HexaClus: Interpretable Hexagonal Supervised Spatial Clustering

Yameng Guo[1,*,†], Seppe vanden Broucke[1,2,†]

[1]Department of Business Informatics and Operations Management, Ghent University, Tweekerkenstraat 2, 9000 Gent, Belgium
[2]Research Centre for Information Systems Engineering, KU Leuven, Naamsestaat 69, 3000 Leuven, Belgium

Abstract

Geospatial inference is a critical component in many domains. Conventional methodologies such as Kriging and Geographically Weighted Regression (GWR) exhibit inherent limitations in inference capabilities, while contemporary machine learning approaches demonstrate superior predictive accuracy but significantly lack interpretability within spatial contexts. To bridge this gap, this work introduces HexaClus, a novel interpretable supervised spatial clustering framework based on mergers over a hexagonal grid that effectively resolves the tension between predictive performance and interpretability in geospatial analysis. HexaClus partitions a study area into hexagonal cells, assigns geospatial features to each cell, and iteratively merges adjacent regions based on predictive consistency to form spatially coherent clusters. A key innovation of HexaClus lies in its architecture-agnostic design, enabling seamless integration with any machine learning model. Through an empirical evaluation, we demonstrate that HexaClus substantially outperforms traditional geospatial models and black-box machine learning techniques across dual dimensions of inference capability and interpretability. The complete implementation of HexaClus is publicly accessible for reproducibility and further development[1].

Keywords

geospatial clustering, supervised learning, geospatial interpretation, estates evaluation, hexagonal lattice,

1. Introduction

Geospatial inference plays a crucial role in addressing both environmental and socio-economic issues [1, 2, 3], increasing the demand for accurate yet interpretable geospatial models. Such models enable decision makers to analyze spatial trends, allocate resources efficiently, and mitigate potential risks. However, balancing predictive accuracy and interpretability remains a significant challenge, particularly in the presence of spatial dependencies and heterogeneous environments.

Traditional geospatial inference methodologies, such as kriging [4, 5] and Geographically Weighted Regression (GWR)[6], have been widely used to model spatial relationships by assigning different weights to nearby observations. Although these methods provide solid mathematical and statistical foundations, they face several limitations in inference capability, interpretability, and generalization. For example, both Kriging and GWR assume stationarity, making them less effective in complex terrains and urban environments where spatial relationships vary significantly [7, 8]. Moreover, they are highly sensitive to parameter choices, such as the selection of the bandwidth in GWR [9] and the variogram parameters in Kriging [10], which can significantly affect the performance of constructed models. Another major drawback is their limited ability to incorporate external geospatial features, such as Points of Interest (POIs), which are essential for modeling human activities and land use patterns. Furthermore, their predictive capabilities remain constrained, as they are primarily designed for interpolation rather than robust inference in the presence of varying spatial relationships.

Stronger machine learning (ML) techniques have emerged as a powerful alternative to geospatial prediction due to their robustness and high predictive accuracy. These models can handle large-scale datasets and complex feature interactions, making them well-suited for geospatial tasks. However, this increase in predictive power typically comes at a cost of hampering interpretability, limiting their utility in many practical geospatial inference settings. Ensemble models are often considered to act as "black boxes", making it difficult to understand why certain regions exhibit higher or lower predicted values. Although interpretation techniques such as SHAP (Shapley Additive Explanations) [11] can be used in this context, they generally lack a geospatial-specific perspective, failing to fully capture spatial dependencies and relationships between adjacent areas in an appealing manner.

To address these challenges, we propose ***HexaClus***, a novel hexagonal supervised spatial clustering method that balances interpretability and predictive precision. Our approach starts with dividing the study area into hexagonal cells, and then aggregates geospatial features within each cell. Next, local supervised models are constructed per cell and are used to iteratively merge adjacent hexes until a validation set-based convergence criterion is met. By developing this algorithm, we aim to:

- Improve interpretability of geospatial models by structuring spatial clusters, providing an intuitive visualization of geospatial distributions and predictions.
- Develop a flexible framework that dynamically integrates local geospatial information. Unlike fixed-radius aggregation approaches around given target instances, our method enables the localized incorporation of spatial embeddings on a smaller scale over the hexagonal grid, ensuring a more accurate representation of local geospatial features.
- Remain compatible with various advanced machine learning models whilst maintaining an interpretable clustering structure.
- Provide an interpretable and adaptive approach to geospatial inference by leveraging hexagonal spatial partitioning and supervised clustering.

