Improving grade estimation using machine learning: A comparative study of ordinary kriging against machine learning algorithms

by A. Akpabio[1], R.C.A. Minnitt[1]

Affiliation:
[1]School of Mining Engineering, University of the Witwatersrand, South Africa

Correspondence to:
R.C.A. Minnitt

Email:
Richard.Minnitt@wits.ac.za

Dates:
Received: 20 Feb. 2025
Revised: 25 Sept. 2025
Accepted: 19 Dec. 2025
Published: March 2026

How to cite:
Akpabio, A., Minnitt, R.C.A. 2026. Improving grade estimation using machine learning: A comparative study of ordinary kriging against machine learning algorithms. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 126, no. 3, pp. 143–156

DOI ID:
https://doi.org/10.17159/2411-9717/3664/2026

ORCiD:
A. Akpabio
https://orcid.org/0009-0004-1667-8614
R.C.A. Minnitt
https://orcid.org/0000-0002-0267-8152

Abstract

This study presents a rigorous comparison between ordinary kriging and commonly used machine learning algorithms, those being, linear regression, support vector regression, decision trees, random forests (RF), and k-nearest neighbours for spatial interpolation of platinum grade estimates in a complex ore body within the Bushveld Igneous Complex. Using only X and Y coordinates as predictors, both ordinary kriging and machine learning models were evaluated at point and block supports under traditional and spatial block cross validation frameworks. While naive validation results suggested superior performance for k-nearest neighbour and random forest (R^2 = 0.92 and 0.86, respectively), these were revealed to be overly optimistic under spatial dependence. Spatial block cross validation results demonstrated substantial declines in model performance, with R^2 often falling below zero, particularly for decision trees and k-nearest neighbour, indicating strong overfitting and limited generalisability. Ordinary kriging exhibited more stable, albeit modest, performance under spatial validation, reflecting its strength in geostatistical interpolation when contextual geological variables are unavailable. The study underscores the critical importance of spatially aware validation in resource estimation and highlights that machine learning models constrained to spatial coordinates behave as interpolators rather than true learners of geological variability. Recommendations are provided for future work incorporating geological information to enhance predictive robustness.

Keywords

machine learning, ordinary kriging, grade estimation, geostatistics, platinum group elements, spatial block cross validation

Introduction

The life of a mining project is typically assessed during feasibility studies well before production (Sinclair, Blackwell, 2002). Because mining is capital-intensive, reliable estimation of grades and tonnages are essential to technical and economic decision-making, alongside mine planning, scheduling, and processing capacity considerations (Deutsch, Rossi, 2014). Within this context, ordinary kriging (OK) remains a cornerstone. It exploits spatial autocorrelation via variogram modelling and provides linear-unbiased point predictions with quantifiable estimation variance. Comparative studies, however, show mixed outcomes. For example, in platinum group element (PGE) deposits, OK and simple kriging can perform differently across grade ranges (Mpanza, 2015), while for strongly skewed gold, nonlinear disjunctive kriging outperformed OK. For moderately skewed copper, performances were similar and implementation choices were decisive (Hekmatnejad et al., 2017). These findings imply that an algorithm's performance is data and implementation dependent, especially in geologically complex settings such as the Bushveld Igneous Complex, where sharp grade fluctuations, variable seam thickness, and structural disruptions complicate modelling.

At the same time, there is growing interest in the use of machine learning (ML) for grade and resource estimation (Dumakor-Dupey, Arya, 2021). With claims that kriging requires substantial expert input, particularly in variogram modelling and parameter selection, ML models are increasingly being adopted because: (1) they can capture complex features, (2) they do not rely on assumptions about the spatial distribution of grades, and (3) they require comparatively less expert knowledge (Erten et al., 2021).

A recent systematic review and comparative study by Mahboob et al. (2022), documents the increasing use of ML models, often reporting competitive or superior accuracy to inverse-distance/kriging baselines in specific cases. The study highlights inconsistencies across deposits, data regimes, and evaluation protocols. The first key lesson is that, how models are validated, strongly conditions the conclusions drawn about their relative performance.

This introduces the adoption of spatially-aware cross validation, so that the cross validated error(s) better reflects prediction at new locations rather than near-replicates of the training data (Roberts et al., 2017). Framing the study around spatial validation clarifies both the methodological choices made and how the results were interpreted.

A second lesson from Mahboob, et al. (2022), is the importance of feature engineering. Many successful ML applications integrate geological, structural, geochemical, geophysical, and spatial predictor variables to capture nonlinear relationships beyond pure spatial interpolation. In contrast, using only spatial coordinates (X, Y, [Z]) effectively constrains ML models to act as interpolators, which is conceptually closer to kriging than to 'learning' geological features.

In this study, predictor variables for PGE grade were restricted to only X and Y coordinates because they are consistently available across the domain and representative of common early-stage datasets, allowing ML models to be framed explicitly as nonparametric spatial interpolators. This design choice is deliberate but limiting.

Against this backdrop, the study objective was to provide a transparent, side-by-side evaluation of OK and several widely used ML algorithms: Linear regression (LR), support vector machines (SVR), decision trees (DT), random forests (RF), and k-nearest neighbour (kNN), at point and block supports.

Data for the study

The study was conducted on a platinum deposit, Project X, located in the eastern limb of the Bushveld Igneous Complex. Confidentiality of the site location is preserved by rotation and translation of the 570 borehole data of which the relative positions are shown in Figure 1. The estimation domain in the X direction ranges from 56837 m to 64759 m, in the Y direction ranges from −185780 m to −178136 m, in the Z direction ranges from 289 m to 1102 m. The average borehole spacing is 324 m.

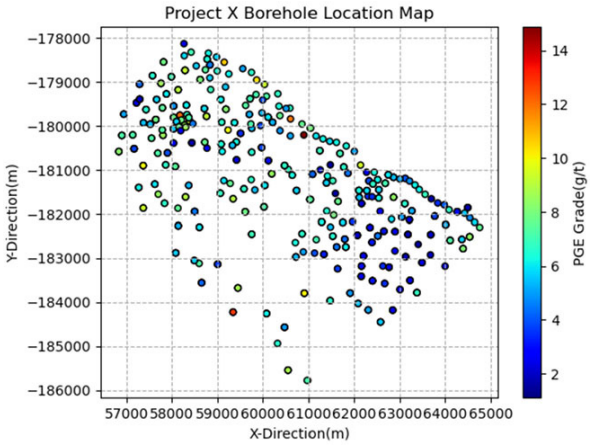

Figure 1—Borehole location map of the Project X deposit

Traditional vs. spatial cross validation

Cross validation (CV) is a fundamental technique for model evaluation and selection, traditionally relying on random splits of data into training and validation sets. Traditional cross validation like leave-one-out or random *k*-fold partitions the data into training and test sets under the implicit assumption that samples are independent and identically distributed (Roberts et al., 2017). When this assumption holds, it yields approximately unbiased estimates of generalisation error and supports fair model comparison. In spatial datasets, however, observations that are geographically close (as can be seen in Figure 1) tend to be autocorrelated. Randomly assigning nearby points to different folds creates train-test leakage where information from the test point can inadvertently influence the model through its spatially close neighbours in the training set, leading to optimistic error estimates and potentially biased model selection (Roberts et al., 2017).

Spatial cross-validation modifies the resampling scheme to respect spatial dependence by withholding contiguous blocks of space or buffered neighbourhoods rather than individual points. Common designs include spatially blocked *k*-fold, leave-location-out, and buffered CV that excludes training samples within a chosen distance of each test location (Wang et al., 2023). Block or buffer sizes are typically set in relation to the spatial autocorrelation range inferred from variograms or correlograms, and block orientation or shape can be aligned with known anisotropy so that folds are separated along the dominant direction of continuity (Stock, 2025).

These designs reduce leakage and produce error estimates that better reflect the intended use case, which is, predicting at new locations rather than at points adjacent to the training data. A practical trade-off is that overly large or misaligned blocks can induce unintended extrapolation between folds and thus conservative errors; consequently, fold geometry should be tuned to the prediction task and the measured correlation scale.

Research methodology

OK, ML implementation and validation

Both the ML and OK workflows were implemented using open-source Python libraries. Ordinary kriging (OK) estimation employed the GeostatsPy package, which integrates GSLIB (Deutsch, Journel, 1998) functions into Python. Developed by Pyrcz et al. (2021), GeostatsPy facilitates spatial modelling workflows. Two modules were used: GeostatsPy.geostats, which reimplements GSLIB functions for variogram analysis, data transformation, and spatial estimation; and geostatspy. GSLIB, which provides simple wrappers for visualisation and numerical tools. Using GeostatsPy, OK was applied to mineral grades based on the spatial coordinates of sample points for both point and block estimates.

Machine learning (ML) models were implemented with Scikit-learn, an open-source library offering efficient tools for data preprocessing, analysis, and modelling. Built on NumPy, SciPy, and Matplotlib, Scikit-learn provides a consistent interface for the algorithms used in this study. Predictions were made by splitting the dataset into training and testing subsets and applying Scikit-learn's prediction framework.

