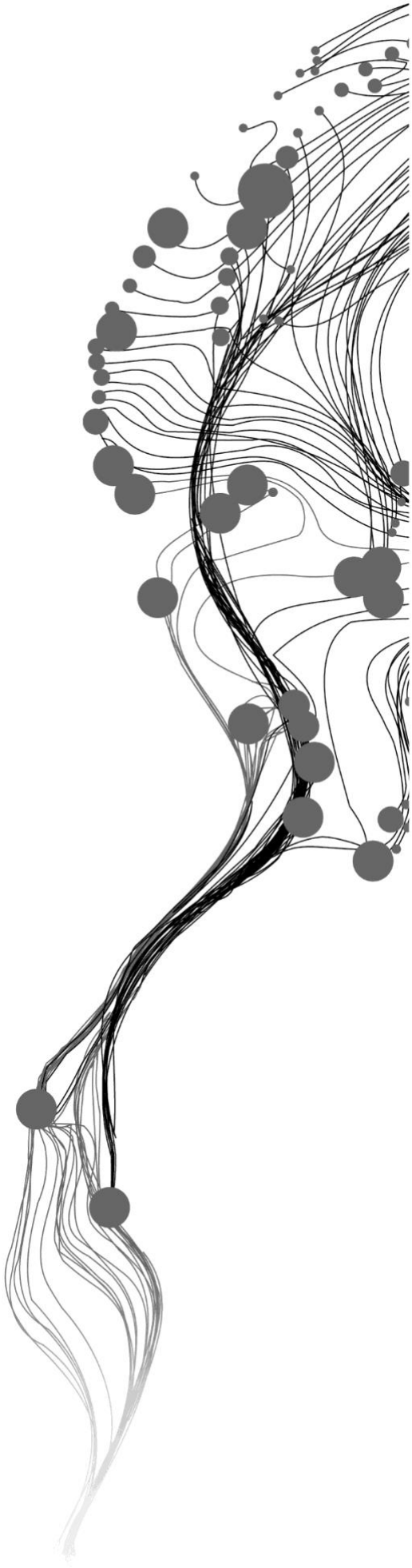
INCORPORATING SPATIAL AUTOCORRELATION IN MACHINE LEARNING

XIAOJIAN LIU
June, 2020

SUPERVISORS:
Dr. Ourania Kounadi
Prof. Dr. Raul Zurita Milla



INCORPORATING SPATIAL AUTOCORRELATION IN MACHINE LEARNING

XIAOJIAN LIU
Enschede, The Netherlands, June, 2020

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: Geoinformatics

SUPERVISORS:
Dr. Ourania Kounadi
Prof. Dr. Raul Zurita Milla

THESIS ASSESSMENT BOARD:
Prof. Dr. M.J. Kraak (Chair)
Dr. E. Izquierdo-Verdiguier (External Examiner, University of Natural Resources and Life Sciences, Vienna, Austria)

ABSTRACT

Applications of machine learning algorithms have witnessed substantial increases in the geoscientific field. However, the predictive performance of these algorithms can be biased if the existing spatial autocorrelation in geospatial data is unattended. This study investigates the approach to account for spatial autocorrelation by introducing additional spatial features in machine learning. We explore the incorporation of two spatial features, i.e. spatial lag and eigenvector spatial filtering (ESF) features, with the widely used random forest (RF) algorithm. Least absolute shrinkage and selection operator (LASSO) selection is introduced to determine the best subset among multiple spatial features that would be included in machine learning. The effects of these spatial features are illustrated on two public datasets of varying sizes (Meuse dataset and California housing dataset). Normal and spatial cross-validation are applied to hyper-parameter tuning and performance evaluation. We utilize Moran's I and local indicators of spatial association (LISA) to assess whether the spatial autocorrelation is captured at both global and local scales. The results show that RF models combined with either spatial lag or ESF features generally yields lower training errors (up to 38% in difference) than the model with no spatial features included. The global spatial autocorrelation of residuals is reduced (up to 95% decrease in Moran's I) when spatial features are included. The local patterns, especially for homogeneous clusters, are weakened as well. However, the generalized error of spatial models increases considerably in spatial cross-validation compared to the error estimated from normal CV (up to 43% in average difference). Normal cross-validation generally returns a lower generalized error which indicates a potential over-optimistic estimate. It can be concluded that the two proposed spatial features are able to account for spatial autocorrelation in machine learning. The differences between normal and spatial cross-validation should be considered whenever a spatial model is evaluated. This study reveals the effectiveness of spatial features in capturing spatial autocorrelation, and provides insights on the usage of spatial cross-validation in performance estimation.

Keywords: spatial autocorrelation, spatial features, machine learning, spatial cross-validation

ACKNOWLEDGEMENTS

I convey my grateful acknowledgement and thanks to my supervisors, Dr. Ourania Kounadi and Prof. Dr. Raul Zurita Milla, for their professional expertise and guidance during my thesis. Their encouragement and kindness will long be treasured in my memories.

Many thanks to Yizhuo Li, Junzheng Zhang, Ran An and many other friends who have always been there for me despite the miles between us.

I thank my family for being unconditionally supportive for my studies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SAC	Spatial autocorrelation
GWR	Geographically weighted regression
EDF	Euclidean distance fields
LISA	Local indicators of spatial association
HH, LL, LH, HL	LISA clusters: High-High, Low-Low, Low-High, High-Low
ESF	Eigenvector spatial filtering
RF	Random forest
LASSO	Least absolute shrinkage and selection operator
CV	Cross-validation
RMSE	Root mean square error
m_{try}	The size of feature subset used for node splitting in random forest

1. INTRODUCTION

The volume of data generated in recent years is increasing tremendously and a large proportion of big data is georeferenced (e.g. remote sensing imagery, GPS trajectories, weather measurements) (Goodchild, 2013). Spatial big data bears the same properties as normal big data like huge volume, high velocity, and high variety. Big data provides new opportunities to uncover previously unknown insights of our world. However, one of the associated challenges with spatial big data lies in developing new methods to handle and analyze complex datasets where traditional approaches may fail (Kitchin, 2013).

Machine learning has demonstrated its versatility for data analysis in different scenarios including face detection, speech recognition, and machine translation. Machine learning methods allow computers to learn from experience. It is powerful to extract information and identify structures from large and high-dimensional datasets (Hoffmann et al., 2019). With unprecedented volumes of geospatial data available in recent years, machine learning has been universally employed in geoscientific research such as land cover classification, soil mapping and atmospheric dynamics (Reichstein et al., 2019). Four major tasks of machine learning include classification, regression, clustering and dimensionality reduction. One of the main utilization of machine learning on geospatial data is spatial prediction where a model is built using training samples to predict unknown values at specific locations (Kanevski, Timonin, & Pozdnukhov, 2009; Shekhar et al., 2015).

In contrast with machine learning, which represents a generic toolset for data analysis, spatial methods specifically aim to analyze data in a spatial context. The nexus of these methods are built upon the first law of geography which states that "everything is related to everything else, but near things are more related than distant things" (Goodchild, 1992; Miller, 2000; Tobler, 1970). Such characteristics of spatial phenomena imply the underlying spatial dependence or spatial autocorrelation (SAC). The presence of this spatial relationship violates the assumption of identical and independent distribution (i.i.d.) upon which many non-spatial statistical methods are predicated. Spatial methods distinguish themselves in explicitly dealing with spatial dependence or spatial autocorrelation that is not addressed by non-spatial models.

Spatial autoregressive (Anselin, 1988) and geographically weighted regression (GWR) (Brunsdon, Fotheringham, & Charlton, 1996) are two commonly used spatial models. Spatial autoregressive models can be configured differently depending on where spatial autocorrelation are introduced (Anselin, 1988; Löchl & Axhausen, 2010). For instance, spatial lag model assumes spatial autocorrelation in the response variable and spatial error model specifies spatial dependencies in the error term. GWR represents a localized linear regression that aims to model spatial heterogeneity by estimating spatially varying parameters (Wheeler, 2014). Another research field that deals with spatial autocorrelation is geostatistics. Kriging is a classic technique in this field that covers a family of methods to interpolate or predict spatial autocorrelated variables. It captures spatial autocorrelation by determining the spatial covariance of samples using a variogram model. However, all these methods mentioned above suffer from divergent drawbacks. Spatial autoregressive and GWR mainly focus on linear relationships. Kriging usually requires assumptions about spatial distribution (e.g. second-order stationary) which may be unrealistic in practice (Fouedjio & Klump, 2019). Additionally, it is difficult to scale kriging and GWR for big spatial computation because of estimation complexity (Kleijnen & van Beers, 2018; Murakami, Tsutsumida, Yoshida, Nakaya, & Lu, 2019).

Machine learning is generally accurate, flexible and scalable for analyzing complex data but may not recognize spatial context. Spatial methods succeed in capturing spatial effects but are limited for applications of non-linear modeling and large-scale computing. Analyses that directly apply machine learning algorithms to spatial data without accounting for the potential spatial autocorrelation could lead to biased outcomes (Dormann et al., 2007; Hengl, Nussbaum, Wright, Heuvelink, & Gräler, 2018; Meyer, Reudenbach, Wöllauer, & Nauss, 2019; Pohjankukka, Pahikkala, Nevalainen, & Heikkonen, 2017).

1.1. Problem statement

The adaptation of machine learning in a geospatial context is a topic that receives attention along with the prominence of artificial intelligence and big data. Present research concerning the incorporation of machine learning and spatial analysis is still preliminary. Existing approaches that have been explored in this field could be roughly categorized in four directions: inclusion of spatial features in original algorithms (Behrens et al., 2018; Hengl et al., 2018; Li, Shen, Yuan, Zhang, & Zhang, 2017), hybrid models with geostatistics (Chen et al., 2019; Foresti, Pozdnoukhov, Tuia, & Kanevski, 2010; Hengl et al., 2015; Hengl, Heuvelink, & Rossiter, 2007), cluster-based methods where cluster analysis on independent variables is introduced as a preprocessing procedure (Mueller, Sandoval, Mudigonda, & Elliott, 2018), and other algorithms exclusively designed for spatial problems such as spatial predictive clustering trees (PCTs) (Stojanova, Ceci, Appice, Malerba, & Džeroski, 2013) and SpaceGAN (Klemmer, Koshiyama, & Flennerhag, 2019).

The aforementioned four directions manifest diverse advantages and unique research values, but it is not feasible to cover all in one attempt. This thesis will investigate the inclusion of spatial features. Feature is a machine learning terminology that is similar to the notion of explanatory variables in statistics. Spatial features refer to relevant variables that can reflect geographical connectivity and spatial relations between observations, thus potentially accounting for spatial autocorrelation (Hengl et al., 2018). Feature engineering represents a crucial process in machine learning which aims to extract and formulate suitable features from raw data for the expected model. Multiple options exist to specify spatial features in present literature: Euclidean distance fields (EDF) which include buffer distances (distance to sampling locations) and coordinates (Behrens et al., 2018), spatial lag based on a definition of neighborhood (Kiely & Bastian, 2019; Li et al., 2017; Zhu, Zhang, Xu, Sun, & Hu, 2019). Adding spatial variables into feature sets aligns with an intuitive and normal procedure in training machine learning models. This technique belongs to a generic feature engineering procedure that it could be extended for various algorithms.

The inclusion of spatial features in machine learning is less explored compared with the combination of machine learning and kriging which can be exemplified by the various applications of regression kriging, neural network with kriging and random forest with kriging (Chen et al., 2019; Foresti et al., 2010; Hengl et al., 2007). Cluster-based methods mainly focus on model interpretability rather than predictive ability. The clusters are usually specified empirically which lacks rigorous justification. The major advantage of the spatial feature approach over exclusively spatial algorithms is that it does not require direct modification of the original algorithm, thus reviving non-spatial machine learning in geographical contexts and maintaining the variety of models that are already established scientifically. In addition, relevant research using spatial features for prediction is fragmented and mostly case-specific. Varying quantifications of spatial features are employed to account for spatial autocorrelation. Hengl et al. (2018) proposed a random forest framework to incorporate distance variants including EDF as geographical covariates for spatial prediction. To our best knowledge, no studies have been made to explicitly figure out how the other spatial features such as spatial lag could be adopted in a general machine learning prediction context, which will serve as the fundamental research problem of this study.

1.2. Objectives

Considering the research gaps and problems mentioned above, the general objective of this study is to investigate the utilization of spatial features in spatially-aware machine learning for prediction.

1.2.1. Sub-objectives & research questions

1) To develop methods for building spatial features.
   Q 1.1: What spatial features can be constructed to potentially account for spatial autocorrelation?
   Q 1.2: How can the spatial features be properly configured? (e.g. standard spatial autoregressive models demand the configuration of a spatial weight matrix to define spatial relationships.)

2) To investigate methods to train machine learning models with spatial features.
   Q 2.1: What effects do cross-validation have on the performance of models with spatial features? (e.g. Brenning, (2012), Pohjankukka et al. (2017), Ruß and Kruse (2010) stated that the presence of spatial autocorrelation can potentially cause standard cross-validation to underestimate the model error.)

3) To evaluate the performance of machine learning models with spatial features.
   Q 3.1: Which spatial features can help to capture spatial autocorrelation and improve prediction accuracy?
   Q 3.2: What variations, if any, do the proposed spatial features have on small and large datasets in terms of the abilities to help with spatial autocorrelation and model accuracy?

2. LITERATURE REVIEW

Adopting machine learning for spatial prediction covers broad concepts from both machine learning and spatial domain. This chapter reviews the basic theoretical aspects and methods that are related to this thesis. Section 2.1describes the notion of spatial autocorrelation and corresponding quantitative measurements for detecting the correlation. In section 2.2, available spatial features that can be extended to machine learning are discussed. Section 2.3 explains fundamental methods involved in the modeling process such as feature selection (2.3.2) and cross-validation (2.3.3). Arguments for the choice of machine learning algorithm and feature selection method used in this study are also provided in section 2.3.1 and 2.3.2 respectively. The last section (2.4) reviews current developments concerning the incorporation of spatial features in machine learning.

2.1. Spatial autocorrelation

Spatial autocorrelation can be considered as a special case of statistical correlation. It assumes that observations at different locations are not independent. It specifically describes the correlation within variables through space (Getis, 2008). For a spatially distributed variable, positive spatial autocorrelation indicates that similar values occur between target location and surroundings while negative spatial autocorrelation implies dissimilar values observed in such locations.

To test spatial autocorrelation, Moran's *I* statistic is the most widely used measurement that is analogous to the Pearson correlation coefficient. For n observations of variable x, it is formulated as follows:

$$I = \frac{n}{S_0}\frac{\sum_i \sum_j w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\sum_i (x_i - \bar{x})^2} \tag{2.1}$$

where i and j are location indices, $\bar{x}$ is the mean of the variable, w_{ij} is the spatial weight between location i and j, and S_0 is the sum of all spatial weights: $S_0 = \sum_i \sum_j w_{ij}$. Moran's *I* varies from -1 to +1. A positive value indicates positive spatial autocorrelation and a negative value indicates otherwise. Zero value means no spatial autocorrelation.

Moran's *I* evaluates the degree of spatial autocorrelation on a global level, but it does not consider the potential local instabilities. Built from decomposition of Moran's *I*, local indicators of spatial association (LISA) was introduced by Anselin (1995) to assess local spatial autocorrelation. This statistics for location i is calculated as:

$$I_i = (n-1)\frac{(x_i - \bar{x})\sum_j w_{ij}(x_j - \bar{x})}{\sum_i (x_i - \bar{x})^2} \tag{2.2}$$

Four groups with significant local spatial autocorrelation (High-High, Low-Low, Low-High, High-Low) can be captured by LISA. High-High (HH) and Low-Low (LL) indicate clustering of high and low values respectively. Low-High (LH) denotes low values surrounded by high values, and the High-Low (HL) group denotes high values surrounded by low values.

2.2. Spatial features

Spatial features are variables that represent spatial relationships of the phenomenon under study. This section introduces two variables from spatial autoregressive models, i.e. spatial lag and eigenvector spatial filtering (ESF).

2.2.1. Spatial lag

Spatial lag is a similar term to the lagged dependent variable in autoregressive time series analysis. Spatial lag linear regression popularized in spatial economics is built on this idea where spatial lag is considered as an additional regressor together with other explanatory variables (Arbia, 2014). A spatial lag model is expressed as follows.

$$y = \rho W y + X\beta + \varepsilon \tag{2.3}$$

The lag term Wy denotes the influence of values from neighboring locations given a target spatial variable (Anselin, 1988). The neighbor structure is expressed through a spatial weight matrix where its element describes the spatial interactions between each paired location in sample data. The spatial weight matrix is usually row standardized. Thus, spatial lag is numerically a weighted variable that captures spatial autocorrelation of the dependent variable in surrounding areas. The spatial lag of location i then is calculated as:

$$Lag_i = \sum_j w_{ij} x_j \tag{2.4}$$

A spatial weight matrix is necessary to construct lag features. In principle, the construction of such a spatial weight matrix involves two procedures: definition of a neighborhood, and calculation of spatial weights. The neighborhood determines which areas are linked. The spatial weights determine the strength of links. It is either determined by binary settings or calculated through distance-based functions such as inverse distance and kernel functions. Different specifications of the matrix represent varying spatial structures. But there does not exist a consensus on the choice of a spatial weight matrix (Bauman, Drouet, Dray, & Vleminckx, 2018).

Three typical approaches of specifying neighborhood include contiguity-based neighbors, k-nearest neighbor, and distance band neighbors.

- Contiguity indicates whether two spatial units share a common border or not. Rook contiguity (Figure 2.1.1) defines neighbors as spatial units sharing a common edge. Queen contiguity (Figure 2.1.2) defines neighbors if two spatial units share a common edge or a common vertex. In addition, the scope of spatial influence can be determined by different orders of neighbors. For instance, second-order neighbors are the neighbors of the first-order neighbors.
- K-nearest neighbor defines a fixed number of neighbors for every spatial unit through the parameter *k*. For a target spatial unit, distances between this unit and all the other units are calculated and ordered. The *k* closest units are considered as the neighbors.
- Distance band defines a fixed distance threshold. When the distance between two spatial units falls under the threshold, these two units are considered as neighbors.

a) Rook contiguity b) Queen contiguity

Figure 2.1. Contiguity-based neighbors

Weight values can also be specified in various forms, and three typical weighting schemes for spatial weight matrix are listed in Table 2.1.

Table 2.1. Examples of weighting schemes for spatial weight matrix

Weighting scheme	Description
Binary weights	$w_{ij} = \begin{cases} 1, & if\ i\ and\ j\ are\ neighbors \\ 0, & otherwise \end{cases}$
Power distance weights	$w_{ij} = d_{ij}^{-\alpha}, \alpha > 0$
Exponential distance weights	$w_{ij} = \exp(-\alpha d_{ij}), \alpha > 0$

2.2.2. Eigenvector spatial filtering (ESF)

Eigenvector spatial filtering (ESF) is a regression technique proposed by Getis and Griffith (2002) to enhance the model results in the presence of spatial dependence. This idea is originated from Moran's *I* where the spatial weight matrix is used to capture the spatial covariations. ESF decomposes a transformed spatial weight matrix and extracts eigenvectors that furnish the underlying latent map patterns (Griffith & Chun, 2014). The spatial weight matrix W is centered by:

$$(I - 11^T/n)W(I - 11^T/n) \tag{2.5}$$

where I is an identity matrix, 1 is a n-by-1 vector of ones. Eigenvectors corresponding to large positive eigenvalues denotes the structure with greater positive spatial dependence (Dormann et al., 2007). These orthogonal and uncorrelated eigenvectors are further utilized as synthetic variables in regression to enable the model to account for spatial autocorrelation (Cupido, Jevtic, & Paez, 2019; Getis & Griffith, 2002; Paez, 2019; Zhang et al., 2018). Constructing ESF also requires the determination of a smaller subset from n eigenvectors. A subsequent stepwise regression is usually used for selection but it suffers from slow computation. LASSO (the least absolute shrinkage and selection operator) can be utilized as a faster alternative (Seya, Murakami, Tsutsumi, & Yamagata, 2015).

Eigen-decomposition is essential for ESF, which is computationally intensive for large samples. To improve computing efficiency, Murakami and Griffith (2018) proposed to approximate the first L ($L \ll n$) eigenvectors using Nyström extension. They employed k-means clustering on the spatial coordinates, and the cluster centers are regarded as the knots for the Nyström extension. At least 200 eigenvectors were advised to be calculated to effectively remove positive spatial autocorrelation with small approximation errors. It was proven that the approximation was able to capture spatial characteristics successfully. ESF, usually used as an exploratory technique, can also be extended to perform prediction on unknown locations by using the approximation. However, this approximation technique cannot deal with negative spatial dependence and is only limited to spatial weight matrices that are based on positive semidefinite kernels such as Gaussian or exponential kernels.

2.3. Machine learning

Machine learning is a broad research field covering numerous algorithms. In terms of prediction tasks, different algorithms can be applied to the same problem, but most of them share similar analyzing procedures which typically involve data collection, data processing, model training, and model evaluation. This research aims to explore the incorporation of spatial features which essentially resemble a feature engineering work in machine learning. Without losing generality, random forest serves as our basis for

discussion because of its general accuracy and successful applications in diverse geoscientific problems. (Foresti et al., 2010; J. Li, Heap, Potter, & Daniell, 2011; Zhu et al., 2019). It has also been used as a framework recently to integrate distance variables in spatial prediction (Hengl et al., 2018). Random forest stands for a reasonable starting point for this study. But it can be changed to any suitable algorithms for specific research requirements.

2.3.1. Algorithm: random forest

Random forest (RF) (Breiman, 2001) is an ensemble of classification and regression trees (CART) that make predictions by aggregating the outputs from all the trees. It is built upon one of the fundamental ideas in ensemble learning called bagging. For each single tree in the ensemble, the training data is reconstructed by iteratively resampling the original dataset with replacement until the new dataset is of the same size as the original one. This resampling process is called bootstrapping. Randomness is also introduced in node splitting of a CART tree. Normally, a split is examined on all features to determine the optimal one that helps the tree learn the best based on a chosen criterion. While in random forest, the optimal splitting is only searched among a randomly selected subset of the whole feature set. The size of the subset is determined by a fixed hyper-parameter which predefines the number of features to be randomly selected. Bagging and random feature subsets in splitting enable random forest to reduce the model's variance without increasing the bias. Random forest is capable to deal with high-dimensional data and robust to noise (Breiman, 2001). The construction of trees of the *originally proposed* random forest algorithm can be summarized as follows:

a) Assuming the dataset contains N samples, resample the dataset with replacement until N "new" samples are retrieved.
b) For each splitting in a tree, randomly select m features from the total M features ($m \ll M$).
c) Assuming the number of trees is T, repeat the above procedures for T times.