An implementation of our work is made publicly available[1].

STRL'25: Fourth International Workshop on Spatio-Temporal Reasoning and Learning, 16 August 2025, Montreal, Canada
*Corresponding author.
†These authors contributed equally.
yameng.guo@ugent.be (Y. Guo); seppe.vandenbroucke@ugent.be (S. vanden Broucke)
0000-0003-2719-1356 (Y. Guo); 0000-0002-8781-3906 (S. vanden Broucke)

[1]See: https://github.com/ArmonGo/HexaClus

2. Related Works

2.1. Geospatial Inference Techniques

Geospatial modeling relies on capturing spatial dependencies to improve inference accuracy. Existing approaches can be broadly categorized into three main groups: statistical models, machine learning-based methods, and deep learning-based methods.

Traditional statistical approaches such as Kriging [4, 5] and Geographically Weighted Regression (GWR) [6] are widely used in geospatial modeling. These methods leverage spatial dependencies by assigning weights to nearby observations. Kriging, a geostatistical interpolation technique, predicts unknown values based on the spatial correlation of known data points. However, its effectiveness is generally limited in heterogeneous landscapes due to the assumption of stationarity [7, 8].

Similarly, GWR extends traditional regression modeling by allowing coefficients to vary spatially, offering insights into how relationships change across different regions. Though GWR improves spatial interpretability, it is sensitive to bandwidth selection [9] and struggles with high-dimensional feature spaces. Furthermore, both Kriging and GWR have limited predictive capabilities, as they are primarily designed for interpolation rather than robust inference.

In practical applications, both methods also impose strict mathematical conditions, leading to complex nonlinear computations that are often computationally expensive [12], causing these models to also be less suitable for large-scale datasets.

Next, machine learning (ML) techniques, particularly tree-based ensemble methods like Random Forest [13] and Gradient Boosting [14, 15, 16], have gained popularity in geospatial inference due to their robustness and predictive accuracy. These models efficiently handle non-linear relationships and large datasets while also offering flexibility in terms of being able to incorporate local Points of Interest (POIs). However, integrating POIs often requires extensive feature engineering [17] to query the surrounding area around targets of interest and then appropriately aggregate POI counts. Typically, a predefined fixed-radius approach is used to define a restricted area (in the shape of a box or circle) to incorporate spatial context, but this may fail to capture dynamic spatial interactions effectively.

A promising direction is the combination of machine learning and spatial correlation modeling. Some hybrid approaches [18, 19] integrate both spatial statistic models and machine learning models to improve geospatial generalization. Yet, the lack of explicit spatial structure often results in reduced interpretability compared to traditional geostatistical models.

Deep learning, in particular Graph Neural Networks (GNNs), has emerged as a promising approach for geospatial modeling [20, 21]. GNNs represent spatial structures as graphs, enabling models to learn complex spatial relationships. However, despite their theoretical advantages, GNN-based geospatial inference faces significant practical challenges related to graph construction and training. First, there is currently no standard well-accepted method for defining an appropriate spatial graph, as most approaches rely on arbitrary distance thresholds. Second, empirical studies have shown that GNNs can struggle to outperform classical machine learning methods in geospatial inference while requiring significantly higher computational resources and tuning efforts [22].

2.2. Interpretability and Visualization

Compared to other inference tasks, geospatial inference places greater emphasis on interpretability due to its involvement with complex spatial relationships, heterogeneous data sources, and real-world decision-making processes, all of which require a high level of trust and transparency.

A common approach to interpreting model mechanisms is assessing feature importance using methods such as SHAP (Shapley Additive Explanations)[11] and LIME (Local Interpretable Model-agnostic Explanations)[23]. While these techniques provide an initial understanding of feature attributions—including e.g. coordinate features—they fail to show local spatial dependencies and variations in an appealing manner as the spatial coordinates are regarded as normal features.

Beyond feature importance, prediction results also play a crucial role in interpretability. Since output values are often tied to specific locations, researchers commonly aggregate raw predictions and visualize them spatially using heatmaps or region-based summaries [24, 25] within administrative boundaries. Additionally, methods such as GWR, Kriging, and Gaussian Processes [26] enable spatially varying coefficients or spatial uncertainty estimation, offering a deeper insight into local spatial variations and uncertainties.