Block estimation followed the microblocking approach of Nwalia et al. (2024), which predicts values at fine scales (microblocks) and aggregates them into larger blocks (macroblocks). Model performance for both OK and ML was evaluated using R^2, RMSE, and MAE, each offering complementary insights into prediction error.

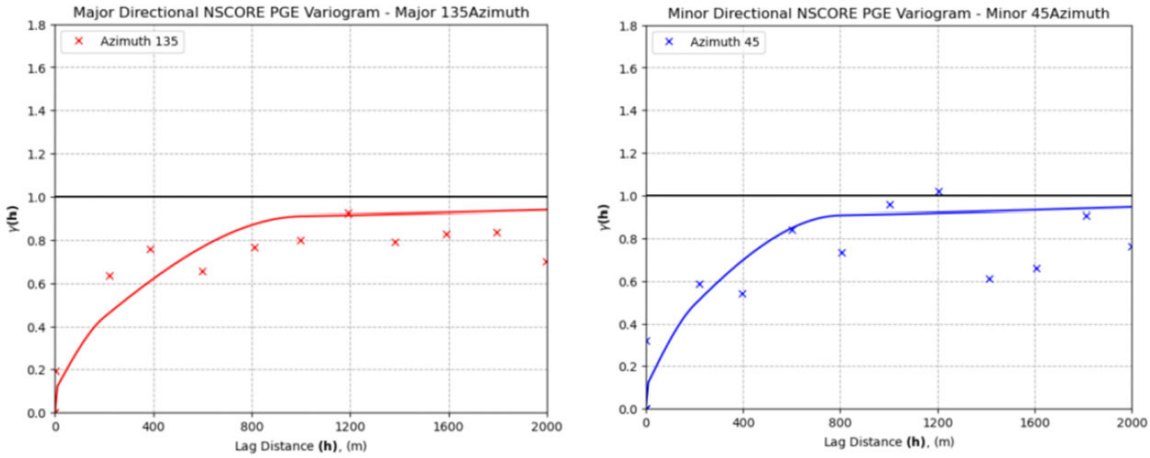

Figure 2—Model variogram for the 135° (major) and 45° (minor) direction of anisotropy for the Project X data

Each metric offers a different perspective on the error between the predicted values by a model and the actual values observed in the data. The formulations and explanations for each are as follows:

R^2 measures the proportion of the variance in the dependent variable that is predictable from the independent variable(s).

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad [1]$$

SS_{res} is the sum of squares of residuals (the differences between observed and predicted values), and SS_{tot} is the total sum of squares (the differences between observed values and the mean of observed values). R-squared values range from 0 to 1, where a higher value indicates a better fit to the data.

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

RMSE is the square root of MSE. It measures the standard deviation of the residuals or predictions errors. By taking the square root of MSE, RMSE converts the error metric back to the same unit as the target variable, making interpretation easier. Like MSE, RMSE penalises larger errors more heavily.

$$MAE = \frac{1}{n}\sum_{i}^{n}|y_i - \hat{y}_i| \quad [3]$$

MAE calculates the average of the absolute errors between the predicted values and the actual values. Unlike RMSE, MAE gives a linear weight to all errors, which means it is less influenced by the occasional, but very large errors. It is a straightforward measure of average error magnitude. Scatterplots, histograms of residuals, and swath plots provided useful visual summaries of results obtained from the analyses.

Variogram modelling

Experimental variograms were computed after cell-declustering and normal-score transformation. Guided by the average data spacing, lags of 200 m (tolerance 100 m), 25 lags, bandwidth 100 m, and azimuth tolerance 22.5° were adopted. Directional variograms at 0°, 45°, 90°, and 135° indicated geometric anisotropy with a preferred continuity along a NW-SE direction (135°). Thereafter, a nested spherical model with a nugget of 0.10 and three structures of which the partial sills were 0.15, 0.65, and 0.10 (unit total sill on the transformed scale). Major/minor ranges for the three structures were approximately 200/200 m, 1000/800 m, and 5000/4500 m, respectively, all oriented at 135° (as shown in Figures 2 and 3).

```
nug = 0.1; nst = 3
it1 = 1; cc1 = 0.15; azi1 = 135; hmaj1 = 200; hmin1 = 200
it2 = 1; cc2 = 0.65; azi2 = 135; hmaj2 = 1000; hmin2 = 800
it3 = 1; cc3 = 0.1;azi3 = 135; hmaj3 = 5000;hmin3 =4500
```

Figure 3—Model variogram parameters

```
search_radii = [3400,4000]
ndmin_values = [3,5,8,12]
ndmax_values = [10,15,20]
nxdis_values = nydis_values = [2,4]
```

Figure 4—Block model variogram parameters

The characterisation of kriging as a minimum variance estimator is valid only when the neighbourhood is appropriately defined (Vann et al., 2003), underscoring the importance of quantitative kriging neighbourhood analysis (QKNA). QKNA identifies the optimal combination of estimation parameters, namely: block size, number of informing samples, search range, and discretisation points, which minimise conditional bias (Chanderman et al., 2017). Performance is assessed using kriging efficiency (KE), which measures how well estimates reproduce local grades, and the slope of regression (SLOR), which indicates smoothing effects between estimated and true grades. A grid search exhaustively evaluates all possible combinations within a predefined set of parameter values (Fgure 4).

A custom function namely, grid_search_kriging (see Appendix A1), was developed to optimise ordinary kriging (OK) parameters based on the geostatspy.geostats.kb2d function, which was modified to handle a three-structure variogram model. For South African deposits, measured resources are typically defined using drillholes spaced 250 m – 300 m apart (Zientek et al., 2014). Accordingly, a 250 m × 250 m block model was adopted, with its origin aligned to the minimum X and Y boundary coordinates.

Due to computational constraints, the grid search parameters (Figure 4) were set up to balance efficiency and coverage ranging from conservative configurations to broader ones. A limited number of discretisation points was used to maintain computational efficiency while attempting to adequately capture spatial variability.

ML and hyperparameters

Sci-kit learn has the *GridSearchCV* function that, like any grid search strategy, exhaustively considers all parameter combinations

Table 1

ML Hyperparameter search space

Algorithm	Hyperparameter (pipeline key)	Grid values
SVR (RBF, γ='scale')	C	0.1, 1, 10
	epsilon	0.1, 0.4, 0.8, 1.0
Random forest	n_estimators	50, 100, 150, 200
	max_depth	3, 10, 20
	min_samples_split	2, 3, 4, 5, 6, 7, 8, 9, 10
	criterion	squared_error
kNN	knn__n_neighbours	3, 5, 7, 9
	knn__weights	uniform, distance
	knn__metric	euclidean, manhattan
Decision tree	max_depth	3, 30, 50
	min_samples_split	2, 5, 10
	criterion	squared_error
	min_samples_leaf	1, 5, 10, 15, 20
	max_leaf_nodes	10, 50, 100, 200

provided in a grid. Table 1 depicts a conservative hyperparameter search that was adopted to reduce computational demand. Given that the predictors comprised only spatial coordinates, increasing model hyperparameters would not alleviate the under-variation revealed by spatial cross-validation unless additional spatial or geological predictors were introduced.

Minimal preprocessing was applied to preserve the spatial characteristics of the data. The features consisted solely of X and Y coordinates, with no engineered or derived variables. Missing value imputation (median-based) was implemented as an optional step but remained disabled, as there were no missing values. Feature scaling using standardisation was applied only to kNN and LR models, while DT and RF were trained on raw coordinates. This approach ensured consistency and interpretability in comparing ML estimators with ordinary kriging. Linear regression is included as a baseline and has no tuneable hyperparameters in this setup.

Validation framework

Traditional CV (leave-one-out and random k CV)

Leave-one-out cross-validation (LOOCV) was first applied to evaluate the accuracy of the OK model by sequentially withholding each observation as a validation sample and averaging the resulting prediction errors. For the ML models, k-fold cross-validation was used, where the dataset was partitioned into 5 folds; each fold was validated once, while the model was trained on the remaining k–1 folds.

Both LOOCV and random k-fold CV assume independent and identically distributed data, an assumption that is violated with spatial datasets. This spatial dependence can cause information leakage and overly optimistic accuracy estimates making it reasonable to make use of traditional CV for point estimation. Results traditionally CV were retained as naïve baselines to illustrate the optimism introduced by ignoring spatial dependence. Following the guidance of Roberts et al. (2017), Stock (2025), and Wang et al. (2023), this study implemented spatial block cross-validation (SBCV) to ensure independence between training and testing through spatial partitioning.

