EIS

Exploration Information System

D 3.3: Documentation Report

Final Version of Additional Algorithms Implementation

Version 2.0

Lead Beneficiary: UTU
04 / 2024

Paavo Nevalainen[1], Jukka Heikkonen[1], Fahimeh Farahnakian[1], Maija Lespinasse[1], Dipak Nidhi[1], Iiro Seppä[1], Rami Ilo[1]

[1] *UTU*

Funded by
the European Union

Disclaimer

The content of this report reflects only the author's view. The European Commission is not responsible for any use that may be made of the information it contains.

Document information

Grant Agreement / Proposal ID	101057357
Project Title	Exploration Information System
Project Acronym	EIS
Scientific Coordinator	Vesa Nykänen (vesa.nykanen@gtk.fi) – GTK
Project starting date (duration)	1 May 2022 (36 months)
Related Work Package	WP 3
Related Task(s)	Task 3.2
Lead Organisation	UTU
Contributing Partner(s)	
Due Date	31.10.2023
Submission Date	09.11.2023, Revision: 05.04.2024
Dissemination level	PU

History

Date	Version	Submitted by	Reviewed by	Comments
17.03.2022	0.1	Andreas Knobloch		Template
08.11.2023	0.9	Paavo Nevalainen		Draft
08.11.2023	1.0	Andreas Knobloch		Revision
09.11.2023	1.0	Hafsa Munia		Submission
21.03.2024	2.0	Paavo Nevalainen		Revision
04.04.2024	2.0	Andreas Knobloch		Review
05.04.2024	2.0	Vesa Nykänen		Review
05.04.2024	2.0	Andreas Knobloch		Revision
05.04.2024	2.0	Hafsa Munia		Re-Submission

Table of Contents

List of Figures

Funded by
the European Union

Funded by
the European Union

Abbreviations and Acronyms

Acronym	Description
WP	Work Package
RF	Random Forest
CV	Cross-validation
GIS	Geographic Information System
EIS	Exploration Information System
PCA	Principal component analysis
PCP	Parallel coordinate plot
CNN	Convolutional neural networks
MPM	Mineral prospectivity mapping
MLP	Multilayer perceptron
ML	Machine learning
SCV	Stratified cross-validation
LOOCV	Leave one out cross-validation
SMOTE	Synthetic over-sampling techniques

Summary

Mineral prospectivity mapping is a vital tool for the exploration and mining industry. It enables geologists and mining companies to identify areas with the highest potential for new mineral deposits, which can guide their future exploration programs and investment decisions. This process involves collecting, compiling, and analysing various geochemical, geological, and geophysical data. Once the data is collected, different techniques, such as data pre-processing, data visualization, feature selection, and data augmentation, are used to identify patterns and relationships in data. Prospectivity modelling is then used to create a model that predicts the likelihood of finding new mineral deposits based on the collected data. The model's performance is evaluated using different cross-validation techniques.

Additional algorithms cover data visualization, unsupervised learning, feature importance estimation, oversampling and uncertainty estimates. First two are about data exploration and useful in comparing various mineral prospectivity mapping projects to each other. The last three are about reducing costs of future mineral prospectivity mapping projects, adapting many existing algorithms to mineral prospectivity mapping task, and serving the decision making after analysis, respectively. Each algorithm serves as an example of its task category. Some future prospects like including a complete spatial cross-validation functions, are also presented.

Keywords

Mineral prospectivity mapping, Machine Learning, Data visualization, Multilayer perceptron, Cross-validation

1. Executive summary

The additional algorithms developed for integration into the EIS Toolkit comprise of a set of typical general data analysis tools adopted to the mineral prospectivity mapping use case. Included are methods such as random forest, logistic regression, multilayer perceptron, convolutional neural network (CNN), uncertainty estimates, and over-sampling and sample balancing techniques.

These are intended to convey a systematic formalization, which turns spatial information sources to conventional Machine Learning (ML) tasks. This way, two goals are accomplished:

- Demonstration of the GIS integrability and validity of work cycles supported by GIS
- Ability to convert a large-scale mineral prospectivity mapping problem to an ordinary ML problem with data formats subjectable to programs and ML experts outside the GIS community.

Although some important aspects of this programme, e.g. spatial cross-validation, are not included, examples provided (in theory, in code, and with results) aid geo-information workers and geology-specialists to implement more methods in the same, in systematic manner.

The existing ML methods, visualization techniques and summarization tools are already in a rather high standard in general ML community and within the project participants, but risks and uncertainties included in having only few mineral prospectivity mapping sites (positive samples in ML genre) have not been studied completely yet. Experiments done by UTU may open a scientific co-operation pipeline, where some important and scientifically established tools like Bayesian models and spatial cross-validation become available on GIS platforms.

2. Introduction

2.1. Objective of task 3.2

The main goal of Task 3.2 is to develop efficient machine learning methods for mineral prospectivity workflow. For this purpose, we divided this task into the following sub-tasks:

- Data pre-processing: Methods for spatial feature extraction from spatial data. We tried to find self-explanatory features as well as more abstract feature spaces.

- Data visualization algorithms: Several clustering methods were tested to produce visualizations of mineral prospectivity and a limited set of auxiliary properties, which are of interest to geologists.

- Prospectivity modelling: Several methods were tested starting from simple and robust ones like random forest and ending to deep learning methods. Special interest are validation methods considering the spatial autocorrelation of data. The order of experimentation is such that it guarantees a satisfying performance within the given resources.

- Algorithms for estimating the uncertainty of prediction: Computationally simple methods e.g., k-nearest neighbours were used to estimate the uncertainty of the models. Advanced data variation approaches were used, too.

- Optimization of training data: Such methods are sought after, which determine new field sample site candidates, which improve the uncertainty or spatial generalization capability of the prospectivity models.

2.2. State of the art

Predictive modelling of mineral prospectivity mapping (MPM) using geographical information systems helps to create maps of mineral potential by incorporating geospatial data from multiple sources. The workflow of GIS-based MPM comprises the spatial correlations between geological, geochemical, and geophysical with known mineral deposits. MPM is a data-driven procedure, which highly depends on geoscientific data and statistical correlations between geospatial patterns and known annotated mineral points.

There have been many reviews of work on mineral identification for specific dataset types. Since 1980, logistic regression (Chung and Agterberg, 1980a, b) and WofE (Agterberg, 1989a, b) have been used in MPM. Bonham-Carter (1994) explained the working idea of these principles in his work. The WofE model introduced by Agterberg and Bonham-Carter (1990) into the field of mapping mineral prospectivity (Agterberg, 1989a, b; Agterberg and Bonham-Carter, 1990; Bonham-Carter, 1994), is popular among these techniques because it is simple to apply and interpret (Porwal and Carranza, 2015). WofE is a probabilistic model that uses conditional probability theory to figure out how geospatial patterns and known mineral deposits are related to each other in the spatial domain. Carranza and Hale investigated how fuzzy logic and logistic regression could be used to map gold mineralization potential in the Philippines' Baguio district (Carranza, E.J.M., and Hale, M. 2001). Their method demonstrated how fuzzy logic could increase the precision of MPM predictions by accounting for uncertainty in the input data. Based on the Dempster-Shafer theory, Evidential belief functions manage uncertainty and incorporate various evidence from geological, geophysical, and geochemical data in mineral prospectivity mapping (Carranza, 2008).

Machine learning is getting a lot of attention in mineral prospectivity mapping because it can handle large and complex datasets, improve the accuracy of predictions, and automate the process of finding out latest information (Carranza, 2011). Since 1990, simulating the human brain's learning process by adjusting weights between interconnected neurons, ANN has been applied in the field of MPM (Cracknell & Reading, 2014). In mineral exploration, ANNs have been shown to be able to model complex, nonlinear relationships between geological, geophysical, and geochemical variables (Singer & Kouda, 1996). However, ANNs have some problems, like the difficulty of estimating the optimal network parameter, the risk of overfitting, and the fact that the learned models are hard to understand (Cracknell & Reading, 2014). SVMs, a class of ML techniques, have been used for MPM due to their ability to manage high-dimensional data and their robustness towards overfitting (Zuo et al., 2017). The main objective of SVM is to find the optimal decision boundary (hyperplane) that can separate the positive (mineralized) and negative (non-mineralized) samples with the maximum margin (Granek, J. 2016). However, with large datasets, the kernel function and hyperparameters selected may impact the model's performance (Zuo et al., 2017). Random forest is a type of ensemble learning that can process high-dimensional data, resist overfitting, and give variable importance measures (Rodriguez-Galiano et al., 2015). RF combines numerous decision trees to create predictions, each of which was built using a different random subset of the training data and variables (Breiman, 2001). Another ensemble method, Gradient Boosting Machine (GBM), has demonstrated promise in MPM by successively integrating many weak learners (often decision trees), with each new learner rectifying the mistakes of the prior one (Oliveira et al., 2019). With better performance than other ML techniques, GBMs have been successfully used to map out the gold reserves in Brazil.

Deep Learning is a state of art technique that has emerged as a promising approach in MPM due to its capability to model complex relationships between input features and target variables (Chen et al., 2020). CNNs have gained a lot of attention in MPM because they can handle spatial data like remote sensing and geophysical images well and learn hierarchical features on their own. The authors showed that CNNs are better than traditional machine learning methods at handling multi-scale spatial data and making accurate predictions. Autoencoders are a type of unsupervised DL technique used in MPM to find features and reduce the number of dimensions (Luo, 2020). Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), a Recurrent Neural Network (RNN) type, has shown potential for MPM due to its ability to model ordinal dependencies in time-series data (Zhao, 2020) (Yin, 2022).

2.3 Novel ML methods in mineral prospectivity mapping

All the implemented methods listed in chapter 3 of this deliverable are established in other fields of general data analysis and machine learning. Some of them might be novel in respect to mineral prospectivity mapping, but actual new algorithms did not make to this delivery. What has been unique in this work, is the collection and re-developed under one template and goal – to be implemented into the EIS Toolkit.

Since this deliverable represents the results of the initial – beta release – development, we expect the following algorithms to be developed, validated and scientifically published and to be available by April 2024 with the final release within the EIS Toolkit:

- Extension of synthetic minority oversampling (SMOTE), (Chawla et.al. 2002) which extends the collinearity assumption to affine subspace spanned by a subset of positives.

- Contrastive learning applied to mineral prospectivity mapping (Jiang et.al. 2021). This will be an experiment starting on May-June 2024.

2.3. Mineral prospectivity mapping workflow and provided ML functions

The following table is provides an overview of typical stages in a mineral mineral prospectivity mapping process versus the potential support they can get from various ML functions implemented in EIS in the frame of Task 3.2 by UTU.

The table is a subjective, temporary view of the additional ML algorithms team and can change later, when EIS is being applied routinely in daily work. Also any new ML and/or GUI functionality may change behaviour of users and become a focal point linking tasks and functions differently.

We have roughly approximated the geological aspect to the following tasks:

- desk study (maintaining general awareness of various potential projects),
- proxy selection (evaluating new data channels and their potential to substitute old ones),
- feature mapping (somewhat vague stage for geologically uninitiated person, but it quite likely involves EIS),
- result evaluation (a stage where one reasonably needs everything the ML data visualization genre can offer), and
- resource estimation (quite likely both addressing similarities between known sites and potential sites).

The listed ML functions in the columns of the table are the same as accounted in chapter 3.

Funded by
the European Union

Table 1: Overview of developed additional algorithms and their fields of application.

