

Spatial cross-validation for GeoAI

Kai Sun ¹, Yingjie Hu ¹, Gaurish Lakhanpal ^{1,2}, and Ryan Zhenqi Zhou ¹

¹*GeoAI Lab, Department of Geography, University at Buffalo, Buffalo, NY, USA*

²*Stevenson High School, Lincolnshire, IL, USA*

Abstract: Cross-validation (CV) has been widely used in GeoAI research to evaluate the performance of machine learning models. Often, a labeled data set is randomly split into training and validation data, and a machine learning model is trained on the training data and then evaluated on the validation data in an iterative manner. Such a random CV approach could lead to an overestimate of model performance on geographic data, due to the existence of spatial autocorrelation. Random CV can generate many training and validation data instances that are spatially close, and a model trained on such training data can be considered as having already “peeked” into the nearby validation data, given their spatial closeness and likely attribute similarity. Spatial CV can help address this issue by splitting the data spatially rather than randomly, thereby increasing the independence between the training and validation data. While a number of spatial CV methods have been developed, they are scattered in the literature across multiple disciplines, including ecology, remote sensing, GIScience, and computer science. This chapter discusses four main spatial CV methods identified from the multidisciplinary literature, and uses two examples based on real-world data to demonstrate these methods in comparison with random CV.

Keywords: cross-validation; spatial cross-validation; spatial autocorrelation; spatial dependence; machine learning; GeoAI.

1. Introduction

Cross-validation (CV) has been widely used in GeoAI research to evaluate the performance of machine learning models. Often, a labeled data set is split into a number of subsets, with one subset held out for validation and the remaining subsets used for model training; the training-and-validation process is then iterated multiple times until all subsets of data are used for training and validation (Browne 2000). The commonly used *k-fold CV* splits data into *k* subsets (or *k* folds), and uses one subset for validation and *k-1* subsets for model training. The process is then repeated *k* times, and the obtained *k* results are averaged to compute a final score for the machine learning model (Burman 1989). When *k* equals to the total number of data instances *n* in the data set, *k-fold CV* becomes *leave-one-out CV*, in which only one data instance is held out for validation each time and the process is repeated *n* times. As an evaluation technique, CV often provides more robust evaluation scores for machine learning models since

the models are trained and evaluated on different subsets of data (Géron 2019). Accordingly, CV avoids the potential bias when only one particular subset of the data is used for validation.

While increasing evaluation robustness, CV could lead to another type of bias when applied to geographic data. CV typically randomly splits data into training and validation subsets, and such a random CV approach can generate many training and validation data instances that are spatially close. As widely discussed in the spatial study literature, spatially close data instances are likely to share similar attribute values due to the existence of spatial dependency (Tobler 1970; Diniz-Filho, Bini, and Hawkins 2003; Fotheringham 2009; Hijmans 2012). A model trained on such training data, therefore, can be considered as having already “peeked” into the nearby validation data, given their spatial closeness and likely attribute similarity. Consequently, we may obtain overly optimistic evaluation results that do not represent the true performance of the model (Roberts et al. 2017; Pohjankukka et al. 2017; Karasiak et al. 2019; Meyer and Pebesma 2022; Karasiak et al. 2022).

Spatial cross-validation (spatial CV) is a spatially explicit CV method that can help address this evaluation bias by splitting data spatially rather than randomly. Spatial CV uses strategies, such as spatial buffering and grid-based splitting (Valavi et al. 2019), to create subsets of data that are more independent from each other, thereby decreasing the spatial autocorrelation between training and validation data. As machine learning models have been increasingly applied to geographic data, researchers have developed a number of methods and software packages for spatial CV. However, these methods and software packages are scattered in the literature across multiple disciplines, including ecology (Roberts et al. 2017; Meyer et al. 2018; Valavi et al. 2019), remote sensing (Brenning 2012; Karasiak et al. 2019; Karasiak 2020), GIScience (Pohjankukka et al. 2017; Crosby, Damoulas, and Jarvis 2020; Brenning 2022), and computer science (Airola et al. 2019; Da Silva, Parmezan, and Batista 2021). Consequently, it is difficult for researchers new to spatial CV to quickly grasp the main methods and existing software packages from this multidisciplinary literature. This book chapter fills this gap by presenting four main spatial CV methods identified from the literature, and uses two examples based on real-world data to demonstrate these spatial CV methods by comparing them with random CV.

Before we move on to discussing spatial CV methods, it is worth noting that spatial CV is not always better than random CV when machine learning models are applied to geographic data. This can be understood based on two common situations in GeoAI research: within-area prediction and between-area prediction (Roberts et al. 2017; Goodchild and Li 2021). For within-area prediction or *interpolation* (in Figure 1(a)), a model is trained on the known values in the study area and is then used to predict unknown values in the same area. In such a case, random CV is in fact preferred since the real test data can be spatially close to known training data too, and we want to leverage the existence of spatial autocorrelation to generate more accurate predictions. For between-area prediction or *extrapolation* (in Figure 1(b)), a model is trained on known values in one geographic area and is then used to predict unknown values in

another area (which could be adjacent to the training data area as shown in Figure 1(b) or be further away). In such a case, spatial CV is likely to provide a more realistic evaluation of model performance since the spatially-split validation data can better mimic the real test scenario that the trained model will be used for (Crosby, Damoulas, and Jarvis 2020; Meyer and Pebesma 2021). In short, whether we should choose spatial CV or random CV depends on which strategy better mimics the real application scenario of the trained machine learning model.