Despite the availability of various methods for interpreting geospatial predictions, they share common challenges. For example, most spatial visualization rely on predefined administrative boundaries, which may not align with actual spatial variations that transcend these divisions. Furthermore, existing map-based techniques, such as hotspot maps and density-based visualizations, are often sensitive to data point density variations, making them susceptible to data collection bias rather than reflecting true spatial patterns. For instance, urban areas may appear as high-risk zones simply due to there being more instances in those areas rather than there being genuine geospatial variation.

To address the previously mentioned challenges, we propose a straightforward hexagonal supervised spatial clustering framework that provides an adaptive and interpretable solution to balance predictive accuracy and spatial interpretability. Unlike fixed-radius approaches, our framework dynamically integrates spatial embeddings, enabling a more context-aware representation of geospatial features. Additionally, HexaClus supports a variety of machine learning models while preserving an interpretable spatial structure, ensuring strong predictive performance. Most importantly, it facilitates clear geospatial visualization and enables local variation analysis beyond administrative boundaries by structuring spatial clusters in a hexagonal format.

3. Methodology

The HexaClus algorithm is built on hexagonal partitions and operates by iteratively clustering geospatial regions using a supervised hierarchical merging strategy. The cell shape of a grid system is an important consideration. We opt for hexagonal cell shapes over e.g. square or other covering partitioning strategies due to their appealing properties. For simplicity, it should be constant-area polygon that tiles regularly, i.e. the triangle, the square, or the hexagon. Of these,

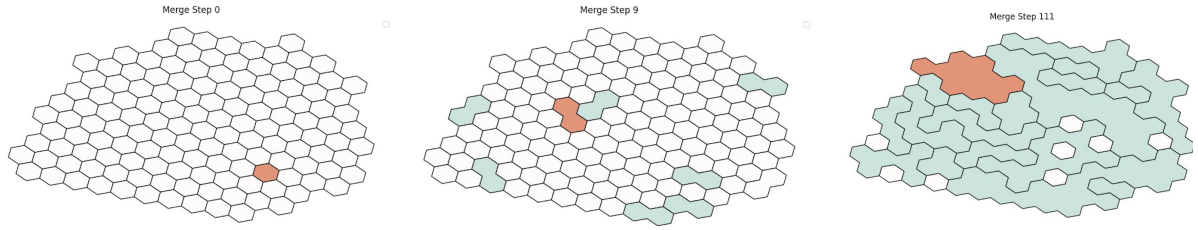

Figure 1: Hexagonal cell clustering and merging process

The area under study is initially divided into hexagonal cells based on a specified resolution. Then, pairs of touching polygons that yield the greatest reduction in training MSE are identified and merged into a single polygon. This process is repeated iteratively until a convergence condition is met.

triangles and squares have neighbors with different distances. Triangles have three different distances, and squares have two different distances. For hexagons, all neighbors are equidistant and additionally have the property of expanding rings of neighbors approximating circles. Hexagons are also optimally space-filling. On average, a polygon may be filled with hexagon tiles with a smaller margin of error than would be present with square tiles. Particularly, in the context of our clustering task, the uniform structure helps reduce directional bias and prevents potential distortions in spatial patterns when aggregating features among all neighbors. Note that other polygonal covering strategies, such as e.g. based on Voronoi maps, could be considered without significant effort, but are left out of scope for this work.

Algorithm 1 and Figure 1 outline the steps to construct a clustering solution solution.

Given a training and validation set (the latter used for stopping when convergence is reached), as well as a set of initial (hexagonal) cells, the algorithm begins by removing all cells which do not contain any train instances. To construct the set of initial hexagonal cells, we utilize Uber's h3 library [2], which can construct hexagonal cells for each location on the globe with different sizes, using a resolution parameter r.

Next, for each initial cell, one can optionally construct a spatial representation vector containing point of interest (POI) information, land use data information, or terrain attributes describing the initial cell. Whilst this is a flexible component which can be either skipped or easily extended by end users, we by default utilize the Python library SRAI to obtain a vector containing spatial information given an initial hexagonal cell [27], similarly to what is done in e.g. Hex2Vec [28].