For classification, the majority voting strategy is employed to aggregate results from T trees. In regression, the aggregation is conducted by averaging the prediction of all the trees. Two major hyper-parameters are involved in random forest (although other parameters can also be introduced in further variants such as the minimum number of observations in a leaf node): the number of trees T and the size of feature subset m. To obtain optimal performance and stable outputs, it is argued that T should be set to a sufficiently large value (Oshiro, Perez, & Baranauskas, 2012; Probst & Boulesteix, 2017). The default value for the number of feature candidates m in several implementations is usually set to $\sqrt{M}$ and it can be further optimized.

2.3.2. Feature selection

Feature selection aims to reduce the number of features in machine learning models. Using a subset of features allows better interpretability of the model and helps to run the algorithm faster. But feature selection is not an indispensable and obligatory procedure for building data-driven models. Feature selection is especially important for high-dimensional data when the number of features exceeds the available samples. The least absolute shrinkage and selection operator (LASSO) is a regularization method developed by Tibshirani (1996) that is widely used in machine learning for feature selection. It sets a L1 constrain on linear regression and penalizes the coefficients by shrinking a part of them to exactly zero. In a linear regression model denoted as $Y = X\beta + \epsilon$, the ordinary least square (OLS) estimate of β is equivalent to:

$$minimize \frac{\sum_{i=0}^{n}(Y_i - X_i\beta)^2}{n} \tag{2.6}$$

The solution of LASSO is given by minimizing the above equation plus the L1 constraint, which is:

$$\underset{\beta}{\operatorname{argmin}} \left[\frac{\sum_{i=0}^{n} (Y_i - X_i\beta)^2}{n} + \lambda \|\beta\|_1 \right] \tag{2.7}$$

The L1 regularization term is calculated as $\|\beta\|_1 = \sum_{j=0}^{k} |\beta_j|$, mathematically equal to the sum of absolute values of model coefficients. A hyper-parameter $\lambda \geq 0$ controls the strength of L1 penalty in LASSO, which can be tuned by cross-validation. Features with non-zero coefficients are preserved in the final model. The largest value of lambda such that the error is within one standard-error of the minimum is often used for the best model (Friedman, Hastie, & Tibshirani, 2010). Particularly, Seya et al. (2015) prove that LASSO can be efficiently used as an alternative to stepwise eigenvector selection in ESF. Feature selection in this study specifically refers to the selection of spatial lag features and eigenvectors of ESF rather than the original features or their combination, as this procedure only aims to identify the spatial features describing the underlying spatial structures.

2.3.3. Spatial cross-validation

Many machine learning models involve hyper-parameters that cannot be directly learned from data. To obtain robust results, these hyper-parameters have to be tuned (Schratz, Muenchow, Iturritxa, Richter, & Brenning, 2019). Cross-validation (CV) is a technique for model evaluation and is widely applied for hyper-parameter optimization. The fundamental idea of CV is to iteratively divide the data into two parts, i.e. training set and test set. The model is trained on the training set, yet the test set is reserved for evaluation. The basic approach, called k-fold CV, randomly partitions the data into k groups. Each fold will serve as the test set once and the training proceeds on the remaining folds. These k performance measurements are averaged to derive the final evaluation value. Each candidate of hyper-parameter settings is evaluated on the same k folds, then the optimal one can be determined by comparing the averaged evaluation scores.

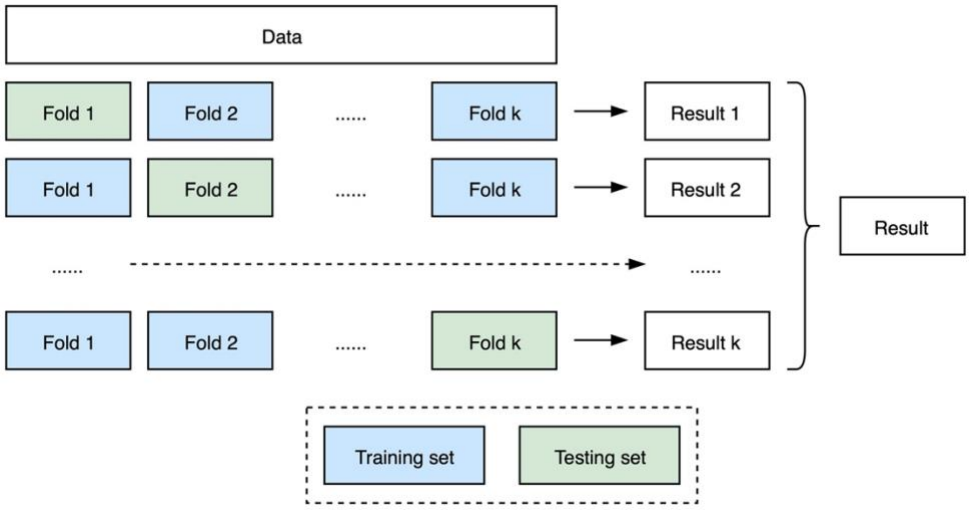


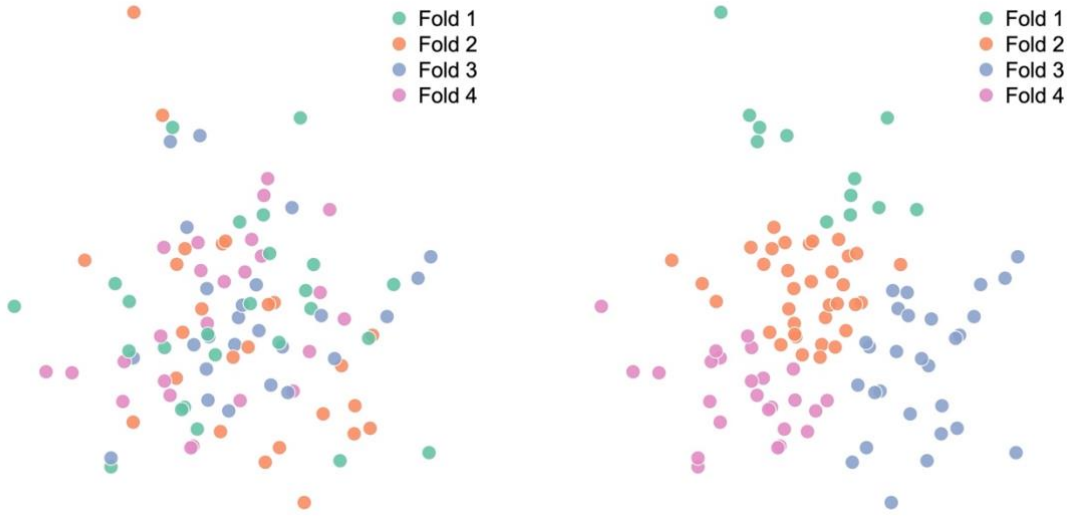
Figure 2.2. k-fold cross-validation

However, cross-validation assumes sample independence. Data samples are randomly assigned to different folds in normal k-fold cross validation. The training and testing samples are spatially close if the data is geo-referenced. With the presence of spatial autocorrelation, training and testing samples are not independent as they inherit similar spatial information. A model fitted on training samples in this scenario will lead to better results on testing samples. Thus, the cross-validation estimates are biased and overoptimistic (Schratz et al., 2019).

To account for spatial autocorrelation, clustering (e.g. k-means) can be used as a preliminary process before resampling (Brenning, 2012; Ruß & Kruse, 2010). Based on k-means clustering on sample coordinates, the dataset is partitioned into spatially contiguous clusters. Then standard cross-validation is performed on these defined clusters. Every cluster obtained from k-means will serve as the testing set once. The above steps are proven to be able to prevent model overfitting on spatial data (Ruß & Brenning, 2010; Schratz et al., 2019). Figure 2.3 shows the difference between normal CV and spatial CV on simulated data.

To be specific, this spatial cross-validation process (k-fold) is described as below:

a) For a predetermined value *K*, perform k-means clustering using the sample coordinates;
b) For each cluster *k*=1, 2, …, *K*:
    1) Take samples in cluster *k* as testing set *test*; take the remaining samples as training set *train*.
    2) Fit a model on *train*, and evaluate the model on *test*.
c) Average the testing results across *K* clusters and report the mean value.



a) Normal cross-validation b) Spatial cross-validation

Figure 2.3. Illustration of normal and spatial cross-validation (4 folds)

2.3.4. Nested cross-validation

The fundamental idea of cross-validation for performance estimates is to separate the dataset into different parts: training and testing. The information of testing samples remains unknown when a model is trained. The result given by cross-validation therefore represents an objective estimate of how the model will generalize on future data. However, the model building process usually involves multiple steps that utilize cross-validation for optimization. When cross-validation is used for multiple times, the information from previous modeling steps is likely to be disclosed subsequent steps. Consequently, the performance estimate may be biased for evaluation. Nested cross-validation is a suitable approach to evaluate the generalization abilities of a model that prevents bias in estimates (Cawley & Talbot, 2010).

Two layers of k-fold cross-validation are included in nested CV (Figure 2.4). The outer CV only serves for estimation while inner CV takes care of other procedures such as hyper-parameter tuning. Inner folds are obtained by splitting the outer training folds. The hyper-parameters are determined by inner CV, then the optimal values are used to fit a model on outer training set. The generalized performance reported by

nested CV is the average over all outer testing folds. The detailed description of a *K*-by-*L* nested CV is illustrated as follows:

a) Split the dataset into *K* outer folds.
b) For each outer fold *k*=1, 2, …, *K*: outer loop for model evaluation
    1) Take fold *k* as outer testing set *outer-test*; take the remaining folds as outer training set *outer-train*.
    2) Split the *outer-train* into *L* inner folds.
    3) For each inner fold *l*=1, 2, …, *L*: inner loop for hyper-parameter tuning
        i. Take fold *l* as inner testing set *inner-test* and the remining as *inner-train*.
        ii. For each hyper-parameter candidate, fit a model on *inner-train* with the combined feature set.
        iii. Evaluate the model on *inner-test* with the assessment metric.
    4) For each hyper-parameter candidate, average the assessment metric values across *L* folds, and choose the best hyper-parameter.
    5) Train a model with the best hyper-parameter on *outer-train*.
    6) Evaluate the model on *outer-test* with the assessment metric.
c) Average the metric values over *K* folds, and report the generalized performance.

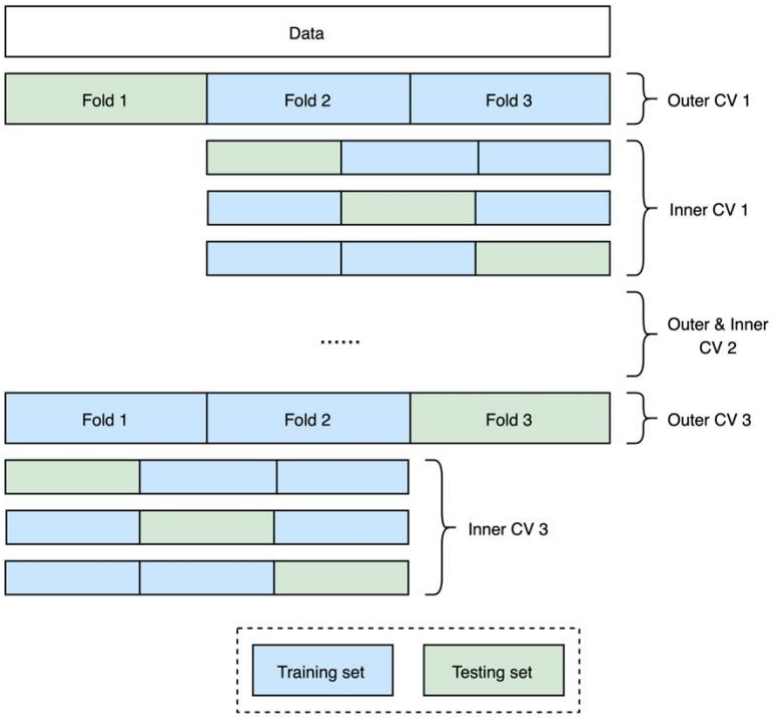


Figure 2.4. Nested cross-validation (3 outer folds, 3 inner folds)

2.4. Current developments

Research on the combination of spatial features and machine learning is emerging these years. Behrens et al. (2018) introduce a spatial modeling framework with generic Euclidean distance fields (EDF) as additional spatial covariates. They combined EDF with other commonly used environmental covariates in the case of digital soil mapping. Six machine learning algorithms were chosen to compare against a reference obtained from regression kriging. The inclusion of EDF enables machine learning to infer spatial autocorrelation when predicting at new locations without an additional step to correct residuals using kriging. Hengl et al. (2018) presented a random forest framework for spatial prediction (RFsp) which

accounts for spatial effects by involving geographical covariates. Multiple distance-based quantifications were proposed including EDF. They evaluated the effectiveness of buffer distances on five environmental datasets. The results demonstrate that RFsp can produce similar predictions as ordinary kriging and regression kriging while RFsp does not demand strict assumptions about distribution and stationarity. The authors also point out that it would be difficult to derive buffer distance variables for datasets that contain a large number of sample points.

Apart from explicit distance-based features, studies on the incorporation of other spatial features and machine learning mainly concentrate on spatial lags. Li et al. (2017) proposed a Geo-intelligent deep learning approach where spatially and temporally lagged PM2.5 terms were combined with satellite-derived and socioeconomic indicators in a deep belief network model. Site-based leave-one-out cross-validation was applied to evaluate the spatial performance. Their analysis proved that including spatial lag as a representation of geographical relations significantly improves the accuracy of PM2.5 estimations. Kiely and Bastian (2019) incorporated spatial lag features into multiple machine learning algorithms to predict real estate sales. The comparison results indicated an enhanced predictive performance of spatially-aware models over non-spatial counterparts. In the work of Zhu et al. (2019), the authors followed the same technique as Li et al. (2017) to include lagged features in several machine learning algorithms. The modified algorithms showed great improvement in terms of accuracy when reconstructing the surface air temperature across China.

However, the research mentioned above adopted varying specifications of spatial lag features and no unified way is proposed to incorporate these spatial features other than distance-based ones. Additionally, none of the studies above except Li et al. (2017) and Zhu et al. (2019) considered the influence of spatial autocorrelation when tuning their models with cross-validation. Kiely and Bastian (2019) discovered overfitting of their models with spatial lag features. Thus, the utilization of spatial cross-validation would be a necessary technique in determining hyper-parameters when spatial features are employed.

In summary, current research confronts three major limitations:

a) Buffer distance features cannot fully satisfy the requirements for all spatial problems especially the ones involving large amounts of data samples, which necessitates the investigation of other possible spatial features.
b) The utilization of spatial lag in machine learning is case-oriented currently. No consistent configuration of this feature is presented such as the specification of spatial weight matrix. It is worthy of adopting this feature under a generic machine learning scenario.
c) The problem of cross-validation is generally neglected when tuning machine learning hyper-parameters with spatial data. The resulting spatial model may suffer from overfitting and under-estimated errors.

3. METHODOLOGY

Previous chapters explain fundamental concepts and related approaches of spatial prediction. Though researchers have made progress in adapting machine learning for spatial data, there still exist certain limitations: first, some spatial features from previous research cannot be effectively applied for large dataset; second, existing spatial lag applications lack proper configuring procedure; third, the issues of cross-validation are not well considered when spatial features are incorporated. This chapter introduces the proposed methodology targeted at these drawbacks. Figure 3.1 illustrates the complete procedure that guides the structure of this chapter. Details are elaborated in subsequent sections. Section 3.1 describes the data on which the experiments are based. The other sections explain how the two spatial features (spatial lag feature and eigenvector spatial filtering) are incorporated in machine learning. Specifically, construction and processing of the two features are illustrated in section 3.2; how to train and evaluate random forest models with spatial features is demonstrated in section 3.3. These two spatial features represent different experiments but share certain procedures. Differences between the two experiments are distinguished separately while the common processes are explained together to avoid verbosity. Specifications of the experiments are summarized in section 3.4. Section 3.5 illustrates the tools or platforms used in this study, and describes what approach is adopted to reproduce this study.

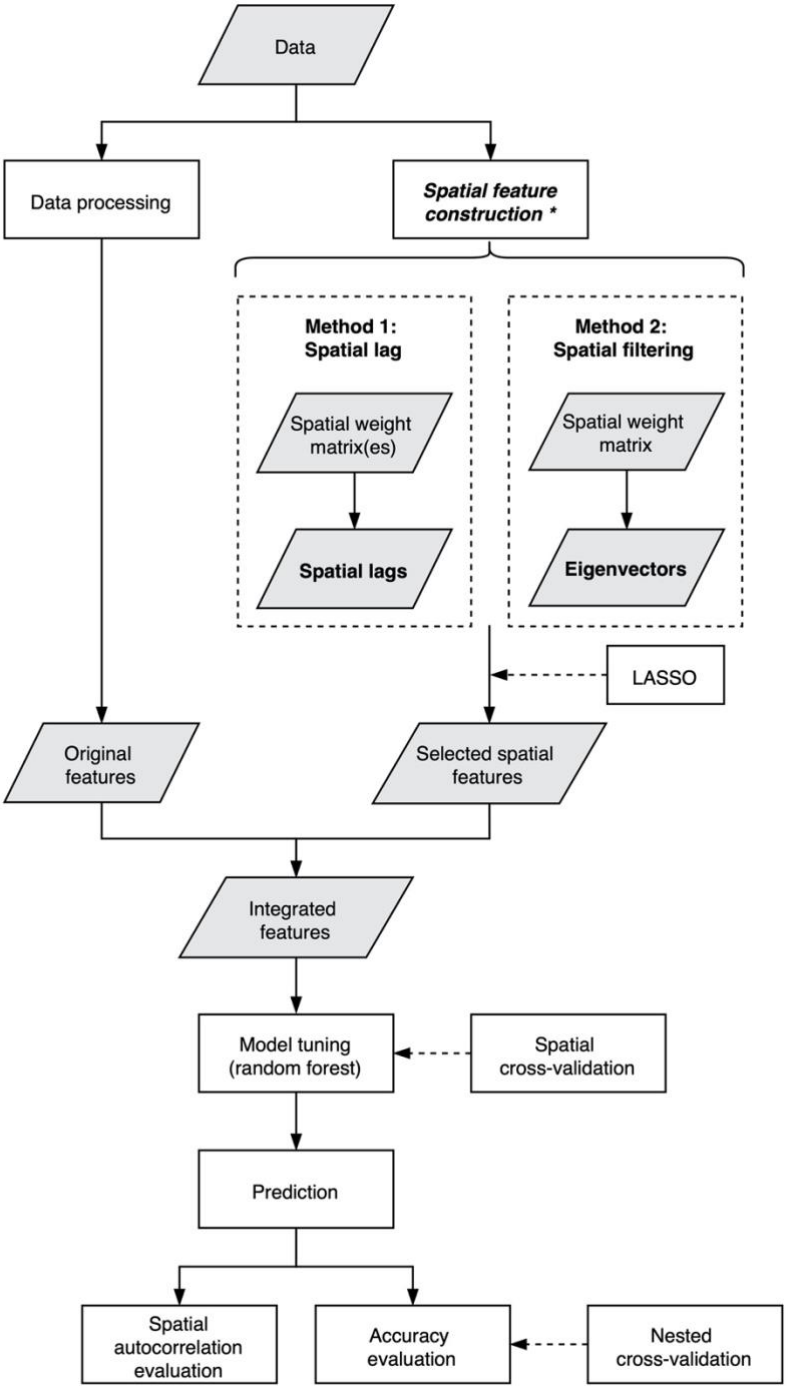


Figure 3.1. Procedures for spatial prediction

3.1. Data sources

Data is the fundamental and essential element for machine learning where models are built to learn the properties of training data. As a method-oriented study that aims to explore possibilities of applying new spatial features in machine learning models, it is important to select representative and effective datasets to demonstrate validity of the proposed methods. Using public and open datasets as benchmark such as Boston housing dataset (Harrison & Rubinfeld, 1978) and ImageNet (Deng et al., 2009) represents a common practice to obtain standardized evaluation of algorithms in machine learning and especially deep learning (LeCun, Bengio, & Hinton, 2015; Russakovsky et al., 2015). Benchmark datasets are usually

carefully collected and curated by professionals to guarantee data quality. Two public spatial datasets with different sizes are used in this study to test the usability of proposed methods.

3.1.1. Meuse river dataset

Meuse is a classical spatial dataset in geostatistics which consists of 155 samples collected in a flood plain of the river Meuse in the Netherlands. Hengl et al. (2018) used Meuse dataset for one of the experiments where distance-based spatial features are introduced in machine learning models. It is internally integrated with several R packages such as 'gstat' (Pebesma, 2004) and 'sp' (Bivand, Pebesma, & Gómez-Rubio, 2013). Four heavy metal concentrations are measured for each sample. Geographical locations are also included together with a number of soil and landscape variables. Details about the data variables are described in Table 3.1. Interpolation of zinc concentration is usually the main focus of this dataset. Flooding frequency and distance to the river can be considered as covariates in regression kriging to predict zinc concentration with the assumption that the river is the main source of zinc. Figure 3.2 shows the distribution of zinc concentrations. Each category has approximately equal number of samples which is determined by quantiles. A higher concentration of zinc is observed along the western riverbank (Figure 3.2). In this study, three samples with missing values are removed which leave 152 samples in total.