Tool	Principal Component Analysis (PCA)	Parallel coordinates	Scatterplot	Self-Organizing Maps (SOM)	RF	Logistic Regression	Artificial Neural Network (MLP)	Convolutional Neural Network (CNN)	Feature Importance	SMOTE	Auto Encoder	Uncertainty Measure	Cross Validation (SCV)
Desk study			X	X								X	
Proxy selection	X	X	X	X					X				X
Feature mapping					X	X	X	X			X		X
Result evaluation	X	X	X	X	X				X	X	X	X	X
Resource estimation							X	X			X	X	X

3. Software documentation

3.1. Data pre-processing

The main task of pre-processing was to convert the multiple geophysical attribute GeoTiff raster layers into a csv table, where one row corresponds to the location of one pixel in the datasets and columns correspond to values of one geophysical attribute. This was only done for the visualizations and analysis that required it, otherwise the GeoTiffs would be used. This conversion should be eliminated from the final version of the software, and everything should use just the GeoTiffs directly. Some additional columns are added, such as the coordinates of each pixel and whether a known ore deposit is located there. The raster sampling can be done easily with conventional GIS software, such as QGIS, or using rasterio and geopandas python packages. In this case, GIS software was used. In QGIS, the workflow is like the following:

"Raster pixels to points"—the tool can be used to convert pixels of one raster layer to a points layer, with each pixel having the value of the raster layer in its attribute table. Other layers values can be added to this layer as new columns using the "sample raster values" tool. The "Rasterize (vector to raster)" tool can be used to create a raster with a value for known ore deposit locations and zero elsewhere. This can then be sampled the same way as all the other raster layers. Finally, the north and east coordinates (in the projection of the data) can be added as fields in the field calculator with the $x and $y commands, respectively. Then, the resulting vector point layer can be saved as a csv. The resulting csv format allows using any generic ML software and environment for rapid experimentation and development of new algorithms, when a need arises.

3.2. Data visualization

3.2.1 Principal component analysis

Principal component analysis (PCA) is a statistical method that uses dimensionality reduction techniques that may give the geologist or mineral prospectivity mapping expert a unified view over data or parts of it. It is possible to detect clusters from this view, which the user then can try to explain, and direct further analysis steps accordingly.

This unsupervised algorithm is often used in ML to find solid patterns while emphasizing the variation in the dataset. The fundamental concept behind PCA is to find dominant directions in the dataset that contain the highest variance. The orientation of the dominant plane is such that clustering structure is sometimes revealed (Abdi & Williams, 2010). The dominant plane is defined by so called principal components (1,2 or 3) of the data. Subsequent principal components account for a decreasing amount of capability to explain the variance in data (Jolliffe & Cadima, 2016).

The PCA is a well-known and common method with many variants, and often used with multi-source geolocational data as a preliminary tool. There are formulations such as the iterative power method, the covariance matrix eigen decomposition, or singular value decomposition (SVD). The covariance matrix of the given data is calculated, and eigenvectors (principal components) and eigenvalues are found. The eigenvalues give the variance in each principal component, while eigenvectors show the direction of the new coordinate system (Wold, Esbensen, & Geladi, 1987). After the calculation of principal components, the calculated data is projected onto the new coordinate system for visualization, classification, or regression tasks (Van Der Maaten, Postma, & Van den Herik, 2009).

Typically, the PCA is a one standard routine done to any ML problem with normalized data. The PCA view is considered in the ML genre just a visualization of its dominant hyperplane (2 or 3 dimensional). This view characterizes the data: a tight ball with initial eigenvalues similar to each other tells there are no clear explainability inherent in the data, whereas dominant initial eigenvalues indicate a linear subspace and possibly small intrinsic dimensionality. In that respect, it is customary to compare PCA's of totally different problems to acquire understanding of the problem difficulty and of the methods required.

There are many options for visualizing PCA. With geospatial raster data, one of the most effective ways is to set different principal components to red, green, and blue bands of a raster (Figure 1). This way, areas of bedrock with comparable properties are highlighted with similar colours. Sometimes it might also be useful to assign principal component 4, 5 or even 6 into one of the colour bands. However, this is only useful if these components have eigenvalues that are not much smaller than principal components 1, 2 and 3.

Figure 3 shows ten different raster layers were used as starting data, each of them containing information about radioactivity, electrical properties, or magnetic and gravitational anomalies in the area. PCA was done for the layers, and the first principal component was assigned to red band, second principal component to green band and third component to blue band. The resulting raster is very colourful since the principal components do not correlate with each other. Known ore deposits are shown with white rings. It is notable that all of them in the area are located at or close to similar hues of dark red and violet.

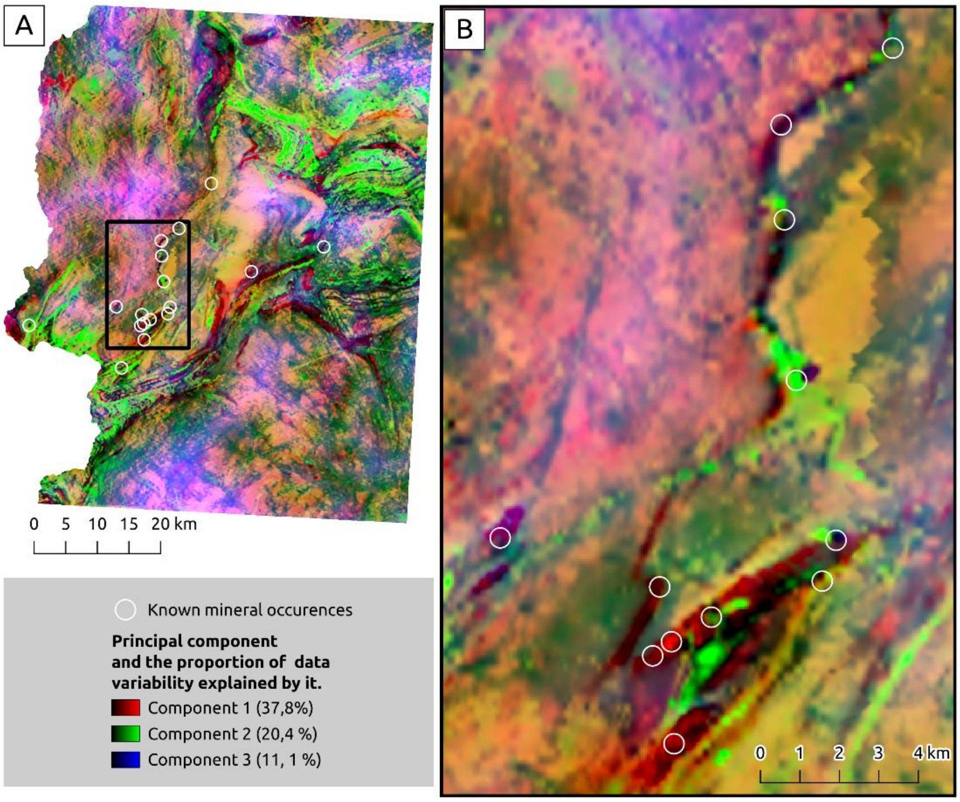
A
0 5 10 15 20 km
○ Known mineral occurences
Principal component and the proportion of data variability explained by it.
🟥 Component 1 (37,8%)
🟩 Component 2 (20,4 %)
🟦 Component 3 (11, 1 %)
B
0 1 2 3 4 km

Figure 1: Example of PCA visualization with geophysical data.

3.2.2 Parallel coordinate plot

A parallel coordinate plot (PCP) is a well-known, powerful visualization tool for analysing and exploring high-dimensional datasets in two-dimensional space. PCP is also known as parallel axes plot or parallel coordinates. PCP represents each row in the data table as a line or profile, with a point on the line representing each row's attribute. Alfred Inselberg first introduced PCP in the 1980s, and it has since become widely used in various fields of machine learning (Inselberg, A., 1985). In some cases, a cluster structure in data can be observed, especially when a certain value of a certain feature dominates a cluster. (Sansen, J., Richer, G., Jourde, T., Lalanne, F., Auber, D., & Bourqui, 2017).

PCP is particularly good in our case because it gives the analyst a compact and straightforward way to explore the data. Secondly, it gives an intuitive overall view of the data (Inselberg, A., & Dimsdale, B., 1990, Figure 2).

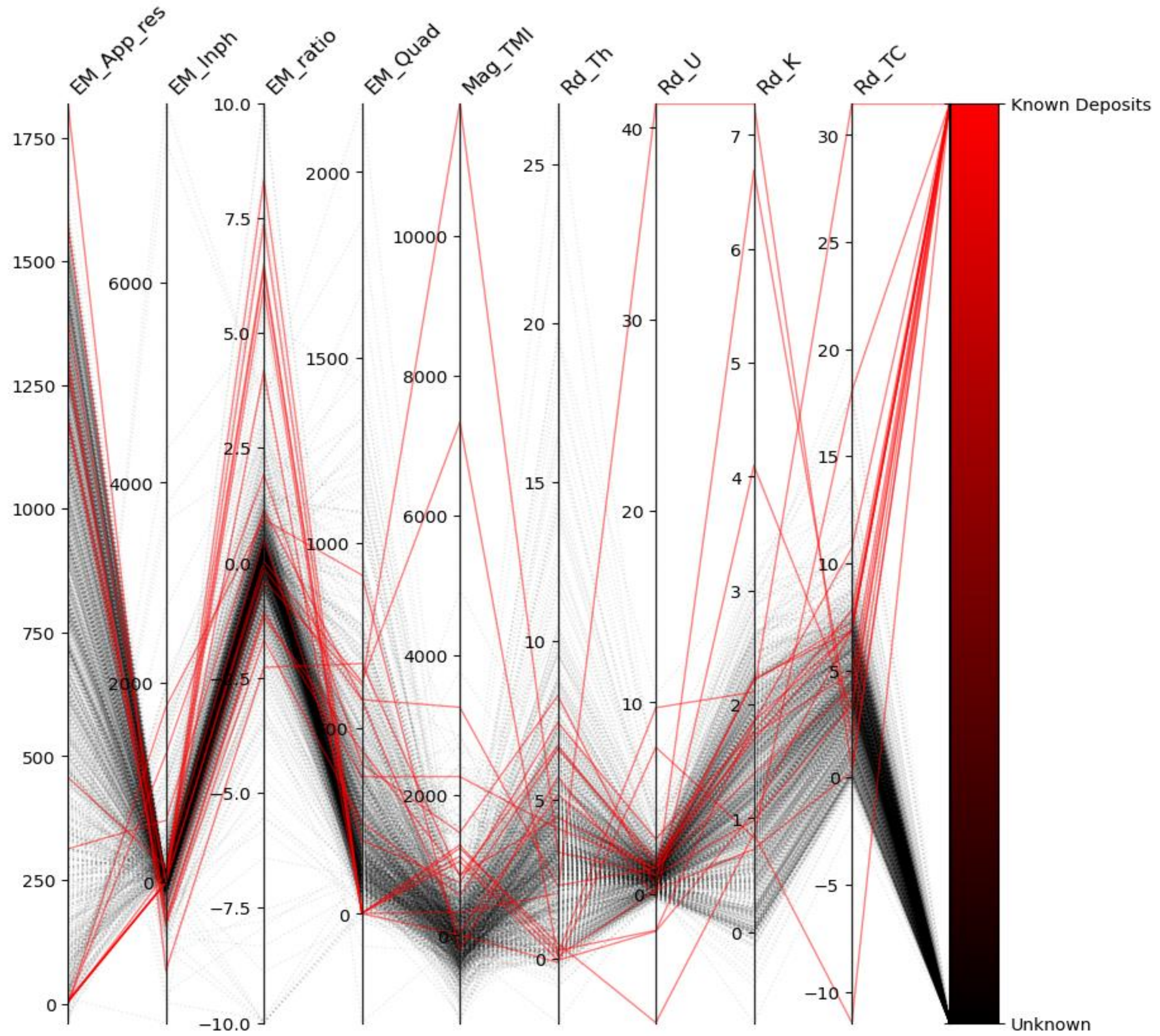

Figure 2: Example of parallel coordinate plot with geophysical data.

In Figure 4, the value of known ore deposits is shown in red, while values from other places are shown in shades of grey with darker areas showing higher densities than lighter areas.

Funded by
the European Union

3.2.3 Heatmap

Heatmaps are a type of plot that visualizes the strength of relationships between numerical variables. Correlation plots can be used to understand which variables are related to each other and the strength of this relationship (Figure 3).