Next, a prediction model base learner should be specified which is used to train over a set of spatially located instanced to predict their target value. We assume target values to be continues so that regression models are constructed, but our approach can be easily extended to a binary classification setting as well. Before the merging procedure of cells can start, a local regression model is hence trained for each of the initial cells. To avoid issues of overfitting and make the resulting local models more interpretable, we use Ridge Regression as our default choice to construct the models, as it applies the $L2$ penalty on large coefficients to prevent overfitting. The objective is defined as

$$\min_{\boldsymbol{\beta}} \ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 \tag{1}$$

where λ controls regularization strength. A single regularization strength value for all constructed models is used (which can be globally tuned if desired).

Next, the algorithm enters an iterative merging phase, where spatially **neighboring** pairs of polygons (hexagonal cells initially) are evaluated for a potential merge. For this, a neighborhood function should be specified which can indicate whether a pair of polygons are valid neighbors. By default, we simply consider touching polygons to be valid. For each candidate pair, with their associated local models, a new model is trained using the combined set of train instances, and the reduction of *train loss* is utilized to select the best pair to merge. The merge is finalized, and the process continues until no further improvement is observed on the *validation loss* of the solution so far as a whole for a predefined number of iterations (early stopping with patience), ensuring that the final clusters are both spatially coherent and predictive.

Note that to obtain predictions for instances, the polygon-local model is retrieved, corresponding with the polygon the instance falls into. Note here that by design, HexaClus does not support inference for geospatial extrapolated points; instead, it restricts predictions to regional instances to prevent spatial variation from affecting extrapolation performance (alternatively, a default prediction corresponding to the mean train target value can be returned). Also remark that in the case where spatial representation vectors where constructed for the initial hexagonal cells, the features of the given instance need to be concatenated with the representational vector corresponding with *the initial hexagonal cell* the instance falls into.

Therefore, for a new unseen data point, we identify both its original hexagonal cell (before merging) and its corresponding merged polygon, and then concatenate the representational vector derived from the **initial cell**, but use the local model fitted on the **merged polygon** to make the prediction.

Note that instead of using a strong internal leaner, we deliberately use Ridge Regression for a couple of reasons. First, individual cells often contain a limited number of observations, making complex models prone to overfitting, while Ridge Regression, with its L2 regularization, can mitigate this effect. Additionally, Ridge is computationally efficient, allowing the merging process to scale effectively. Finally, Ridge provides an interpretable framework, where feature coefficients can be analyzed per polygon to explain spatial variations, addressing a major limitation of black-box machine learning methods in geospatial inference. While Ridge Regression serves as the default model, HexaClus is flexible

[2]See: https://h3geo.org/.

Algorithm 1 HexaClus: Supervised Hexagonal Spatial Clustering

Input: $\mathcal{D}_{\rangle} = \{(x_i, y_i, \mathbf{f}_i, t_i)\}_{i=1}^{N}$: train dataset with spatial coordinates (x_i, y_i), feature vectors $\mathbf{f}_i \in \mathbb{R}^n$, and target values t_i
Input: $\mathcal{D}_{|}^{\sqsubseteq} = \{(x_j, y_j, \mathbf{f}_j, t_j)\}_{j=1}^{N}$: validation dataset
Input: $\mathcal{F}$: regression model learner
Input: $\mathcal{H}$: initial set of cells (e.g. hexagonal cells with r the hexagonal resolution)
Input: $p_{\max}$: maximum patience threshold
Input: E: maximum number of epochs
Input: $N(p_1 \in \hat{\mathcal{H}}, p_2 \in \hat{\mathcal{H}})$ boolean neighborhood function indicating whether two polygons are valid neighbors
Input: $R(p \in \mathcal{H})$: function returning optional non-empty feature vector for an initial cell
Output: $\hat{\mathcal{H}}^*$: optimized hexagonal partitions
Output: $\hat{\mathcal{M}}^*$: polygon-local models

function ConcatFeats($\mathcal{D}, \mathcal{H}$)
  $\mathcal{D}' = \emptyset$
  for $H_n \in \mathcal{H}$ **do**
    for $(x_i, y_i, \mathbf{f}_i, \dots) \in contains(H_n, \mathcal{D})$ **do**
      $\mathcal{D}' \leftarrow \mathcal{D}' \cup \{(\mathbf{f}_i \mid R(H_n), \dots)\}$
        ▷ Concatenate spatial features to instance features
    end for
  end for
  return $\mathcal{D}'$
end function

function MseLoss($\mathcal{D}, \hat{\mathcal{H}}, \hat{\mathcal{F}}$)
  for $H_n, F_n \in \hat{\mathcal{H}}, \hat{\mathcal{F}}$ **do**
    for $(x_i, y_i, \mathbf{f}_i, t_i) \in contains(H_n, \mathcal{D})$ **do**
      $L \leftarrow L + (F_n(x_i, y_i, \mathbf{f}_i) - t_i)^2$
    end for
  end for
  return $L/|\mathcal{D}|$
end function