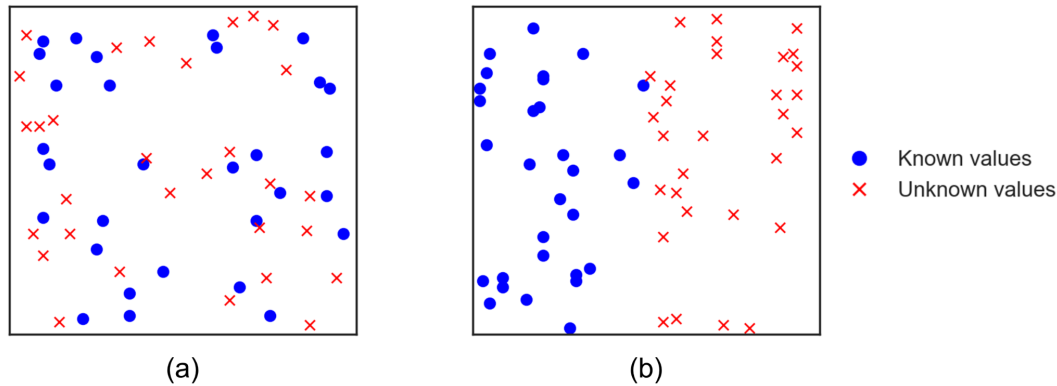


Figure 1. An illustration of two common situations under which machine learning models are applied to geographic data: (a) within-area prediction or interpolation; (b) between-area prediction or extrapolation.

The remainder of this chapter is organized as follows. Section 2 presents four main spatial CV methods and software packages that implement these methods. In section 3, we use two examples based on real-world data to demonstrate the use of these four spatial CV methods in comparison with random CV. Finally, Section 4 concludes this chapter.

2. Four main spatial CV methods

While various spatial CV methods have been developed, they can be organized into four main types: *clustering-based spatial CV*, *grid-based spatial CV*, *geo-attribute-based spatial CV*, and *spatial leave-one-out CV*. These four methods share the common idea of spatially splitting training and validation data while differing in their specific ways of splitting the data. A number of software packages have been developed to implement these spatial CV methods. Given the popularity of the R programming language in the ecology research community where spatial CV has received much attention, a lot of these packages are developed in R which include:

- *sperrorest*: <https://cran.r-project.org/web/packages/sperrorest/index.html> (Brenning 2012)
- *spatialsample*: <https://cran.r-project.org/web/packages/spatialsample/index.html>
- *blockCV*: <https://cran.r-project.org/web/packages/blockCV/index.html> (Valavi et al. 2019)
- *ENMeval*: <https://cran.r-project.org/web/packages/ENMeval/index.html> (Muscarella et al. 2014)
- *Mr3spatiotempcv*: <https://cran.r-project.org/web/packages/mlr3spatiotempcv/index.html> (Schratz et al. 2021)
- *CAST*: <https://cran.r-project.org/web/packages/CAST/index.html> (Meyer et al. 2018)

There also exist two Python-based spatial CV packages, which are:

- *spacv*: <https://github.com/SamComber/spacv>
- *MuseoToolbox*: <https://museotoolbox.readthedocs.io/en/latest/> (Karasiak 2020)

Each package implements its own set of spatial CV methods, which may overlap with some other packages. In the following, we present each of the four main spatial CV methods and the software packages that have implemented the method. Interested readers can choose the software package based on the spatial CV methods needed and their preferred programming language.

2.1 Clustering-based spatial CV

One method to split data spatially is to perform spatial clustering. Based on the geospatial coordinates of the data and leveraging a clustering algorithm (e.g., K-means clustering), we can split data into several subsets with data instances within the same subset being spatially continuous and those in different subsets being relatively separated from each other. Figure 2 illustrates this clustering-based spatial CV using K-means clustering ($k=3$). With the split data in Figure 2(b), one can then use, e.g., the data in *Cluster 1* for validation and the data in *Cluster 2* and *Cluster 3* for model training, and then iterate this process. In addition, we are not limited to using only one cluster for validation but can use multiple clusters too. For example, we could split the data into ten clusters using a clustering algorithm, and then use three clusters for validation and seven clusters for model training.

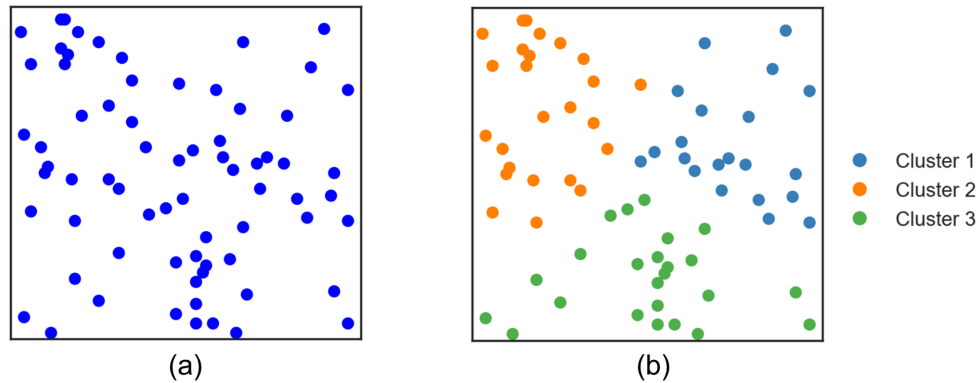


Figure 2. An illustration of clustering-based spatial CV: (a) original training data; (b) training data split into three folds using K-means clustering.