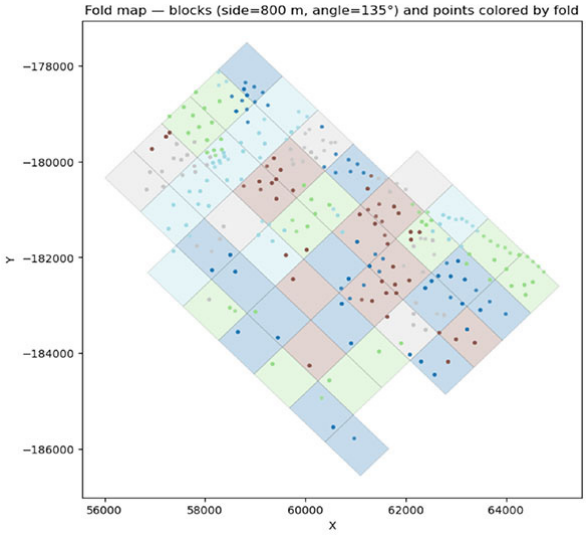

Figure 5—Fold distribution and block geometry

Spatial block cross validation

In this study, k-fold spatial block cross-validation (SBCV) was performed with k = 5, where each fold comprised multiple non-contiguous spatial blocks assigned in a round-robin manner. Guided by the nested spherical variogram model, anisotropic blocks of 1000 m x 800 m (representing 65% of the partial sill) were oriented along the direction of major continuity (Figure 5).

An 800 m buffer was applied around each test fold and any training sample whose nearest test sample lay within this distance was removed. This buffer radius is approximately equal to the minor direction range of the dominant structure, beyond which the semi variance is close to the sill and the remaining correlation is weak. This is depicted in Figure 6 where training and test samples were well balanced, with approximately 160 – 220 training points and 100 test points per fold. Each test cluster (orange) is located within one or more spatial blocks, while the surrounding training samples (blue) form a halo separated by the 800 m exclusion buffer (indicated by the circles), ensuring spatial independence between training and validation data.

For block estimation, prior to cross-validation, an evaluation grid of centroids was generated to define the spatial prediction support for both kriging and machine learning models. As shown in Figure 7, the blue points represent grid centroids used as prediction locations, while the orange points show the distribution of observed samples. This grid ensured that predictions for both OK and ML were made over an equivalent spatial framework, allowing for direct comparison of model performance at consistent support.

The evaluation grid was then partitioned into spatial folds using the same anisotropic block geometry and orientation. In Figure 8, each colour denotes a distinct fold containing multiple dispersed blocks, ensuring that every portion of the domain was represented in both training and validation phases across the five iterations of cross-validation.

This SBCV framework was implemented identically for both OK and ML to guarantee methodological parity. In OK, predictions were made at block centroids corresponding to the evaluation grid, while ML algorithms were trained and validated using the same spatial folds and exclusion buffers. This approach was to ensure that both methods were evaluated under the same spatial independence assumptions, i.e., block geometry, and variogram guided anisotropy, providing a fair and directly comparable validation of their predictive performance.

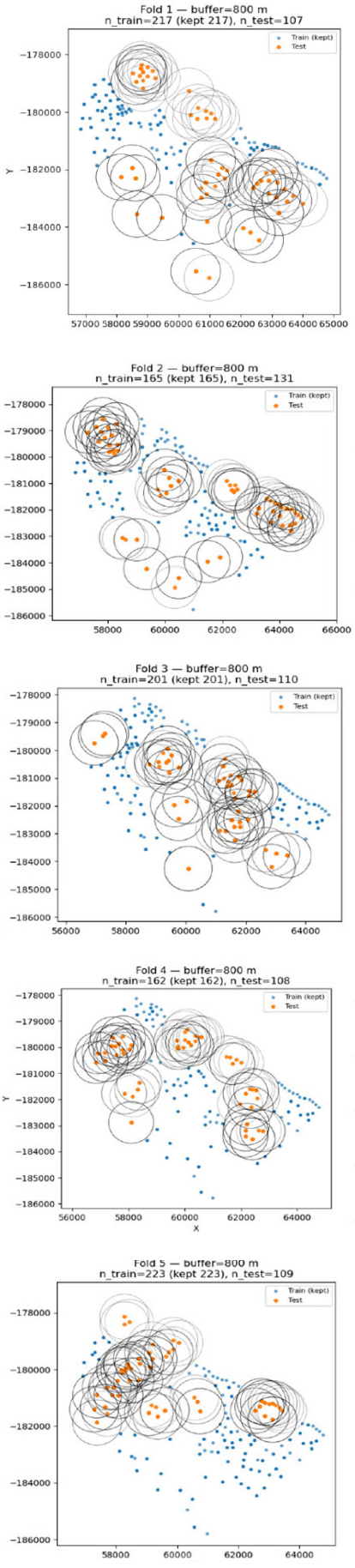

Figure 6—Fold-by-fold sample allocation

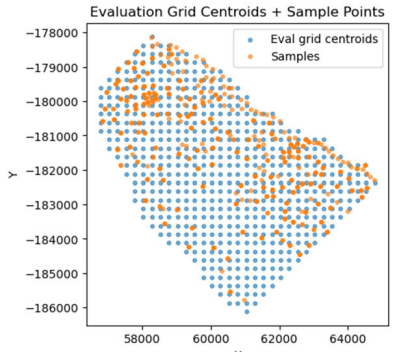

Figure 7—Initial setup of the block estimation SBCV framework

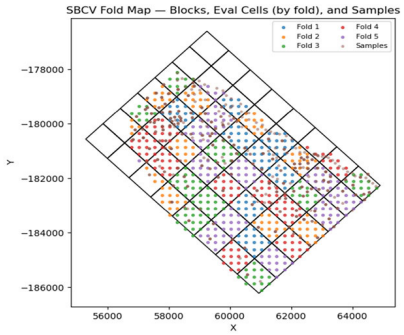

Figure 8—Fold partitioning stage of the block SBCV process

Results and discussion

Naïve validation

Table 2 summarises the validation metrics for all models under the naive CV scheme. All models predict a mean grade very close to the true mean (5.76 g/t). The kNN model stands out with the lowest errors (RMSE = 0.644, MAE = 0.211) and highest R^2 (0.92). The RF model also performs very well (RMSE = 0.846, R^2 = 0.86). The OK and DT models achieve intermediate performance (OK: RMSE = 0.996, R^2 = 0.76; DT: RMSE = 1.180, R^2 = 0.72). In contrast, SVR and LR have much larger errors and lower R^2 (SVR: RMSE = 1.709, R^2 = 0.40; LR: RMSE = 2.061, R^2 = 0.12). These results indicate that kNN and RF provide the most accurate point estimates under naive validation, while LR and SVR perform poorly.

It is important to recognise that, because nearby samples tend to be similar, LOOCV and random k-fold splits allow spatially proximate points to appear in both training and test sets, effectively leaking spatial information. This violates the independence assumption and implies that results obtained are overly optimistic (Roberts et al., 2017). It can further be inferred that naive CV results therefore represent the best case scenario under spatial information leakage; true generalisation performance in unsampled areas is likely to be substantially worse.

Table 2

Summary results for comparative metrics of estimated grade and errors

Model	Estimated grade (g/t)	RMSE	MAE	R-squared
OK	5.69	0.996	0.748	0.76
DT	5.77	1.180	0.775	0.72
SVR	5.72	1.709	1.293	0.40
RF	5.77	0.846	0.551	0.86
LR	5.75	2.061	1.629	0.12
kNN	5.76	0.644	0.211	0.92

Influence of model characteristics on naive validation performance

Each algorithm learns spatial structure in a distinct way, and this influences the extent to which conventional validation overestimates model accuracy:

- LR assumes a global linear relationship between spatial coordinates and grade, because it cannot capture local spatial variation. Its low complexity prevents it from overfitting to local spatial clusters, so the degree of performance inflation under naïve validation is minimal.
- SVR incorporates nonlinear mapping via a radial basis function (RBF) kernel, enabling it to represent smooth spatial trends. Its moderate performance (R^2 = 0.40) arises from capturing continuous spatial patterns, yet its kernel structure allows partial memorisation of local relationships when nearby samples appear in both training and test folds. Consequently, SVR results under naïve CV are modestly optimistic.
- DT models partition the input space into discrete spatial regions based on threshold splits in X and Y. This inherently local structure leads to strong apparent accuracy (R^2 = 0.72) under naïve validation because nearby training and test samples fall within the same or adjacent partitions. However, such models generalise poorly when validation regions are spatially separated.
- RF, an ensemble of decision trees, captures complex nonlinear and local spatial dependencies by averaging across multiple decision structures. Under naïve validation, RF achieves very high apparent accuracy (R^2 = 0.86), largely because it benefits from repeated exposure to spatially correlated train–test pairs. The ensemble averaging smooths local noise but cannot eliminate spatial leakage, making its naïve performance among the most inflated.
- The kNN, which predicts grades as the average of nearby samples, shows the highest apparent accuracy under naïve validation (R^2 = 0.92). This is a direct consequence of its design; it exploits spatial autocorrelation explicitly. When test points lie within the neighbourhood of training samples, predictions are almost exact. However, kNN cannot extrapolate beyond the spatial domain of its neighbours. Its inflated accuracy under naïve validation is therefore almost entirely attributable to information leakage.
- OK models spatial dependence explicitly through the variogram and yields moderately high accuracy (R^2 = 0.76) under LOOCV. Although LOOCV introduces minor spatial dependence between test and training samples, kriging's variogram-based weighting limits overfitting compared to purely distance-based ML models.