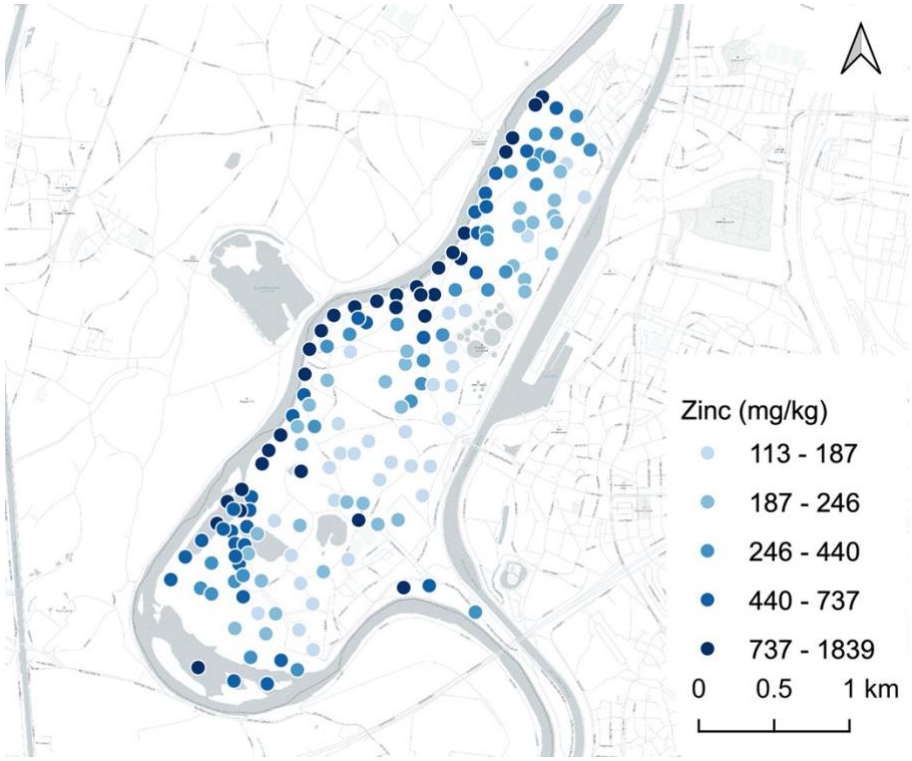


Figure 3.2. Distribution of Meuse data samples (quantile breaks)

Table 3.1. Variable description of Meuse river dataset

Variable	Description	Variable	Description
x	X coordinate (EPSG: 28992)	ffreq	Flooding frequency class
y	Y coordinate (EPSG: 28992)	soil	Soil type
cadmium, copper, lead, zinc	Top soil heavy metal concentration (mg/kg)	landuse	Land use class
elev	Relative elevation above local river bed	lime	Lime class
om	Organic matter	dist	Distance to river Meuse

3.1.2. California housing dataset

This dataset contains 20,640 observations of California housing prices based on 1990 California census data. Each row represents a census block group or district (the smallest geographical unit for which the U.S. Census Bureau publishes sample data). It was originally used by Pace and Barry (1997) to build spatial autoregressive models, and it is considered as a standard example dataset with spatial autocorrelation (Klemmer et al., 2019). Median house price, location of the samples and 6 other explanatory variables are described in Table 3.2. The price values are classified by quantiles in Figure 3.3. Coastal regions usually hold higher house prices, especially for districts around metropolitan cities like San Francisco and Los Angeles (Figure 3.3). Because different districts are populated with varying amounts of households, the total number of rooms or bedrooms will be divided by the number of households in this study to obtain the average variable. The major task is to create a model that can predict the housing price of this region with improved accuracy.

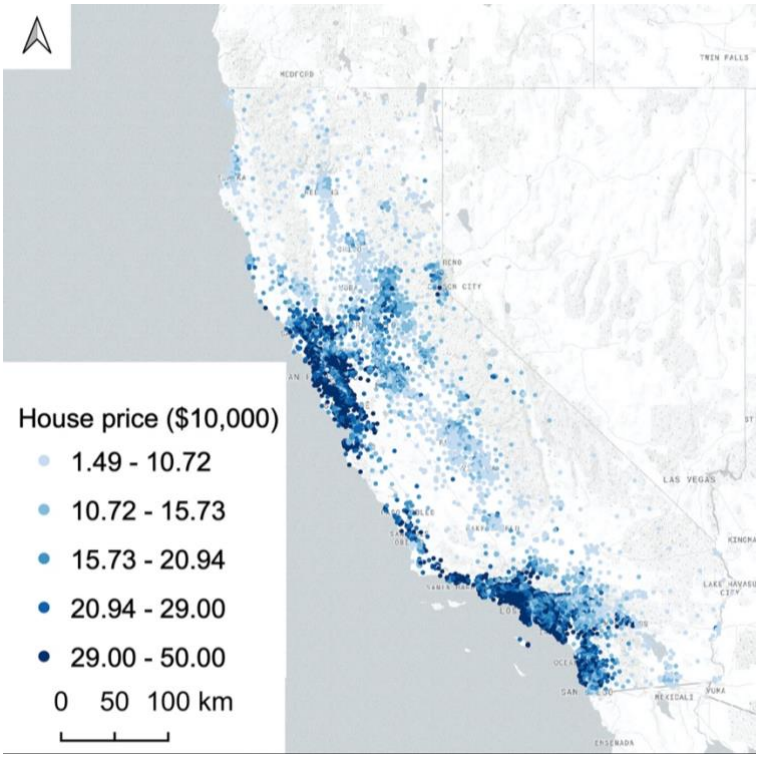


Figure 3.3. Distribution of California housing data samples (quantile breaks)

Table 3.2. Variable description of California housing dataset

Variable	Description	Variable	Description
longitude	WGS 84 coordinate	population	Total population in the district
latitude	WGS 84 coordinate	households	Total households in the district
housing_median_age	Median house age in the district	median_income	Median income of the district
total_rooms	Total rooms in the district	median_house_value	Median house price of the district
total_bedrooms	Total bedrooms in the district		

3.2. Construction and processing of spatial features

3.2.1. Spatial lag features

Many efforts have been invested in selecting an appropriate spatial matrix for spatial autoregressive regression. Rather than one single matrix, different spatial weight matrices can be used to include multiple spatial lags in one regression model aiming to capture different types of dependence (Debarsy & LeSage, 2018). Similarly, we propose to include different spatial lag features in machine learning to accommodate diverse possibilities of spatial representations.

It is possible that not all lag features are representative of the unknown spatial phenomenon, which makes it beneficial to exclude unnecessary ones in terms of model interpretability and complexity. LASSO is further utilized to select a subset of lag features. In our setting, the target variable is regressed only on the constructed lag features. The lag features with non-zero coefficients are selected and further combined with other original features in random forest. However, it should be noted that lag features for a test sample or location with unknown target value can only be derived from a re-built spatial weight matrix, which describes the spatial relations between this single testing location and all the training samples.

In this study, k-nearest neighbor is utilized as it provides a convenient interface to construct spatial weight matrix by changing the value of parameter k. k-nearest neighbor also introduces an adaptive connectivity configuration, in which the number of neighbors is constant but the distance range between neighbors is not fixed compared with the distance-band option. Besides, the weight matrix is row-standardized such that lag features represent the average of surrounding values. Thus, the weight values are:

$$w_{ij} = \begin{cases} 1/k, & if\ i\ and\ j\ are\ neighbors \\ 0, & otherwise \end{cases} \tag{3.1}$$

For Meuse data, an increasing sequence of 5, 10, 15 is used for parameter k to indicate spatial properties at different scales. Three corresponding spatial lag features are constructed in this case. As California housing data covers a larger area, 5, 10, 15, 50 nearest neighbors are employed to generate four spatial lag features. This study follows a data-driven approach and these k values are empirically configured with the purpose to include different possibilities of the neighbors. The subsequent LASSO procedure bears the responsibility to select the best subset of the lag features. The number of neighbors can be changed depending on the data characteristics. K-nearest neighbor can also be replaced by other types of spatial weight matrices to accommodate different needs of other spatial problems.

3.2.2. Eigenvector spatial filtering

As we stated in chapter 2, ESF for prediction is currently only valid for a positive semidefinite spatial weight matrix because of the constraints of Nyström approximation. This study adopts a common exponential kernel from Murakami and Griffith (2018) where the authors demonstrated the usability of ESF in large datasets. The element of the matrix is calculated as:

$$w_{ij} = \exp\left(\frac{-d_{ij}}{r}\right) \tag{3.2}$$

where d_{ij} is the distance between location i and j, and r is given by the maximum length in the minimum spanning tree that connects all the samples. The exponential kernel can be substituted with any kernel function to meet the requirements of other problems as long as the kernel is semidefinite. ESF is capable to explain the spatial patterns by multiple eigenvectors of one spatial weight matrix. Similar to the process of lag features, LASSO is conducted with the extracted eigenvectors. The eigenvector features with non-zero coefficients are selected. Due to the sample size and computational concern of eigen-decomposition, only the first 200 eigenvalues are approximated for California housing data. For Meuse dataset, the exact eigenvalues are calculated without approximation. Eigenvector features of testing samples can only be approximated by Nyström extension for both datasets.

3.3. Machine learning models and evaluation

After finishing the construction of spatial features, this section proceeds to explain how these features are used in building spatial models. The section is organized in three parts. First, sub-section 3.3.1 describes how spatial and non-spatial models are trained. Then, prediction accuracy is examined in 3.3.2. Sub-section 3.3.3 aims to assess whether spatial autocorrelation is successfully accounted in spatial models.

3.3.1. Hyper-parameter tuning

Random forest is used in this study for its general effectiveness and efficiency. To incorporate spatial relationships, original predictors and selected spatial features are combined for spatial prediction. The most influential hyper-parameter in random forest is the number of sub-features used in node splitting ('m_{try}'). As normal cross-validation suffers from spatial autocorrelation, spatial CV is also used for parameter tuning where training and testing set are determined by k-means clusters. The models in our experiments are tuned twice with both normal CV and spatial CV to examine their difference. The number of trees is kept at a moderate size of 200 trees for a balance between computational efficiency and predictive stability. Non-spatial random forest shares the same process such as spatial cross-validation but without spatial features.

To be specific, three models are built in this study: a spatial model with spatial lag features, a spatial one with ESF features, and a non-spatial model. The lag model uses lag features created from multiple spatial weight matrices. The ESF model uses a subset of eigenvector extracted from one kernel matrix.

3.3.2. Accuracy evaluation

The proposed method consists of multiple steps spanning from spatial feature construction, feature selection to hyper-parameter tuning. To retrieve more objective performance estimates of the method, we adopt the idea of nested cross-validation. The optimal hyper-parameters are used in the outer fold to re-train a model. It has to be stressed that nested cross-validation just provides a generalized estimate for the whole procedure. It does not provide optimal hyper-parameters, or produce a model for practical use.

As stated at the start of this chapter, the proposed approach in this study can be generalized into four major processes: construction of spatial features, selection of spatial features (LASSO), model training with specified hyper-parameter values, and evaluation. The nested CV follows the four steps without exception, but implements these in an iterative and nested manner. The nested CV starts from inner loops. To be specific, the spatial features are first generated on the inner training folds. Then LASSO is performed on these spatial features where another intrinsic CV process is used to determine the lambda λ parameter in LASSO with one-standard-error rule. The spatial features with non-zero coefficients are combined with original features. The random forest model is trained with the combined feature set and a specific hyper-parameter setting. In this case, we only consider the number of features for node splitting. After fitting to the inner training folds, the selected spatial features are generated for the inner testing fold. The model predicts on inner testing samples, and the assessment metric is calculated. This inner process is iterated for every inner fold. For every hyper-parameter candidate setting, the assessment values from all inner folds are averaged. The candidate with the best average metric value is considered as the best hyper-parameter for outer training folds. Now the spatial features are re-constructed and re-selected from the outer training folds. A random forest model is trained with the best hyper-parameter identified from the inner CV. Evaluation is conducted on the outer testing fold. The final generalized performance is the averaged metric value across all outer folds.

The nested cross-validation process used in this study is further summarized as follows:

a) Split the dataset into K outer folds.
b) For each outer fold k=1, 2, …, K: outer loop for model evaluation
    1) Take fold k as outer testing set *outer-test*; take the remaining folds as outer training set *outer-train*.
    2) Split the *outer-train* into L inner folds.
    3) For each inner fold l=1, 2, …, L: inner loop for hyper-parameter tuning
        i. Take fold l as inner testing set *inner-test* and the remining as *inner-train*.
        ii. Calculate spatial features on *inner-train*.

iii. Perform cross-validated LASSO on *inner-train* with spatial features, and determine the lambda λ with 'one-standard-error' rule; Select the spatial features with non-zero coefficients.
iv. For each hyper-parameter candidate, fit a model on *inner-train* with the combined feature set.
v. Calculate the selected spatial features on *inner-test*.
vi. Evaluate the model on *inner-test* with the assessment metric.

4) For each hyper-parameter candidate, average the assessment metric values across *L* folds, and choose the best hyper-parameter.
5) Calculate spatial features on *outer-train*.
6) Perform cross-validated LASSO on *outer-train* with spatial features, and determine the lambda λ with 'one-standard-error' rule. Select the spatial features with non-zero coefficients.
7) Train a model with the best hyper-parameter on *outer-train*.
8) Calculate the selected spatial features on *outer-test*.
9) Evaluate the model on *outer-test* with the assessment metric.

c) Average the metric values over *K* folds, and report the generalized performance.

To retrieve a final model that can be used, the four major processes (i.e. spatial feature construction, spatial feature selection, hyper-parameter tuning and evaluation) have to be conducted on all the samples in the dataset. The final model is tuned and trained on the whole dataset. The accuracy value calculated from the final model demonstrates how well the model fits this specific dataset, while the accuracy estimate from nested CV can be considered as an indication of how the final model would perform on potential unseen data.

Various metrics are available to evaluate a model's accuracy. This study employs the commonly used root mean square error (RMSE) which is formulated as:

$$RMSE = \sqrt{\frac{\sum_i^n (y_i - \hat{y}_i)^2}{n}} \tag{3.3}$$

The actual value and predicted value of a sample are denoted as y_i and $\hat{y}_i$ respectively. The difference between the actual value and predicted value, i.e. $(y_i - \hat{y}_i)$, is denoted as residual. n is the number of samples.

3.3.3. Spatial autocorrelation evaluation

Apart from the RMSE for prediction accuracy, another essential aspect for evaluating spatial models is to investigate whether spatial autocorrelation is successfully considered. Traditional prediction models like linear regression assume an independent and normally distributed error or noise term. When such models are directly applied to spatial data without considering spatial effects, the residuals will retain as spatially autocorrelated. Based on this property, the effectiveness of the final spatial models can be examined through the mitigation or elimination of spatial autocorrelation in model residuals (Behrens et al., 2018; Ganapathi Subramanian & Crowley, 2018; Schratz et al., 2019; Zhang et al., 2018). Thus, Moran's I can be used to detect and quantify global spatial autocorrelation in residuals. LISA clusters of residuals are utilized to further examine the existence of local patterns. Both the global Moran's I and LISA are tested under Monte Carlo simulation (Anselin, 1995). In theory, a significant spatial autocorrelation should be observed in residuals from the non-spatial model. The spatial autocorrelation of the residuals from the spatial counterpart is supposed to approximate zero.

3.4. Expriment summary

This study involves experiments of different spatial features on two datasets. Although the tests on two datasets follow similar procedures as indicated in section 3.2 and 3.3, it is necessary to further provide a clear description of all the experiments. Each dataset involves the evaluation of three models (non-spatial, spatial lag, and ESF). The experiments of this study use the 5x3 nested CV (5 outer folds, 3 inner folds) to test the generalization ability. 5-fold CV is used for tuning the final models. Additionally, all the experiments are conducted twice with normal CV and spatial CV separately. In order to examine spatial autocorrelation, the global Moran's *I* of the model residuals is assessed, and then local patterns are further tested as well based on LISA clusters.

Table 3.3. Experiment specification

		Models		
		Non-spatial model	**Spatial lag model**	**ESF model**
Data	Meuse	• Normal CV, spatial CV	• K-nearest neighbor (k = 5, 10, 15) • Normal CV, spatial CV	• Exponential kernel • Normal CV, spatial CV
	California housing	• Normal CV, spatial CV	• K-nearest neighbor (k = 5, 10, 15, 50) • Normal CV, spatial CV	• Exponential kernel • Normal CV, spatial CV

3.5. Software & tools

The implementation of the method is mainly conducted in R 3.6.1 (R Core Team, 2019). The major external R packages utilized in this study are listed in the following table. Python packages from PySAL library are used for calculating local Moran's I.

Table 3.4. Tools and packages

	Name	**Package version**	**Usage**
R	FNN	1.1.3	K nearest neighbor
	foreach	1.4.8	Parallel computing
	glmnet	3.0-2	LASSO
	mlr	2.17.0	Spatial cross-validation
	ranger	0.12.1	Random forest
	sf	0.8-1	Spatial data processing
	spdep	1.1-3	Global Moran's I
	spmoran	0.1.7.2	Eigenvector spatial filtering
Python	libpysal	4.0.1	Spatial weights
	esda	2.0.0	Local Moran's I

3.5.1. Reproducibility

Interactive notebooks, such as Jupyter notebooks, are gaining popularity in both science and industry to facilitate reproducible studies (Perkel, 2018; Shen, 2014). The approaches of communicating procedures and outcomes in geoscientific research should be similar to the overall guidelines of reproducibility regardless of distinguishable characteristics of geographic data (Kray, Pebesma, Konkol, & Nüst, 2019).

Jupyter notebooks with R extensions are used in this study to document the implementing details. All the scripts are available on GitHub (Figure A3 in appendix). The datasets of this project only involve publicly available ones and are under open license which eliminates the obstacles of data availability when rerunning the code.

4. RESULTS & DISCUSSION

This chapter illustrates the performance of the methods and models described in chapter 3. The results on two spatial datasets are presented separately, but they follow the same structure. The first two sections (4.1, 4.2) present the results of three models (non-spatial, spatial lag and ESF) for both datasets. Section 4.3 summarizes the results from two datasets and compares the performance of different models.

4.1. Meuse river dataset

Figure 4.1 shows the distribution and Moran's I value of the zinc concentration for reference. The following sub-sections are dedicated to three models respectively. Within each model, normal cross-validation and spatial cross-validation are both presented. The fold division is random in normal cross-validation while spatial cross-validation considers the geographic distribution of the samples during splitting. The resulting folds from spatial CV are spatially contiguous regions without overlapping. An illustration of the 5 outer folds in nested CV is presented in Figure 4.2. To ensure comparability, the same splitting folds are maintained across the three models, which means the models are tuned and evaluated on the same subset of samples.

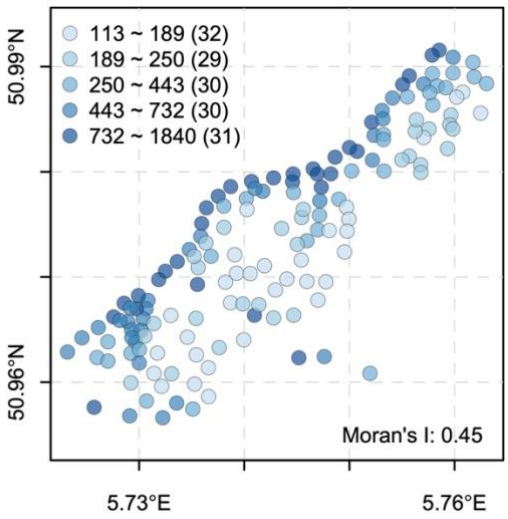
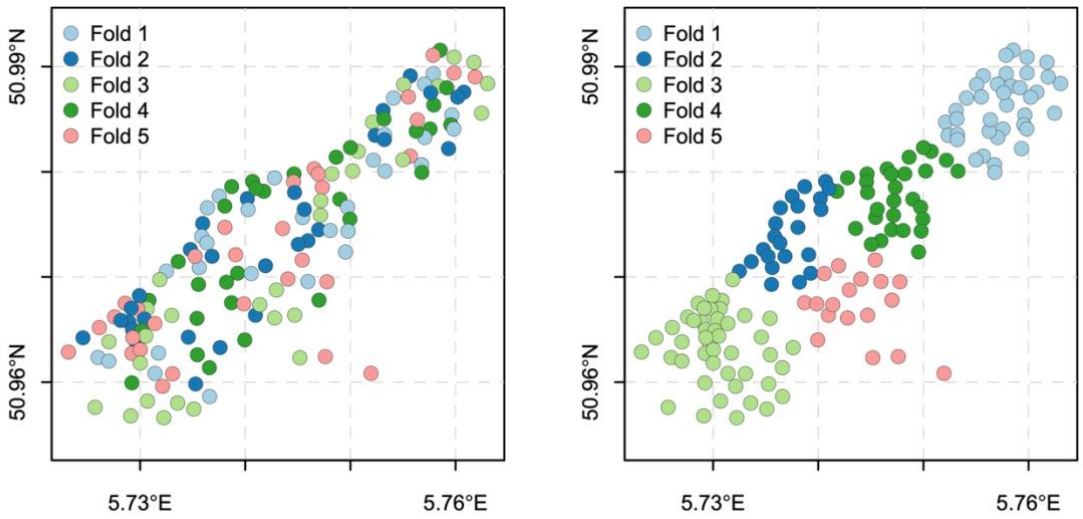


Figure 4.1. Spatial characteristics of Meuse data (quantile breaks). The integer in parentheses refer to the number of samples within each category.



Normal CV (5 folds) Spatial CV (5 folds)

Figure 4.2. Cross validation of Meuse data

4.1.1. Non-spatial model

In this setting, no spatial features are incorporated. Only the original feature set is used for building the model. Table 4.1 illustrates the results from the nested CV. The generalized error is the averaged RMSE over the outer folds. The RMSE of each outer fold from spatial CV and normal CV is listed together for illustration. It by no means indicates that the samples from normal CV are the same as those in the same fold from spatial CV. The generalized error from spatial CV is higher than that from normal CV, which indicates potential over-optimistic estimate of normal CV.