Clustering-based spatial CV has been implemented in the following packages: *spatialsample* (R), *sperrorest* (R), *Mr3spatiotempcv* (R), *blockCV* (R), and *spacv* (Python). K-means is the most commonly used clustering algorithm in these packages. The package *spatialsample* also implements another clustering algorithm, hierarchical clustering, for clustering-based spatial CV.

2.2 Grid-based spatial CV

Another method to spatially split data is to use spatial grids. By dividing the entire geographic area of the data into n rows and m columns, we can split the data into $n \times m$ grid cells. Figure 3 illustrates grid-based spatial CV using nine grid cells. We can then use the data in one grid cell

for validation and those in the remaining grid cells for model training, and then iterate this process. Similar to clustering-based spatial CV, data from multiple grid cells can be used for validation too. In addition to squares or rectangles, grid cells can be in other shapes too, such as hexagons which have often been used in GIScience research (Montello, Friedman, and Phillips 2014; Gao et al. 2017).

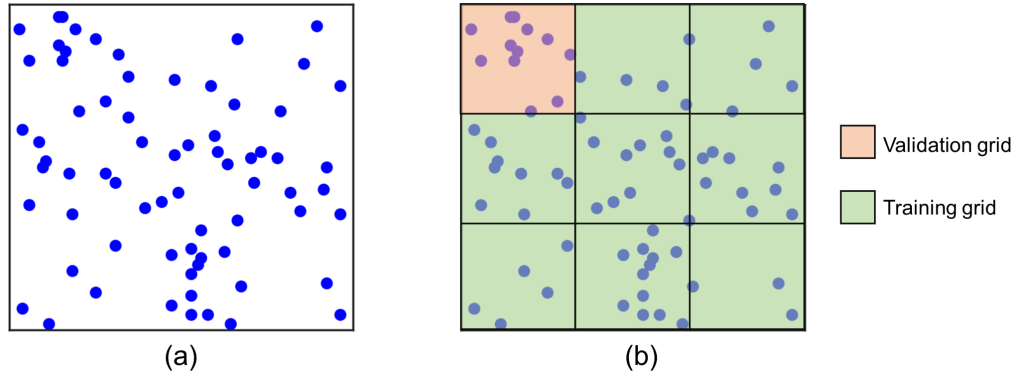


Figure 3. An illustration of grid-based spatial CV: (a) original training data; (b) training data split into nine grid cells.

Grid-based spatial CV has been implemented in the following packages: *ENMeval* (R), *spatialsample* (R), *sperrorest* (R), *Mlr3spatiotempcv* (R), *blockCV* (R), and *spacv* (Python). These packages implement rectangle-based grids, and two of them, *spatialsample* and *spacv*, also implement hexagonal grids. In addition, *sperrorest* provides an extra feature that allows users to specify a minimum number of data points inside a grid cell, and grid cells with smaller numbers of data instances will be merged into their neighboring cells.

2.3 Geo-attribute-based spatial CV

When a geo-attribute is available, such as an attribute containing county names or city district names, the data can be split based on the spatial regions defined by the geo-attribute. Figure 4 illustrates this spatial CV using four geographic regions. Since we are using pre-defined geographic regions to split the data, the evaluation results can be communicated relatively easily using the names of the geographic regions. For example, we can clearly describe the geographic regions whose data are used for model training and the region(s) whose data are used for validation in each iteration of the CV.

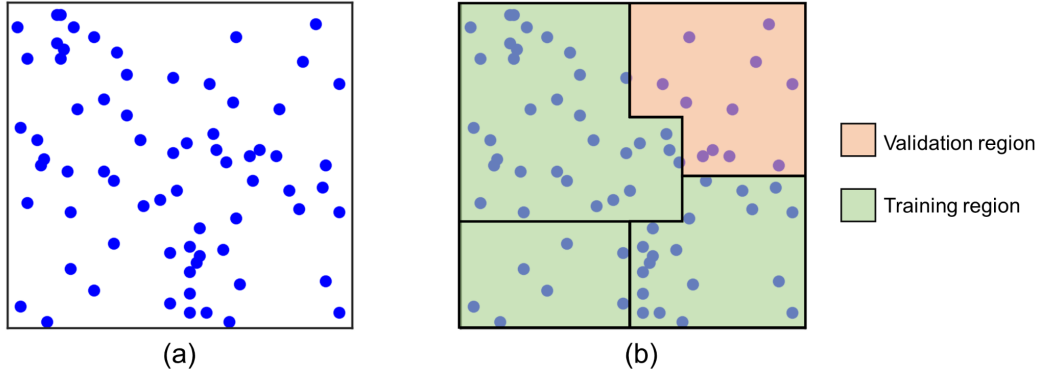


Figure 4. An illustration of geo-attribute-based spatial CV: (a) original training data; (b) training data split based on four geographic regions defined by the geo-attribute.

Geo-attribute-based spatial CV has been implemented in *Mlr3spatiotempcv* (R) and *spacv* (Python). *Mlr3spatiotempcv* requires the user to specify a target attribute, while *spacv* expects the user to provide polygons related to the geo-attribute. In addition, we can also use typical spatial data processing packages, such as *geopandas* (Python), to implement geo-attribute-based spatial CV by splitting data based on the geo-attribute.

