman, J. Gabry, Practical bayesian model evaluation using leave-one-out cross-validation and waic, Statistics and computing 27 (5) (2017) 1413–1432.

[9] A. Vehtari, J. Gabry, M. Magnusson, Y. Yao, Paul-Christian, Diagnostics for Pareto smoothed importance sampling (PSIS), https://mc-stan.org/loo/reference/pareto-k-diagnostic.html, Last accessed on 2022-11-30 (2022).

[10] A. Vehtari, M. Paasiniemi, Bayesian data analysis - RStan demos, http://avehtari.github.io/BDA_R_demos/demos_rstan/rstan_demo.html, Last accessed on 2022-11-30 (2022).