Table 4.1. Accuracy evaluation of non-spatial models (Meuse)

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Generalized error
Normal CV	179.54	123.25	191.55	201.07	259.77	191.04
Spatial CV	265.47	229.24	151.72	268.66	172.75	217.57

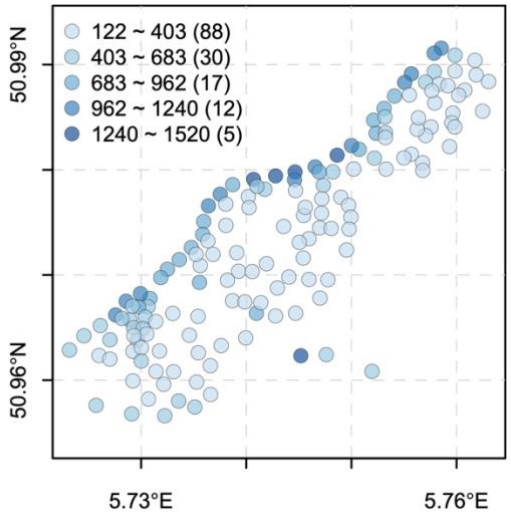
The hyper-parameter of the final model is tuned on all the samples available using non-nested cross-validation. The final model is fitted with the best hyper-parameter. 'm_{try}' represents the number of variables randomly selected as candidates at each split. The training error is the RMSE of the final model. The Moran's I of the residuals is calculated under a spatial matrix of 5-nearest-neighbour. P-value of the Moran's I listed in parenthesis (Table 4.2) is approximated under Monte Carlo simulation of 1000 times. Random forest involves an intrinsic bootstrapping procedure. Usually, the random forest model will be different when retrained each time, even with the same parameter setting. For consistency, we set a sample seed when training the final model, which denotes that the model should be exactly the same whenever the same hyper-parameter, training samples and feature set are used.

Table 4.2. Final evaluation of non-spatial models (Meuse)

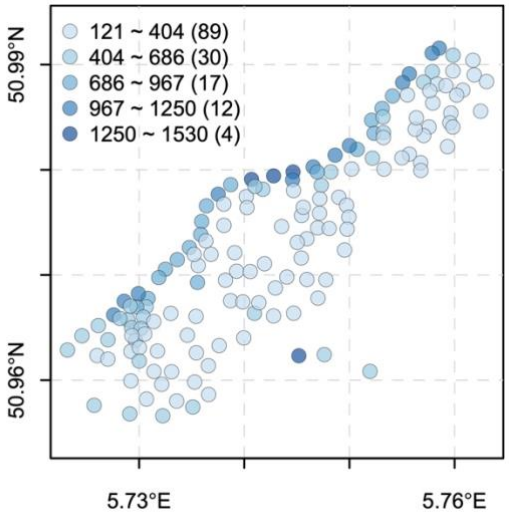
	Optimal m_{try}	Training error	Moran's I of residuals
Normal CV	5	83.59	0.20 (0.001)
Spatial CV	4	86.58	0.18 (0.001)

P-value of the Moran's I listed in parenthesis is approximated under Monte Carlo simulation of 1000 times.

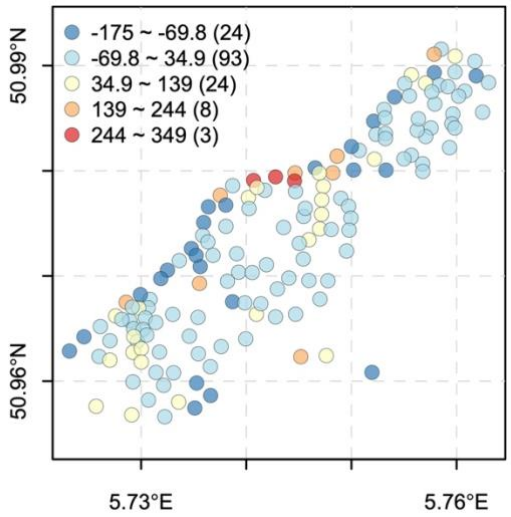
Table 4.2 indicates that the best 'm_{try}' values given by normal CV and spatial CV are slightly different. The residuals from both models still exhibit significant spatial autocorrelation. To investigate the spatial patterns of residuals, the LISA clustering map is presented in Figure 4.3 together with the distribution of predictions and residuals from the final models. The significance level of LISA clustering is set to 5%. All the distributions from the two final models are similar as shown in Figure 4.3.



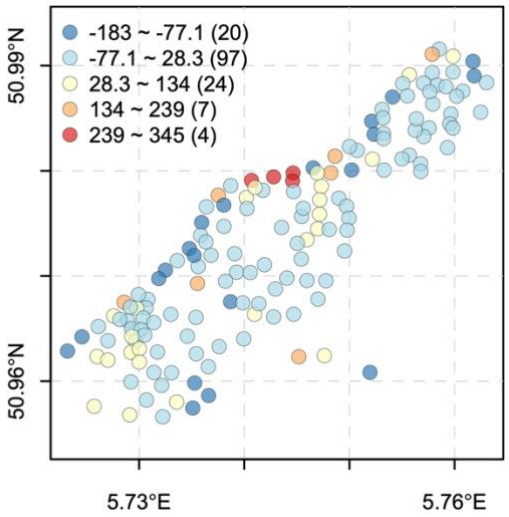
a) Prediction - Normal CV



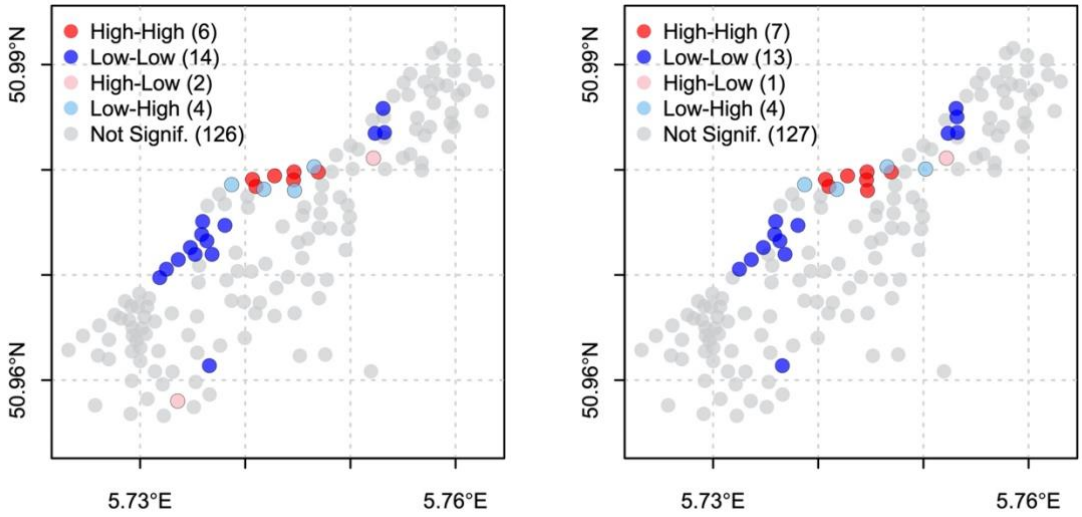
b) Prediction - Spatial CV



c) Residual - Normal CV



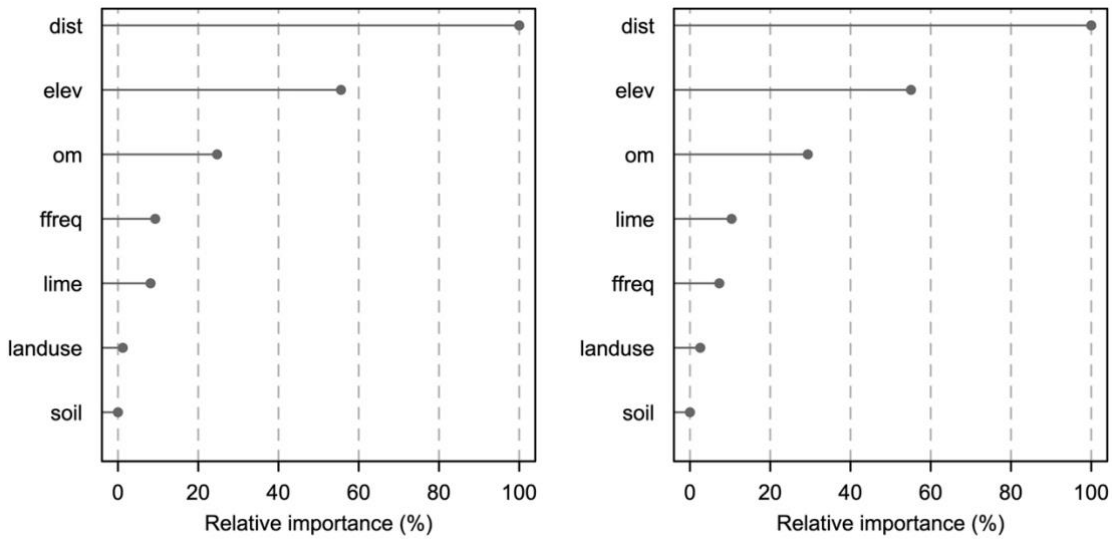
d) Residual - Spatial CV



e) LISA cluster - Normal CV f) LISA cluster - Spatial CV

Figure 4.3. Spatial evaluation of non-spatial models (Meuse). The integer in parentheses refer to the number of samples within each category.

Feature importance of the final model is illustrated in Figure 4.4. Distance to the river has the largest influence on the model and elevation comes at the second place. The soil type has the least influence in the final model.



a) Non-spatial (normal CV) b) Non-spatial (spatial CV)

Figure 4.4. Feature importance of final non-spatial models (Meuse). Relative feature importance is obtained by scaling the original values to 0-100%.

4.1.2. Spatial lag model

Three spatial lag features are constructed initially with the number of nearest neighbors equal to 5, 10, 15 respectively. These lag features are then fed to LASSO for selection. As stressed earlier, the same CV division is retained across three model settings, i.e. the same folds are used for spatial models and the non-spatial counterpart. Table 4.3 and 4.4 describe the results of the nested CV and final lag model respectively.

To avoid potential confusion, it should be noted that the final lag model from normal CV incorporates spatial lag features and is different from the non-spatial model shown in Table 4.2 although the best 'm_{try}' values are both equal to 5.

Table 4.3. Accuracy evaluation of spatial lag models (Meuse)

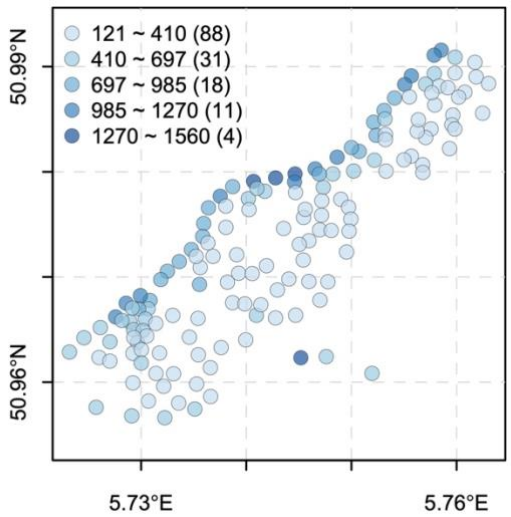
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Generalized error
Normal CV	181.05	120.44	195.43	187.23	229.00	182.63
Spatial CV	250.13	218.40	135.15	284.00	227.22	222.98

Table 4.4. Final evaluation of spatial lag models (Meuse)

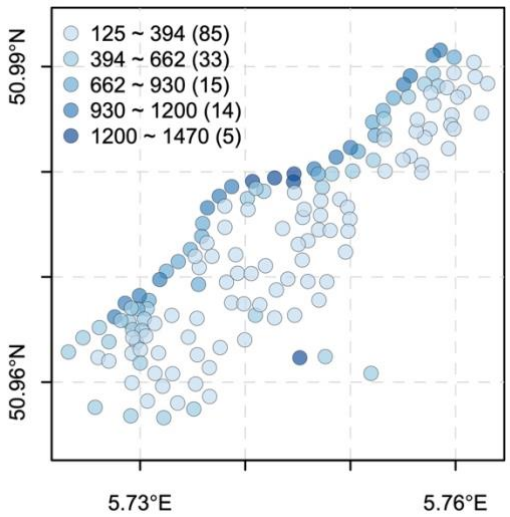
	Constructed spatial features	Selected spatial features	Optimal m_{try}	Training error	Moran's I of residuals
Normal CV	lag_k5, lag_k10, lag_k15	lag_k5	5	79.69	0.029 (0.227)
Spatial CV	lag_k5, lag_k10, lag_k15	lag_k5	2	97.85	0.12 (0.006)

The spatial lag features are differentiated by the k value used for k-nearest-neighbour spatial weight matrix. For instance, 5-nearest-neighbour is used to build "lag_k5". P-value of Moran's I listed in parenthesis is approximated under Monte Carlo simulation of 1000 times.

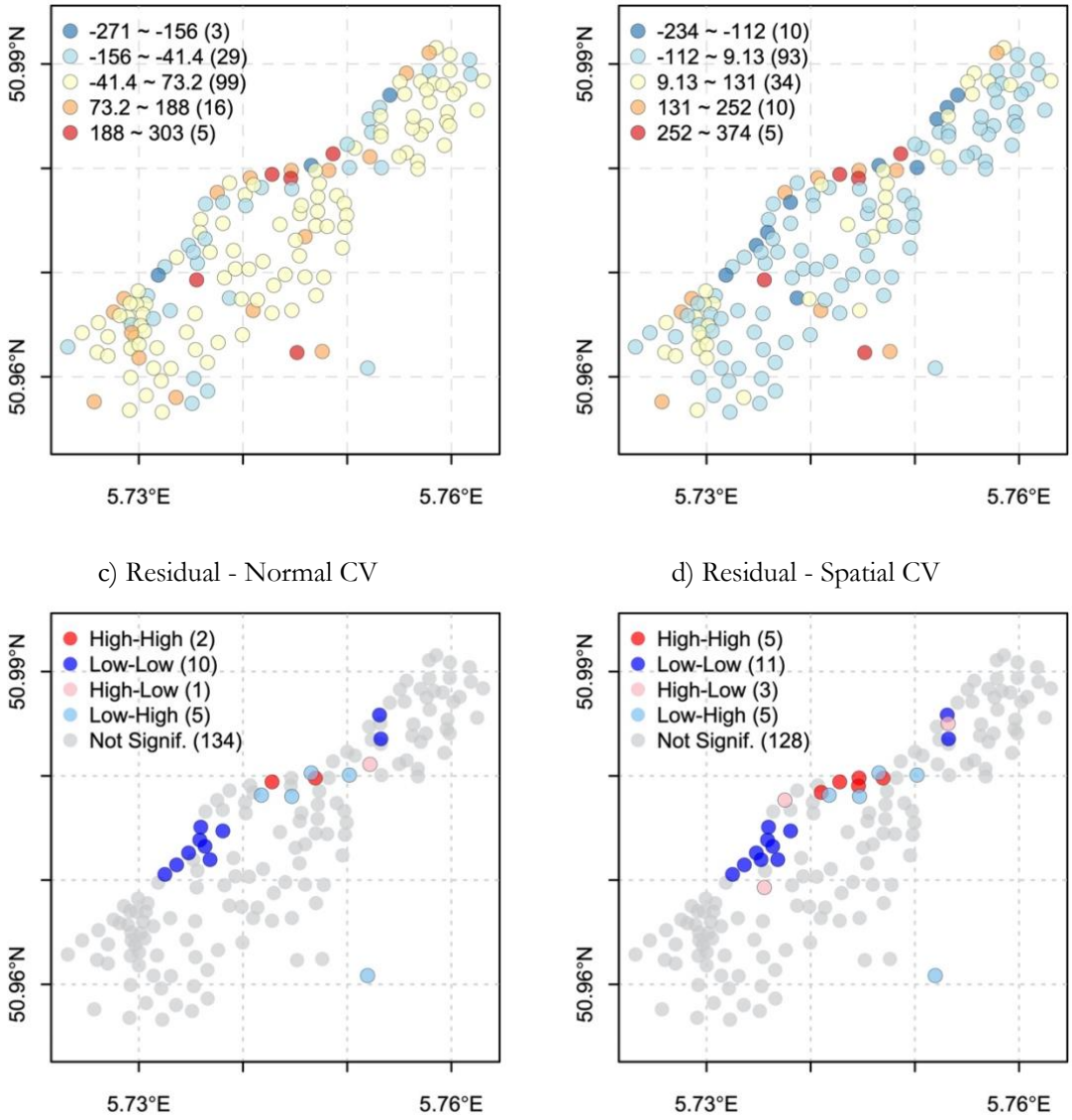
Nested CV results with spatial CV indicate a higher generalized error. The p-value of the final model tuned from normal CV is too large to reject the null hypothesis. There is not enough evidence to claim the residuals have significant spatial autocorrelation for the final model from normal CV in this case. However, the final model with the spatial CV setting still demonstrates an evident spatial autocorrelation in residuals. The residuals of the normal CV model are less skewed than those of the spatial CV when we compare the sample sizes in different intervals (Figure 4.5). The LISA map from normal CV also presents a slight decrease in the size of HH and LL clusters.



a) Prediction - Normal CV



b) Prediction - Spatial CV



c) Residual - Normal CV

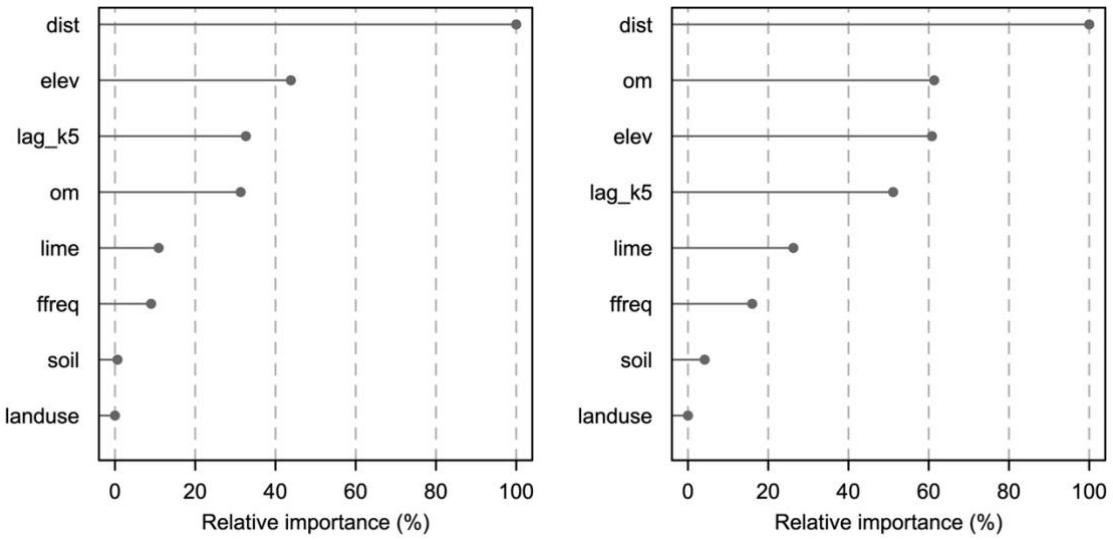
d) Residual - Spatial CV

e) LISA cluster - Normal CV

f) LISA cluster - Spatial CV

Figure 4.5. Spatial evaluation of spatial lag models (Meuse). The integer in parentheses refer to the number of samples within each category.

Distance to the river is still dominant in spatial lag models (Figure 4.6). The importance of the selected spatial lag feature (lag_k5) is approximately at the same level with variable elevation and organic matter.



a) Spatial lag (normal CV) b) Spatial lag (spatial CV)

Figure 4.6. Feature importance of final spatial lag models (Meuse). Relative feature importance is obtained by scaling the original values to 0-100%.

4.1.3. Eigenvector spatial filtering model

An exponential kernel is used to construct semi-definite weight matrix as described in Section 3.2.2. Meuse dataset contains less than 200 samples, so the eigenvalues of the weight matrix are not approximated but rather precisely calculated. The ESF features are then selected by LASSO. Although the eigenvectors for training samples are calculated explicitly, the eigen-features for testing samples can only be approximated through Nyström extension. The accuracy evaluation of ESF models is presented in Table 4.5 and 4.6. The tuned 'm_{try}' value through normal CV is still the same with non-spatial and lag models, but it must be mentioned that these three models are distinct as they are trained on different feature sets. Non-spatial models do not include any spatial features; spatial lag models incorporate spatial lag features; ESF models involve eigenvector features.

Table 4.5. Accuracy evaluation of ESF models (Meuse)

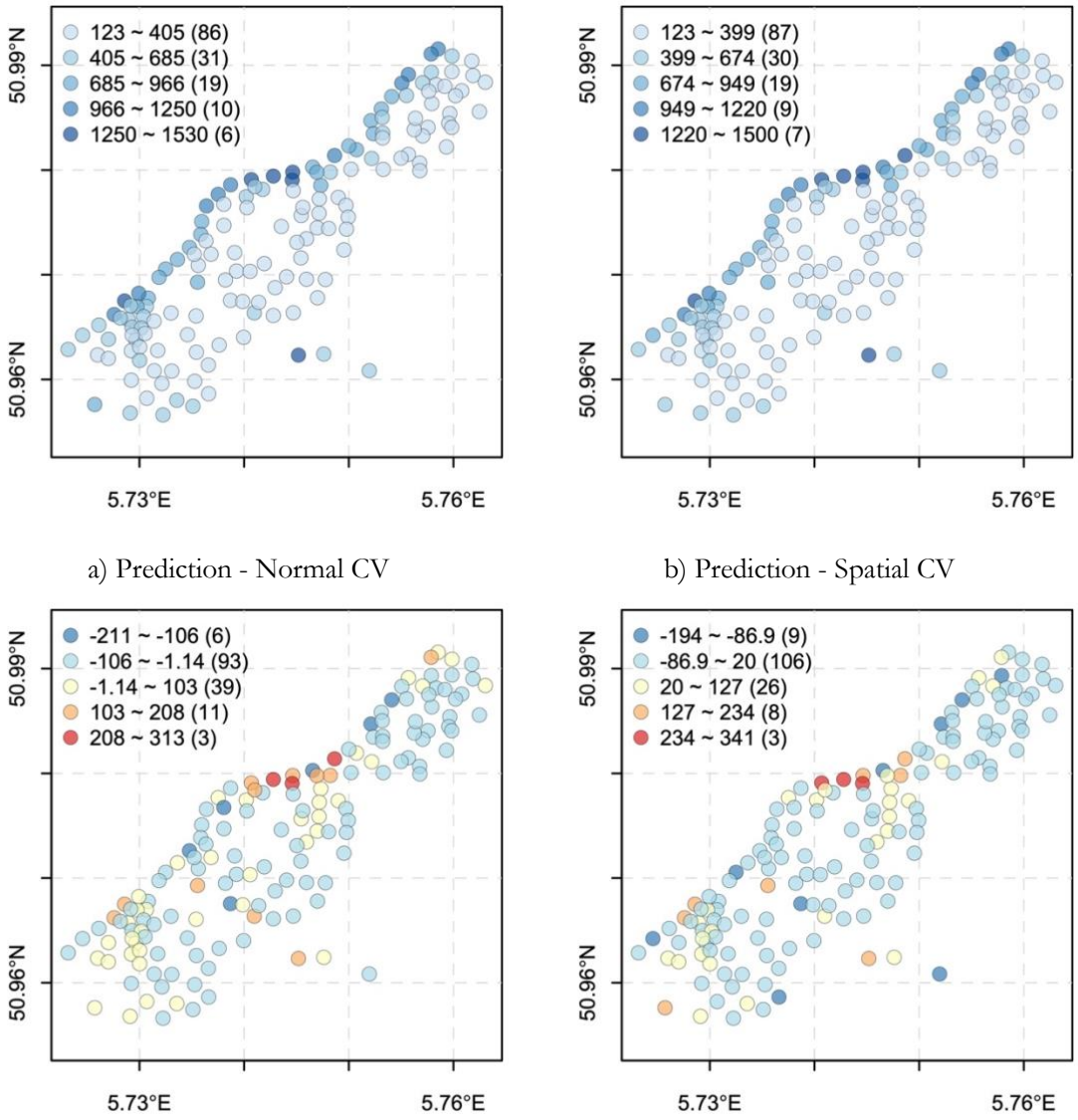
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Generalized error
Normal CV	149.87	109.02	182.29	176.88	241.04	171.82
Spatial CV	265.47	229.24	151.72	268.66	172.75	217.57

Table 4.6. Final evaluation of ESF models (Meuse)

	Constructed spatial features	Selected spatial features	Optimal m_{try}	Training error	Moran's I of residuals
Normal CV	ev1 ~ ev152	ev8, ev11, ev12	5	75.52	0.19 (0.001)
Spatial CV	ev1 ~ ev152	ev8, ev11, ev12, ev34	6	78.10	0.15 (0.001)

Eigenvector features are indicated by "ev" and a number. "ev1" represents the eigenvector corresponding to the largest eigenvalues, and likewise. P-value of the Moran's I listed in parenthesis is approximated under Monte Carlo simulation of 1000 times.