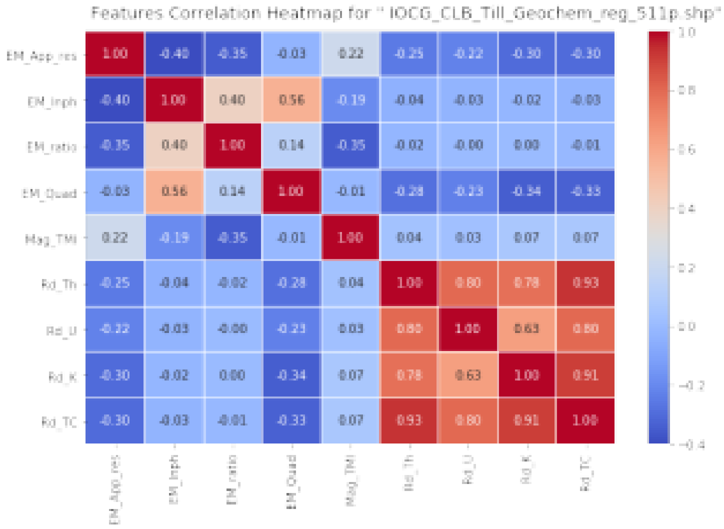

Figure 3: **Pairwise correlations between different geophysical variables.**

The radioactivity measures are strongly correlated with each other in the test data, but otherwise the variables are mostly not strongly correlated.

3.2.4 Scatter plots

A scatterplot is useful tool for understanding the pairwise relationship between different variables in a dataset. They are particularly effective when they are interactively combined with other data visualization tools.

Example with geophysical data

For this example, the PCA visualization introduced previously is used as a starting point. A sub-region of the data with similar colours to the known ore deposits is also demarcated (Figure 4).

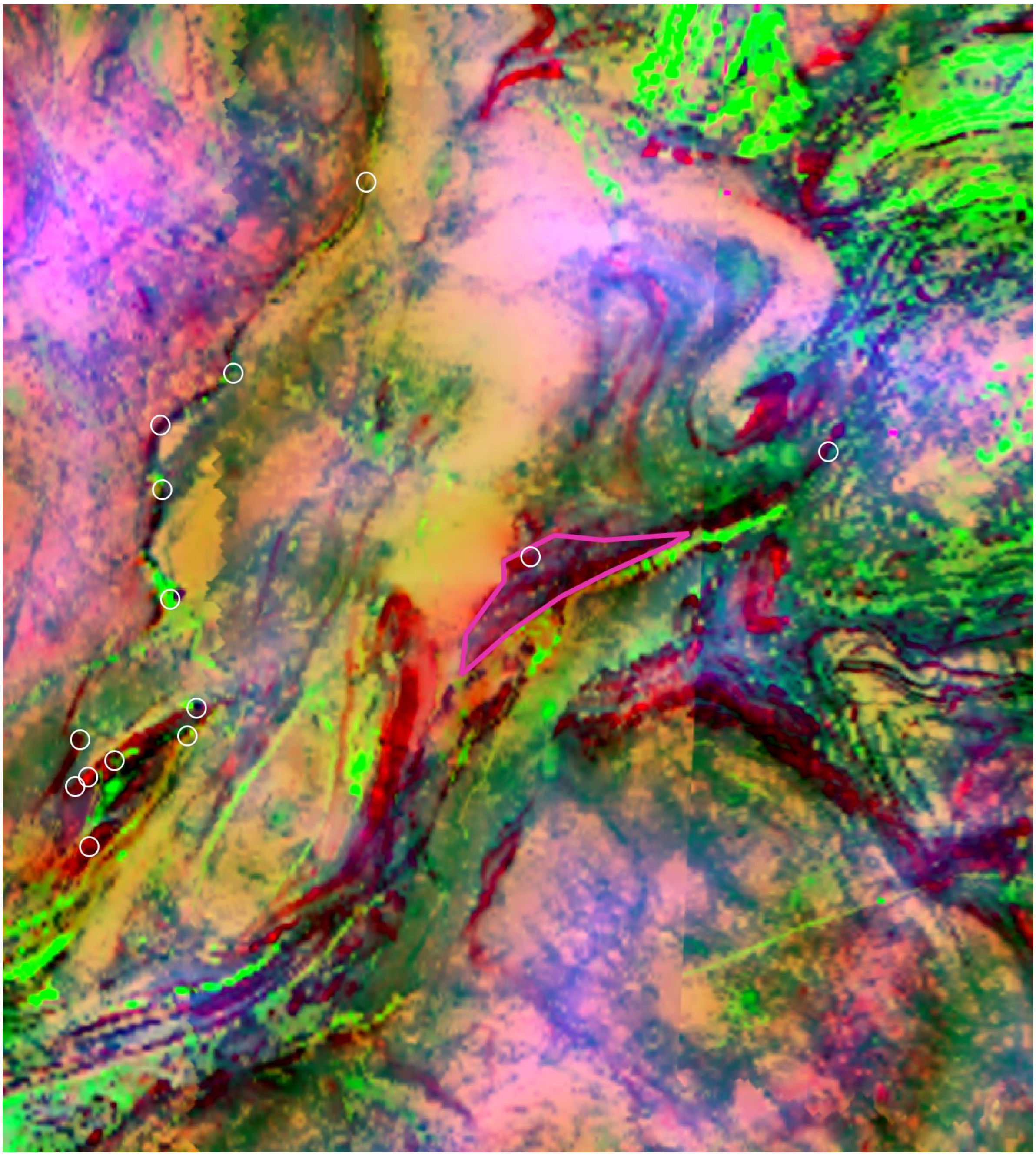

Figure 4: **An example sub-region (pink) that is similar colour to the known ore deposits (white rings) has been digitized, and its values from bands 1 and 2 have been plotted on Figure 5 along with the values of all the data. The data is PCA from multiple geophysical data sources, where principal component (PC) 1 is assigned to red, PC 2 to green and PC 3 to blue.**

Principal components 1 and 2 were chosen for the scatterplot (Figure 5). The sub-region was plotted with a distinct colour than the rest of the data. Raster files often have millions of pixels, and it is thus important to show point density with colour in the densest parts of the scatterplot.

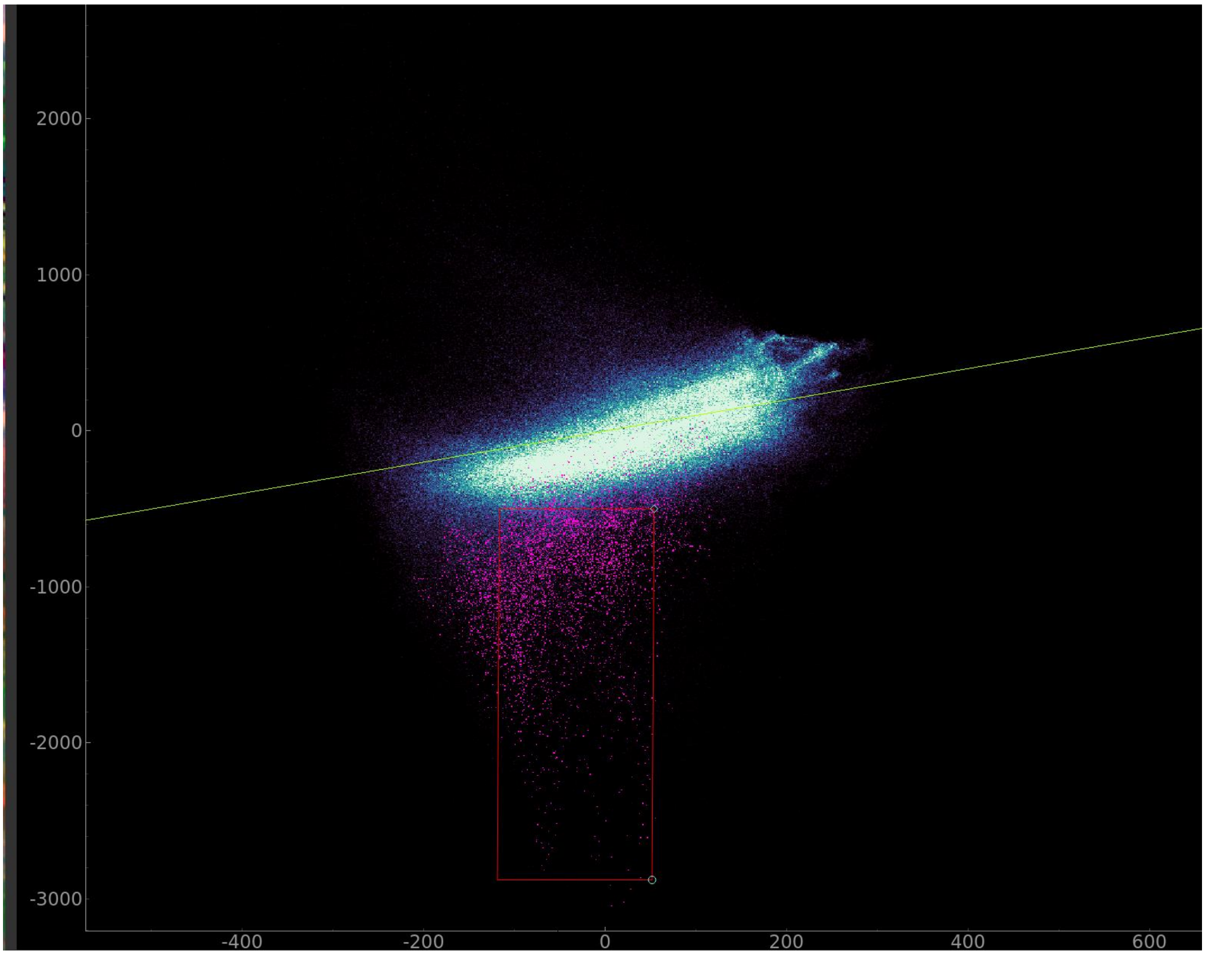

Figure 5: **Scatterplot of principal components 1 (horizontal) & 2 (vertical). The colours indicate the sample density (white: dense, pink: sparse).**

The sub-region shown in Figure 5 is shown in pink. The red box shows the approximate value range of the sub-region. All the other values are shown in shades of blue. Lighter regions correspond to higher density.

Finally, the locations with values within the demarcated region in the scatterplot can be added back to the map (Figure 6). These are areas with similar geophysical features, and ore deposit locations since they are like known ore deposits. The areas of interest could be narrowed further by looking at different band combinations in the scatterplot.