Evidence of overfitting

The swath plots seen in Figure 9 illustrate the predicted versus observed PGE grade variation along a NW–SE direction at 200 m intervals. For ML models, particularly kNN, DT, SVR, and RF, the predicted curves reproduce the observed grade fluctuations almost exactly, indicating an overly tight fit to local spatial variations, rather than generalised trends. This behaviour reflects overfitting due to spatial information leakage under naïve validation, where spatially proximate samples appear in both training and testing subsets. The DT model exhibits abrupt fluctuations consistent with its piecewise structure, while SVR shows slightly smoother behaviour, yet still mirrors short-range oscillations. A well-generalising model would smooth local noise while preserving broad spatial trends, thus, the high apparent accuracy of these models primarily reflects their exploitation of spatial autocorrelation, rather than true predictive capability.

Spatial cross validation

Point estimation

All six algorithms exhibited strong sensitivity to the spatial structure imposed by the 800 m buffered SBCV, with model performance varying markedly between folds. A consistent pattern in R^2 in Fold 4 emerged across all ML methods and OK.

For DT, error metrics shown in Figure 10 varied substantially across folds, with RMSE ranging from 1.72 to 2.60 and MAE from 1.46 to 2.10, indicating strong sensitivity to spatial location. R^2 values were negative in four of the five folds (ranging from -0.89 to 0.32). Plots in Figure 11 show that predicted values were tightly clustered around the mid-range of the grade distribution, leading to systematic underestimation of high grades and overestimation of low grades. Residuals were widely dispersed, with large errors occurring throughout the domain.

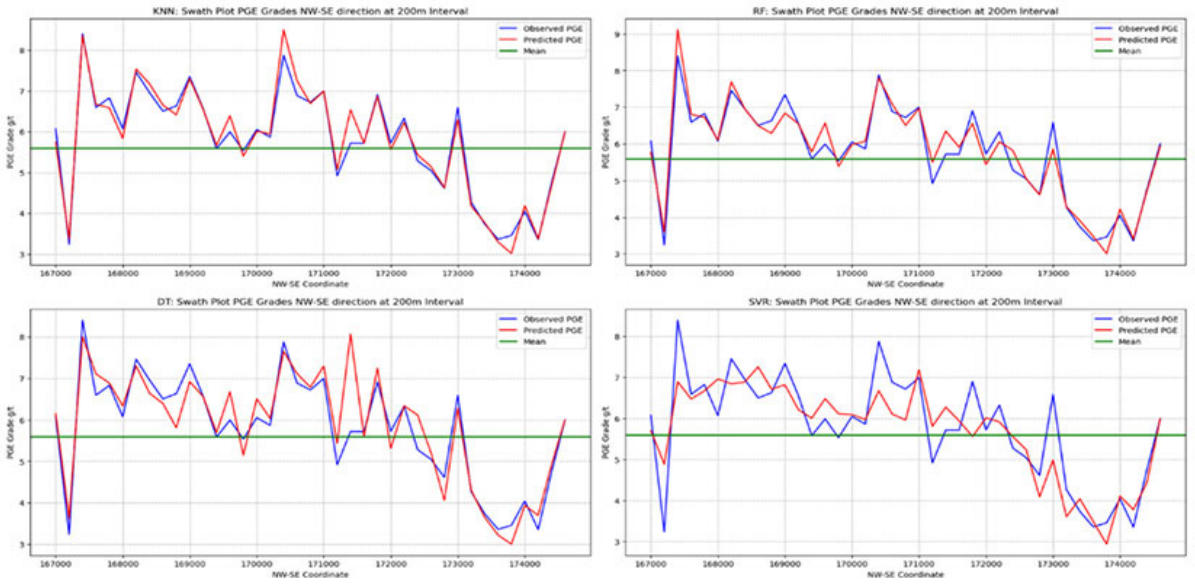

Figure 9—Naïve swath plot at 200 m intervals, kNN (top left), RF (top right), DT (bottom left), SVR (bottom right)

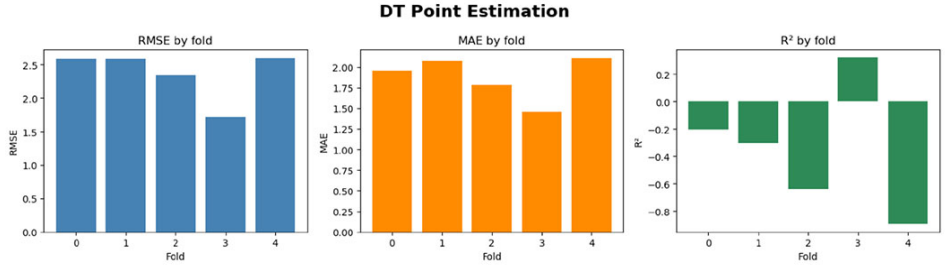

Figure 10—DT fold-wise evaluation metrics

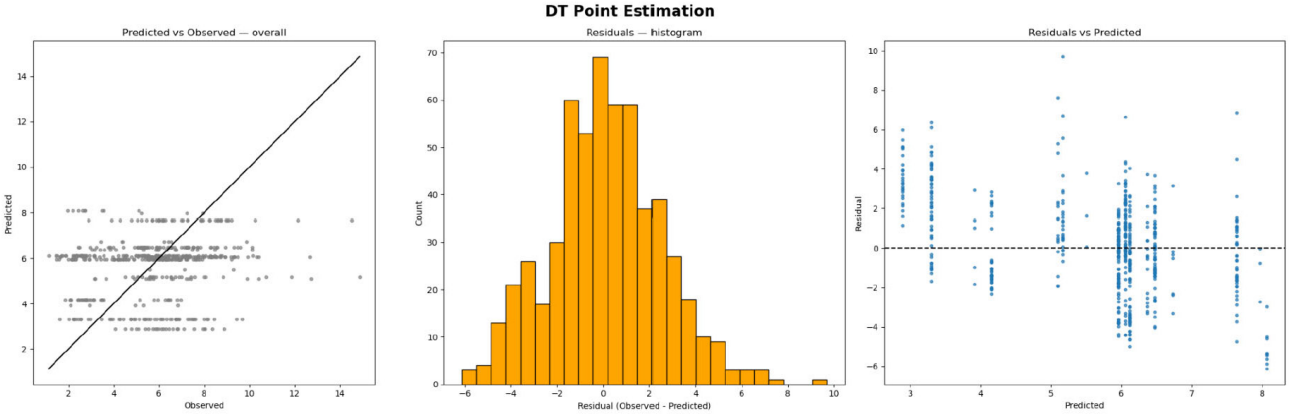

Figure 11—DT results, predicted vs. observed scatter plot (left), residual histogram (middle), residual vs. predicted scatter (right)

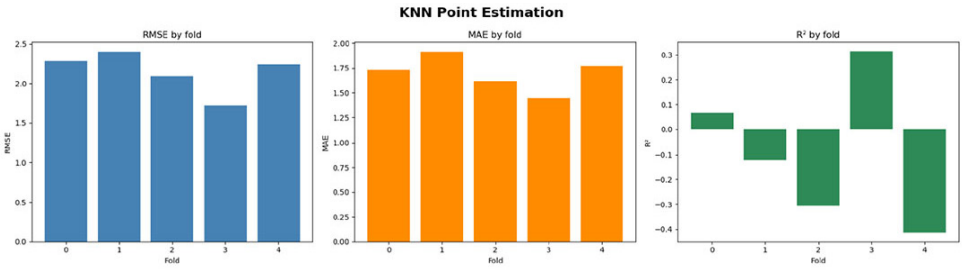

Figure 12: kNN fold-wise evaluation metrics

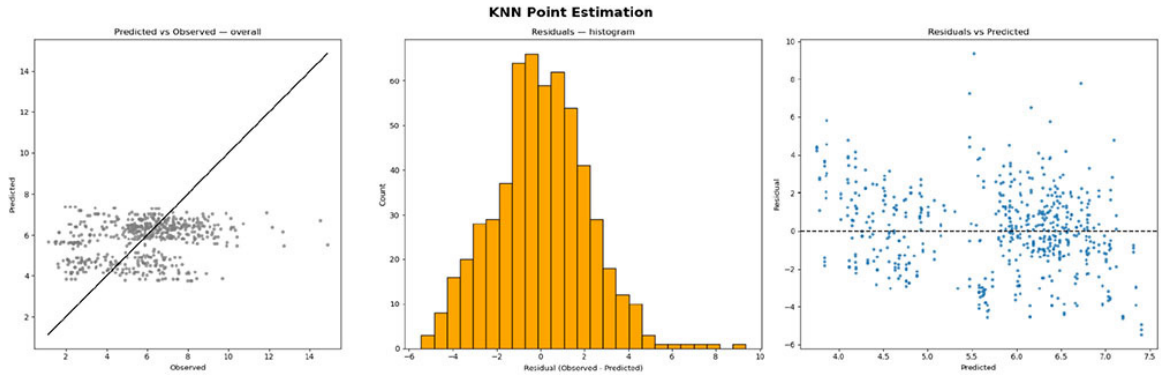

Figure 13—kNN results, predicted vs. observed scatter plot (left), residual histogram (middle), residual vs. predicted scatter (right)

The kNN model demonstrated moderate predictive performance across the spatial cross-validation folds, with error magnitudes comparable to those observed for DT, but showing slightly more stability. Similar to DT, kNN produced mixed R^2 scores across folds, with fold 4 attaining a positive value (0.30), but the majority falling below zero, resulting in an overall average of -0.26 (Figure 12). In Figure 13, the residual histogram displays a roughly symmetric but slightly right-skewed distribution, and the residuals-versus-predicted plot shows larger positive residuals occurring at higher predicted grades.