Initialization
$\hat{\mathcal{H}} \leftarrow \{H_n | H_n \in \mathcal{H} : \exists D_i \in \mathcal{D}_{\rangle}, contains(H_n, D_i)\}$
$\hat{\mathcal{M}} \leftarrow \{F_n \leftarrow \mathcal{F}(ConcatFeats(contains(H_n, \mathcal{D}), \mathcal{H})) | H_n \in \hat{\mathcal{H}}\}$

Iterative Merging
$p \leftarrow 0$ ▷ Initialize patience
for $e = 1$ to E **do**
  $\mathcal{L}_{\text{val}}^{\text{prev}} \leftarrow MseLoss(\mathcal{D}^{\sqsubseteq}, \hat{\mathcal{H}}, \hat{\mathcal{F}})$ ▷ Initial validation loss
  $\mathcal{C} = \emptyset$ ▷ Initialize candidate merge set
  for $H_i \in \hat{\mathcal{H}}, H_{j>i} \in \hat{\mathcal{H}} : N(H_i, H_j)$ **do**
    $H^* \leftarrow H_i \cup H_j$ ▷ Merged polygons
    $D^* \leftarrow contains(H_i, \mathcal{D}) \cup contains(H_j, \mathcal{D})$ ▷ Merged instances
    $F^* \leftarrow \mathcal{F}(ConcatFeats(D^*, \mathcal{H}))$ ▷ Construct model over merged instances
    $\Delta\mathcal{L} = MseLoss(D^*, H^*, F^*) - MseLoss(contains(H_i, \mathcal{D}), \{H_i\}, \{F_i\}) - MseLoss(contains(H_j, \mathcal{D}), \{H_j\}, \{F_i\})$ ▷ Calculate train loss reduction
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{(H_i, H_j, H^*, F^*, \Delta\mathcal{L})\}$
  end for
  $H_i, H_j, H^*, F^*, \Delta\mathcal{L} = \arg\max_{(H_i, H_j, H^*, F^*, \Delta\mathcal{L}) \in \mathcal{C}} \Delta\mathcal{L}$
  Update partitions: $\hat{\mathcal{H}} \leftarrow \hat{\mathcal{H}} \setminus \{H_i, H_j\} \cup \{H^*\}$
  Update models: $\hat{\mathcal{M}} \leftarrow \hat{\mathcal{M}} \setminus \{F_i, F_j\} \cup \{F^*\}$
  $\mathcal{L}_{\text{val}}^{\text{new}} \leftarrow MseLoss(\mathcal{D}^{\sqsubseteq}, \hat{\mathcal{H}}, \hat{\mathcal{F}})$ ▷ New validation loss
  if $\mathcal{L}_{\text{val}}^{\text{new}} < \mathcal{L}_{\text{val}}^{\text{prev}}$ **then** ▷ Save the best solution so far
    $p \leftarrow 0$
    $\hat{\mathcal{H}}^* \leftarrow \hat{\mathcal{H}}$
    $\hat{\mathcal{M}}^* \leftarrow \hat{\mathcal{M}}$
  else
    $p \leftarrow p + 1$
  end if
  if $p \geq p_{\max}$ **then**
    break for
  end if
end for

to this regard and can integrate more complex supervised learning algorithms when desired.

4. Experimental Setup

In this section, we compare our proposed method in terms of predictive performance across multiple real-world datasets with the baseline default strategy of considering a single global model.

We utilize three publicly available property valuation datasets covering London, New York, and Paris. All datasets are sourced from the Kaggle data hub, with links being available on our GitHub repository de quo. Each dataset is partitioned into a training set (70%), validation set (10%), and test set (20%). Comparisons between our method and the global Ridge model use the same data partitions.

To evaluate the performance in the setup, we compare a global Ridge Regression model with HexaClus. For the global model, we tune the regularization strength hyperparameter α over a range of 0.1 to 0.9 in increments of 0.1. For HexaClus, we tune the hexagonal resolution parameter r, varying from 5 to 9, and do not tune the regularization strength hyperparameter α, instead simply keeping it constant to 0.1.