The generalized error for nested spatial CV is still larger. The spatial autocorrelation of the residuals is significant for both the final ESF models. Similarity is observed in the spatial distributions of predictions, residuals and LISA clusters (Figure 4.7).

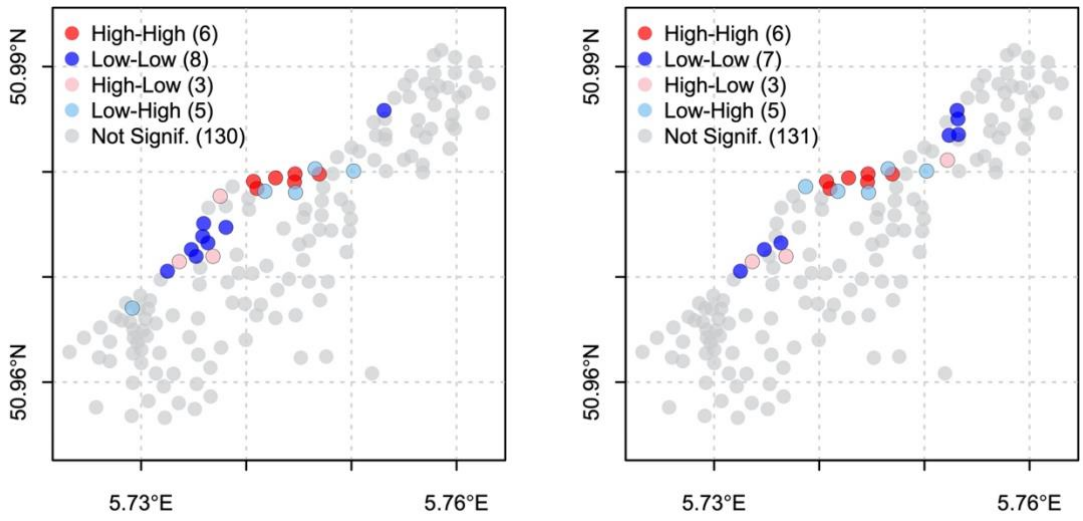


a) Prediction - Normal CV

b) Prediction - Spatial CV

c) Residual - Normal CV

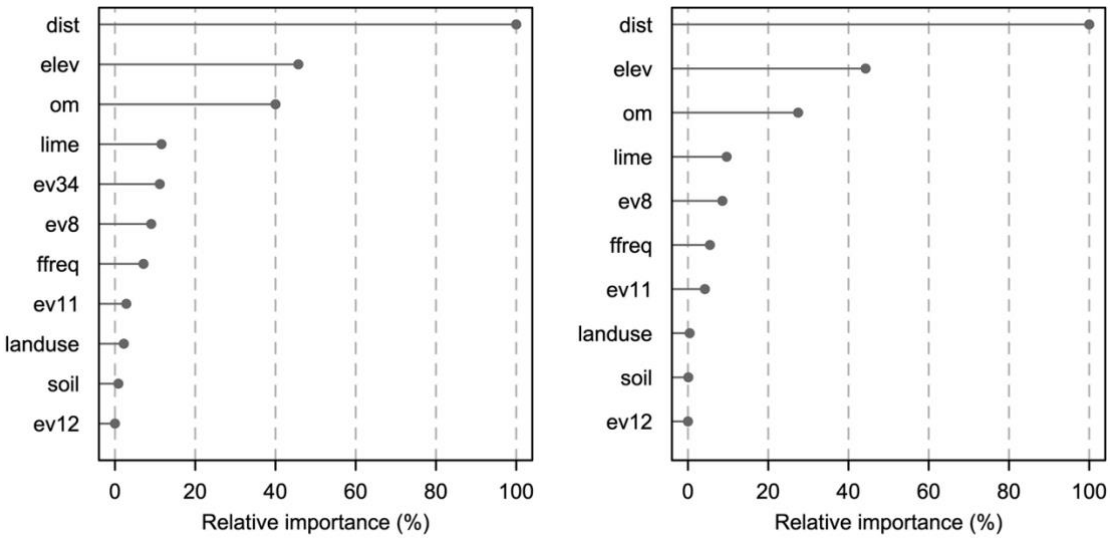
d) Residual - Spatial CV



e) LISA cluster - Normal CV f) LISA cluster - Spatial CV

Figure 4.7. Spatial evaluation of ESF models (Meuse). The integer in parentheses refer to the number of samples within each category.

The top three important features of ESF models are the same as non-spatial models (Figure 4.8). The ESF features are not that influential when fitting the models.



a) ESF (normal CV) b) ESF (spatial CV)

Figure 4.8. Feature importance of final ESF models (Meuse). Relative feature importance is obtained by scaling the original values to 0-100%.

4.2. California housing dataset

This section presents the results for California housing dataset in the same structure as the Meuse models. Figure 4.9 shows the distribution of housing prices and the Moran's *I*. Non-spatial models are firstly described followed by spatial lag and ESF models. The generalized performance is evaluated by 5x3 nested CV and the final model is tuned by 5-fold CV. The division of outer folds are presented in Figure 4.10.

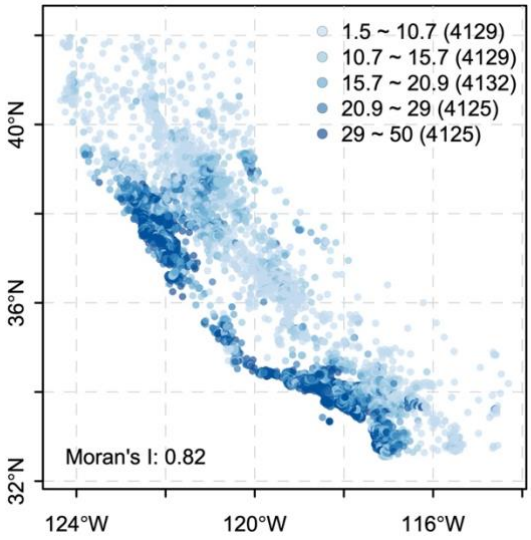


Figure 4.9. Spatial characteristics of California housing data ($10,000). The intervals are determined by quantile breaks. The integer in parentheses refer to the number of samples within each category.

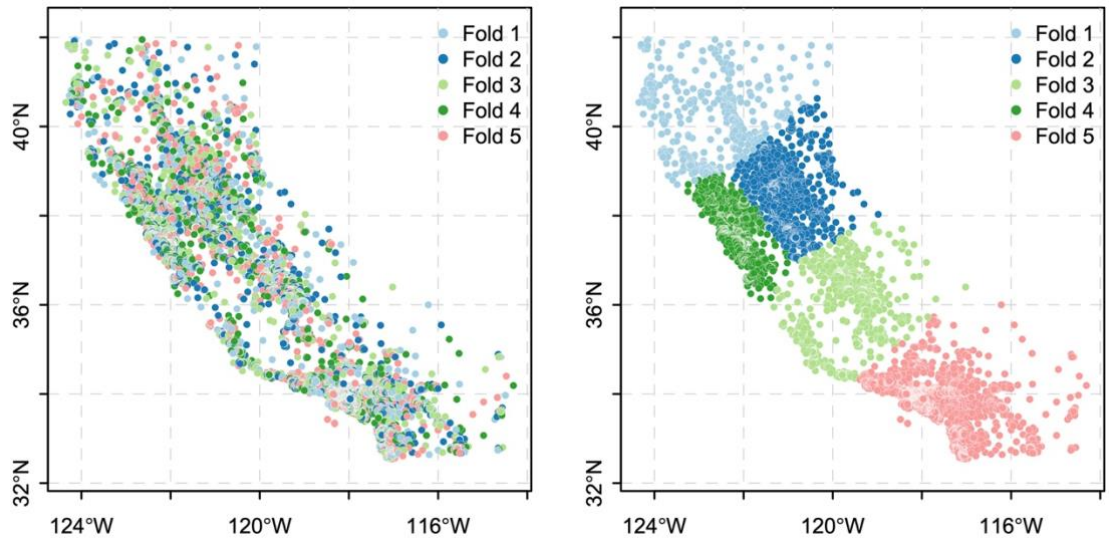


Figure 4.10. Cross validation of California housing data

4.2.1. Non-spatial model

The non-spatial model is built with the explanatory variables except the latitude and longitude as shown in Table 4.7. The final model is fitted on all data samples (Table 4.8). The generalized errors from the two CV methods are approximately at the same level. However, the RMSEs from outer folds in spatial CV show more volatility while the RMSEs from normal CV remains stable. The spatial autocorrelation persists in residuals of the two final models.

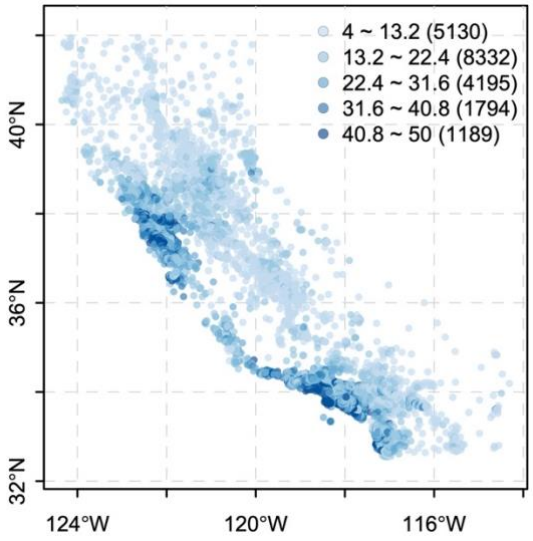
Table 4.7. Accuracy evaluation of non-spatial models (CA)

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Generalized error
Normal CV	65589.35	64799.53	66965.33	68654.93	63721.71	65946.17
Spatial CV	47419.24	58712.81	72584.38	75477.13	71218.24	65082.36

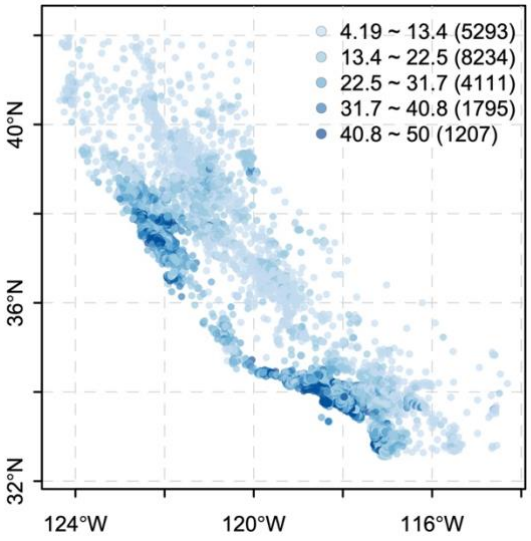
Table 4.8. Final evaluation of non-spatial models (CA)

	mtry	Training error	Moran's I of residuals
Normal CV	2	29857.57	0.42 (0.001)
Spatial CV	3	29086.55	0.40 (0.001)

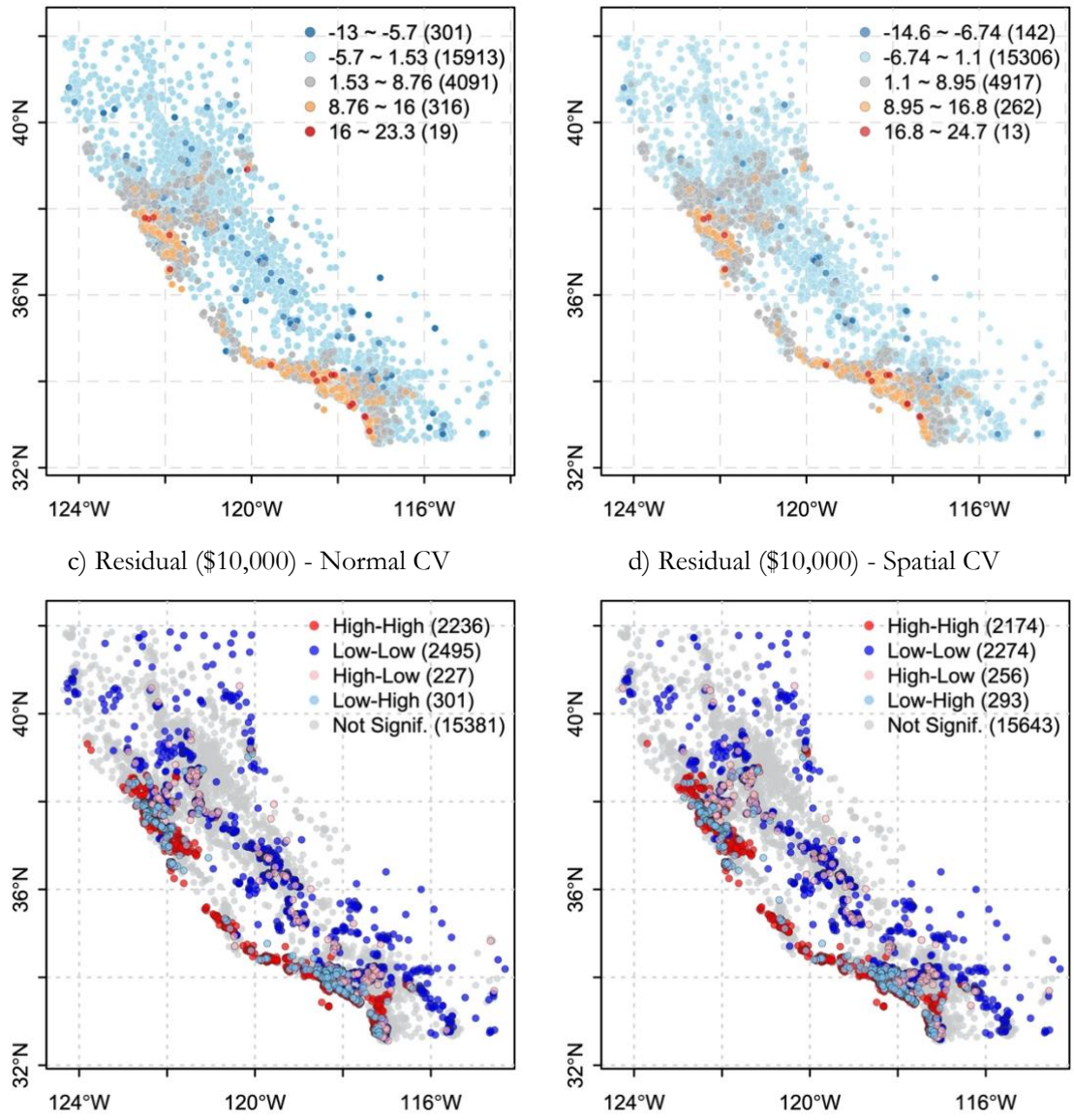
Figure 4.11 demonstrates the distribution of predictions, residuals and LISA clusters from the final model. The patterns are essentially similar. The coastal region presents more clusters of high housing prices (HH). The LL clusters are dispersed across eastern areas.



a) Prediction ($10,000) - Normal CV



b) Prediction ($10,000) - Spatial CV



c) Residual ($10,000) - Normal CV

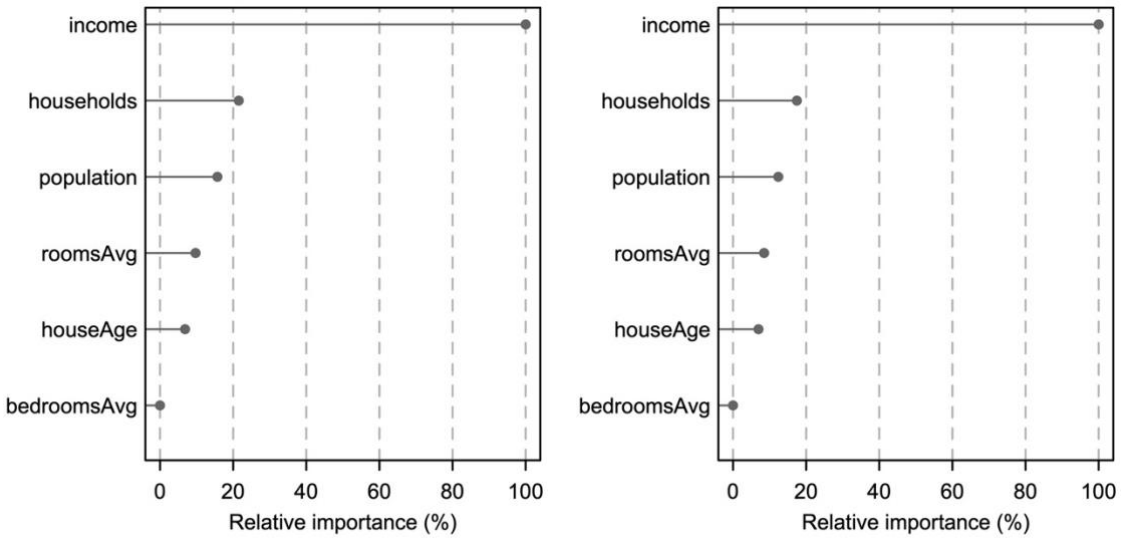
d) Residual ($10,000) - Spatial CV

e) LISA cluster - Normal CV

f) LISA cluster - Spatial CV

Figure 4.11. Spatial evaluation of non-spatial models (CA). The integer in parentheses refer to the number of samples within each category.

In Figure 4.12, income has a major impact on the final model. The other variables are much less influential compared with income.



a) Non-spatial (normal CV) b) Non-spatial (spatial CV)

Figure 4.12. Feature importance of final non-spatial models (CA). Relative feature importance is obtained by scaling the original values to 0-100%.

4.2.2. Spatial lag model

For the spatial weight matrix, lag features of 5, 10, 15, 50 nearest neighbors are employed. When training the final model, the identical lag features are selected from the two CV methods and the optimal hyper-parameters are also the same (Table 4.10). Consequently, the final models from these two settings share the same results. The generalized error from nested spatial CV is nearly twice the error from normal CV (Table 4.9). The RMSEs of outer fold 4 and 5 in spatial CV are extremely high when these two folds are evaluated as testing samples. These two folds correspond to the regions of San Francisco and Los Angeles respectively where the samples are more densely distributed (Figure 4.9, 4.10). In terms of final models, the training error of the final spatial lag model is lower than that of the non-spatial one indicated in section 4.2.1. Besides, the residuals do not show significant spatial autocorrelation anymore. The residuals appear to be more symmetrically distributed in statistics (Figure 4.13b). The HH and LL clusters are greatly reduced as well (Figure 4.13c).