RF achieved the most favourable performance of the tree-based models, though similar limitations persisted under spatial cross-validation. In Figure 14, it can be observed that fold-level RMSE ranged from 1.74 to 2.39, and MAE varied between 1.47

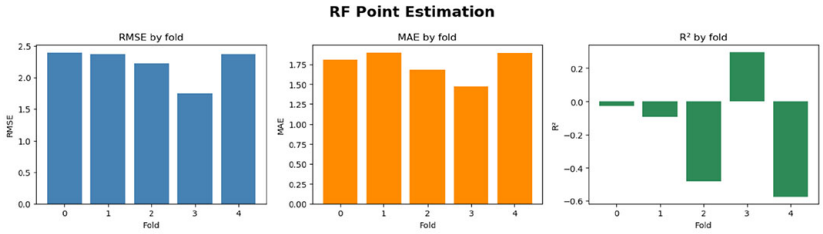

Figure 14—RF fold-wise evaluation metrics

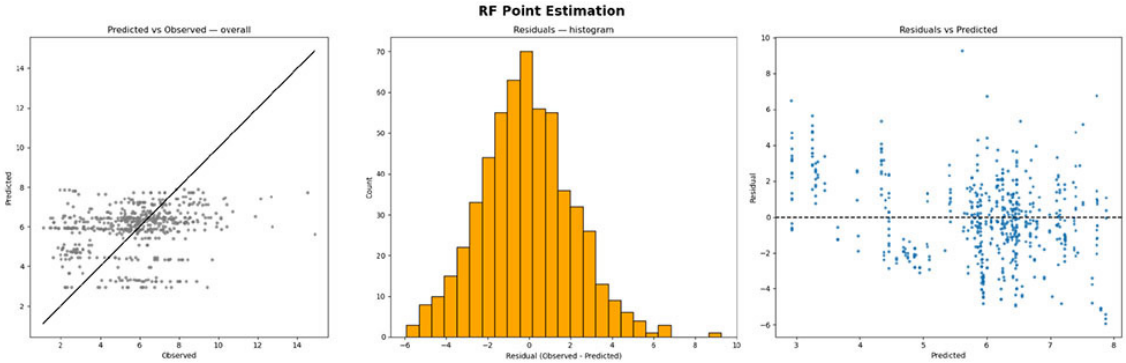

Figure 15—RF results, predicted vs. observed scatter plot (left), residual histogram (middle), residual vs. predicted scatter(right)

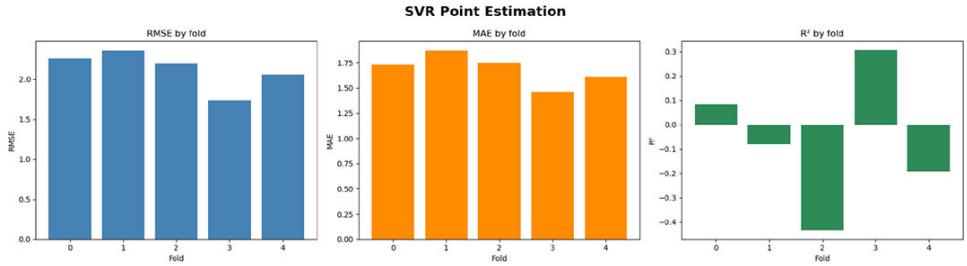

Figure 16—SVR fold-wise evaluation metrics

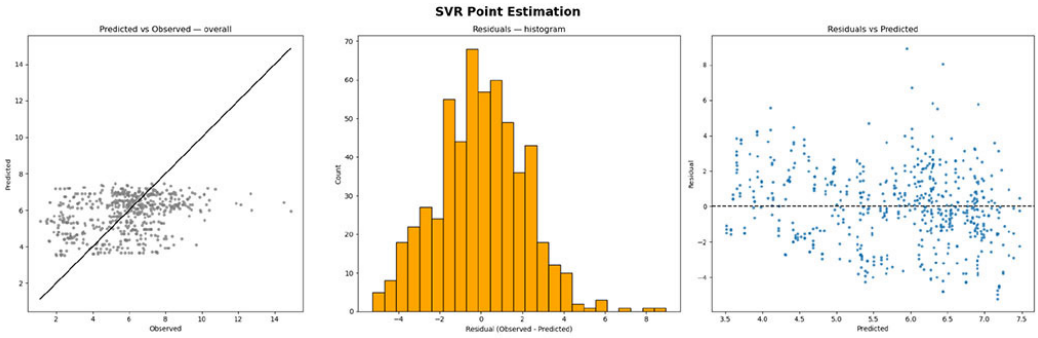

Figure 17—SVR results, predicted vs. observed scatter plot (left), residual histogram (middle), residual vs. predicted scatter(right)

and 1.90, with Fold 4 again yielding the strongest performance. R^2 values spanned 0.57 to 0.30, reflecting partial, but inconsistent, ability to reproduce spatial variability. Scatterplots and residual diagnostics in Figure 15 further reveal the model's tendency to regress predictions toward 5 g/t – 7 g/t, capturing general trends but like residuals, which remained broadly symmetric and persistently underestimated higher grade values.

SVR produced results broadly consistent with the other ML algorithms. As with the other models, in Figure 16, Fold 4 consistently yielded the strongest performance, achieving both the lowest errors and the only clearly positive R^2. Overall, looking at Figure 17, SVR performs moderately well for central grade values but struggles to reproduce variability at the higher end of the distribution, in line with patterns observed across the other machine-learning approaches.

Model performance of LR across the spatial folds remained relatively weak and broadly consistent with the trends observed for the other ML algorithms (Figures 18 and 19). The R^2 values fluctuated considerably, from slightly positive in Fold 4 to moderately negative in the remaining folds. Overall, LR produced

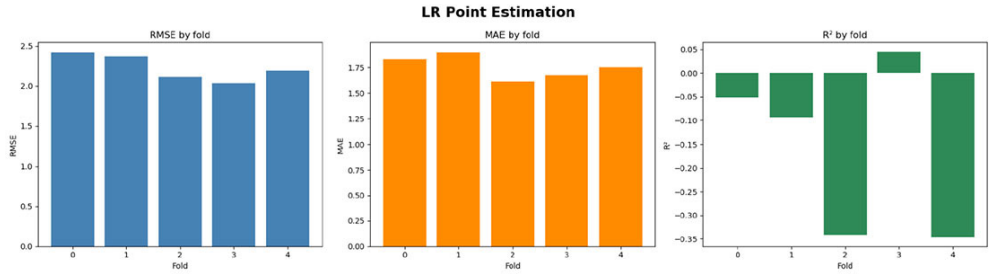

Figure 18—LR fold-wise evaluation metrics

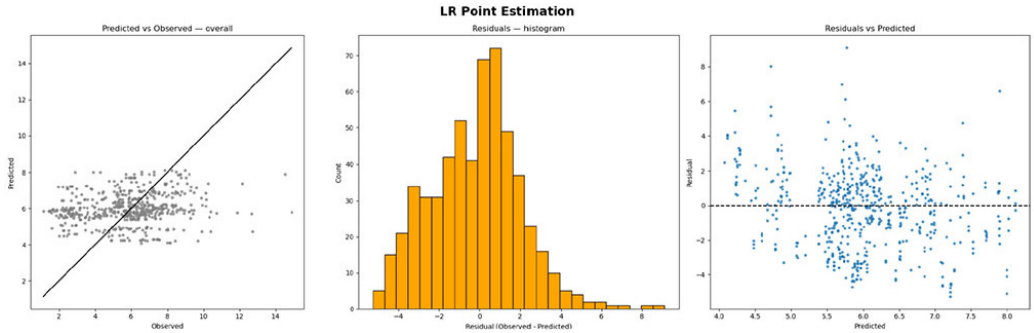

Figure 19—LR results, predicted vs. observed scatter plot (left), residual histogram (middle), residual vs. predicted scatter (right)