5. Results

Table 1 presents a comparison of RMSE (Root Mean Squared Error) between the global model and our proposed approach across the three datasets. The results clearly demonstrate that HexaClus outperforms the global approach on all three datasets, highlighting the effectiveness and predictive capability of our clustering approach across different urban environments.

Dataset	Global Ridge Reg.	HexaClus
London	0.0823	**0.0706**
New York	0.0909	**0.0743**
Paris	0.0213	**0.0205**

Table 1
Performance comparison between RMSE of the global Ridge Regression model and Hexaclus across datasets.

In addition, we provide three visualized clustering solution maps (Figure 2, 3, and 4) for London, New York, and Paris, colored based on the local predictions generated by HexaClus. These maps represent property prices on top of the clustered cells, where blue indicates comparatively lower prices and red represents higher prices.

Notably, the figures reveal irregular boundaries of high-price areas that do not strictly conform to administrative regions. By adopting a hexagonal cell-based interpretation paradigm, local variations in target values become more apparent. For instance, in London, distinct high-price areas emerge, which might otherwise be obscured by over-smoothed summary maps.

6. Conclusions

In this work, we proposed HexaClus, a novel interpretable hexagonal supervised spatial clustering method that effectively balances predictive performance and interpretability in geospatial inference. By leveraging a hexagonal grid

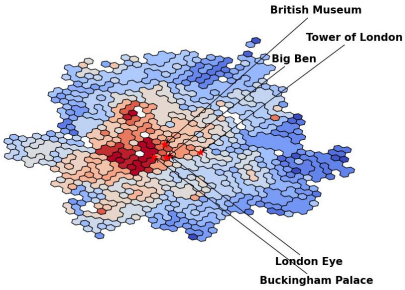

Figure 2: London property prices clustering solution visualized with landmarks

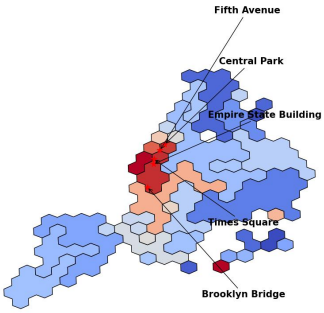

Figure 3: New York property prices clustering solution visualized with landmarks

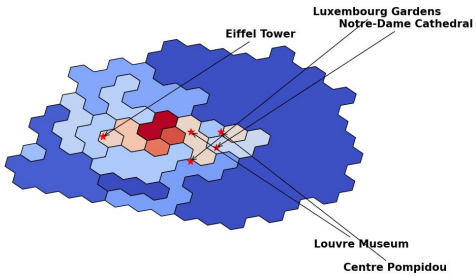

Figure 4: Paris property prices clustering solution visualized with landmarks

structure, our approach dynamically merges adjacent regions based on prediction consistency, ensuring that the final clusters reflect meaningful spatial patterns, but meanwhile provide reliable predictions. Unlike traditional geospatial methods which rely on strong stationarity assumptions, HexaClus provides a flexible framework to incorporate local geospatial features while maintaining spatial interpretability. Furthermore, in contrast to black-box machine learning models, HexaClus offers a transparent and explainable clustering process, making it particularly useful for geospatial decision making applications.

Several areas for future improvement exist. First, optimizing the computational efficiency of the merging process remains an important direction, particularly for large-scale datasets. Second, while we employ Ridge Regression as a simple and interpretable local model, future iterations can incorporate more complex models, such as tree-based methods, to improve predictive performance. Third, considering different initialization strategies other than the hexagonal approach considered in this work can be an interesting avenue for future work as well. Finally, extensive evaluation across diverse geographic areas will be necessary to assess the generalizability of HexaClus across different spatial contexts.

References

[1] P. K. Rai, V. N. Mishra, P. Singh, et al., Geospatial Technology for Landscape and Environmental Management, Springer, 2022.

[2] J. K. Thakur, S. K. Singh, A. Ramanathan, M. B. K. Prasad, W. Gossel, Geospatial techniques for managing environmental resources, Springer Science & Business Media, 2012.

[3] N. N. Kussul, B. V. Sokolov, Y. I. Zyelyk, V. A. Zelentsov, S. V. Skakun, A. Y. Shelestov, Disaster risk assessment based on heterogeneous geospatial information, Journal of Automation and Information Sciences 42 (2010).