Table 4.9. Accuracy evaluation of spatial lag models (CA)

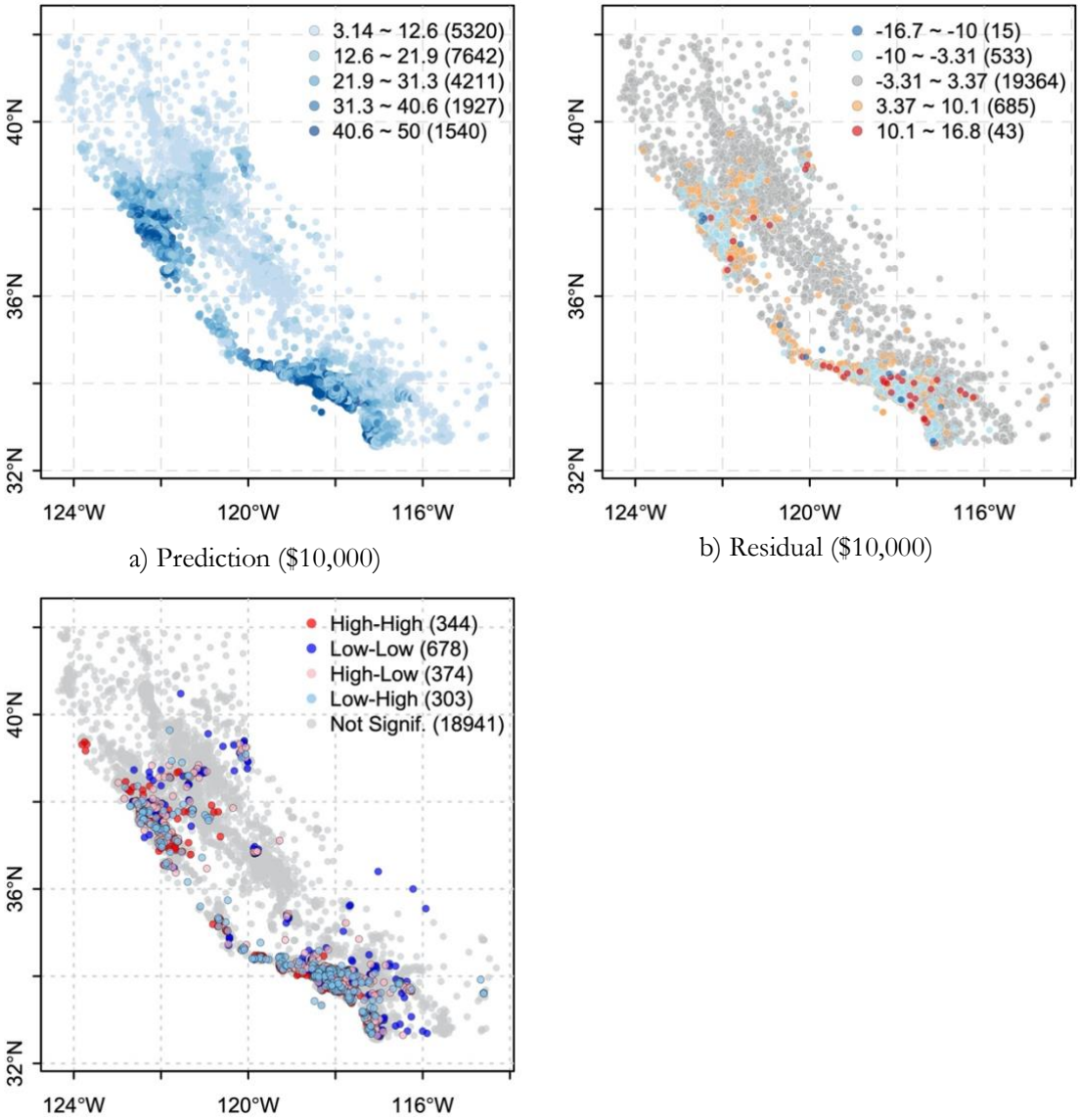
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Generalized error
Normal CV	44018.01	43306.16	45092.36	44457.47	43300.77	44034.95
Spatial CV	44206.38	76967.17	60853.46	135877.61	104218.33	84424.59

Table 4.10. Final evaluation of spatial lag models (CA)

	Constructed spatial features	Selected spatial features	Optimal m_{try}	Training error	Moran's I of residuals
Normal CV	lag_k5, lag_k10, lag_k15, lag_k50	lag_k5, lag_k10, lag_k15	6	17949.20	0.023 (0.999)
Spatial CV	lag_k5, lag_k10,	lag_k5,	6	17949.20	0.023 (0.999)

	lag_k15, lag_k50	lag_k10, lag_k15

The spatial lag features are differentiated by the k value used for k-nearest-neighbour spatial weight matrix. For instance, 5-nearest-neighbour is used to build "lag_k5". P-value of Moran's I listed in parenthesis is approximated under Monte Carlo simulation of 1000 times.



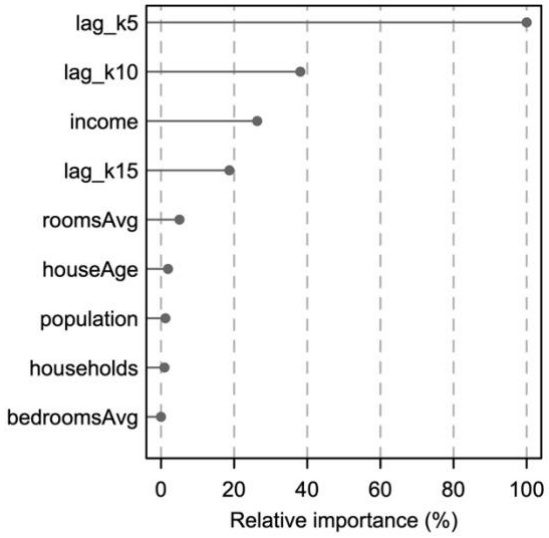
a) Prediction ($10,000)

b) Residual ($10,000)

c) LISA cluster

Figure 4.13. Spatial evaluation of spatial lag models (CA). The integer in parentheses refer to the number of samples within each category. The final spatial lag models tuned from normal and spatial CV are equivalent.

The spatial lag feature built from 5 nearest neighbor demonstrates dominance in the final model (Figure 4.14). The other two lag also rank above original explanatory features except for income.



a) Spatial lag (normal CV & spatial CV)

Figure 4.14. Feature importance of final spatial lag models (CA). Relative feature importance is obtained by scaling the original values to 0-100%.

4.2.3. Eigenvector spatial filtering model

As the dataset contains more than 20,000 samples, it is impractical and unnecessary to calculate eigenvalues of the full spatial weight matrix. 200 eigenvalues are approximated from the exponential kernel matrix (Murakami & Griffith, 2018). Due to the content limits, the selected ESF features are shown in the appendix (Table A5). The final models tuned from two CV methods are still equivalent (Table 4.12). Similar observations can be made from the ESF models compared with the spatial lag ones. The nested spatial CV presents a higher generalized error, and the training error is lower than that of the non-spatial models (Table 4.11). The p-values of Moran's I approach 1 which indicates the spatial autocorrelation is not statistically significant. Additionally, the size of HH and LL clusters decreases considerably (Figure 4.15), which denotes weakening local patterns.

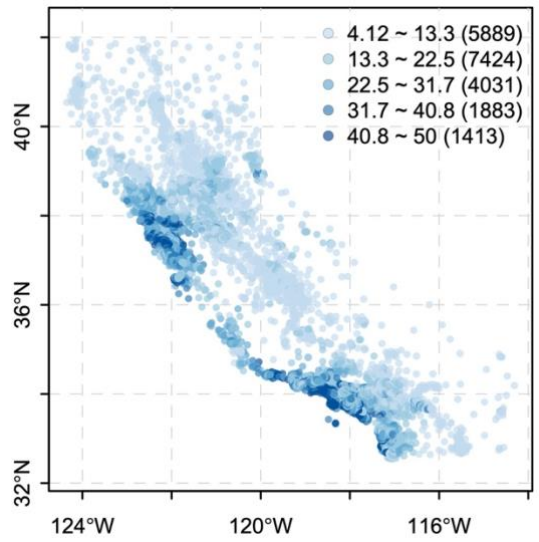
Table 4.11. Accuracy evaluation of ESF models (CA)

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Generalized error
Normal CV	70264.71	67756.02	66949.00	66348.53	69475.80	68158.81
Spatial CV	68705.30	88940.59	108915.96	110953.71	95698.05	94642.72

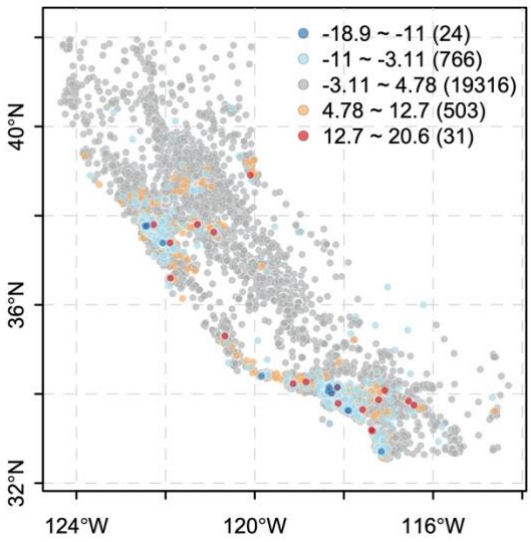
Table 4.12. Final evaluation of ESF models (CA)

	Optimal m_{try}	Training error	Moran's I of residuals
Normal CV	6	20825.50	0.019 (0.999)
Spatial CV	6	20825.50	0.019 (0.999)

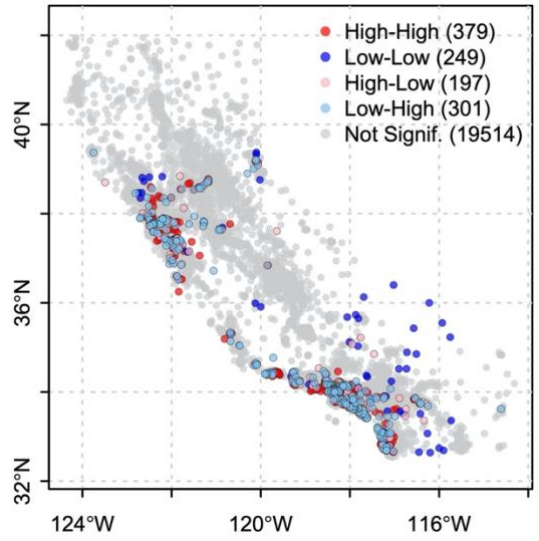
The constructed and selected ESF features are listed in appendix (Table A5). P-value of the Moran's I listed in parenthesis is approximated under Monte Carlo simulation of 1000 times.



a) Prediction ($10,000)



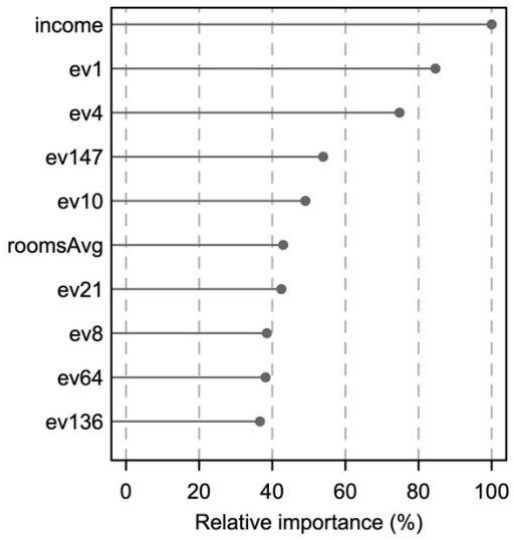
b) Residual ($10,000)



c) LISA cluster

Figure 4.15. Spatial evaluation of ESF models (CA). The integer in parentheses refer to the number of samples within each category. The final ESF models tuned from normal and spatial CV are equivalent.

The income variable shows the highest features importance value (Figure 4.16). The eigenvector with the largest eigenvalue (ev1) ranks second. Other ESF features exhibit varying levels of impact on the final model.



a) ESF (normal CV & spatial CV)

Figure 4.16. Feature importance of final ESF models (Meuse). Relative feature importance is obtained by scaling the original values to 0-100%. Only the top ten ranks are listed for conciseness.

4.3. Model comparison

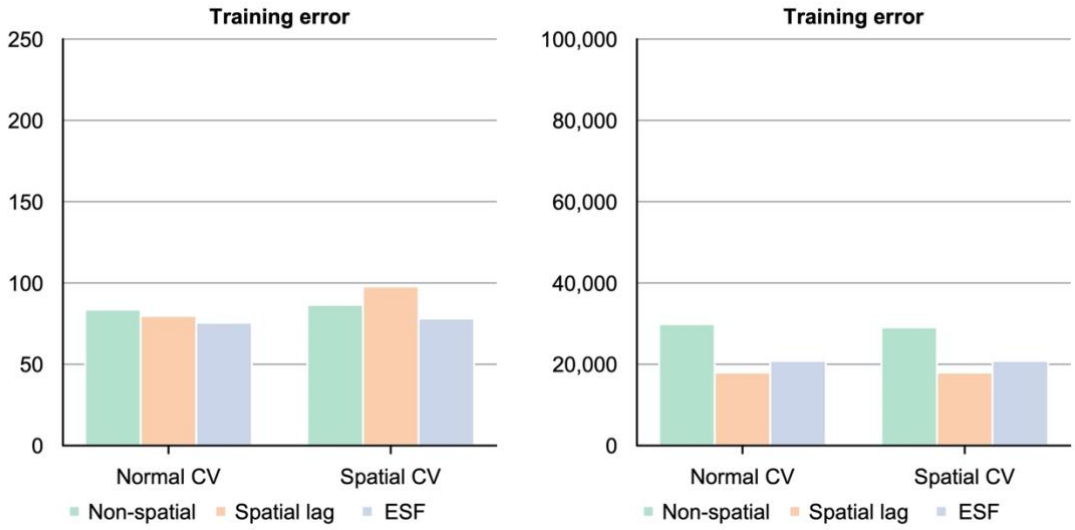
Two spatial models (i.e. spatial lag and ESF) and the non-spatial counterpart are implemented in this study with two different cross-validation settings. The models are developed on two spatial datasets with different sizes. The evaluation consists of two parts: accuracy evaluation (Table 4.13) and spatial evaluation (Table 4.14).

Within the accuracy evaluation, the training error of the final model assesses how well the model can fit on existing data. The generalized error obtained from nested cross-validation estimates the model performance on future unseen data. Figure 4.17 further elucidates the training and generalized errors for the ease of comparison.

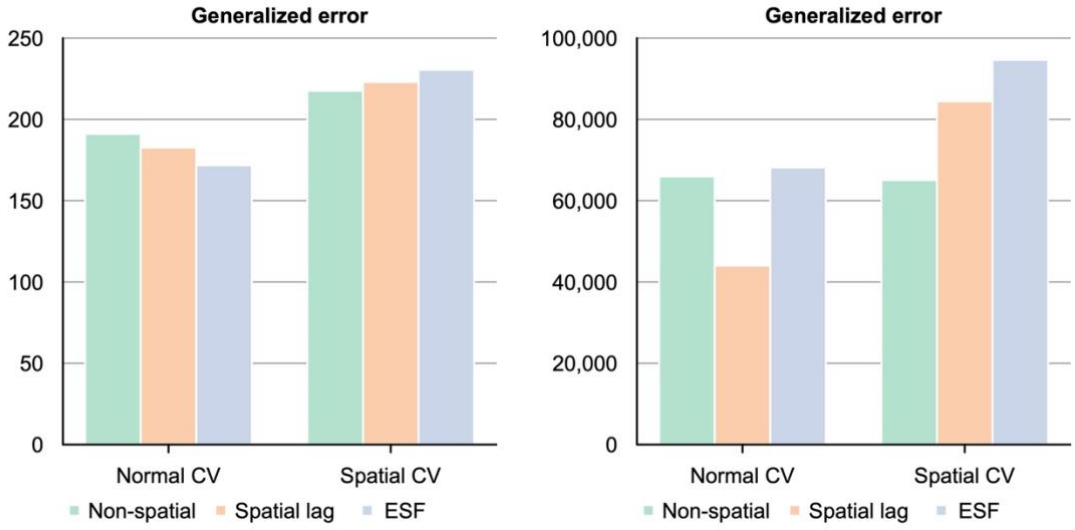
Table 4.13. Accuracy evaluation of different models

	Meuse			**California housing**		
	Optimal m_{try}	**Training error**	**Generalized error**	**Optimal m_{try}**	**Training error**	**Generalized error**
Normal CV + Non-spatial	5	83.59	191.04	2	29857.57	65946.17
Normal CV + Spatial lag	5	79.69	182.63	6	17949.20	44034.95
Normal CV + ESF	5	75.52	171.82	6	20825.50	68158.81
Spatial CV + Non-spatial	4	86.58	217.57	3	29086.55	64995.47
Spatial CV + Spatial lag	2	97.85	222.98	6	17949.20	84424.59

Spatial CV + ESF	6	78.10	230.50	6	20825.50	94642.72



a) Traning error - Meuse b) Traning error - CA



c) Generalized error - Meuse d) Generalized error - CA

Figure 4.17. Accuracy evaluation of different models

For spatial evaluation, both global and local spatial autocorrelation are inspected on residuals. The number of residuals with non-significant LISA values under the level of 0.05 is listed in Table 4.14 as another indicator of how well the local patterns are reduced.

Table 4.14. Spatial evaluation of different models

	Meuse		**California housing**	
	Moran's I	**# LISA cluster (not signif.)**	**Moran's I**	**# LISA cluster (not signif.)**

Normal CV + Non-spatial	0.20***	126	0.42***	15381
Normal CV + Spatial lag	0.029	134	0.022	18941
Normal CV + ESF	0.19***	130	0.019	19514
Spatial CV + Non-spatial	0.18***	127	0.40***	15643
Spatial CV + Spatial lag	0.12**	128	0.022	18941
Spatial CV + ESF	0.15***	131	0.019	19514

*# LISA cluster (not signif.): the number of samples with a non-significant local Moran's I under the level of 0.05. *: $p \leq 0.05$, **: $p \leq 0.01$, ***: $p \leq 0.001$.*

The difference across models are compared from two perspective: the type of spatial features (Table 4.15) and the type of cross-validation methods (Table 4.16). The difference value is calculated as percentage. When comparing the type of spatial features (Table 4.15), the mean results from the two CV methods on generalized error are not provided. Normal and spatial CV are distinct ways for estimating model performance. Averaging the generalized error over two CV methods does not yield a more objective and unbiased view of how the spatial features generally perform.

Table 4.15. Comparison based on the type of spatial features

	Meuse				California housing			
	Training error	**Moran's I**	**# LISA cluster (not signif.)**	**Generalized error**	**Training error**	**Moran's I**	**# LISA cluster (not signif.)**	**Generalized error**
Spatial lag vs. non-spatial								
Normal CV	-5%	-86%	+6%	-4%	-40%	-95%	+23%	-33%
Spatial CV	+13%	-33%	+1%	+2%	-38%	-95%	+21%	+30%
Mean	**+4%**	**-59%**	**+4%**	-	**-39%**	**-95%**	**+22%**	-
ESF vs. non-spatial								
Normal CV	-10%	-5%	+3%	-10%	-30%	-95%	+27%	+3%
Spatial CV	-10%	-17%	+3%	+6%	-28%	-95%	+25%	+46%
Mean	**-10%**	**-11%**	**+3%**	-	**-29%**	**-95%**	**+26%**	-

The positive sign (+) indicates spatial lag/ESF has a higher value compared to the non-spatial one. Likewise, the negative sign (-) indicates a lower value of spatial lag/ESF. For instance, the final spatial lag model tuned from normal CV on the Meuse data has a lower training error than the non-spatial model tuned from normal CV with a difference of around 5%. The decimals are rounded to the integer level.

Table 4.16. Comparison based on the type of cross-validation methods

	Meuse				California housing			
	Training error	**Moran's I**	**# LISA cluster (not signif.)**	**Generalized error**	**Training error**	**Moran's I**	**# LISA cluster (not signif.)**	**Generalized error**
Spatial CV vs. normal CV								
Non-spatial	+4%	-10%	+1%	+14%	-3%	-5%	+2%	-1%
Spatial lag	+23%	+314%	-4%	+22%	0	0	0	+92%
ESF	+3%	-21%	+1%	+34%	0	0	0	+39%
Mean	+10%	+94%	-1%	**+23%**	-1%	-2%	+1%	**+43%**

The positive sign (+) indicates spatial CV has a higher value compared to normal CV. Likewise, the negative sign (-) indicates a lower value of spatial CV. For example, the final non-spatial model tuned from spatial CV on Meuse data has a higher training error than the non-spatial model tuned from normal CV with a difference of 4%. The decimals are rounded to the integer level.

The following subsections discuss the comparison results from two perspectives. Section 4.3.1 examines the influence of two spatial features on model performance. The second subsection 4.3.2 analyzes how different tuning methods (normal and spatial CV) affects the models.

4.3.1. Effects of spatial features

From the perspective of spatial features, the major results are as follows:

1. For most experiments of our study, incorporating either spatial lag or ESF features helps the model to fit better on all the data samples, which is reflected by the lower training error (Table 4.13, 4.15).
    1.1. An anomaly is observed for spatial lag model in Meuse data. The final model built from spatial CV shows a much higher training error than the non-spatial counterpart.
2. Global Moran's I value of residuals in spatial models decreases, and the number of non-significant LISA values increases (Table 4.14, 4.15). Both of the two spatial features are capable to reduce global and local spatial autocorrelation in residuals.
    2.1. The decrease of global Moran in residuals is limited in the Meuse case except for the spatial lag model from normal CV (Table 4.14).
    2.2. The local spatial patterns of residuals display significant changes mostly in HH and LL clusters. The HL and LH patterns are persistent although ESF models show slightly better effects than spatial lag models on these two clusters.

The results above accord with our expectation that the inclusion of spatial information is supposed to be helpful in capturing the spatial autocorrelation and increasing fitting accuracy. These results share a consensus with previous research where spatial features are included as well (Kiely & Bastian, 2019; Li et al., 2017; Zhang et al., 2018; Zhu et al., 2019). The observations in 1.1 and 2.1 only pertain to the Meuse dataset. Meuse has a limited number of samples and illustrates a less stable outcome compared with the California housing data. For the spatial lag model of Meuse, different hyper-parameters result in significant divergence in terms of training errors and the global spatial autocorrelation despite that the same spatial feature is selected (Table 4.4, 4.13). The data size could be the main contributor to this anomaly in observation 1.1. Additionally, the feature importance results show that some original features in Meuse (such as distance to the river and elevation) have higher importance values than either spatial lag or ESF features (Figure 4.6, 4.8). For the California housing data, spatial lag and eigenvectors features are much more dominant in final models than original features in terms of features importance (Figure 4.14, 4.16). The influence of spatial feature in Meuse is not as powerful as that in California housing data. Different spatial mechanisms between zinc concentration and housing price could be another explanation for observation 2.1. Investigation on more datasets with varying data sizes would uncover a more complete understanding of the performance of the two proposed spatial features.

Observation 2.2 is a common issue for both the two datasets. HH and LL represent a positive spatial autocorrelation with the clustering of similar values which is of the major concern for most spatial problems. Spatial lag features essentially express the quantitative characteristics of surrounding regions. When positive spatial autocorrelation dominates a spatial process, which is usually the case in the real world, the spatial lag features helps the model to learn the property of homogeneous clusters. The HL and LH information encoded in spatial lag can lead to confusion in the model which consequently performs worse on these clusters. Although ESF can generate eigenvectors representing both positive and negative spatial autocorrelation, the approximation only produces the first 200 eigenvectors for a large dataset due to computational concerns. Additionally, the subsequent LASSO selection procedure may eliminate the eigenvectors representing negative spatial autocorrelation. Thus, the ESF model does not effectively help with the local heterogeneity (HL and LH) either. Current experiments show the effectiveness of the two spatial features in capturing global spatial autocorrelation. How to represent the underlying local patterns

requires further research. It is promising that representing the regional negative spatial autocorrelation in a more explicit manner would help models fit better on spatial data.