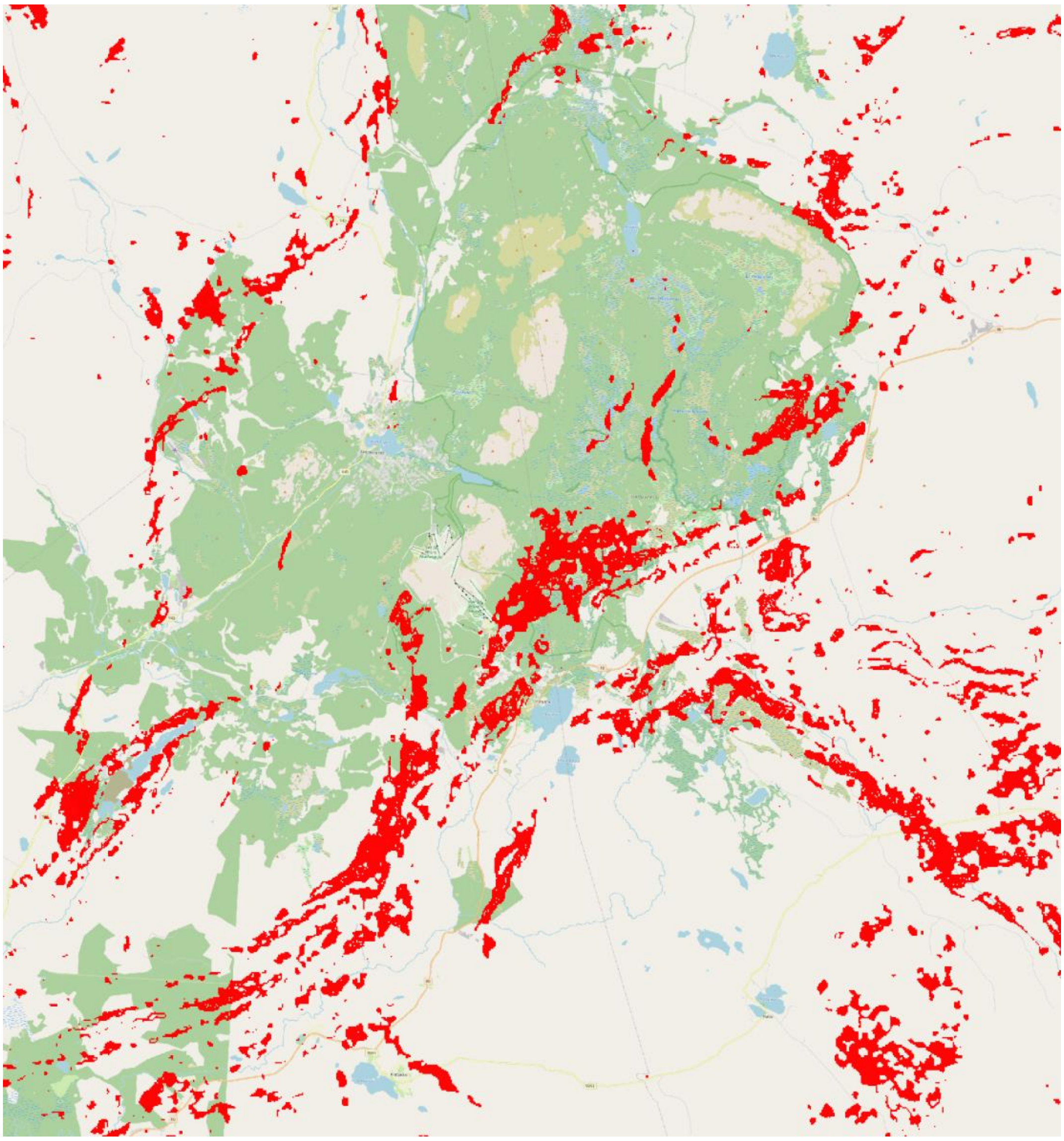

Figure 6: Locations of pixels with values of principal components 1 and 2 in the value range demarcated in Figure 7.

3.2.5 Self-organizing maps (SOM)

Self-organizing maps are a type of neural network that were invented in the 1980's (Kohonen 1980). SOM can be used to compress multiple layers of data into a simpler feature space, which is useful for finding groups in the data and visualizing patterns. It works by initializing a grid of values with either random or PCA-initialized values. Then for all the data the best matching unit (BMU) is found, and the values of that cell and ones close to it are moved towards the presented data point. SOM results can be concisely represented in a U-matrix, where the value describes each cell's "feature distance" from the neighbouring cells (Figure 7A). Additionally, SOM can also be looked at per variable basis. These can then be changed to geographical maps by finding the BMU for all the pixels of a Geospatial raster (Figure 7B and Figure 7C). SOM has shown promise with exploration of Geological datasets (Torppa et. al. 2021). The SOM can also be k-means clustered (Figure 8).

Funded by
the European Union

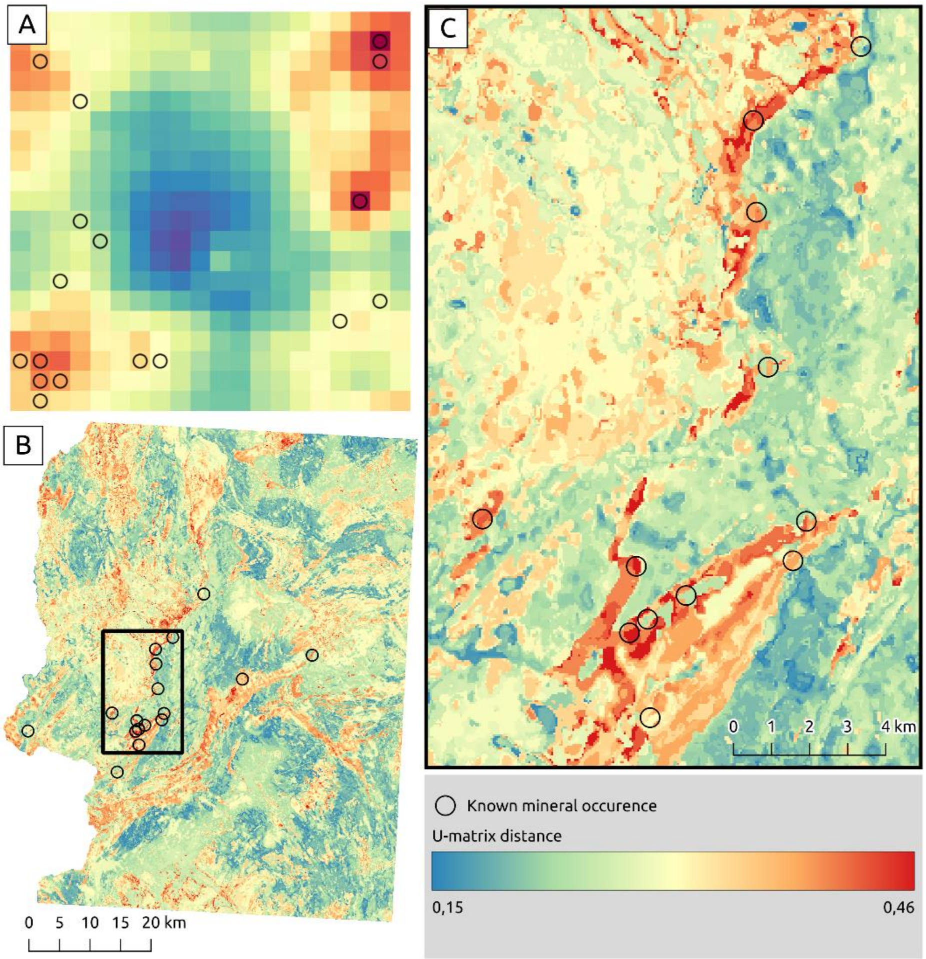

Figure 7: SOM with U-matrix representation. Known mineral occurrences are shown with black circles. A) The raw SOM U-matrix representation. Please note that the topology is toroidal, which means that top and bottom rows are next to each other as well as leftmost and rightmost columns. B and C) Geospatial U-matrix.

Funded by
the European Union

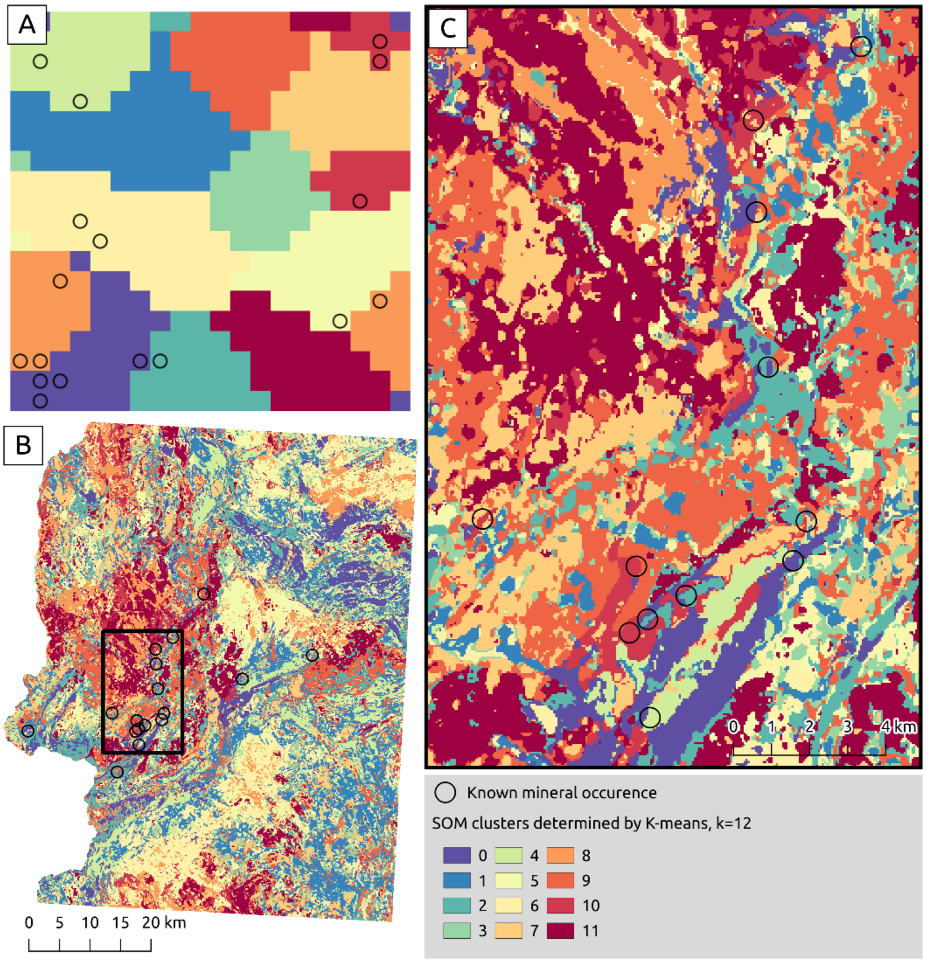

Figure 8: K-means clustered SOM.

3.2.6 Mahalanobis similarity

Mahalanobis similarity is standard deviation extended to multiple variables. Main idea is to measure similarity of all locations in a given geospatial raster to a (small) group of known mineral occurrences (Figure 9). The similarity is measured compared to the averaged features of the mineral occurrences, meaning that even the known mineral occurrences do not get full similarity.

However, there are multiple reasons for why Mahalanobis may have limited usefulness with geological data. Mahalanobis requires multivariate normal distribution, which is rare in geological data. Mahalanobis may thus require several transformations of data or simply not be possible. Only important and relevant geophysical measurements should be included in the analysis, which requires previous feature importance analysis.

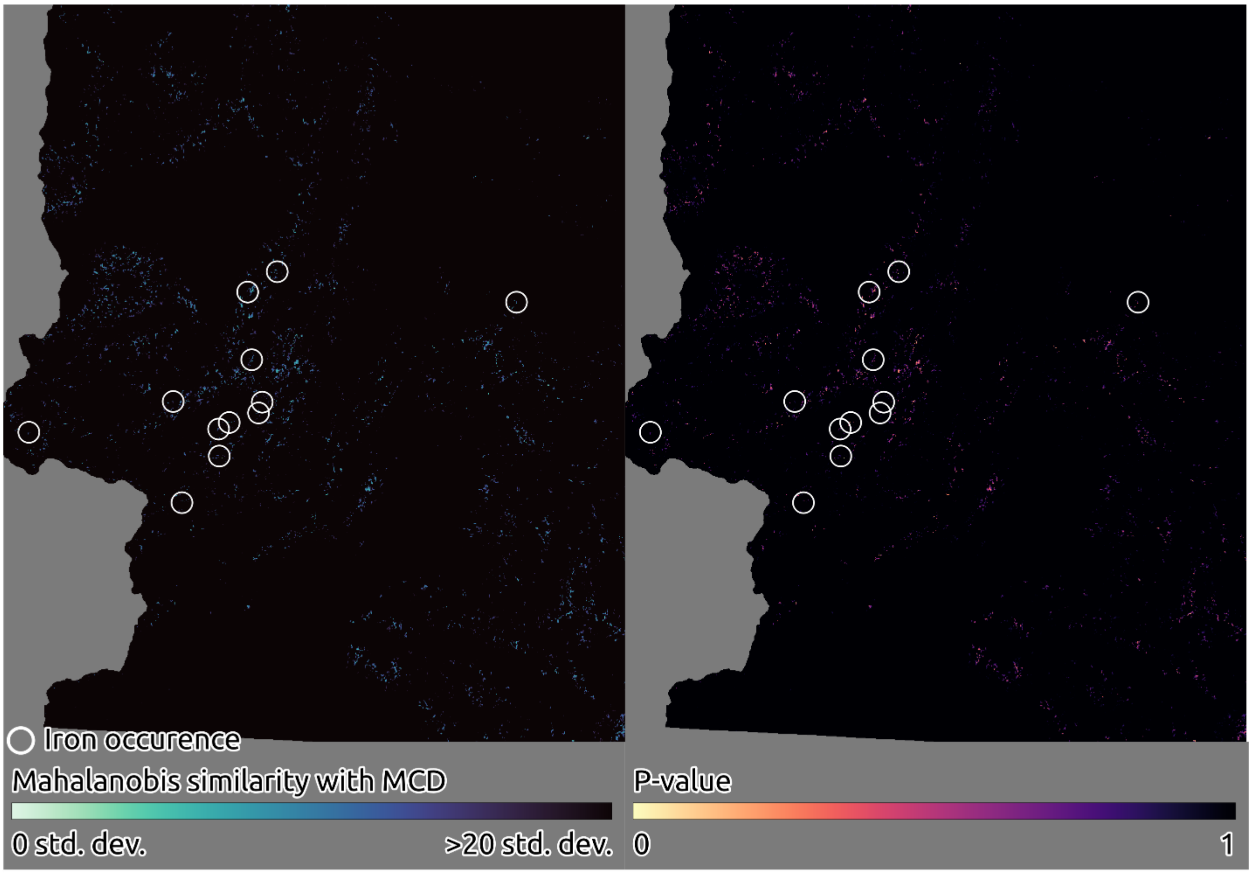

Figure 9: Mahalanobis distance and p-values.