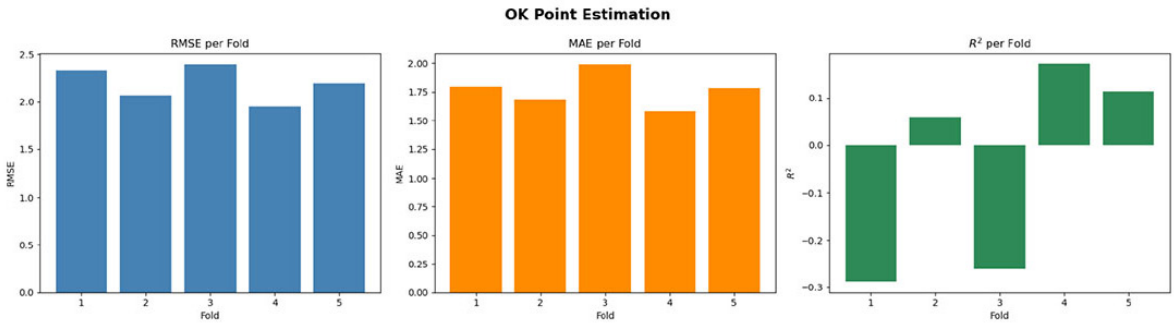

Figure 20—OK fold-wise evaluation metrics

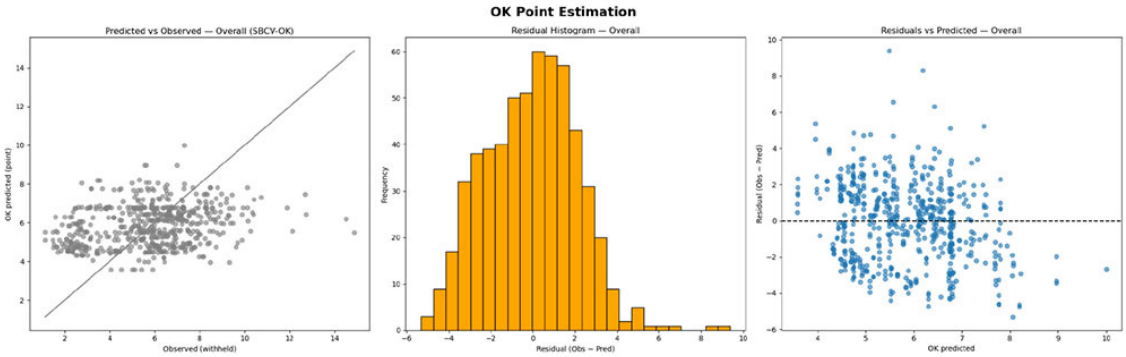

Figure 21—OK results, predicted vs. observed scatter plot (left), residual histogram (middle), residual vs. predicted scatter (right)

stable, but comparatively low predictive accuracy, reinforcing the broader pattern across all algorithms that spatially informed CV imposes a more stringent and realistic assessment of model generalisation.

OK exhibited moderate variability in predictive performance across the spatial folds. RMSE values ranged from 1.95 g/t to 2.40g/t. The R^2 values fluctuated around zero, spanning from -0.29 to 0.17, with Fold 4 having the best performance (Figure 20). Residuals followed an approximately symmetric distribution, centred slightly above zero, with no strong systematic patterns when plotted against predicted values (Figure 21). OK achieved performance comparable to the ML algorithms in magnitude of error, but with similarly constrained ability to reproduce the full variability of the assay grades under strict spatial validation.

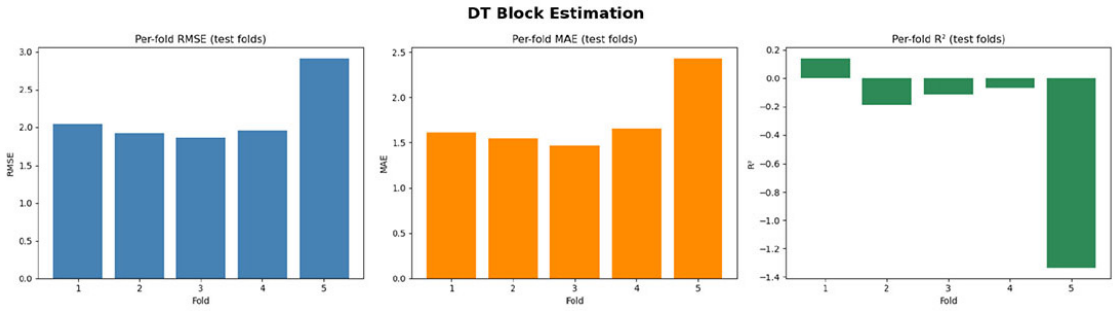

Figure 22—DT fold-wise evaluation metrics

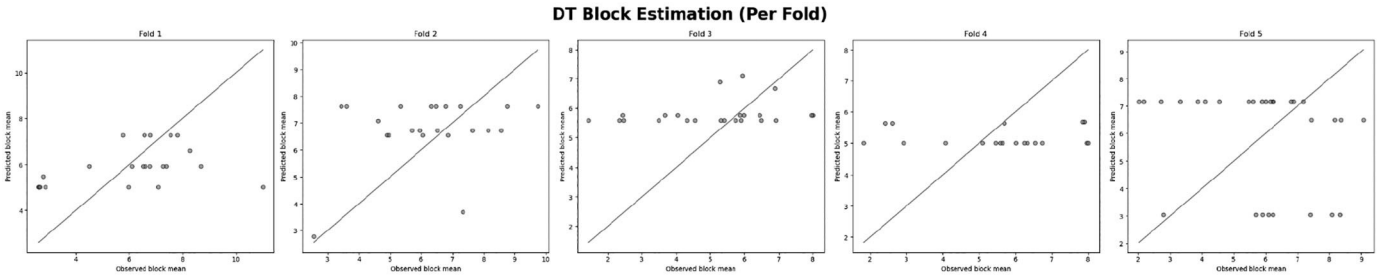

Figure 23—DT ,-wise predicted vs. observed scatter plot

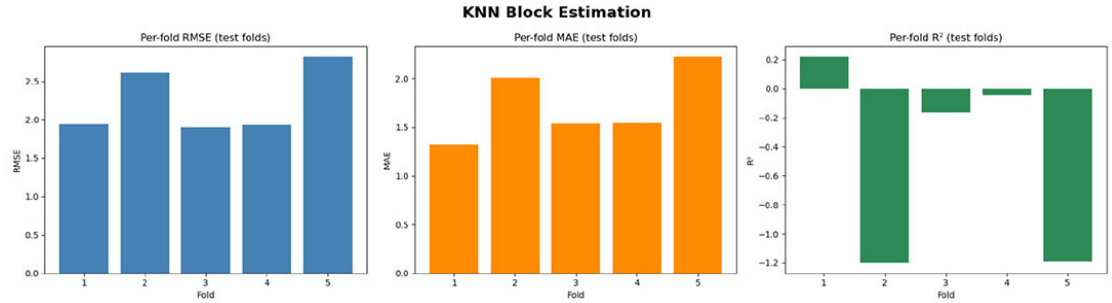

Figure 24—kNN fold-wise evaluation metrics

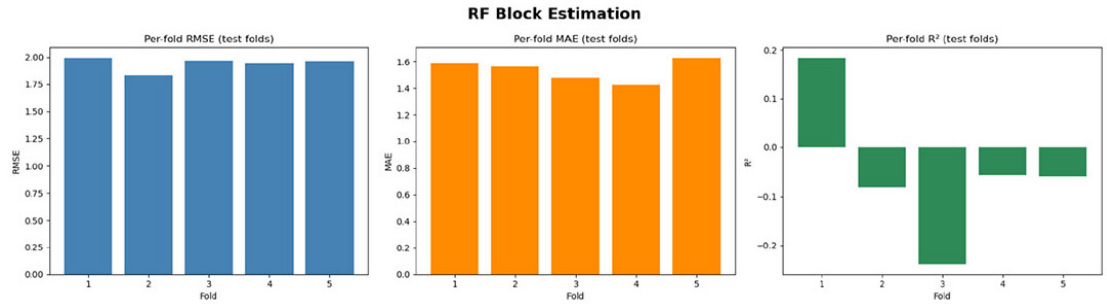

Figure 25–RF fold-wise evaluation metrics

Block estimation

Both the OK and ML block estimation workflows operate on the same spatial support, ensuring that their predictions are directly comparable. Each method uses an evaluation grid composed of 250 m × 250 m blocks, with the ML grid explicitly aligned to the OK block-model origin and geometry. Although OK applies micro-discretisation (2 × 4) to integrate the block estimate, while ML predicts directly at the block centroid, both approaches ultimately generate estimates for an equivalent block volume, making the results support-consistent.

With the DT model across the five SBCV folds, RMSE ranged from about 1.87 to 2.04, with a marked degradation in Fold 5 (Figure 22). Fold-level R^2 values varied from weakly positive in Fold 1 (0.14), to strongly negative in Fold 5 (-1.34). The per-fold predicted-versus-observed block plots in Figure 23 shows that DT is tending to compress block means toward a narrow grade band and failing to reproduce extremes, particularly in the poorest fold.

The kNN model performed similarly to, but slightly worse than, the DT model under SBCV. Fold-level RMSE values ranged from about 1.90–1.95 in the better folds (1, 3, and 4) to 2.61 and 2.82 in Folds 2 and 5, with corresponding MAE values increasing from 1.3–1.6 to just over 2.0 (Figure 22). Only Fold 1 achieved a modest positive R^2 (0.22); Folds 2 and 5 produced strongly negative R^2 (around –1.2), indicating severe overfitting to local training patterns that do not generalise to withheld blocks (Figure 24).