4.3.2. Effects of cross-validation methods

From the perspective of tuning spatial models, two main observations are derived:

1. Generalized error from normal CV is usually lower than the error from spatial CV (Table 4.13, 4.16).
    1.1. For California housing data, the generalized error of non-spatial model from normal CV is slightly higher than the value of spatial CV.
2. The influence of different CV methods on final models (training error, Moran's I, and LISA clusters) are not obvious (Table 4.16).
3. In normal CV, spatial models with lag or ESF usually show lower generalized errors (Table 4.13, 4.15).
    3.1. For California housing data, the ESF model from normal CV presents a higher generalized error than the non-spatial model.
4. In spatial CV, the generalization ability of spatial models decreases considerably and the non-spatial model displays the lowest generalized error (Table 4.13, 4.15).

These results illustrate the distinction between normal CV and spatial CV. In normal CV, the random sampling leads to the mixture of training and testing samples in geographical space (Figure 4.2a, 4.10a). Spatial CV considers the spatial distribution of data samples which divides the dataset into spatially disjoint regions (Figure 4.2b, 4.10b). The model is trained and tested on separate areas. In that sense, spatial CV is analogous to the procedure of extrapolation while normal CV is more consistent with interpolation.

Many studies (Brenning, 2012; Pohjankukka et al., 2017; Schratz et al., 2019) have found that normal CV would give overoptimistic estimates when spatial autocorrelation exists. Observation 1 reflects the design principles of spatial CV and consolidates the claim by previous research. However, observation 1.1 demonstrates an opposite effect of the claim. In appendix Table A3 and A4, the RMSE values of outer fold 1 and 2 have decreased for non-spatial models from normal CV to spatial CV. The errors of outer fold 3, 4, 5 do not change extremely. It could be argued that the original features possess certain abilities in explaining the regional variations of housing prices. When the results are averaged, the generalized errors of non-spatial models from the two CV methods roughly remain at the same level for California housing data. Previous research pointed out that this counter result is also possible and spatial CV would possibly present better outcomes than normal CV depending on data properties (Schratz et al., 2019). Although normal CV and spatial CV demonstrate substantial differences in terms of estimating generalized errors, observation 2 shows that no obvious differentiations are present when these two methods are used to tune the final models. For California housing data, the optimal 'm_{try}' values are relatively consistent (Table 4.13). The changes between two CV methods are minimal on training error, Moran's I and LISA clusters (Table 4.16). For Meuse, the mean changes on training error, Moran's I and LISA clusters are distorted by the final spatial lag model tuned from spatial CV. The effects of different cross-validation methods on tuning the hyper-parameters of final models need experiments on more datasets for future studies.

The training and testing samples in normal CV still share similar spatial properties because they are randomly scattered in space. Compared with non-spatial models, spatial models have learnt additional spatial information of the same area from training samples. The generalized errors of spatial models, therefore, are expected to be lower than non-spatial ones, which is represented by observation 3. The result of observation 3.1 is inconsistent with the outcomes from Meuse dataset. It is hard to conclude the exact cause with experiments on just two datasets. The incorporation of ESF features shows a lower

training error than non-spatial models for California housing. The generalized error is supposed to be lower in normal CV as seen in the Meuse case. But the RMSEs of outer folds do not decrease in Table A3 compared with non-spatial models. The errors are calculated when the outer fold is considered as the testing set. The eigenvectors for testing samples cannot be explicitly calculated. More than 4000 samples are contained in every fold for California housing, while the outer fold from normal CV in Meuse data has around 30 samples. The overall inaccuracy may increase when this approximation is applied to too many locations. Further research with other datasets is critical to understand the robustness of ESF methods for spatial prediction.

Observation 4 demonstrates the exaggerated effects of spatial CV on spatial models. Spatial CV splits the study area into different sub-regions. The segregation would potentially disrupt the spatial properties of training and testing samples. The spatial information encoded in training regions is likely to be different from testing regions, which indicates spatial models may not generalize well. As the features used in non-spatial models are more spatially-agnostic in contrast with explicit spatial features, the influence of spatial CV is less drastic for non-spatial models. This differentiation between normal CV and spatial CV is more notable in the California housing case where the RMSE across different outer folds manifest substantial variance in spatial CV settings. When examining spatial lag model from nested normal CV, the error values are approximately uniform across different outer folds (Table A3). By contrast, the errors in nested spatial CV demonstrates substantial variations (Table A4). For instance, the fold 4 and 5 are the regions of two major cities (San Francisco and Los Angeles) where high housing prices are more likely to present. When these two folds with unique characteristics are used as testing sets, the model trained on other areas cannot generalize well to patterns that have not been seen during training. Besides, the random forest algorithm could be another source of high extrapolation error. Decision-tree based methods cannot generate predictions that exceed the value range of existing data (Hengl et al., 2018). Other algorithms could possibly exhibit better performance in extrapolation tasks.

5. CONCLUSION

5.1. Conclusion

This study investigated the incorporation of two spatial features, i.e. spatial lag and eigenvector spatial filtering, in machine learning to account for spatial autocorrelation. We compared the proposed methods against non-spatial equivalents using random forest on two public spatial datasets (Meuse dataset, and California housing dataset). Moreover, two approaches of cross-validation (normal CV and spatial CV) were explored to tune the hyper-parameters and estimate the model performance. The models were evaluated from both accuracy and spatial perspectives. For accuracy evaluation, generalization and fitting ability were assessed by generalized error from nested CV and training error respectively. For spatial evaluation, global Moran's I and LISA clusters were used to examine the global and local patterns in residuals. The experiments show that the training errors of spatial models are mostly lower than the non-spatial ones. The incorporation of spatial features helps the model to fit better on the data. The generalized errors of spatial models from spatial CV are higher than the values from normal CV. Normal CV yields over-optimistic error estimates than spatial CV, which is in agreement with the findings of previous studies (Ruß & Brenning, 2010; Schratz et al., 2019). From the spatial perspective, the global spatial autocorrelation in residuals has been decreased in spatial lag or ESF models. The homogeneous clusters of local spatial autocorrelation are reduced as well. Spatial features enable machine learning models to capture the spatial autocorrelation. The outcomes in general are within the expectation. In retrospect of the research questions, the answers are can be explained as follows.

Q 1.1: What spatial features can be constructed to potentially account for spatial autocorrelation?
Two spatial features are investigated in this study: spatial lag and eigenvector spatial filtering (ESF). Both the features have been used to capture spatial autocorrelation in linear regression (Anselin, 1988; Arbia, 2014; Getis & Griffith, 2002; Griffith & Chun, 2014). This study extends these two approaches to machine learning and incorporates each of these features as additional variables. Spatial lag represents the average value of surrounding areas. ESF uses the eigenvectors from the spatial weight matrix as auxiliary features.

Q 1.2: How can the spatial features be properly configured?
Rather than one single spatial lag feature used by previous studies (Kiely & Bastian, 2019; Li et al., 2017; Zhu et al., 2019), multiple spatial lag features are constructed for our experiments. Various k values of the k-nearest-neighbor are used to indicate different possibilities of spatial weight matrix. A data-driven LASSO procedure is introduced to select the most informative subset of spatial lag features. For ESF features, a classic exponential kernel is employed to create a positive semidefinite weight matrix (Murakami & Griffith, 2018). The eigenvectors extracted from the weight matrix represent varying map patterns. To reduce the number of eigenvectors, the same LASSO procedure is adopted to select a parsimonious subset of ESF features (Seya et al., 2015).

Q 2.1: What effects do cross-validation have on the performance of models with spatial features?
As regular cross-validation may be biased by spatial autocorrelation (Brenning, 2012; Pohjankukka et al., 2017; Ruß & Brenning, 2010), we applied the spatial cross-validation method developed by Brenning (2012). The spatial CV considers the locations of data samples and splits the data into disjoint regions based on the k-means algorithm. Normal CV resembles the process of interpolation within the same

region while spatial CV is similar to extrapolation. Both normal CV and spatial CV are employed in the experiments to examine their potential differences. The generalized error of spatial models in spatial CV increases up to 92% compared to that in normal CV. Overall, normal CV generates optimistic estimates than spatial CV, which agrees with previous research (Brenning, 2012; Schratz et al., 2019). This difference between normal and spatial CV becomes more significant when spatial features are included. The purpose of the model should be taken into consideration with respect to the choice of CV methods for performance estimation especially when spatial features are included.

Q 3.1: Which spatial features can help to capture spatial autocorrelation and improve prediction accuracy?

The global spatial autocorrelation is successfully reduced in residuals (up to 95% in the California housing case) when either spatial lag or ESF is applied. The size of high-high and low-low clusters has shrunk, and the number of non-significant LISA values have increased. The training errors of spatial models has dropped for most of the experiments. The incorporation of spatial features helps the model to fit better on existing data in general. The generalization ability of the spatial model is influenced by which CV method is employed for evaluation. Spatial CV gives a less optimistic estimate. Spatial models present lower generalized error in normal CV than non-spatial models, while the generalization ability of spatial models considerably decreases if spatial CV is utilized for evaluation.

Q 3.2: What variations, if any, do the proposed spatial features have on small and large datasets in terms of the abilities to help with spatial autocorrelation and model accuracy?

Although our experiments demonstrate the expected ability of both the spatial features in capturing spatial autocorrelation and improving fitting accuracy, different characteristics of the datasets can influence the effectiveness of these spatial features. In Meuse dataset, the reduction of spatial autocorrelation is not substantial in residuals. The original features are more dominant than either spatial lag or ESF features in terms of feature importance in final models. Moreover, Meuse contains limited samples. The results of spatial autocorrelation and training error are closely related to the choice of hyper-parameters. California housing has a much larger number of samples than Meuse. The feature importance results show that spatial features are more influential in the final models. The effects of reducing spatial autocorrelation in residuals are more obvious and consistent. The global spatial autocorrelation in residuals decreases more than 90% and is not significant anymore. The training error of final models also demonstrates great improvements with the average 39% and 29% decrease for spatial lag and ESF respectively.

5.2. Future work

The presence of spatial autocorrelation introduces lurking problems in data analysis when the spatial effects are not explicitly addressed. Incorporating spatial features that express spatial properties represents a promising and extensible approach, which enables the original non-spatial models to account for spatial autocorrelation. Two existing spatial features are extended and the effects of cross-validation are emphasized in this study. Since only two spatial datasets with relatively extreme sizes are investigated, future research is recommended for more representative outcomes. Further studies on the following directions may contribute to more profound insights.

a) This study shows spatial lag or ESF demonstrates the ability of capturing global spatial autocorrelation. Homogeneous local patterns are greatly reduced as well. However, the effects on reducing heterogeneous local clusters (HL and LH clusters in LISA) are marginal. The regional negative spatial autocorrelation still persists. How to explicitly express local negative spatial autocorrelation remains unexplored and requires further research.
b) The generalized error has increased dramatically in spatial CV when the model incorporates spatial features. This study alone is not enough to realize the comprehensive effects of spatial CV.

For instance, since k-means is involved in spatial CV, different choices of k and initialization would divide the data into different regions. Experiments with varying configurations of spatial CV would help to understand how the estimates of spatial CV would change under different settings.

c) Spatial lag and ESF features allow the incorporation of spatial autocorrelation in machine learning when used individually. Spatial lag is easy to calculate and effective in capturing global spatial autocorrelation, and ESF has the potential to represent negative spatial autocorrelation by certain eigenvectors. It is worth the effort for future studies to explore whether the combination of spatial lag and ESF features would yield a better model performance.

d) In this study, the spatial features are tested on two public datasets with different sizes. Variations of model performance are observed among the two datasets. In addition, random forest is chosen for our experiments because of its wide application and general accuracy. New application studies with other machine learning algorithms (like support vector machine, neural networks) and more spatial datasets are needed to understand the robustness of the two spatial features.

LIST OF REFERENCES

Anselin, L. (1988). *Spatial Econometrics: Methods and Models* (Vol. 4). Dordrecht: Springer. https://doi.org/10.1007/978-94-015-7799-1

Anselin, L. (1995). Local Indicators of Spatial Association-LISA. *Geographical Analysis*, *27*(2), 93–115. https://doi.org/10.1111/j.1538-4632.1995.tb00338.x

Arbia, G. (2014). *A Primer for Spatial Econometrics*. London: Palgrave Macmillan UK. https://doi.org/10.1057/9781137317940

Bauman, D., Drouet, T., Dray, S., & Vleminckx, J. (2018). Disentangling good from bad practices in the selection of spatial or phylogenetic eigenvectors. *Ecography*, *41*(10), 1638–1649. https://doi.org/10.1111/ecog.03380

Behrens, T., Schmidt, K., Viscarra Rossel, R. A., Gries, P., Scholten, T., & MacMillan, R. A. (2018). Spatial modelling with Euclidean distance fields and machine learning. *European Journal of Soil Science*, *69*(5), 757–770. https://doi.org/10.1111/ejss.12687

Bivand, R. S., Pebesma, E., & Gómez-Rubio, V. (2013). *Applied Spatial Data Analysis with R*. New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-7618-4

Breiman, L. (2001). Random forests. *Machine Learning*, 5–32. https://doi.org/10.1023/A:1010933404324

Brenning, A. (2012). Spatial cross-validation and bootstrap for the assessment of prediction rules in remote sensing: The R package sperrorest. In *2012 IEEE International Geoscience and Remote Sensing Symposium* (pp. 5372–5375). Munich, Germany: IEEE. https://doi.org/10.1109/IGARSS.2012.6352393

Brunsdon, C., Fotheringham, A. S., & Charlton, M. E. (1996). Geographically Weighted Regression: A Method for Exploring Spatial Nonstationarity. *Geographical Analysis*, *28*(4), 281–298. https://doi.org/10.1111/j.1538-4632.1996.tb00936.x

Cawley, G. C., & Talbot, N. L. C. (2010). On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *Journal of Machine Learning Research*, *11*(70), 2079–2107. Retrieved from http://jmlr.org/papers/v11/cawley10a.html

Chen, L., Ren, C., Li, L., Wang, Y., Zhang, B., Wang, Z., & Li, L. (2019). A Comparative Assessment of Geostatistical, Machine Learning, and Hybrid Approaches for Mapping Topsoil Organic Carbon Content. *ISPRS International Journal of Geo-Information*, *8*(4), 174. https://doi.org/10.3390/ijgi8040174

Cupido, K., Jevtic, P., & Paez, A. (2019). Spatial Patterns of Mortality in the United States: A Spatial Filtering Approach. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3359353

Debarsy, N., & LeSage, J. (2018). Flexible dependence modeling using convex combinations of different types of connectivity structures. *Regional Science and Urban Economics*, *69*, 48–68. https://doi.org/10.1016/j.regsciurbeco.2018.01.001

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248–255). IEEE. https://doi.org/10.1109/CVPR.2009.5206848

Dormann, C. F., McPherson, J. M., Araújo, M. B., Bivand, R., Bolliger, J., Carl, G., … Wilson, R. (2007). Methods to account for spatial autocorrelation in the analysis of species distributional data: a review. *Ecography*, *30*(5), 609–628. https://doi.org/10.1111/j.2007.0906-7590.05171.x

Foresti, L., Pozdnoukhov, A., Tuia, D., & Kanevski, M. (2010). Extreme Precipitation Modelling Using Geostatistics and Machine Learning Algorithms. In P. Monestiez, D. Allard, & R. Froidevaux (Eds.), *geoENV VII – Geostatistics for Environmental Applications* (Vol. 16, pp. 41–52). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-90-481-2322-3_4

Fouedjio, F., & Klump, J. (2019). Exploring prediction uncertainty of spatial data in geostatistical and machine learning approaches. *Environmental Earth Sciences*, *78*(1), 1–24.

https://doi.org/10.1007/s12665-018-8032-z
Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, *33*(1), 1–22. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/20808728
Getis, A. (2008). A History of the Concept of Spatial Autocorrelation: A Geographer's Perspective. *Geographical Analysis*, *40*(3), 297–309. https://doi.org/10.1111/j.1538-4632.2008.00727.x
Getis, A., & Griffith, D. A. (2002). Comparative Spatial Filtering in Regression Analysis. *Geographical Analysis*, *34*(2), 130–140. https://doi.org/10.1111/j.1538-4632.2002.tb01080.x
Goodchild, M. F. (1992). Geographical information science. *International Journal of Geographical Information Systems*, *6*(1), 31–45. https://doi.org/10.1080/02693799208901893
Goodchild, M. F. (2013). The quality of big (geo)data. *Dialogues in Human Geography*, *3*(3), 280–284. https://doi.org/10.1177/2043820613513392
Griffith, D., & Chun, Y. (2014). Spatial Autocorrelation and Spatial Filtering. In *Handbook of Regional Science* (pp. 1477–1507). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-23430-9_72
Harrison, D., & Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, *5*(1), 81–102. https://doi.org/10.1016/0095-0696(78)90006-2
Hengl, T., Heuvelink, G. B. M., Kempen, B., Leenaars, J. G. B., Walsh, M. G., Shepherd, K. D., … Tondoh, J. E. (2015). Mapping Soil Properties of Africa at 250 m Resolution: Random Forests Significantly Improve Current Predictions. *PLOS ONE*, *10*(6), e0125814. https://doi.org/10.1371/journal.pone.0125814
Hengl, T., Heuvelink, G. B. M., & Rossiter, D. G. (2007). About regression-kriging: From equations to case studies. *Computers & Geosciences*, *33*(10), 1301–1315. https://doi.org/10.1016/j.cageo.2007.05.001
Hengl, T., Nussbaum, M., Wright, M. N., Heuvelink, G. B. M., & Gräler, B. (2018). Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables. *PeerJ*. https://doi.org/10.7717/peerj.5518
Hoffmann, J., Bar-Sinai, Y., Lee, L. M., Andrejevic, J., Mishra, S., Rubinstein, S. M., & Rycroft, C. H. (2019). Machine learning in a data-limited regime: Augmenting experiments with synthetic data uncovers order in crumpled sheets. *Science Advances*, *5*(4), eaau6792. https://doi.org/10.1126/sciadv.aau6792
Kanevski, M., Timonin, V., & Pozdnukhov, A. (2009). *Machine Learning for Spatial Environmental Data*. EPFL Press. https://doi.org/10.1201/9781439808085
Kiely, T. J., & Bastian, N. D. (2019). The Spatially-Conscious Machine Learning Model. Retrieved from http://arxiv.org/abs/1902.00562
Kitchin, R. (2013). Big data and human geography. *Dialogues in Human Geography*, *3*(3), 262–267. https://doi.org/10.1177/2043820613513388
Kleijnen, J. P. C., & van Beers, W. C. M. (2018). Prediction for Big Data Through Kriging: Small Sequential and One-Shot Designs. *CentER Discussion Paper*, *2018–022*. https://doi.org/10.2139/ssrn.3210567
Klemmer, K., Koshiyama, A., & Flennerhag, S. (2019). Augmenting correlation structures in spatial data using deep generative models. Retrieved from http://arxiv.org/abs/1905.09796
Kray, C., Pebesma, E., Konkol, M., & Nüst, D. (2019). Reproducible Research in Geoinformatics: Concepts, Challenges and Benefits (Vision Paper). In S. Timpf, C. Schlieder, M. Kattenbeck, B. Ludwig, & K. Stewart (Eds.), *14th International Conference on Spatial Information Theory (COSIT 2019)* (Vol. 142, pp. 8:1--8:13). Dagstuhl, Germany: Schloss Dagstuhl--Leibniz-Zentrum fuer Informatik.

https://doi.org/10.4230/LIPIcs.COSIT.2019.8
LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539
Li, J., Heap, A. D., Potter, A., & Daniell, J. J. (2011). Application of machine learning methods to spatial interpolation of environmental variables. *Environmental Modelling and Software*, *26*(12), 1647–1659. https://doi.org/10.1016/j.envsoft.2011.07.004
Li, T., Shen, H., Yuan, Q., Zhang, X., & Zhang, L. (2017). Estimating Ground-Level PM 2.5 by Fusing Satellite and Station Observations: A Geo-Intelligent Deep Learning Approach. *Geophysical Research Letters*, *44*(23), 11,985-11,993. https://doi.org/10.1002/2017GL075710
Löchl, M., & Axhausen, K. W. (2010). Modelling hedonic residential rents for land use and transport simulation while considering spatial effects. *Journal of Transport and Land Use*, *3*(2), 39–63. https://doi.org/10.5198/jtlu.v3i2.117
Meyer, H., Reudenbach, C., Wöllauer, S., & Nauss, T. (2019). Importance of spatial predictor variable selection in machine learning applications – Moving from data reproduction to spatial prediction. *Ecological Modelling*, *411*, 108815. https://doi.org/10.1016/j.ecolmodel.2019.108815
Miller, H. J. (2000). Geographic representation in spatial analysis. *Journal of Geographical Systems*, *2*(1), 55–60. https://doi.org/10.1007/s101090050030
Mueller, E., Sandoval, J. S. O., Mudigonda, S., & Elliott, M. (2018). A Cluster-Based Machine Learning Ensemble Approach for Geospatial Data: Estimation of Health Insurance Status in Missouri. *ISPRS International Journal of Geo-Information*, *8*(1), 13. https://doi.org/10.3390/ijgi8010013
Murakami, D., & Griffith, D. A. (2018). Eigenvector Spatial Filtering for Large Data Sets: Fixed and Random Effects Approaches. *Geographical Analysis*, *51*(1), 23–49. https://doi.org/10.1111/gean.12156
Murakami, D., Tsutsumida, N., Yoshida, T., Nakaya, T., & Lu, B. (2019). Scalable GWR: A linear-time algorithm for large-scale geographically weighted regression with polynomial kernels. Retrieved from http://arxiv.org/abs/1905.00266
Oshiro, T. M., Perez, P. S., & Baranauskas, J. A. (2012). How Many Trees in a Random Forest? In P. Perner (Ed.), *Machine Learning and Data Mining in Pattern Recognition* (pp. 154–168). Berlin, Heidelberg: Springer Berlin Heidelberg.
Pace, R. K., & Barry, R. (1997). Sparse spatial autoregressions. *Statistics & Probability Letters*, *33*(3), 291–297. https://doi.org/10.1016/S0167-7152(96)00140-X
Paez, A. (2019). Using Spatial Filters and Exploratory Data Analysis to Enhance Regression Models of Spatial Data. *Geographical Analysis*, *51*(3), 314–338. https://doi.org/10.1111/gean.12180
Pebesma, E. J. (2004). Multivariable geostatistics in S: the gstat package. *Computers & Geosciences*, *30*(7), 683–691. https://doi.org/10.1016/j.cageo.2004.03.012
Perkel, J. M. (2018). Why Jupyter is data scientists' computational notebook of choice. *Nature*, *563*(7729), 145–146. https://doi.org/10.1038/d41586-018-07196-1
Pohjankukka, J., Pahikkala, T., Nevalainen, P., & Heikkonen, J. (2017). Estimating the prediction performance of spatial models via spatial k-fold cross validation. *International Journal of Geographical Information Science*, *31*(10), 2001–2019. https://doi.org/10.1080/13658816.2017.1346255
Probst, P., & Boulesteix, A.-L. (2017). To tune or not to tune the number of trees in random forest? *Journal of Machine Learning Research*, *18*, 1–18.
R Core Team. (2019). R: A Language and Environment for Statistical Computing. Vienna, Austria. Retrieved from https://www.r-project.org/
Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., & Prabhat. (2019). Deep learning and process understanding for data-driven Earth system science. *Nature*, *566*(7743), 195–204. https://doi.org/10.1038/s41586-019-0912-1