[4] D. G. Krige, A statistical approach to some basic mine valuation problems on the witwatersrand, Journal of the Southern African Institute of Mining and Metallurgy 52 (1951) 119–139.

[5] G. Matheron, Principles of geostatistics, Economic geology 58 (1963) 1246–1266.

[6] C. Brunsdon, S. Fotheringham, M. Charlton, Geographically weighted regression, Journal of the Royal Statistical Society: Series D (The Statistician) 47 (1998) 431–443.

[7] J. P. Kleijnen, W. C. Van Beers, Robustness of kriging when interpolating in random simulation with heterogeneous variances: some experiments, European Journal of Operational Research 165 (2005) 826–834.

[8] G. De Marsily, F. Delay, J. Gonçalvès, P. Renard, V. Teles, S. Violette, Dealing with spatial heterogeneity, Hydrogeology Journal 13 (2005) 161–183.

[9] L. Guo, Z. Ma, L. Zhang, Comparison of bandwidth selection in application of geographically weighted regression: a case study, Canadian Journal of Forest Research 38 (2008) 2526–2534.

[10] D. J. Toal, N. W. Bressloff, A. J. Keane, Kriging hyperparameter tuning strategies, AIAA journal 46 (2008) 1240–1252.

[11] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 4765–4774.

[12] P. Harris, C. Brunsdon, A. S. Fotheringham, Links, comparisons and extensions of the geographically weighted regression model when used as a spatial predictor, Stochastic environmental Research and Risk assessment 25 (2011) 123–138.

[13] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32.

[14] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[15] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, Lightgbm: A highly efficient gradient boosting decision tree, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Pro-

cessing Systems, volume 30, Curran Associates, Inc., 2017.
[16] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, A. Gulin, Catboost: unbiased boosting with categorical features, Advances in neural information processing systems 31 (2018).
[17] K. Patroumpas, D. Skoutas, G. Mandilaras, G. Giannopoulos, S. Athanasiou, Exposing points of interest as linked geospatial data, in: Proceedings of the 16th International Symposium on Spatial and Temporal Databases, 2019, pp. 21–30.
[18] B. S. Murphy, Pykrige: development of a kriging toolkit for python, in: AGU fall meeting abstracts, volume 2014, 2014, pp. H51K–0753.
[19] G. Erdogan Erten, M. Yavuz, C. V. Deutsch, Combination of machine learning and kriging for spatial estimation of geological attributes, Natural Resources Research 31 (2022) 191–213.
[20] Y. Guo, S. vanden Broucke, Enhancing geospatial prediction models with feature engineering from road networks: a graph-driven approach, International Journal of Geographical Information Science 38 (2024) 1611–1632.
[21] W. Zhang, H. Liu, L. Zha, H. Zhu, J. Liu, D. Dou, H. Xiong, Mugrep: A multi-task hierarchical graph representation learning framework for real estate appraisal, in: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, 2021, pp. 3937–3947.
[22] M. Geerts, J. De Weerdt, et al., Graph neural networks for house price prediction: do or don't?, International Journal of Data Science and Analytics (2024) 1–31.
[23] M. T. Ribeiro, S. Singh, C. Guestrin, "why should I trust you?": Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, 2016, pp. 1135–1144.
[24] J. Yu, Z. Zhang, M. Sarwat, Geosparkviz: a scalable geospatial data visualization framework in the apache spark ecosystem, in: Proceedings of the 30th international conference on scientific and statistical database management, 2018, pp. 1–12.
[25] J. Hynek, J. Kachlík, V. Rusnák, Geovisto: A toolkit for generic geospatial data visualization., in: VISIGRAPP (3: IVAPP), 2021, pp. 101–111.
[26] C. K. Williams, C. E. Rasmussen, Gaussian processes for machine learning, volume 2, MIT press Cambridge, MA, 2006.
[27] P. Gramacki, K. Lesniara, K. Raczycki, S. Wozniak, M. Przymus, P. Szymanski, SRAI: towards standardization of geospatial AI, in: Proceedings of the 6th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery, GeoAI 2023, Hamburg, Germany, 13 November 2023, ACM, 2023, pp. 43–52.
[28] S. Woźniak, P. Szymański, Hex2vec: Context-aware embedding h3 hexagons with openstreetmap tags, in: Proceedings of the 4th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery, 2021, pp. 61–71.