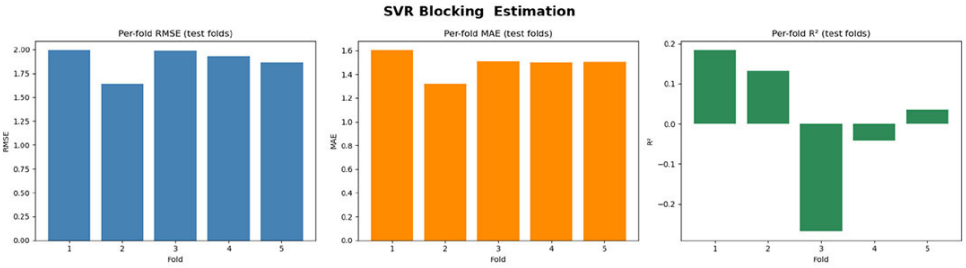

Figure 26—SVR fold-wise evaluation metrics

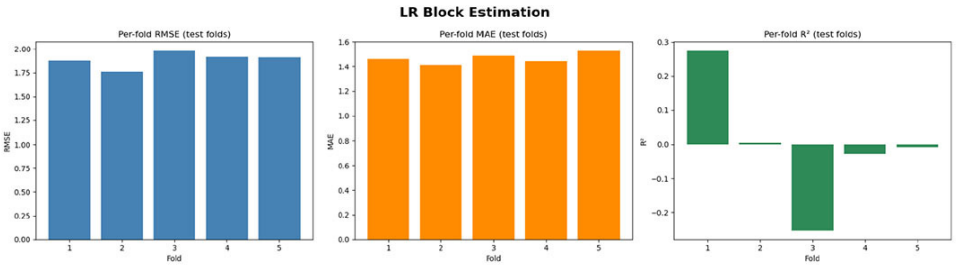

Figure 27—LR fold-wise evaluation metrics

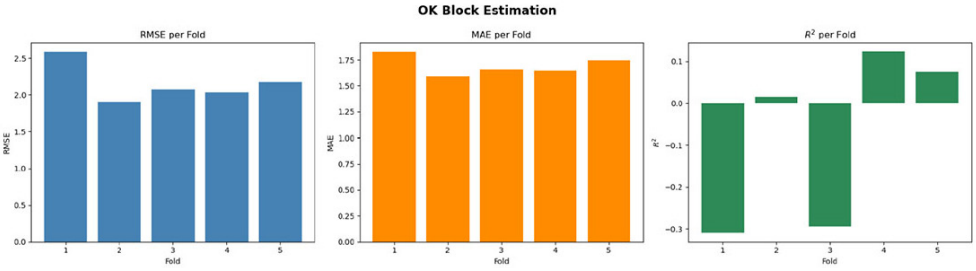

Figure 28—OK fold-wise evaluation metrics

The RF model predictive strength is modest relative to DT and kNN. Test-fold RMSE values ranged narrowly between 1.83 and 1.99, and MAE values between 1.42 and 1.63, indicating stable, but not strongly accurate block predictions (Figure 25). The model exhibited mixed directional accuracy, with Fold 1 yielding positive R^2 (Fold 1 0.18), while the remaining folds were negative. This behaviour suggests that while RF occasionally captured meaningful block-scale trends, it struggled to generalise consistently under the spatially constrained training conditions imposed by SBCV.

With SVR, RMSE values across the five folds ranging from approximately 1.64 to 1.99 and MAE values were stable, indicating consistent absolute deviations in predicted block means (Figure 26). The fold-wise R^2 values fluctuated around zero, reflecting limited predictive strength.

RMSE across folds in LR, have values ranging from approximately 1.76 to 1.98, while MAE values remained within a narrower band of 1.41 to 1.53 (Figure 27), comparable in magnitude to the other ML models. R^2 values fluctuate around zero, with onefold 1 showing a slightly positive coefficient and others showing marginally negative scores, excluding Fold 3.

OK RMSE and MAE values remain relatively stable across folds, as shown in Fgure 28. Unlike the ML models with R^2, Fold 1 emerges as the poorest performer together with Fold 3, whilst the best performance was in Fold 4 followed by Fold 5 and then 2.

Comparative analysis

The contrast of the evaluation results in Table 3 across traditional CV and SBCV mimics real-world applications where predictions are made in new unmapped locations. When assessed under SBCV, all models exhibited a dramatic drop in performance. The R^2 of SVR declined from 0.92 under naïve CV to -0.06 under point SBCV, and marginally improved to 0.01 under block SBCV. RF showed a similar trend, with R^2 dropping from 0.86 to -0.17 (point) and -0.05 (block). The kNN and DT experienced the most severe deterioration, reflecting their strong reliance on localised spatial structure. Despite OK`s results also following this trend, its variogram-based approach offers a more spatially coherent prediction framework, though limited in this study by the spatial context held out from the optimisation parameters. The improvement under block SBCV may be due to the fact that, when residuals average out over spatial blocks, the error structure of LR looks less severe.

The failure of LR across all settings highlights a central issue, that is, the exclusive use of spatial coordinates as predictors. All machine learning models functioned as interpolators, attempting to reconstruct the spatial signal without any geochemical or geological features to inform contextual variation. While models such as SVR and RF can learn smooth functions from data, their generalisation capacity is restricted in the absence of explanatory variables that

Table 3

Naïve and spatial block CV evaluation results

Model	Naïve RMSE	Naïve MAE	Naïve R^2	Point SBCV RMSE	Point SBCV MAE	Point SBCV R^2	Block SBCV RMSE	Block SBCV MAE	Block SBCV R^2
OK	0.996	0.748	0.76	2.190	1.767	0.01	2.157	1.689	-0.04
DT	1.180	0.775	0.72	2.367	1.872	-0.34	2.134	1.740	-0.31
kNN	1.709	1.293	0.40	2.303	1.796	-0.26	2.242	1.727	-0.48
RF	0.846	0.551	0.86	2.220	1.747	-0.17	1.938	1.536	-0.05
LR	2.061	1.629	0.12	2.226	1.754	-0.16	1.849	1.514	0.04
SVR	0.644	0.211	0.92	2.120	1.685	-0.06	1.884	1.488	0.01

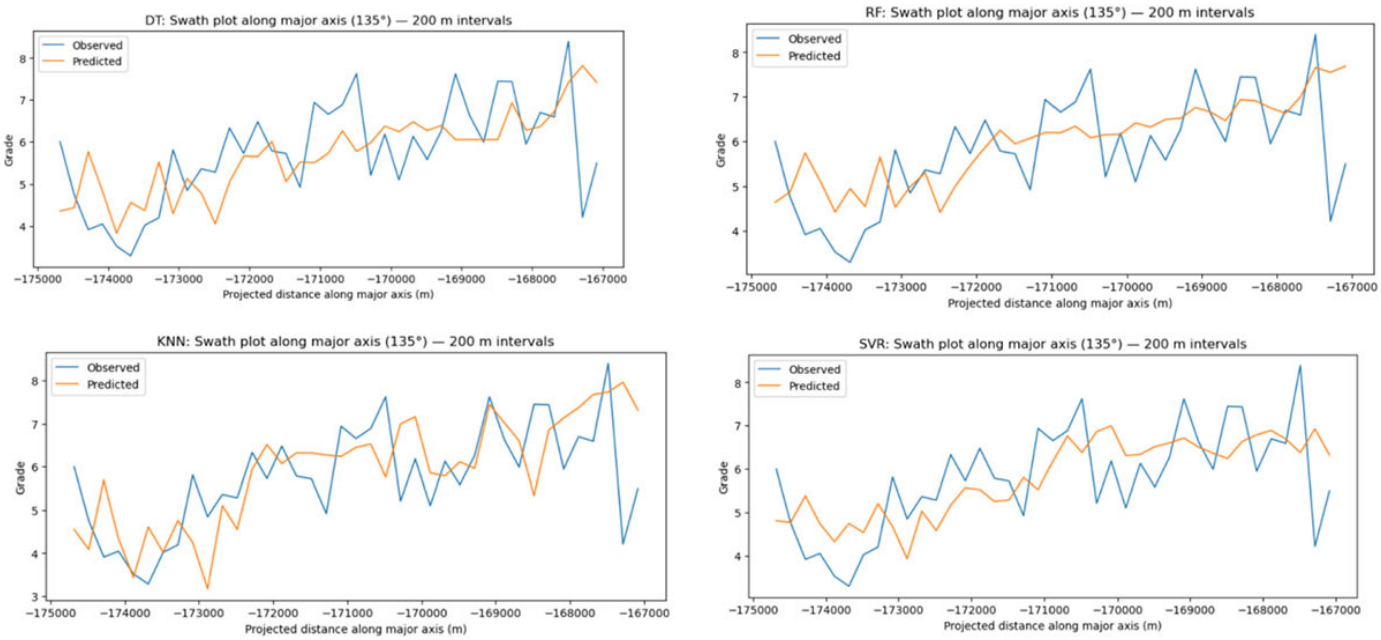

Figure 29—SBCV informed swath plot at 200m intervals, DT (top left), RF (top right), kNN (bottom left), SVR (bottom right)

capture underlying spatial processes. Kriging's performance suggests greater resilience under strict spatial validation, highlighting its suitability for interpolation tasks when only location data is available.