2.4 Spatial leave-one-out CV

The fourth spatial CV method is an extension of the classic leave-one-out CV. Instead of simply using one data instance for validation and the remaining data for model training, spatial leave-one-out CV creates a buffer zone surrounding the validation data. Data falling into the buffer zone will not be used for model training. Figure 5 illustrates spatial leave-one-out CV. Note that this method is not limited to using one validation data instance only in each iteration, but can include more data within the same local region. This extension is called *spatial leave-one-disc-out* (Schratz et al. 2021), where a *disc* is a circular local region for validation, a buffer zone is created around the *disc*, and the remaining data are used for model training.

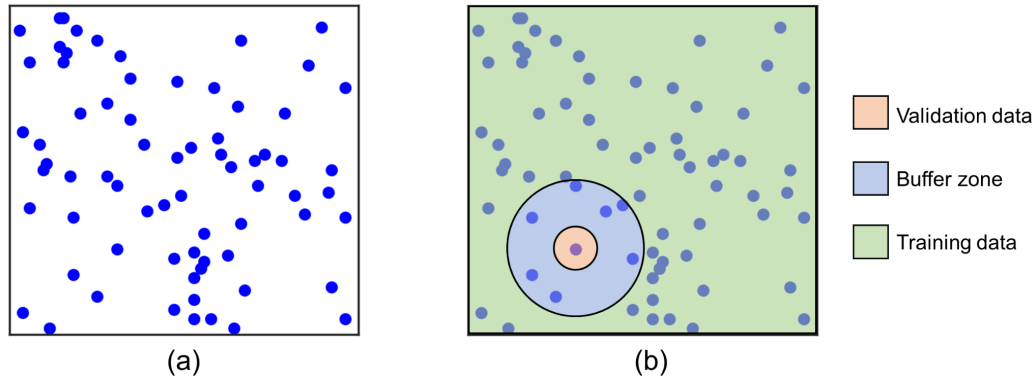


Figure 5. An illustration of spatial leave-one-out CV: (a) original training data; (b) training data split into validation data, a buffer zone (data in which are not used for model training), and training data.

Spatial leave-one-out CV has been implemented in the following packages: *CAST* (R), *Mlr3spatiotempcv* (R), *blockCV* (R), *sperrorest* (R), *spatialsample* (R), *spacv* (Python), and

museotoolbox (Python). The extended spatial leave-one-disc-out CV has been implemented in *Mlr3spatiotempcv* and *sperrorest*. In addition, the technique of adding a buffer zone to further increase the independence between training and validation data is not limited to spatial leave-one-out CV only, but has been implemented in the three other spatial CV methods as well.

2.5 Summary

We have presented four main spatial CV methods and their corresponding software package implementations. In practice, the decision of which spatial CV method to use depends on which method best mimics the real application scenario of the trained model. For example, if the trained model will be used for making predictions in predefined administrative regions, then a geo-attribute-based spatial CV may be more suitable. By contrast, if the trained model will be used to make predictions in new regions defined by grid cells (or tiles), then a grid-based spatial CV may be preferred. The principle of mimicking the real application scenario also guides our choice between spatial CV and random CV (e.g., interpolation). Each spatial CV method involves certain parameters, such as the number of clusters, the size of the grids, and the distance of the buffer zone. There is no single best way for specifying the parameter values, and existing research generally recommends considering the effect of spatial autocorrelation and domain knowledge related to the machine learning task (Karasiak et al. 2019; Valavi et al. 2019; Brenning 2022).

3. Examples based on real-world data

In this section, we use two examples based on real-world data to demonstrate the use of the four spatial CV methods and how they may affect the model evaluation results compared with using random CV. To quantify the performance of the trained models, we use two metrics, R^2 and Root Mean Square Error (RMSE) calculated using the following two equations:

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2}, \quad (1)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2}, \quad (2)$$

where y_i is the observed value of the i th data instance; \hat{y}_i is the predicted value made by the model; \bar{y} is the average of the dependent variable; and N is the total number of data instances.

3.1 Example 1: Predicting domestic violence using random forest

In the first example, we leverage the domestic violence data in the city of Chicago from our previous study (Chang et al. 2022), and use a random forest model to predict domestic violence based on socioeconomic and demographic variables. The geographic unit of analysis for this

example is census block groups (CBGs), and the time period of the data is from December 31, 2018 to January 6, 2020.

The dependent variable is the domestic violence rate at the CBGs level. The global Moran's I of the domestic violence data is 0.648 ($p < 0.001$). The independent variables are 19 socioeconomic and demographic variables in five categories: (1) race and ethnicity, (2) age, (3) social disadvantage level, (4) residential instability, and (5) urbanicity. More details about the dependent and independent variables are available in the original article (Chang et al. 2022).

We use a random forest (RF) model to predict domestic violence rate based on the 19 independent variables. The RF model has two major hyperparameters to be set: $n_estimators$ controlling the number of trees in the RF model, and $max_features$ controlling the number of features to be considered when a node of a tree is split. Here, we use the hyperparameter values identified from our previous study by setting $n_estimators$ to 80 and $max_features$ to $\sqrt{n_v}$ where n_v is the number of independent variables.

3.1.1 Data partition using spatial and random CV

We use the four spatial CV methods and a random CV method to split the data into training and validation sets. The parameters of these five CV methods are set as follows. The data splitting result of each method is visualized in Figure 6.