3.2.7 Spatial cross-validation

Spatial cross-validation (SCV) comes in several different varieties, since the topic has been discovered and addressed over several independent studies during the years 2011-2023. SCV can be divided to three approaches, each one increasingly more pessimistic when compared to standard, non-spatial CV (number 1):

1. Normal CV: data is divided to test and non-test parts. The latter is divided to pairs of teaching and validation. Each teaching data set helps to optimize method parameters, and the performance is tested by validation part. The validation part is used to optimize the hyperparameters. Finally, the test set records the estimated performance, when facing new data. The notable thing is that data is divided to various sets at random, independent of spatial location. A synthetical example of sample locations divided to train and validation (could be test and non-test division as well) is depicted in Fig. 1 (a). The procedure is finished by the final teaching with whole data to achieve the production predictor, estimate of performance was produced by previous arrangement.

2. Zonewise CV (ZCV): just like normal CV, except each set is allotted a solid zone, and zone borders are relatively simple (not zig-zagging), see Figure 10b. This gives some sense top the final estimation of the performance, since spatial auto-correlation (SAC) has a minimal effect (only some sample pairs chosen from training and test sets are geometrically close). (Deville et.al. 2014) and (Towards 2024). Since this approach omits a lot of SAC effects, it tends to create a more pessimistic estimate of the prediction performance when facing similar data channels in a completely new location.

Funded by
the European Union

3. ZCV with a spatial margin (ZCVm): just like the plain ZCV, but a margin of width *r* is added in between different subsets (Beigaite et.al. 2022). The margin makes it more clear how semivariogram subjected to the prediction error would behave over the range *r*, and is more pessimistic than the version 2. See Figure 10c.

4. Normal CV with a spatial margin (SCV): Just like normal CV (with its usual versions of normal CV and leave-one-out (LOO)) but this time each training sample, which is too close from any validation or test sample, is excluded. (Ploton et.al. 2020, Airola et.al. 2019] See Figure 10d. This drastically reduces the training set when r is large, but for practical r values it gives much more stable models over each training set, and estimates prediction semivariogram accurately over much longer ranges. It is able to judge, whether a spatial generalization occurs and if so, how far it reaches. It is also able to predict a degree of spatial invariance, that means, whether a model is valid "everywhere" or not.

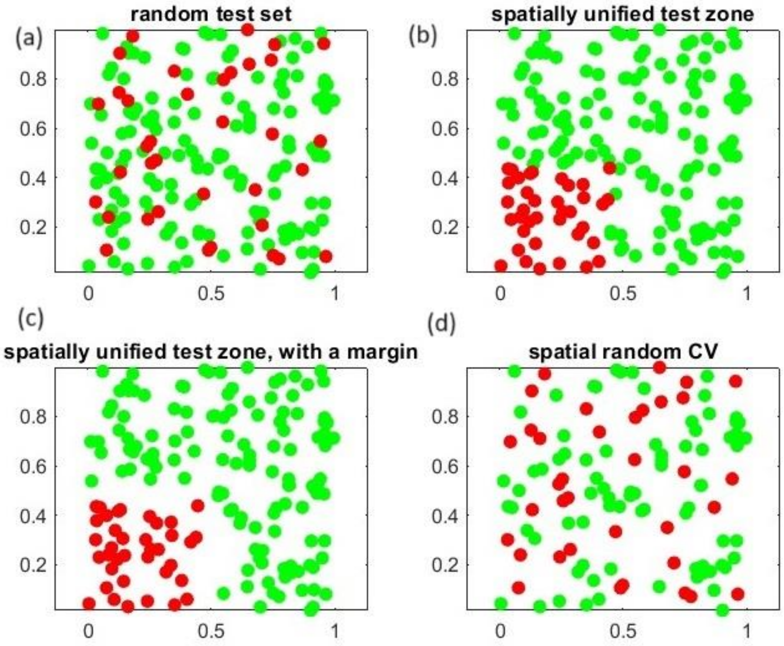

Figure 10: **(a): Spatial data divided to two sets: teaching (green) and validation (red). (b): Training and validation sets divided to different zones. (c): Margin added to (b). (d): Margin used to (a) to eliminate too close points from the training set (green).**

Figure 11a has semivariograms of a prediction over the range r. The definition of semivariance concerns only one variable v, mean of *(v(p) – v(q))^2* of which is taken over all locations *p,q* having a distance *r* ≈ *||p-q||*. In this case we digress from this and subject the semivariogram over the prediction errors $\hat{f} - f$. Typically the SCV semivariogram of error grows fast until it reaches approximately the level of prediction to mean, at which range the model is practically useless. SCV is more pessimistic (error above ordinary CV error expressed as semivariogram), and normal CV shows total ignorance to the distance of the prediction.

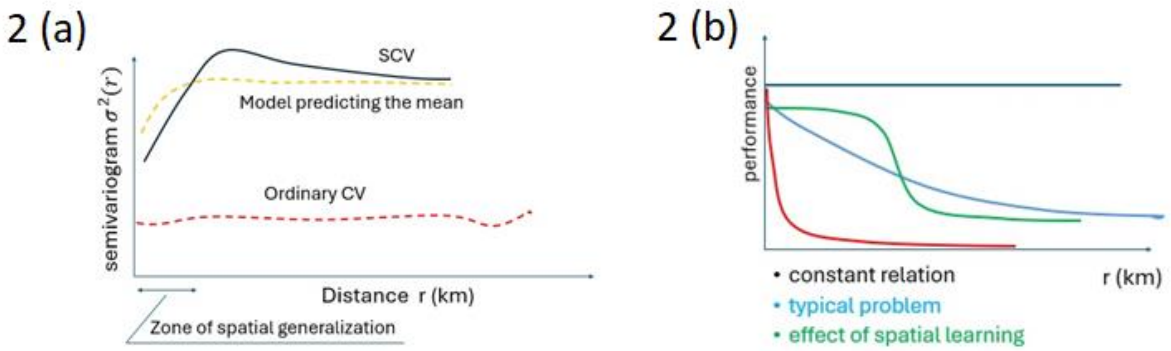

Figure 11: **(a) Semivariance of a prediction error (red dashed) by normal CV stays low and independent of distance. The prediction to mean (yellow) gives mean square error (SQE) measured as semivariogram. Spatial CD (black solid) predicts much higher error, which stabilizes to the prediction of mean at the end of the zone of spatial generalization. (b) Classification performance of four models: spatially independent model (black), a typical problem (blue) with a model without spatial generalization capability, signs of spatial generalization (green), highly localized problem (red). Detail (a) is a recreation of a similar presentation in (Ploton et.al. 2020).**

Figure 11b shows 3 different classification problems, and their classification performance behaviour over range *r*. There is a category of spatially invariant problems (e.g. annual temperature at the mountain top predicted by the mountain height is surprisingly stable over large latitude range). The most common problem type (blue) becomes gradually more difficult over the range to nearest labelled data, and the last problem (red) shows high spatial correlation, and should be approached by Kriging -like methods, and this only nearby any labeled samples. The green curve is a result of applying an expressive enough model successfully to a problem (blue) achieving spatial generalization. Detail (b) examplifies results in (Airola et.al. 2019), where the measure (performance axis) used was an area-under-curve AUC of a binary classifier.

3.3. Mineral prospectivity modelling

3.3.1 Random forest

Random forest (RF) algorithm is a decision tree-based method, which works by testing multi-branched decision trees for the input data set and selecting the decision tree that results in the best prediction. Each node in the decision tree represents testing an input feature, and each branch end represents the result of the test (Breiman 2001). The decision trees allow modelling complex interactions in the feature space formed by the input data, and thus they often work well in high-dimensional feature space. Individual decision trees are often prone to overfitting, which can be kept under control by using multiple decision trees (Stephens 2017). RF typically requires enough observations in the training data. RF is a common method in remote sensing-based applications where a high number of input features are used.

3.3.2 Logistic regression

Logistic Regression (LR) is a widely used statistical model that is mostly employed for binary classification tasks (Hosmer Jr et al., 2013). It is well-known for its simplicity, interpretability, and its capability to generate probabilistic predictions (Agresti, 2002). The LR model is based on a robust mathematical framework, which models the logarithm of the chances of event probabilities (McCullagh and Nelder, 1989). This model is optimized using the Maximum Likelihood Estimation (MLE) technique, which guarantees that the predictions align with the observed data. The adaptability of this technology is seen in its wide-ranging applications across various sectors, including medical diagnostics, marketing analytics, and financial risk assessment. The ongoing relevance and utility of linear regression (LR) in predictive analytics are underscored by its capacity to offer valuable insights through the interpretation of coefficients and its adaptability in various scenarios, including those involving sparse data environments. These strengths persist despite LR's linear nature and the assumption of a linear relationship between features and log-odds. The enhanced interpretability of the model, resulting from its capability to generate probabilities, facilitates nuanced decision-making across many applications (Bishop et al., 2006). Consequently, it has become a vital tool within the domain of statistical and machine learning algorithms.

3.3.3 Multilayer perceptron

Multilayer perceptron (MLP) (Bishop, 1995) is a simple feedforward neural network that is composed of multiple layers of perceptron. MLP typically consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer. In the prediction task, MLP neural network typically trains the neural network to minimize the difference between output and objective by adjusting the weights of the inputs within the layers. MLP, or neural networks in general, are usually trained using some form gradient descent-based backpropagation algorithms which calculate the network’s current performance with the data and then updates the network weights layer by layer by moving backwards (i.e., back-propagating) through the network. Because the number of weights in a neural network are usually exceptionally large, it becomes easily computationally infeasible to perform the iterative updates of the network using the full data available. In these cases, especially in deep neural network context, a stochastic versions of backpropagation algorithms are used which take a random smaller sample of the training data and perform the weight updates using this sample. The training of MLP is easy task, but in remote sensing-based applications MLP has been superseded by other machine learning methods such as genetic algorithms and support vector machines due to their better empirical performance in empirical prediction and classification tasks (see e.g., Frias-Martinez et al., 2005, Pohjankukka et al., 2018).

3.3.4 Multilayer perceptron probabilistic model

This study deploys a Multilayer Perceptron (MLP) neural network to conduct mineral prospectivity mapping. The dataset used in this task is pre-processed with the Synthetic Minority Over-sampling Technique (SMOTE), which ensures that class imbalance issues are handled effectively (Bishop, 1995) (Chawla et al., 2002). The aim of this study is to predict the probability of mineral occurrence in different geographic areas, focusing on a binary classification approach. The Multilayer Perceptron (MLP) is a neural network architecture with interconnected layers: the input, hidden, and output. Its purpose is to effectively capture intricate relationships within the data, facilitating accurate mineral prospectivity predictions. Following the training phase applying SMOTE-enhanced data, the model was subsequently evaluated on mineral class occurrences that were previously unknown mineral class instances. The model computed the probability of each mineral class being present during this evaluation.