In Figure 9, it could be observed that under naïve CV, the predicted PGE grades closely followed observed trends, with minimal lag or attenuation. Under SBCV in Figure 29, predictions across all models became markedly smoother and less responsive to high-frequency spatial variation. The amplitude of predicted values was dampened, with extreme values either under or overestimated depending on model bias. In this instance, DT and kNN, in particular, produced overly simplified or noisy estimates that failed to reflect underlying grade trends. RF, while more stable, showed a reduced dynamic range and consistent underestimation of peaks. SVR predictions were the smoothest but frequently misaligned with observed values, especially in transitional zones. These patterns confirm that SBCV exposes models' inability to reproduce sharp spatial features when deprived of nearby data, again emphasising the limitation of relying solely on coordinate-based interpolation.

For point estimation, Fold 4 consistently yields the best R^2 across all machine learning (ML) models. This suggests that Fold 4 likely corresponds to a spatial region with smoother grade variability, more consistent sampling density, or higher alignment with the training data's feature distribution. For block estimation, the performance peak shifts predominantly to Fold 1 across most ML models. This inversion suggests that Fold 1 encompasses spatial blocks that are more homogeneously sampled or better represented in the training data distribution when cross-validation is spatially blocked. On the other hand, OK continued to perform best on Fold 4, again due to its ability to explicitly incorporate spatial autocorrelation and its reliance on a variogram model. This is unlike ML models that inferred spatial structure only implicitly through coordinate inputs. The folds demonstrate the effectiveness of SBCV in mimicking real-world prediction tasks for grade estimation. However, they also highlight a challenge when training and test sets differ significantly in grade distribution. In this study, model performance declined.

Conclusion

This study presented a comparative evaluation of OK and a suite of ML algorithms for grade predictions using a platinum deposit dataset from the Bushveld Igneous Complex. By employing both

traditional and SBCV frameworks, the analysis highlighted the substantial impact of spatial autocorrelation on model performance and the tendency of traditional CV to produce over-optimistic estimates due to spatial information leakage.

Under naïve CV, ML models such as kNN and RF showed high apparent accuracy, but their performance declined sharply under SBCV, revealing limited generalisation when predicting at spatially independent locations. OK, while not immune to degradation under SBCV, demonstrated relatively stable and interpretable behaviour, owing to its explicit treatment of spatial structure through variogram modelling. Among the ML methods, SVR and RF exhibited the most resilience under spatial validation, though still failing to match OK's coherence at block support.

A critical limitation of this study lies in the use of only spatial coordinates as predictor variables. This design choice constrained all ML algorithms to operate as nonparametric interpolators, limiting their capacity to learn and extrapolate from geological context. As such, the conclusions drawn here relate specifically to interpolation strategies in early-stage or data-constrained scenarios. Additionally, model tuning and validation were carried out under a fixed variogram and buffer structure.

With this study as a baseline, future research will explore the integration of richer (complex) geological, geochemical, and geophysical predictor variables to enhance model expressiveness and predictive accuracy. The study further highlighted that claims of superiority between kriging or ML based models should never be stated in absolute terms. The incorporation of domain-specific knowledge through feature engineering, hybrid kriging-ML frameworks, and advanced deep learning architectures, holds more promise for improving estimation in complex ore bodies. Furthermore, sensitivity analysis on buffer sizes, fold geometries, and variogram configurations in SBCV would aid in developing robust and generalisable validation frameworks. Finally, extending comparisons to conditional simulation or uncertainty-aware models could offer deeper insights into risk and resource classification in geostatistical modelling.

References

Chanderman, L., Dohm, C. & Minnitt, R., 2017. 3D geological modelling and resource estimation for a gold deposit in Mali. *The Journal of the Southern African Institute of Mining and Metallurgy*, vol. 117, pp. 189–197.

Deutsch, C., Journel, A. 1998. *GSLIB: Geostatistical Software Library and User`s Guide*. 2nd ed. s.l.:Oxford University Press.

Deutsch, C., Rossi, M. 2014. *Mineral Resource Estimation*. Berlin: Springer.

Dumakor-Dupey, N.K., Arya, S., 2021. Machine learning—A review of applications in mineral resource estimation. *Energies*, vol. 14, no. 14, pp. 1–29.

Erten, E. G., Yavuz, M., Deutsch, C.V. 2021. Grade estimation by a machine learning model. *Applied Earth Science*, vol. 130, no. 1, pp. 57–66.

Hekmatnejad, A., Emery, X., Alipour-Shahsavari, M. 2017. Comparing linear and non-linear kriging for grade predictions and ore/waste classification in mineral deposit. *International Journal of Mining, Reclamation and Environment*, vol. 33, no. 4, pp. 247–264.

Mahboob, M., Celik, T., Genc, B. 2022. Review of machine learning-based Mineral. *The Journal of the Southern African Institute of Mining and Metallurgy*, vol. 122, no. 11, pp. 655–664.

Mpanza, M. 2015. Wits wiredspace. [Online] Available at: https://wiredspace.wits.ac.za/items/951f024c-f64f-46ee-9be6-027443550b76/full [Accessed 12 November 2022].

Nwalia, G., Zhang, S., Bourdeau, J., Frimmel, H. 2024. Spatial Interpolation Using Machine Learning: From Patterns. *Natural Resources Research*, vol. 33, pp. 129–161.

Pyrcz, M. et al. 2021. PyPI. [Online] Available at: https://pypi.org/project/geostatspy/ [Accessed 3 December 2023].

Roberts, D. et al. 2017. Cross-validation strategies for data with temporal, spatial, hierarchical or phylogenetic structure. *Ecography*, vol. 40, no. 8, pp. 913–929.

Sinclair, A., Blackwell, G. 2002. *Applied Mineral Inventory Estimation*. s.l.:Cambridge University Press.

Stock, A. 2025. Choosing blocks for spatial cross-validation: lessons from a marine remote sensing case study. *Frontiers in Remote Sensing*, vol. 6, pp. 1–14

Vann, J., Jackson, S., Bertoli, O. 2003. Quantitative Kriging Neighbourhood Analysis for Mining Geologist - A Description of the Method With Worked Case Examples. Bendigo. *Australasian Institute of Mining and Metallurgy*, pp. 1–10.

Wang, Y., Khodadadzadeh, M., Zurita-Milla, R. 2023. Spatial+: A new cross-validation method to evaluate geospatial machine. *International Journal of Applied Earth Observation and Geoinformation*, vol. 121. ◆

Appendix A 1

```
def grid_search_kriging(df, xcol, ycol, vcol, tmin, tmax, nx, xmn, xsiz, ny, ymn, ysiz, ktype, skmean, vario):
    # Define search parameters
    search_radii = [3400,4000]
    ndmin_values = [3,5,8,12]
    ndmax_values = [10,15,20]
    nxdis_values = nydis_values = [2,4]

    # Initialize best performance and parameters
    best_performance = -float('inf')
    best_params = {}
    best_kriging_result = None
    best_weights_map = None
    best_cbb = None
    best_s = None
    best_kriging_variance = None

# Iteration count and total iterations
total_iterations = len(search_radii) * len(ndmin_values) * len(ndmax_values) * len(nxdis_values) * len(nydis_values)
iteration_count = 0

# Grid search loop
for radius in search_radii:
    for ndmin in ndmin_values:
        print(f"Current ndmin: {ndmin}")
        for ndmax in ndmax_values:
            print(f"Current ndmax: {ndmax}")
            for nxdis in nxdis_values:
                for nydis in nydis_values:
                    iteration_count += 1

                    # Skip invalid configurations
                    if ndmin > ndmax:
                        continue

# Assuming kb2d_3 is modified accordingly to return cbb, s, and kriging variance
kriged_results, kriging_variance, weights_map, cbb, s,vk = geostats.kb2d_3(df, xcol, ycol, vcol, tmin, tmax, nx, xmn, xsiz, ny, ymn, ysiz, nxdis, nydis, ndmin, ndmax, radius, ktype, skmean, vario)

# Calculate SLOR and KE
slor, ke = calculate_performance_metric(kriged_results, cbb, kriging_variance, s, unest=-999.)

# Use custom scoring to evaluate the overall performance
overall_performance = custom_scoring(KE=ke, SLOR=slor)

# Update best performance and parameters based on overall_performance
if overall_performance > best_performance:
    best_performance = overall_performance
    best_params = {
        'radius': radius,
        'ndmin': ndmin,
        'ndmax': ndmax,
        'nxdis': nxdis,
        'nydis': nydis
    }
    best_ke = ke
    best_slor = slor
    #best_kriging_result = kriged_results
    #best_weights_map = weights_map
    #best_cbb = cbb
    #best_s = s
    #best_kriging_variance = kriging_variance

# Progress update
if iteration_count % 10 == 0 or iteration_count == total_iterations:
    print(f"Iteration {iteration_count}/{total_iterations}, Current Best Performance: {best_performance},Current Best Parameters: {best_params}, Best KE: {best_ke}, Best SLOR: {best_slor}")
```

Appendix A 1: Source code of the *grid_search_kriging* function

Appendix A 2: Project code repository

https://drive.google.com/drive/folders/1bd07hBcjGWfCchkdMnlLD5hEtmdTbGI0?usp=sharing