- *Random CV*: We employ the traditional 10-fold CV by randomly splitting all CBGs into 10 subsets.
- *Clustering-based spatial CV*: We use k-means clustering to split the whole data set also into 10 clusters (i.e., $k = 10$).
- *Grid-based spatial CV*: We use a 3×3 grid to split the data into 9 grid cells. Among these 9 grid cells, 7 of them contain data and therefore a 7-fold cross-validation is used.
- *Geo-attribute-based spatial CV*: Using a geo-attribute of community names in Chicago, we split the data (2,146 CBGs) to 96 communities.
- *Spatial leave-one-out CV*. We set the buffer distance to 0.05 quantile of all distances among CBGs. One CBG is used for each iteration of the validation process.

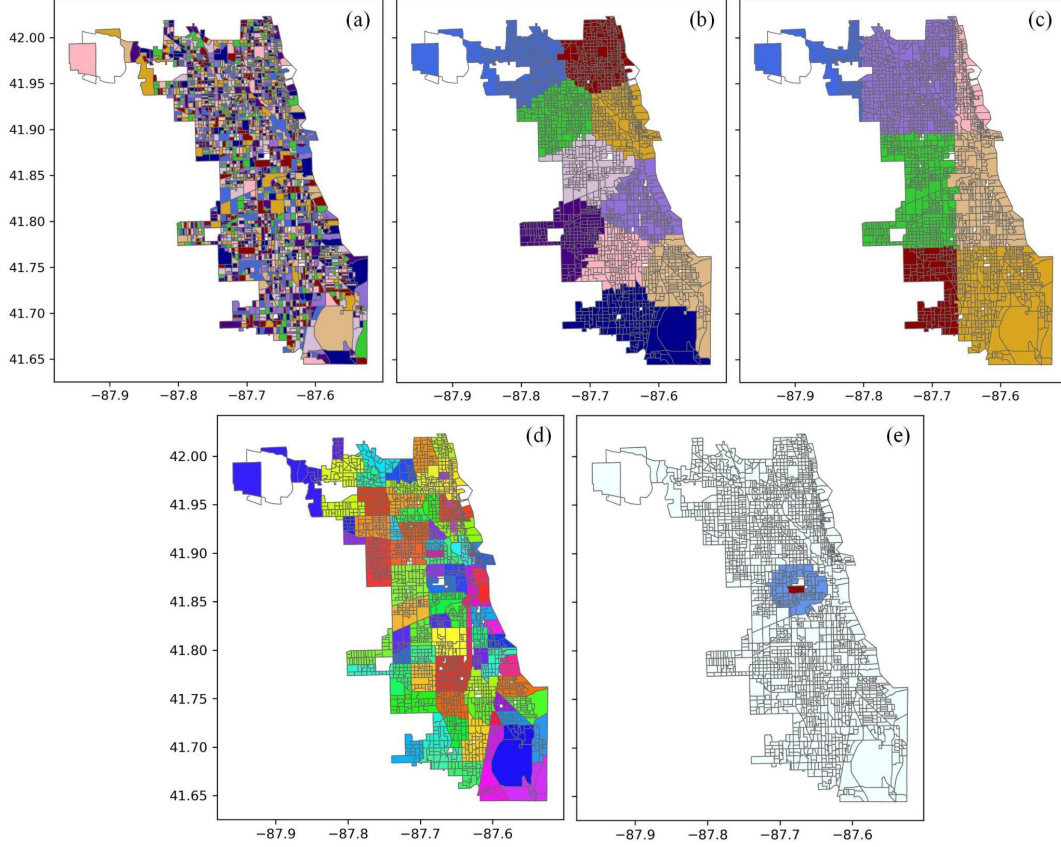


Figure 6. Data splitting results for CBGs in Chicago: (a) *random CV*; (b) *clustering-based spatial CV*; (c) *grid-based spatial CV*; (d) *geo-attribute-based spatial CV*; (e) *spatial leave-one-out CV*.

In Figure 6, each color in subfigures (a)-(d) indicates one subset of data, and subfigure (e) shows one CBG used for validation (in red color) and its surrounding buffer zone (in blue). As can be seen in subfigure (a), the training and validation data generated by random CV are scattered across the entire study area and can be spatially adjacent to each other. By contrast, the training and validation data generated by clustering-based, grid-based, and geo-attribute-based spatial CV (in subfigures (b), (c), and (d)) are more spatially separated from each other. In subfigure (c), i.e., *grid-based spatial CV*, we can also clearly see the horizontal and vertical lines formed by the 3×3 grid.

3.1.2 Results and discussion

Table 1 presents the evaluation results using different CV methods. The hyperparameters of the RF model are fixed during the entire evaluation process to ensure that the performance difference only comes from the way that the data is split.

Table 1. Performance of the RF model using random CV and four spatial CV methods.

| <i>CV method</i> | R^2 | <i>RMSE</i> |
|------------------|--------|-------------|
| <i>Random CV</i> | 0.5952 | 8.9398 |

| | | |
|---------------------------------------|--------|--------|
| <i>Clustering-based spatial CV</i> | 0.5443 | 9.4853 |
| <i>Grid-based spatial CV</i> | 0.5643 | 9.2752 |
| <i>Geo-attribute-based spatial CV</i> | 0.5667 | 9.2501 |
| <i>Spatial leave-one-out CV</i> | 0.5470 | 9.4575 |

As can be seen, different performance scores are obtained when different CV methods are used for splitting the data. The performance scores by themselves do not suggest which CV method should be ideally used; instead we should choose the CV method that best mimics the real application scenario of the trained model. Overall, the results in Table 1 show that the highest performance scores are obtained using random CV, while lower performance scores are obtained using the four spatial CV methods. The highest performance scores from random CV are likely due to the many spatially adjacent training and validation data instances generated by random CV, which allows the RF model to “peek” into nearby validation data during the training process. The performance scores obtained based on random CV are about 5% - 8% higher than those obtained via the four spatial CV methods, but are not radically different in this example. This suggests that the scores of random CV in this example are not entirely invalid even when spatial CV should ideally be used, but will likely present an overestimate of the model performance. If the real application scenario of the trained RF model is to predict domestic violence rate in CBGs of Chicago (i.e., the same study area), the evaluation scores obtained from random CV can still reflect the true performance of the model; if the application scenario is to predict domestic violence rate in nearby cities or regions, using random CV could lead to a 5% - 8% overestimate of the model performance.