The probabilities obtained from the last layer of the Multilayer Perceptron (MLP) were utilized to perform a binary classification of each geographic point. Based on the predetermined threshold criteria, these probabilities facilitated the binary type of each geographical point into two categories: mineral-rich or not mineral-rich.

Subsequently, points categorized as containing minerals are extracted and carefully mapped on a geographical map based on their precise geographic coordinates, which provides a visual representation of the regions with a higher likelihood of mineral presence. Integrating SMOTE pre-processing, MLP modelling, and geographical visualization offers a robust methodology for identifying possible mineral-rich areas, which helps mineral exploration professionals make better strategic decisions.

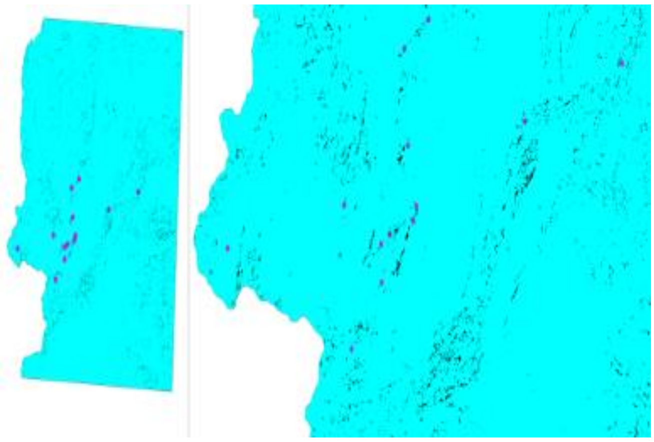

Figure 12: Prediction of minerals with probability threshold (>=) 50%.

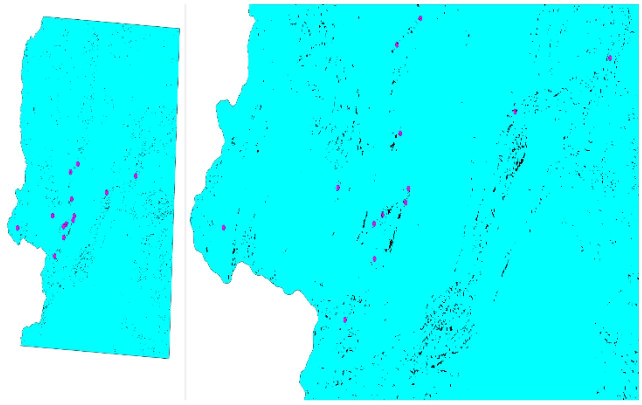

Figure 13: Prediction of minerals with probability threshold (>=) 95%.

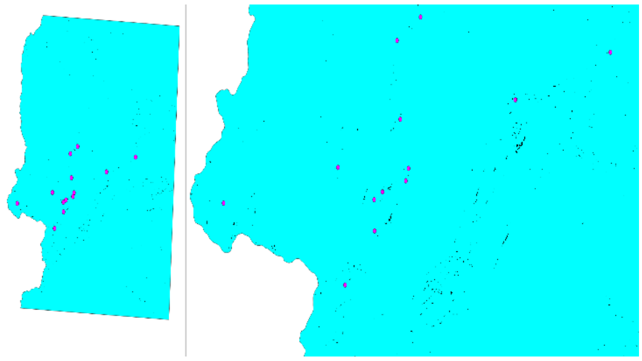

Figure 14: Prediction of minerals with probability threshold (>=) 99.998%.

Figures 12-14 illustrate the spatial visualization of probabilistic model estimation achieved through the Multilayer Perceptron (MLP) classifier. The MLP model, trained on the SMOTE data, predicts the probability of mineral occurrence across the given geographical areas. By applying specific threshold criteria, the geographical points in Figures 12-14 are classified into either mineral deposits or non-deposits. These thresholds correspond to confidence levels of 50%, 95% and 99.998%, respectively.

3.3.5 Convolutional neural network (CNN) classifier

In the last few years, Deep Learning (DL) has made huge progress in remote sensing (Lei et al. 2019). The most common DL applications in remote sensing include image fusion, scene classification, object detection, land-cover classification, and semantic segmentation. However, for remote sensing, several challenges from difficult data acquisition and annotation have not been fully solved yet. Convolutional neural network (CNN) is a most extensively used DL models that can process data in the form of multiple layers (Yao et al. 2019). For this reason, CNN is applicable for processing multi-band remote sensing data. A CNN model consists of different layers of convolution, Rectified Linear Unit (RELU), pooling and fully connected (Dhillon et al. 2020). The Convolutional layer learns feature in an input data. The RELU layer introduces non-linearity through activation function. The pooling layer reduces dimensionality and preserves spatial invariance. All these layers are responsible for feature extraction and then the fully connected layer perform classification.

3.3.6 CNN segmentation with uncertainty estimate

Semantic segmentation with a CNN means classifying images pixel-wise. U-net has become successful and popular segmentator in many fields. However, training it requires large datasets that may be labour-intensive or simply unrealistic to create. This is especially true in Geology, where usually only a handful of occurrences are known. Thus, several measures are implemented to minimize the need of training data. First of all, the used images are small, only 32 x 32 pixels. Secondly, instead of full U-net only the mid-section of the model is used. Thirdly, the point features are exaggerated in size because larger objects are easier to recognize for CNNs. Fourthly, the areas with known locations are oversampled (Figure 15). The known occurrences are oversampled and negative samples are selected randomly. This ensures that both sample types are represented equally in the created dataset. The sampler works well with sparse datasets, where it is relatively rare that the negative samples would contain an unknown occurrence. Even after all these methods, overfitting remains a problem.

Funded by
the European Union

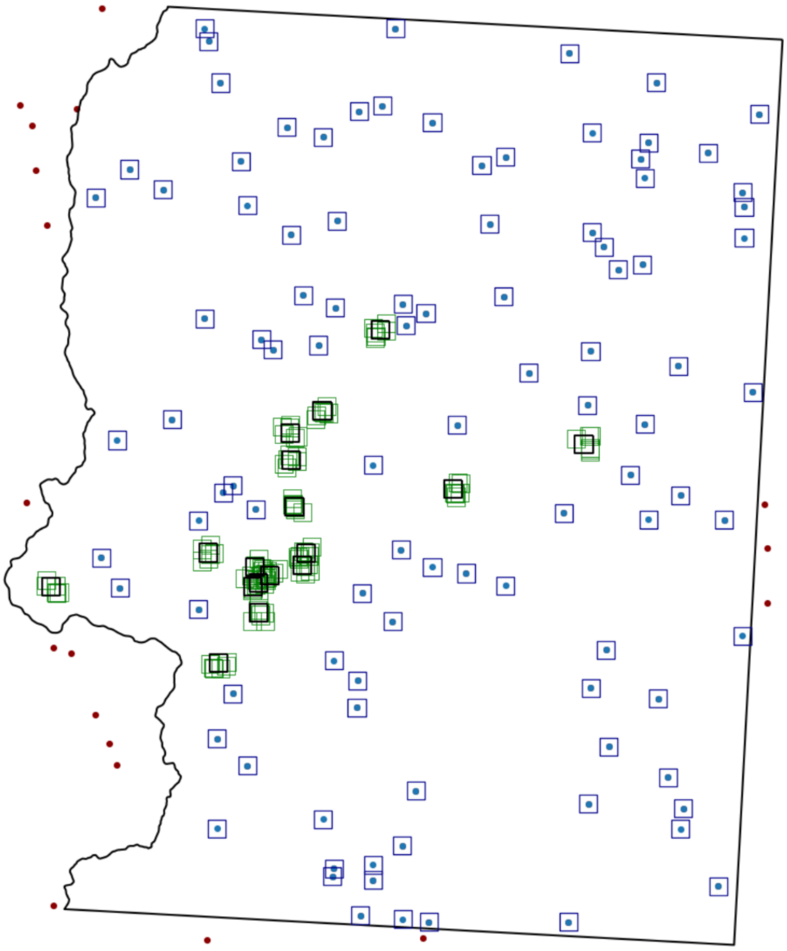

Figure 15: Example of samples created for CNN.

Main idea of uncertainty estimation: the model learns to predict its own uncertainties alongside the actual prediction (DeVries & Taylor, 2018) (Figure 15). When overall certainty is high (close to 1) the model is evaluated as if no certainty estimate was used. When certainty is close to zero at certain pixel, the model prediction does not matter at all for the loss at that pixel, encouraging accurate uncertainty estimate. To prevent model predicting high uncertainty everywhere, it is punished for being uncertain at large areas.

Cons: The model hyperparameters need to be fine-tuned precisely. Even after all the data scarcity mitigation, overfitting remains a problem. However, the uncertainty estimate may reveal potential new areas even when the model has started to overfit (Figure 16).

Funded by
the European Union

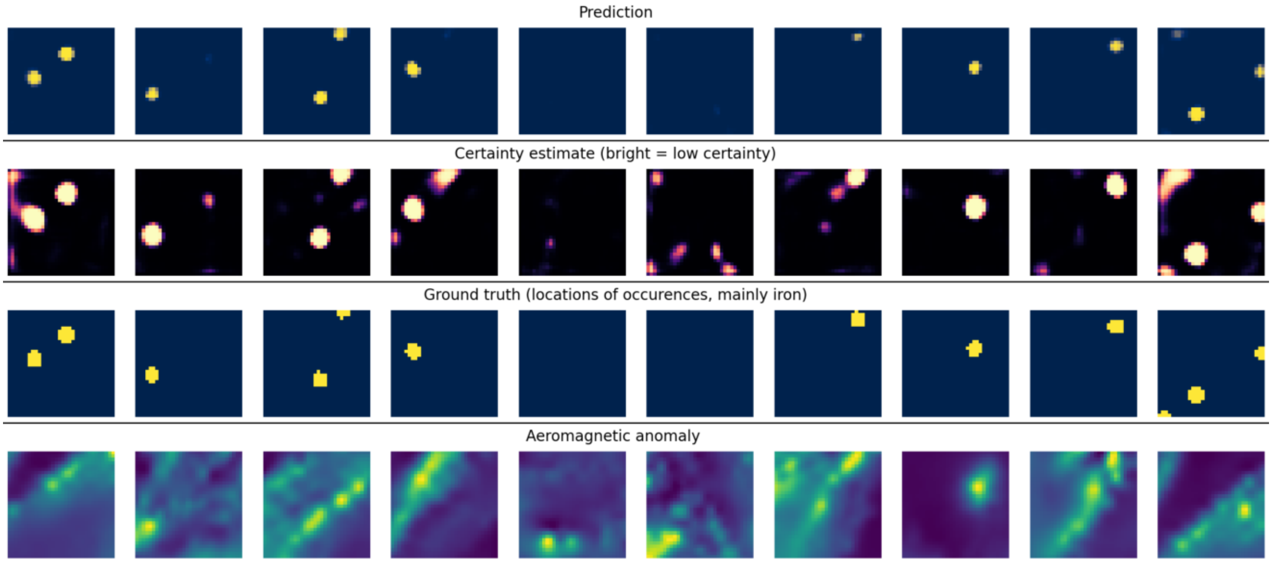

Figure 16: Example of model prediction and uncertainty estimate.

3.3.7 Permutation feature importance

Permutation feature importance is a model-agnostic method for figuring out each feature's importance in a machine learning model. It does this by measuring how the model's performance changes when the values of a particular feature are changed randomly (Breiman, 2001). The basic idea is that if a feature is vital for making predictions, changing its value at random should significantly affect how well the model works. It is model agnostic, allowing it to be applied to any supervised learning model, and it does not rely on assumptions about the data distribution or the model's structure (Fisher et al., 2019). However, it can be computationally expensive, especially for high-dimensional datasets, because it requires evaluating the performance of the model more than once. Also, it might not give an accurate estimate of how important correlated features are since rearranging one feature might not significantly affect how well the model works if another correlated feature still gives similar information (Strobl et al., 2008). Despite these shortcomings, permutation feature importance is still a fashionable way to figure out how important a feature is in different situations. It helps researchers and practitioners understand their models, determine their meaning, and make their models work better by choosing crucial features.