Ruß, G., & Brenning, A. (2010). Spatial Variable Importance Assessment for Yield Prediction in Precision Agriculture. In P. R. CohenNiall, M. AdamsMichael, & R. Berthold (Eds.), *Advances in Intelligent Data Analysis IX* (pp. 184–195). Tucson, AZ, USA: Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-13062-5_18

Ruß, G., & Kruse, R. (2010). Regression Models for Spatial Data: An Example from Precision Agriculture. In P. Perner (Ed.), *Advances in Data Mining. Applications and Theoretical Aspects. ICDM 2010. Lecture Notes in Computer Science, vol 6171* (pp. 450–463). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-14400-4_35

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Schratz, P., Muenchow, J., Iturritxa, E., Richter, J., & Brenning, A. (2019). Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data. *Ecological Modelling*, *406*(April 2018), 109–120. https://doi.org/10.1016/j.ecolmodel.2019.06.002

Seya, H., Murakami, D., Tsutsumi, M., & Yamagata, Y. (2015). Application of LASSO to the Eigenvector Selection Problem in Eigenvector-based Spatial Filtering. *Geographical Analysis*, *47*(3), 284–299. https://doi.org/10.1111/gean.12054

Shekhar, S., Jiang, Z., Ali, R., Eftelioglu, E., Tang, X., Gunturi, V., & Zhou, X. (2015). Spatiotemporal Data Mining: A Computational Perspective. *ISPRS International Journal of Geo-Information*, *4*(4), 2306–2338. https://doi.org/10.3390/ijgi4042306

Shen, H. (2014). Interactive notebooks: Sharing the code. *Nature*, *515*(7525), 151–152. https://doi.org/10.1038/515151a

Stojanova, D., Ceci, M., Appice, A., Malerba, D., & Džeroski, S. (2013). Dealing with spatial autocorrelation when learning predictive clustering trees. *Ecological Informatics*, *13*, 22–39. https://doi.org/10.1016/j.ecoinf.2012.10.006

Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Tobler, W. R. (1970). A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography*, *46*, 234. https://doi.org/10.2307/143141

Wheeler, D. C. (2014). Geographically Weighted Regression. In *Handbook of Regional Science* (pp. 1435–1459). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-23430-9_77

Zhang, J., Li, B., Chen, Y., Chen, M., Fang, T., & Liu, Y. (2018). Eigenvector Spatial Filtering Regression Modeling of Ground PM2.5 Concentrations Using Remotely Sensed Data. *International Journal of Environmental Research and Public Health*, *15*(6), 1228. https://doi.org/10.3390/ijerph15061228

Zhu, X., Zhang, Q., Xu, C.-Y., Sun, P., & Hu, P. (2019). Reconstruction of high spatial resolution surface air temperature data across China: A new geo-intelligent multisource data-based machine learning technique. *Science of The Total Environment*, *665*, 300–313. https://doi.org/10.1016/j.scitotenv.2019.02.077

APPENDIX

Table A1. Results of Meuse data from normal cross-validation

	Final model			**Nested CV (RMSE)**					
	m_{try}	Training error	Moran of residuals	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Generalized error
Non-spatial	5	83.59	0.20***	179.54	123.25	191.55	201.07	259.77	191.04
Spatial lag	5	79.69	**0.029**	181.05	120.44	195.43	187.23	**229.00**	182.63
ESF	5	**75.52**	0.19***	**149.87**	**109.02**	**182.29**	**176.88**	241.04	**171.82**

**: $p \leq 0.05$, **: $p \leq 0.01$, ***: $p \leq 0.001$. The minimum of a column is indicated in bold.*

Table A2. Results of Meuse data from spatial cross-validation

	Final model			**Nested CV (RMSE)**					
	m_{try}	Training error	Moran of residuals	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Generalized error
Non-spatial	4	86.58	0.18***	265.47	229.24	151.72	**268.66**	**172.75**	**217.57**
Spatial lag	2	97.85	**0.12****	**250.13**	**218.40**	**135.15**	284.00	227.22	222.98
ESF	6	**78.10**	0.15***	270.09	220.03	141.22	293.41	227.77	230.50

**: $p \leq 0.05$, **: $p \leq 0.01$, ***: $p \leq 0.001$. The minimum of a column is indicated in bold.*

Table A3. Results of California housing data from normal cross-validation

	Final model			**Nested CV (RMSE)**					
	m_{try}	Training error	Moran of residuals	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Generalized error
Non-spatial	2	29857.57	0.42***	65589.35	64799.53	66965.33	68654.93	63721.71	65946.17
Spatial lag	6	**17949.20**	0.022	**44018.01**	**43306.16**	**45092.36**	**44457.47**	**43300.77**	**44034.95**
ESF	6	20825.50	**0.019**	70264.71	67756.02	66949.00	66348.53	69475.80	68158.81

**: $p \leq 0.05$, **: $p \leq 0.01$, ***: $p \leq 0.001$. The minimum of a column is indicated in bold.*

Table A4. Results of California housing data from spatial cross-validation

	Final model			**Nested CV (RMSE)**					
	m_{try}	Training error	Moran of residuals	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Generalized error
Non-spatial	3	29086.55	0.40***	47419.24	**58712.81**	72149.94	**75477.13**	**71218.24**	**64995.47**
Spatial lag	6	**17949.20**	0.022	**44206.38**	76967.17	**60853.46**	135877.61	104218.33	84424.59
ESF	6	20825.50	**0.019**	68705.30	88940.59	108915.96	110953.71	95698.05	94642.72

**: $p \leq 0.05$, **: $p \leq 0.01$, ***: $p \leq 0.001$. The minimum of a column is indicated in bold.*

Table A5. Selected spatial features of final models (Meuse)

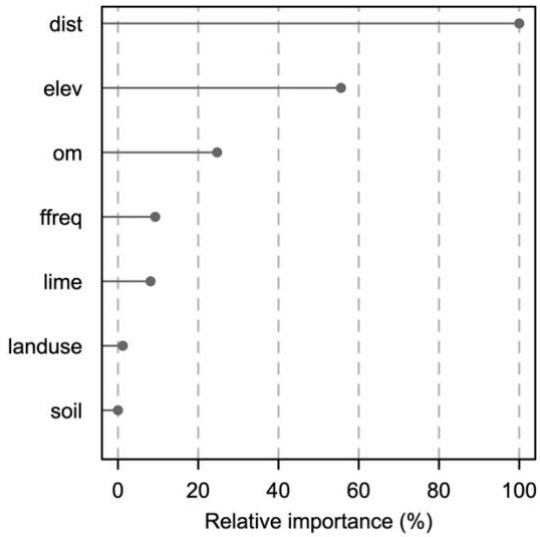
	Constructed spatial features	Selected spatial features
Normal CV + Non-spatial	-	-
Normal CV + Spatial lag	lag_k5, lag_k10, lag_k15	lag_k5
Normal CV + ESF	ev1 – ev152	ev8, ev11, ev12, ev34
Spatial CV + Non-spatial	-	-
Spatial CV + Spatial lag	lag_k5, lag_k10, lag_k15, lag_k50	lag_k5
Spatial CV + ESF	ev1 – ev152	ev8, ev11, ev12

"ev" stands for "eigenvector".

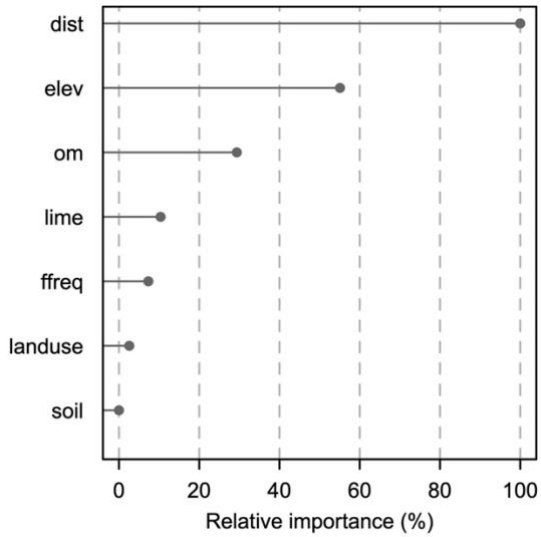
Table A6. Selected spatial features of final models (California housing)

	Constructed spatial features	Selected spatial features
Normal CV + Non-spatial	-	-
Normal CV + Spatial lag	lag_k5, lag_k10, lag_k15, lag_k50	lag_k5, lag_k10, lag_k15
Normal CV + ESF	ev1 - ev200	ev1, ev4, ev8, ev10, ev14, ev19, ev21, ev23, ev30, ev33, ev38, ev40, ev43, ev50, ev51, ev53, ev55, ev57, ev58, ev62, ev63, ev64, ev70, ev76, ev79, ev80, ev81, ev82, ev83, ev88, ev90, ev91, ev98, ev100, ev101, ev103, ev108, ev109, ev110, ev113, ev114, ev115, ev116, ev119, ev120, ev121, ev123, ev128, ev130, ev131, ev132, ev135, ev136, ev139, ev140, ev147, ev149, ev150, ev153, ev156, ev157, ev159, ev161, ev162, ev166, ev170, ev172, ev174, ev175, ev178, ev182, ev183, ev184, ev190, ev191, ev195, ev197 (77 features)
Spatial CV + Non-spatial	-	-
Spatial CV + Spatial lag	lag_k5, lag_k10, lag_k15, lag_k50	lag_k5, lag_k10, lag_k15
Spatial CV + ESF	ev1 - ev200	Same as "Normal CV + ESF"

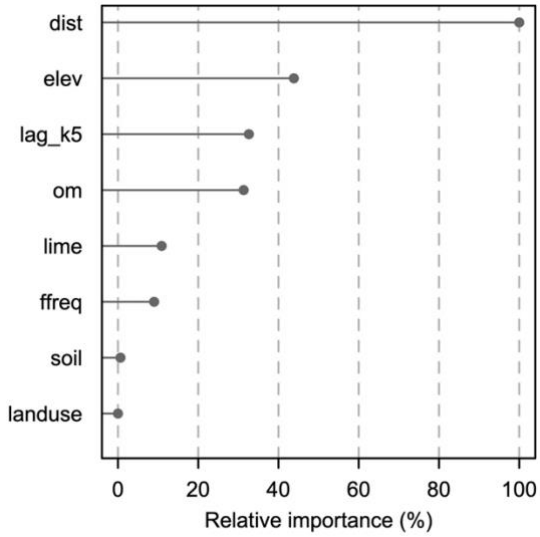
"ev" stands for "eigenvector".



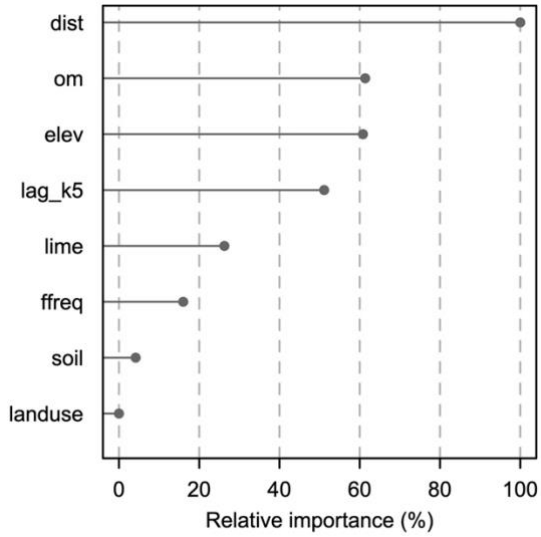
a) Non-spatial (normal CV)



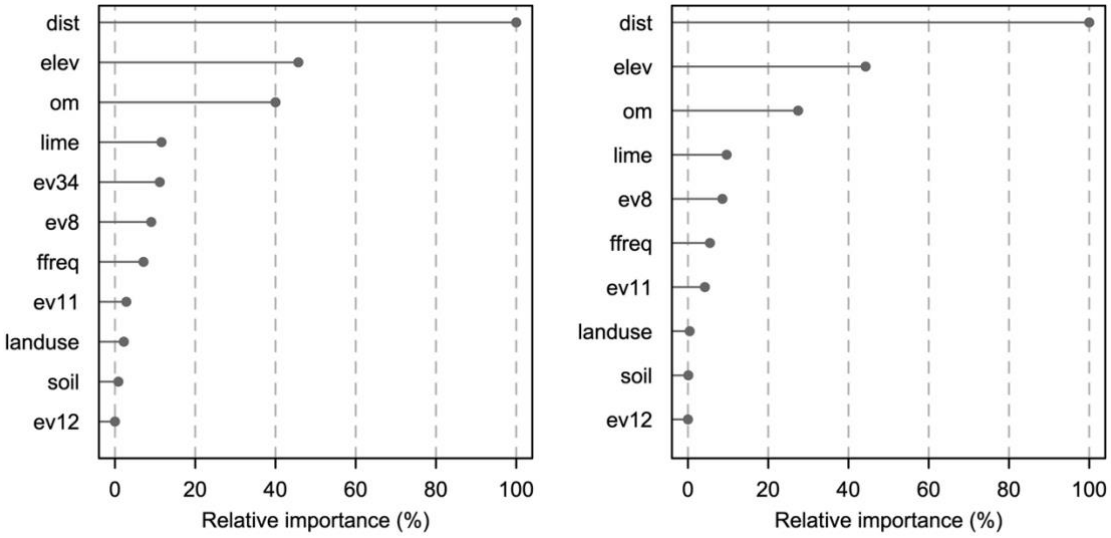
b) Non-spatial (spatial CV)



c) Spatial lag (normal CV)

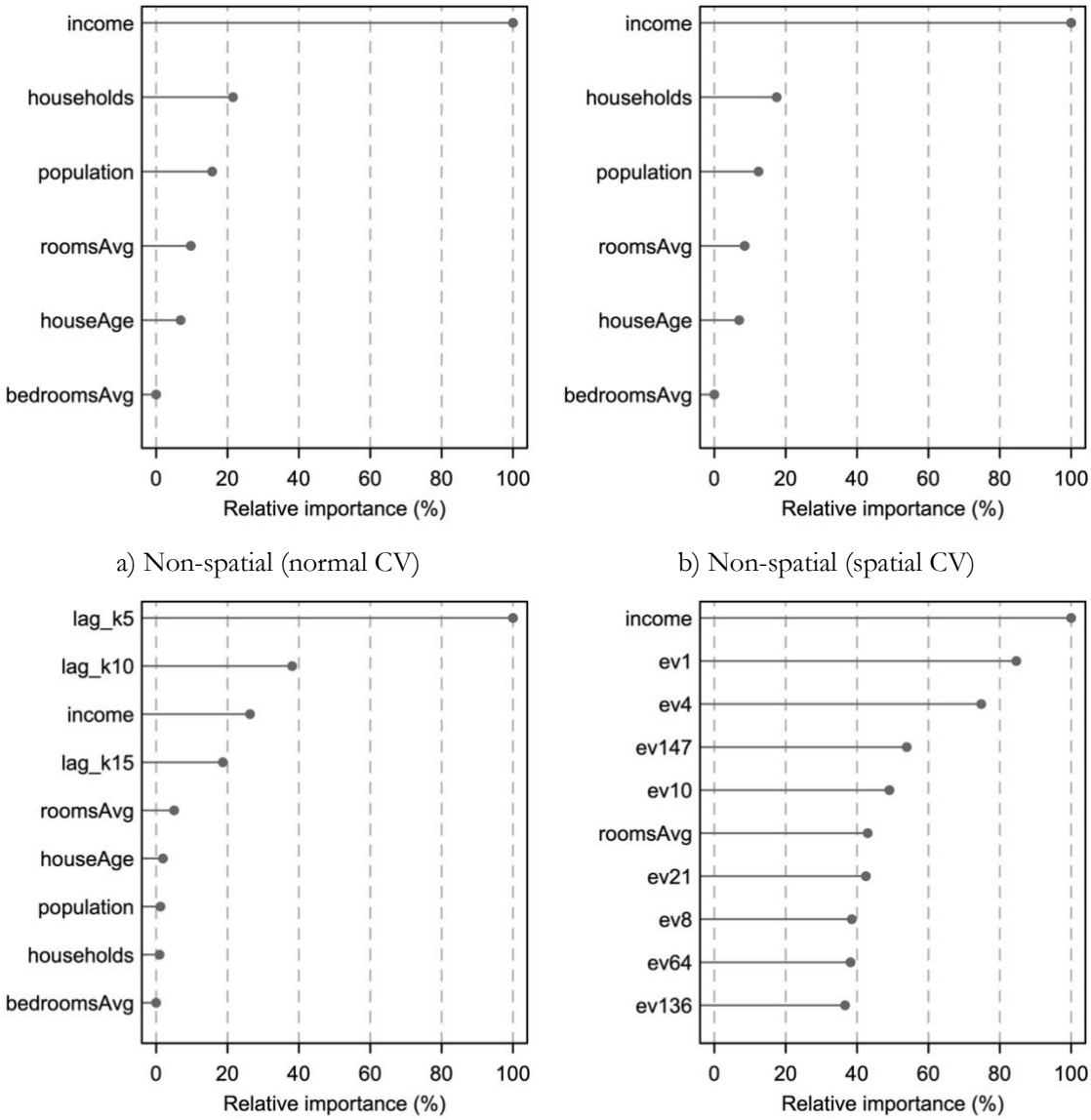


d) Spatial lag (spatial CV)



e) ESF (normal CV) f) ESF (spatial CV)

Figure A1. Feature importance of final models (Meuse). The feature importance is scaled to 0-100%.



a) Non-spatial (normal CV)

b) Non-spatial (spatial CV)

c) Spatial lag (normal CV & spatial CV)

d) ESF (normal CV & spatial CV)

Figure A2. Feature importance of final models (California housing). The feature importance is scaled to 0-100%. The final spatial lag models from normal CV and spatial CV are identical. The same holds for ESF models.

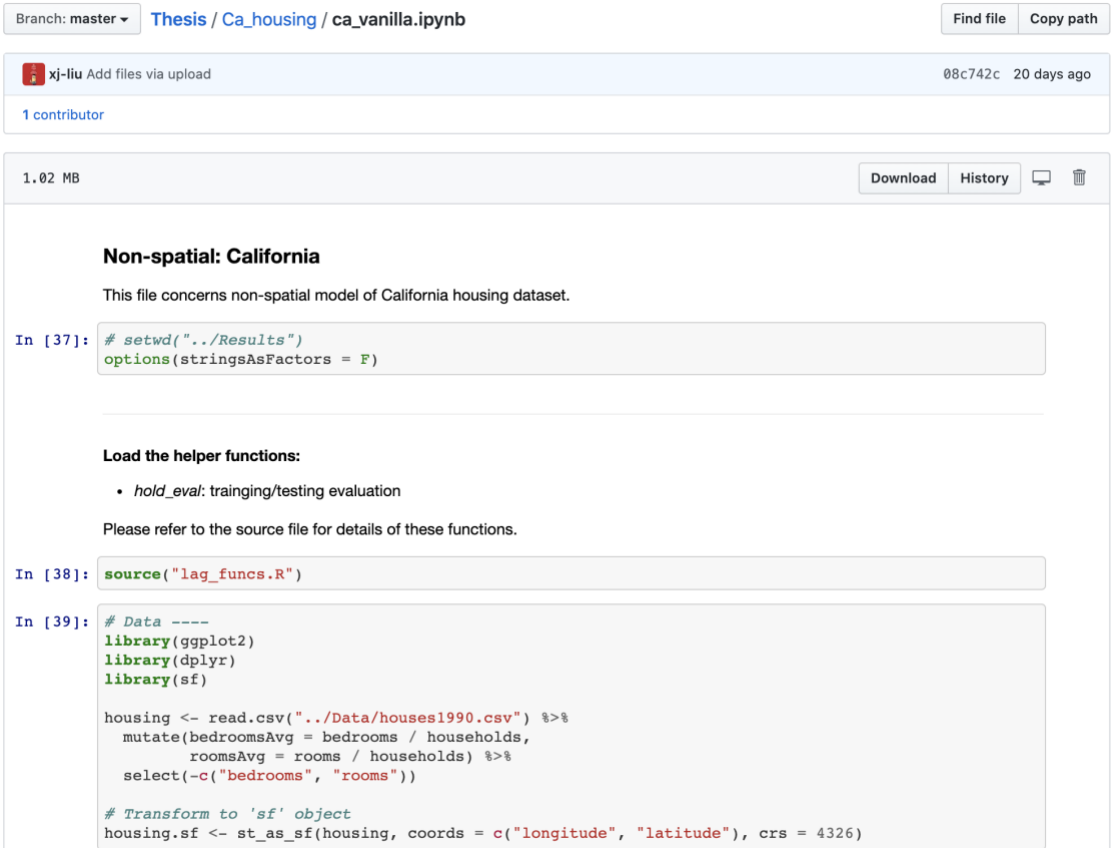


Figure A3. Example of the Jupyter notebook uploaded on GitHub (https://github.com/xj-liu/Thesis).