3.2 Example 2: Predicting obesity prevalence using deep neural network

In this second example, we use the obesity prevalence data in New York City (NYC) from another recent study (Zhou et al. 2022), and use a deep neural network (DNN) model to predict obesity prevalence. The geographic unit of analysis is census tract and the time period of the data is the entire year of 2018.

The dependent variable is the obesity prevalence at the census tract level, and its global Moran’s I is 0.740 ($p < 0.001$). The original data were obtained from the *PLACES* project of the Centers for Disease Control and Prevention (CDC). The independent variables are 21 socioeconomic and demographic factors related to race and ethnicity, gender, age, education, income, and housing conditions. More details about the dependent and independent variables are available in the original article (Zhou et al. 2022).

We use DNN in this example for predicting obesity prevalence. The hyperparameters of the model are set based on our previous study, and the model has 9 hidden layers with 160, 208, 160, 160, 256, 32, 240, 96, and 208 neurons for them respectively.

3.2.1 Data partition using spatial and random CV

Same as the last example, we use the four spatial CV methods and a random CV method to split the data. The parameters of these five CV methods are set as follows. The splitting results are visualized in Figure 7.

- *Random CV*. We use the same classic 10-fold random CV as used in the previous example.
- *Clustering-based spatial CV*. We use the same K-means clustering method ($k=10$) to split the data.
- *Grid-based spatial CV*. We split the NYC area using a 3×3 grid, and 8 of the grid cells contain census tracts.
- *Geo-attribute-based spatial CV*. NYC consists of five boroughs. We split data into these five boroughs.
- *Spatial leave-one-out CV*. Similar to the previous example, the 0.05 quantile of distances among all census tracts is used as the buffer distance.

3.2.2 Results and discussion

Table 2 presents the evaluation results using different CV methods. Similar to the previous example, the hyperparameters of the DNN model are fixed during the entire evaluation process to ensure that the performance difference only comes from the way that the data is split.

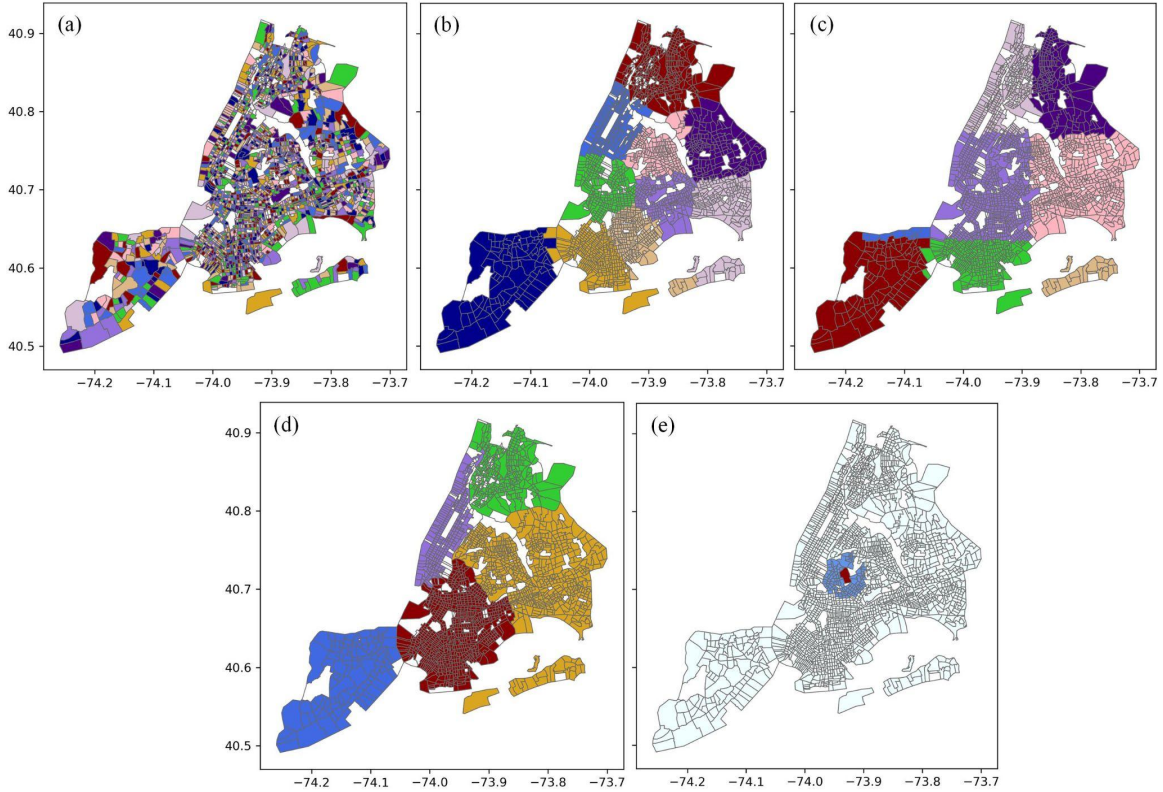


Figure 7. Data splitting results for census tracts in NYC: (a) *random CV*; (b) *clustering-based spatial CV*; (c) *grid-based spatial CV*; (d) *geo-attribute-based spatial CV*; (e) *spatial leave-one-out CV*.