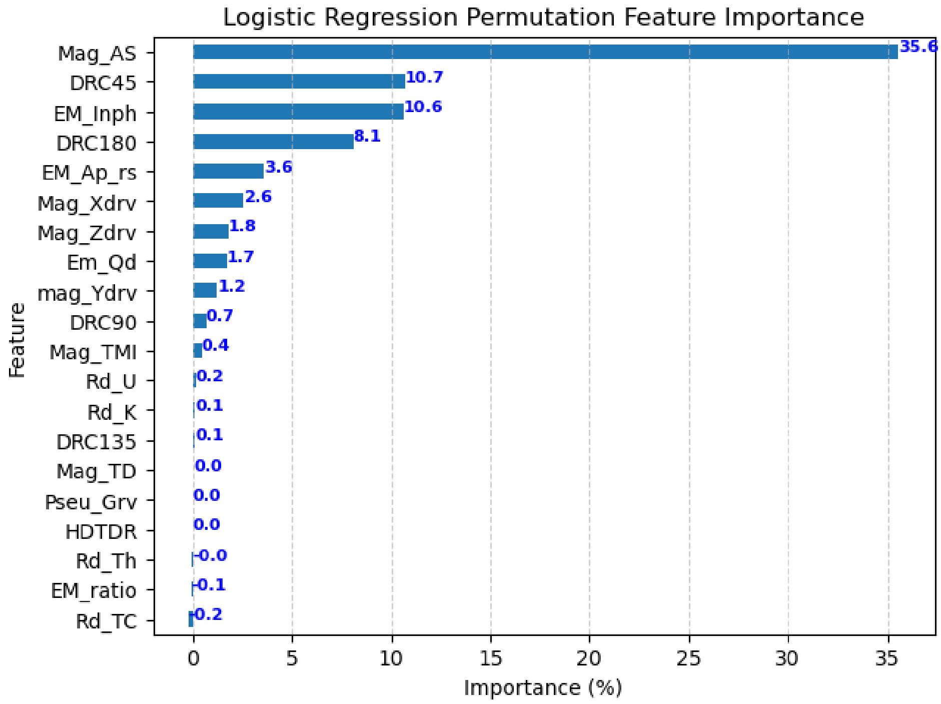

Figure 17: **Permutation Feature Importance with Logistic Regression.**

Figure 17 demonstrates the results of applying permutation feature importance to a logistic regression model. This study helps in the understanding of the features that have the most significant impact on the performance of the model.

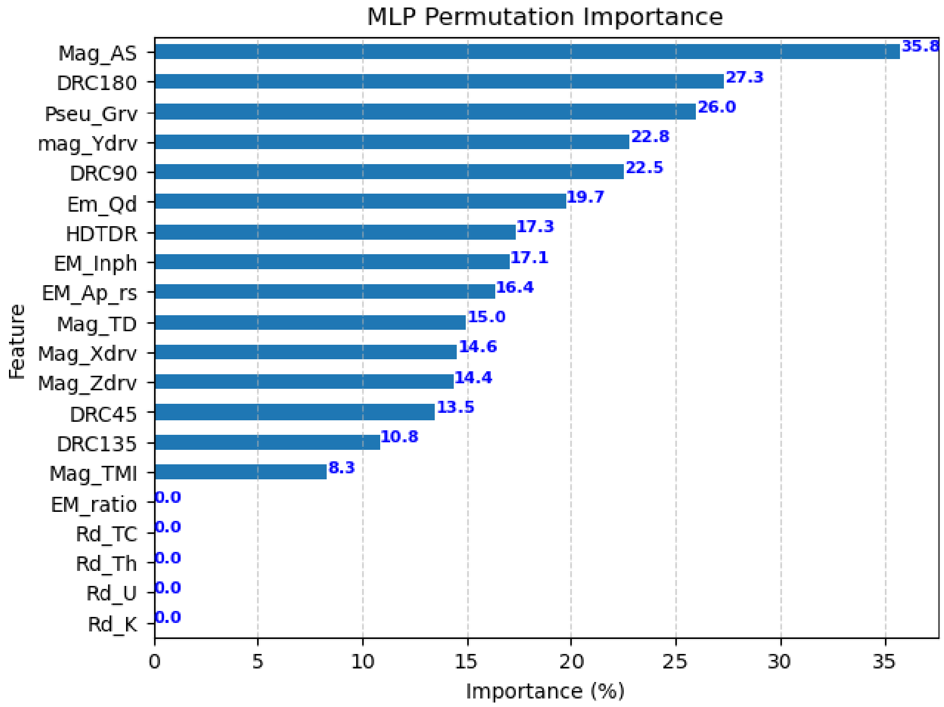

Figure 18: **Permutation Feature Importance with Multilayer Perceptron.**

The study is further expanded to encompass a more intricate model known as multilayer perceptron (MLP) as shown in Figure 18. Permutation feature importance applied to the MLP model illustrated how different features contribute to its predictive power.

The application of permutation feature importance enables user to make decisions on the selection of features, hence improving the interpretability of models and optimizing their performance.

3.3.8 Weighting of class

Weighting of class is a machine learning technique to address imbalanced datasets, where some classes have significantly fewer instances than others. Imbalanced datasets can lead to biased models that perform poorly on the underrepresented classes, as the learning algorithm tends to focus on the majority class to minimize the overall loss (He & Garcia, 2009). By assigning different weights to each class, the learning algorithm can be guided to pay more attention to underrepresented classes during training. Most of the time, larger weights are assigned to the minority classes, whereas lower weights are assigned to the majority classes. This way, misclassifying an instance from the minority class will significantly impact the loss function, encouraging the model to improve its predictions for the underrepresented classes. Many machine learning libraries, like Python's scikit-learn, have built-in features that automatically calculate class weights or let users set their own (Pedregosa et al., 2011). Adding class weights to the learning algorithm makes the model more likely to do better in the minority classes. This class weight makes the classifier more balanced and reliable.

3.3.9 Synthetic minority over-sampling techniques (SMOTE)

Synthetic Minority Over-sampling Technique (SMOTE) is a popular technique used to address the class imbalance problem in machine learning datasets (Chawla et al., 2002). The issue of class imbalance occurs when certain classes have a significantly smaller number of instances than others, which results in biased models that underperform for the underrepresented classes. The problem of class imbalance occurs when some classes have much smaller datasets than others, resulting in biased models. SMOTE works by generating synthetic samples for the minority class, effectively balancing the class distribution, and improving the model's performance on the minority class. In this way, SMOTE generates synthetic instances, providing a more balanced dataset without duplicating any already existing instances. This allows the learning algorithm to capture the decision boundary between the classes better and achieve improved performance on the minority class.

Despite its advantages, SMOTE also has some limitations. It can lead to overfitting, particularly when the minority-class instances are noisy or near the decision boundary. Moreover, SMOTE does not consider the majority class distribution, which may result in synthetic instances being generated in regions where majority class instances dominate. To address these issues, several SMOTE variants have been proposed, such as Borderline-SMOTE (Han et al., 2005) and Adaptive Synthetic (ADASYN) sampling (He et al., 2008), which focus on generating synthetic instances in more challenging regions of the feature space.

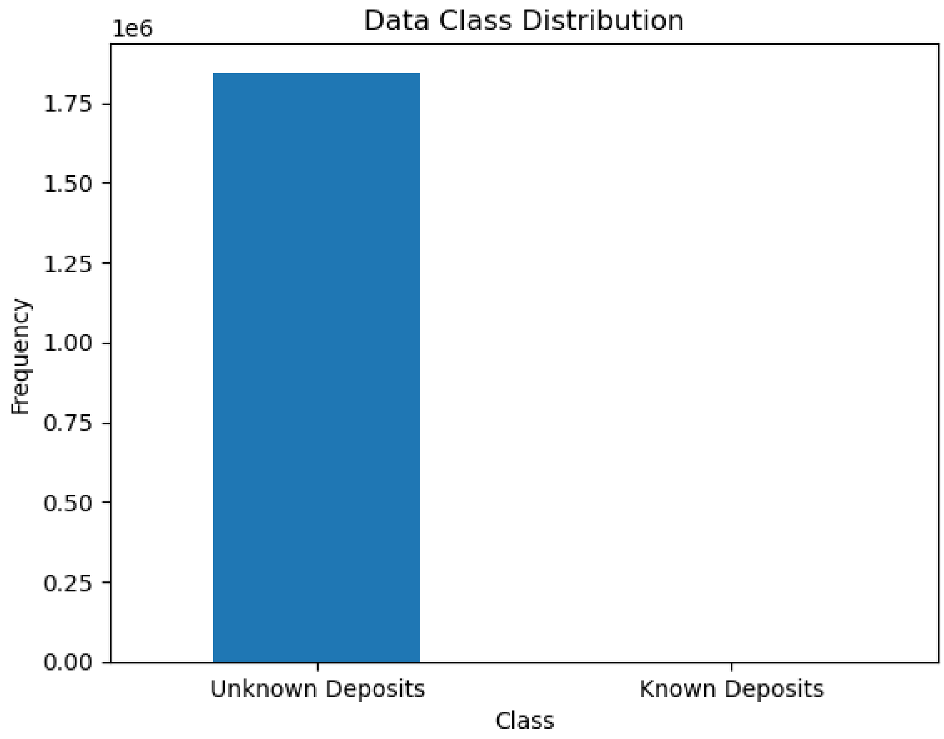

Figure 19: Original Dataset Distribution.

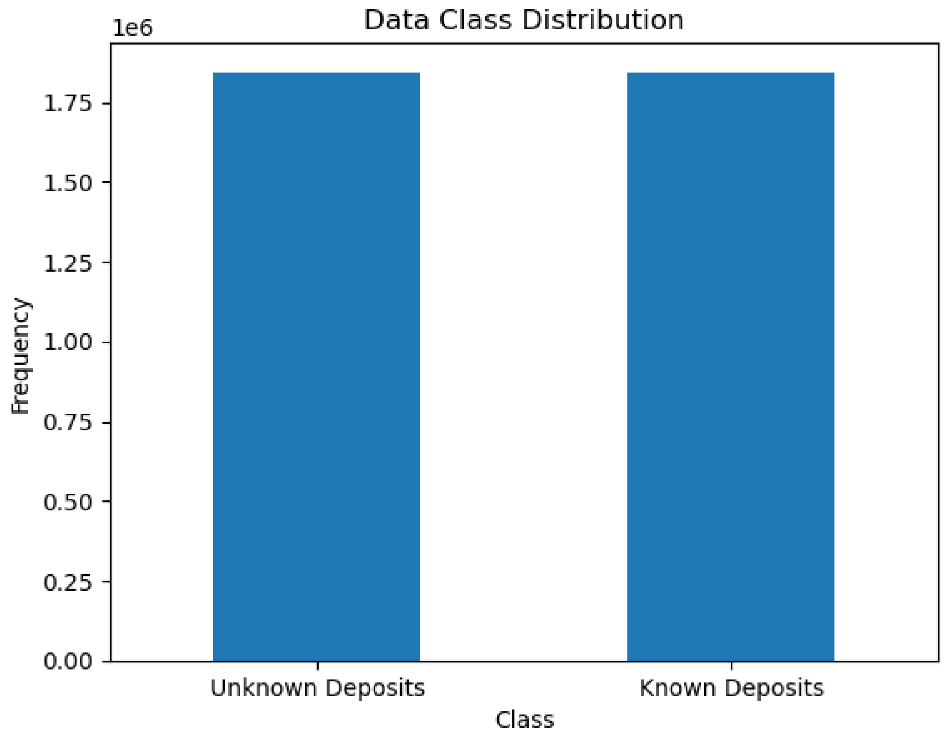

Figure 20: Dataset Distribution after applying SMOTE.