Table 2. Performance of the DNN model using random CV and four spatial CV methods.

| <i>CV method</i> | R^2 | <i>RMSE</i> |
|---------------------------------------|--------|-------------|
| <i>Random CV</i> | 0.8692 | 2.1287 |
| <i>Clustering-based spatial CV</i> | 0.7244 | 3.0899 |
| <i>Grid-based spatial CV</i> | 0.7466 | 2.9624 |
| <i>Geo-attribute-based spatial CV</i> | 0.6613 | 3.4250 |
| <i>Spatial leave-one-out CV</i> | 0.8083 | 2.5766 |

As can be seen from Table 2, the overall pattern of the obtained performance scores holds, i.e., the scores from random CV are the highest compared with the scores obtained via the four spatial CV methods. As shown in Figure 7(a), random CV again generates many training and validation census tracts that are spatially adjacent, allowing the DNN model to “peek” into nearby validation census tracts during the training process. The performance scores from random CV are 8% - 31% higher. In particular, the random CV scores are about 31% higher than the performance scores obtained via *geo-attribute-based spatial CV*. Note that the spatial autocorrelation in the obesity prevalence data is higher than that in the domestic violence data used in the previous example. This higher spatial autocorrelation likely contributes to the higher overestimate of performance scores from random CV. If the real application scenario of the trained model is to predict into other nearby geographic regions not in the training data, using random CV in this case could present a relatively large overestimate of the model performance.

4. Conclusions

Cross-validation is an evaluation approach that has been widely used in GeoAI research. In this chapter, we have discussed the concept of spatial CV and why we may need it over random CV under certain situations (e.g., extrapolation). We then present four main spatial CV methods identified and synthesized from the multidisciplinary literature, which are: *clustering-based spatial CV*, *grid-based spatial CV*, *geo-attribute-based spatial CV*, and *spatial leave-one-out CV*. We also list the software packages that have implemented these methods and their corresponding programming languages to inform the choice of interested readers. Two examples based on real-world data are used to demonstrate the different performance scores obtained via spatial CV and random CV. While the two examples involve different topics, study areas, and machine learning models, the obtained results share a similar pattern, i.e., the performance scores obtained via random CV tend to be higher than performance scores obtained via spatial CV methods. While random CV can still provide some extent of evaluation when spatial CV should ideally be used, we need to be cautious about these scores and know that those evaluation scores from random CV are likely to be overestimates of the true model performance. The ultimate goal of model evaluation is to provide realistic estimates that can reflect the true performance of a model

when it is put into use. We hope that these spatial CV methods and their software packages can help expand the model evaluation toolbox of researchers in GeoAI and spatial data science.

Data and code availability statement

Data and code supporting this work are available at: <https://github.com/geoai-lab/spatialCV>

References:

- Airola, Antti, Jonne Pohjankukka, Johanna Torppa, Maarit Middleton, Vesa Nykänen, Jukka Heikkonen, and Tapio Pahikkala. 2019. “The Spatial Leave-Pair-out Cross-Validation Method for Reliable AUC Estimation of Spatial Classifiers.” *Data Mining and Knowledge Discovery* 33 (3). Springer: 730–747.
- Brenning, Alexander. 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*, 5372–5375. IEEE.
- Brenning, Alexander. 2022. “Spatial Machine-Learning Model Diagnostics: A Model-Agnostic Distance-Based Approach.” *International Journal of Geographical Information Science*. Taylor & Francis, 1–23.
- Browne, Michael W. 2000. “Cross-Validation Methods.” *Journal of Mathematical Psychology* 44 (1). Elsevier: 108–132.
- Burman, Prabir. 1989. “A Comparative Study of Ordinary Cross-Validation, v-Fold Cross-Validation and the Repeated Learning-Testing Methods.” *Biometrika* 76 (3). Oxford University Press: 503–514.
- Chang, Ting, Yingjie Hu, Dane Taylor, and Brian M. Quigley. 2022. “The Role of Alcohol Outlet Visits Derived from Mobile Phone Location Data in Enhancing Domestic Violence Prediction at the Neighborhood Level.” *Health & Place* 73 (January): 102736. doi:10.1016/j.healthplace.2021.102736.
- Crosby, Henry, Theodoros Damoulas, and Stephen A. Jarvis. 2020. “Road and Travel Time Cross-Validation for Urban Modelling.” *International Journal of Geographical Information Science* 34 (1). Taylor & Francis: 98–118.
- Da Silva, Tiago Pinho, Antonio RS Parmezan, and Gustavo EAPA Batista. 2021. “A Graph-Based Spatial Cross-Validation Approach for Assessing Models Learned with Selected Features to Understand Election Results.” In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 909–915. IEEE.
- Diniz-Filho, José Alexandre Felizola, Luis Mauricio Bini, and Bradford A. Hawkins. 2003. “Spatial Autocorrelation and Red Herrings in Geographical Ecology.” *Global Ecology and Biogeography* 12 (1). Wiley Online Library: 53–64.
- Fotheringham, A. Stewart. 2009. “‘The Problem of Spatial Autocorrelation’ and Local Spatial Statistics.” *Geographical Analysis* 41 (4). Wiley Online Library: 398–403.

- Gao, Song, Krzysztof Janowicz, Daniel R. Montello, Yingjie Hu, Jiue-An Yang, Grant McKenzie, Yiting Ju, Li Gong, Benjamin Adams, and Bo Yan. 2017. “A Data-Synthesis-Driven Method for Detecting and Extracting Vague Cognitive Regions.” *International Journal of Geographical Information Science* 31 (6). Taylor & Francis: 1245–1271.
- Géron, Aurélien. 2019. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media.
- Goodchild, Michael F., and Wenwen Li. 2021. “Replication across Space and Time Must Be Weak in the Social and Environmental Sciences.” *Proceedings of the National Academy of Sciences* 118 (35). National Acad Sciences.
- Hijmans, Robert J. 2012. “Cross-validation of Species Distribution Models: Removing Spatial Sorting Bias and Calibration with a Null Model.” *Ecology* 93 (3). Wiley Online Library: 679–688.
- Karasiak, Nicolas. 2020. “Museo ToolBox: A Python Library for Remote Sensing Including a New Way to Handle Rasters.” *Journal of Open Source Software* 5 (48): 1978.
- Karasiak, Nicolas, Jean-François Dejoux, Mathieu Fauvel, Jérôme Willm, Claude Monteil, and David Sheeren. 2019. “Statistical Stability and Spatial Instability in Mapping Forest Tree Species by Comparing 9 Years of Satellite Image Time Series.” *Remote Sensing* 11 (21). MDPI: 2512.
- Karasiak, Nicolas, J.-F. Dejoux, Claude Monteil, and David Sheeren. 2022. “Spatial Dependence between Training and Test Sets: Another Pitfall of Classification Accuracy Assessment in Remote Sensing.” *Machine Learning* 111 (7). Springer: 2715–2740.
- Meyer, Hanna, and Edzer Pebesma. 2021. “Predicting into Unknown Space? Estimating the Area of Applicability of Spatial Prediction Models.” *Methods in Ecology and Evolution* 12 (9). Wiley Online Library: 1620–1633.
- Meyer, Hanna, and Edzer Pebesma. 2022. “Machine Learning-Based Global Maps of Ecological Variables and the Challenge of Assessing Them.” *Nature Communications* 13 (1). Nature Publishing Group UK London: 2208.
- Meyer, Hanna, Christoph Reudenbach, Tomislav Hengl, Marwan Katurji, and Thomas Nauss. 2018. “Improving Performance of Spatio-Temporal Machine Learning Models Using Forward Feature Selection and Target-Oriented Validation.” *Environmental Modelling & Software* 101. Elsevier: 1–9.
- Montello, Daniel R., Alinda Friedman, and Daniel W. Phillips. 2014. “Vague Cognitive Regions in Geography and Geographic Information Science.” *International Journal of Geographical Information Science* 28 (9). Taylor & Francis: 1802–1820.
- Muscarella, Robert, Peter J. Galante, Mariano Soley-Guardia, Robert A. Boria, Jamie M. Kass, María Uriarte, and Robert P. Anderson. 2014. “ENM Eval: An R Package for Conducting Spatially Independent Evaluations and Estimating Optimal Model Complexity for Maxent Ecological Niche Models.” *Methods in Ecology and Evolution* 5 (11). Wiley Online Library: 1198–1205.
- Pohjankukka, Jonne, Tapio Pahikkala, Paavo Nevalainen, and Jukka Heikkonen. 2017. “Estimating the Prediction Performance of Spatial Models via Spatial K-Fold Cross Validation.” *International Journal of Geographical Information Science* 31 (10). Taylor & Francis: 2001–2019.
- Roberts, David R., Volker Bahn, Simone Ciuti, Mark S. Boyce, Jane Elith, Gurutzeta Guillera-Arroita, Severin Hauenstein, José J. Lahoz-Monfort, Boris Schröder, and

- Wilfried Thuiller. 2017. “Cross-Validation Strategies for Data with Temporal, Spatial, Hierarchical, or Phylogenetic Structure.” *Ecography* 40 (8). Wiley Online Library: 913–929.
- Schratz, Patrick, Marc Becker, Michel Lang, and Alexander Brenning. 2021. “Mlr3spatiotempcv: Spatiotemporal Resampling Methods for Machine Learning in R.” *ArXiv Preprint ArXiv:2110.12674*.
- Tobler, W. R. 1970. “A Computer Movie Simulating Urban Growth in the Detroit Region.” *Economic Geography* 46. [Clark University, Wiley]: 234–240. doi:10.2307/143141.
- Valavi, Roozbeh, Jane Elith, José J. Lahoz-Monfort, and Gurutzeta Guillera-Aroita. 2019. “BlockCV: An r Package for Generating Spatially or Environmentally Separated Folds for k-Fold Cross-Validation of Species Distribution Models.” *Methods in Ecology and Evolution*, no. 10. Cold Spring Harbor Laboratory: 225–232.
- Zhou, Ryan Zhenqi, Yingjie Hu, Jill N. Tirabassi, Yue Ma, and Zhen Xu. 2022. “Deriving Neighborhood-Level Diet and Physical Activity Measurements from Anonymized Mobile Phone Location Data for Enhancing Obesity Estimation.” *International Journal of Health Geographics* 21 (1). BioMed Central: 1–18.