Figure 19 illustrates the distribution of the original dataset, highlighting the presence of class imbalance concerns wherein certain classes exhibit a notable scarcity of instances. On the other hand, Figure 20 illustrates the dataset distribution following the implementation of the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE decreases challenges related to class imbalance and it improves the model's capacity to manage instances from the minority class successfully.

3.3.10 Autoencoder

Autoencoders can be used to gain useful information about the data; this is achieved by training an encoder-decoder network that reconstructs the input image. After training, by inspecting the bottleneck layer between the encoder and decoder, the data’s dimensionality is reduced to the most prominent features, as the features have thus been squeezed into a minimized number of computing units. The code allows for different autoencoder configurations, where the user can give the desired number of layers and convolutional filters, which will then build a symmetrical encoder-decoder structure. The used loss metric was mean squared error, along with Structural similarity index for evaluation.

This bottleneck can then be used for varying applications, including:

1) training other models via transfer-learning,

2) non-linear dimensionality reduction, as opposed to linear dimensionality reduction like PCA,

3) Visualizing the features of the bottleneck, which can be done with tools such as t-SNE in 2D or 3D (as shown in Figure 21),

4) Feature extraction by using the bottleneck layer as an input to other models, such as a CNN or as a part of a loss function, such as Perceptual loss (Johnson et al., 2016),

5) Anomaly detection, either with training the model on normal data and then predicting it on new data, as anomalies would have high reconstruction loss, or by inspecting the feature difference and clusters with tools such as t-SNE

6) Denoising the data, which might have its uses with imperfect satellite images,

7) Generative modelling, both with the before-mentioned Perceptual loss function and with Variational autoencoders,

8) prediction tasks, such as predicting the band A given input data of bands B and C.

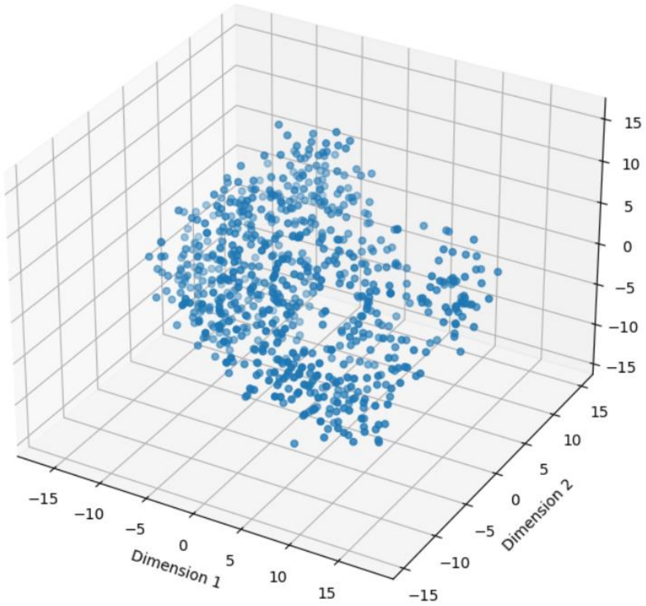

Figure 21: Example of 3D t-SNE on geophysical data.

Included is also an autoencoder builder function with skip-connections and attention blocks. Skip-connections are used in U-Net segmentation models to bypass layers and connect to layers not immediately next or before the current layer, which have been noticed to aid in segmentation tasks to keep small and general feature relationships correlated. Attention is a method to focus on features, that have the strongest effect on the model performance or affect the current segmentation etc. task the most.

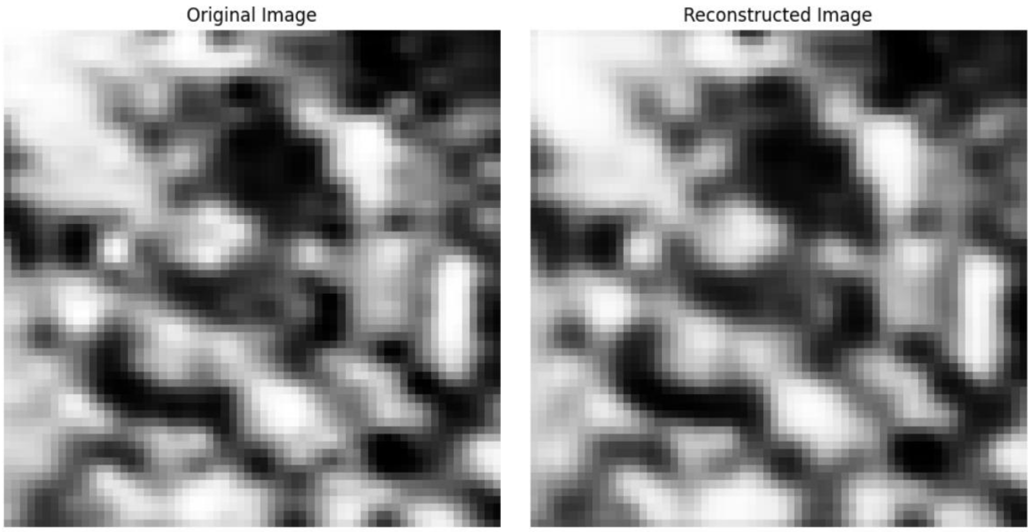

Figure 22: Example of reconstruction of AEM band of geophysical data.

3.3.11 Uncertainty estimation using Bayesian neural network

In traditional deep neural networks (DNN), the trained weights of a DNN are a point estimate for each parameter, thereby producing deterministic network outputs for a given input. On the other hand, Bayesian neural networks (BNN) establish a probabilistic distribution over the weight parameters. The process of estimating the posterior weight distribution enables Bayesian neural networks (BNNs) to capture and quantify uncertainty in their predictions effectively (Olivier et al.,2021). Measuring uncertainty can help identify when a test input's predictions are noisy because it deviates from the training distribution or because the model cannot account for unknown causes. The point predictions generated by Deep Neural Networks (DNNs) do not provide information regarding the inherent features of the input.

The total uncertainty in the prediction, referred to as the predictive uncertainty, is comprised of both epistemic and aleatoric uncertainty (Joachims, 2021). Epistemic uncertainty refers to uncertainty within the model parameters. It is reducible uncertainty, which means that it can be reduced by collecting more data or improving the model. This can be viewed as the posterior weight distribution, in which a peaked posterior distribution reflects lower epistemic uncertainty, while a broader posterior distribution shows higher epistemic uncertainty. For the given input point and fixed weight parameters, higher aleatoric uncertainty denotes a noisy and uncertain estimate of the model's output. In these cases, the model shows a lack of confidence in its forecasts, which is indicative of significant uncertainty or variation in the evaluated results.

The implementation of a Bayesian Neural Network (BNN) involves the utilization of probabilistic (e.g., DenseVariational) layers to define the posterior and prior distributions over the weights of the model, hence enabling the estimation of uncertainty (Chang et al., 2021). This methodology enables the neural network to effectively include the inherent uncertainty present in both the data and the model, resulting in the generation of a probability distribution including a range of potential outputs, as opposed to a singular deterministic prediction. By utilizing probabilistic layers, the neural network is capable of preserving and enhancing distribution over every weight. This approach allows for a more comprehensive representation that encompasses the uncertainty and variability in both the model parameters and predictions. Consequently, it leads to the development of a more resilient and informative model, particularly in situations where there is uncertainty or limited data availability.

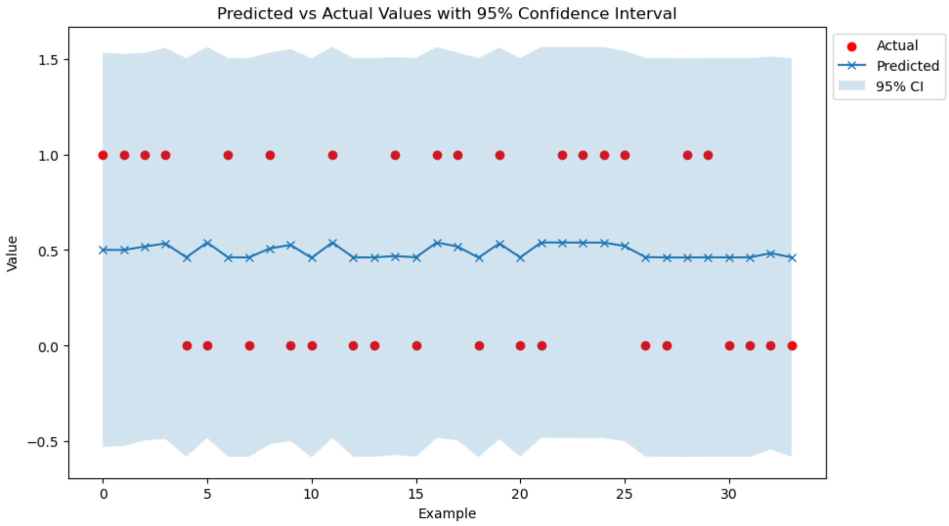

Figure 23: Predicted vs. Actual Values with 95% Confidence Interval.

Funded by
the European Union

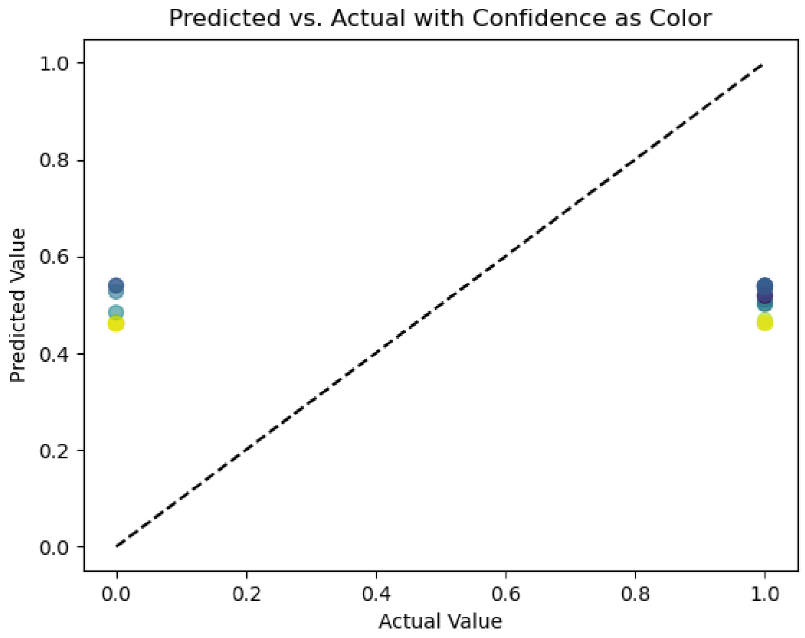

Figure 24: **Predicted vs. Actual Values with Confidence as Colour.**

Figures 23 and 24 provide informative visual representations of the prediction capabilities demonstrated by the Bayesian Neural Network (BNN). Figure 23 illustrates a comparison of predicted and actual values, comprised by a 95% confidence interval. This visual representation successfully communicates the spectrum of possible results and the underlying uncertainty in the model's predictions. The color-coded representation of the projected vs actual values in Figure 24 enables users to distinguish regions of greater or lesser certainty in the model's predictions, as determined by the level of confidence in the predictions. Together, these images offer a comprehensive picture of prediction quality and highlight areas where further investigation may be necessary due to increased levels of uncertainty.

The implementation of Automatic Relevance Detection (ARD) in a Bayesian Neural Network (BNN) involves the utilization of probabilistic layers to establish the posterior and prior distributions relevant to the weights of the network (Burden et al., 2000). Automated relevance determination (ARD) aids in the assessment of the importance of individual input features in relation to the predictive outcomes of the model. By leveraging the probabilistic layers, which maintain and optimize a distribution over each weight, the Bayesian Neural Network (BNN) can estimate the impact of the features. This approach enables the Bayesian Neural Network (BNN) to prioritize essential features, potentially simplifying the model and improving its interpretability, while maintaining the performance.

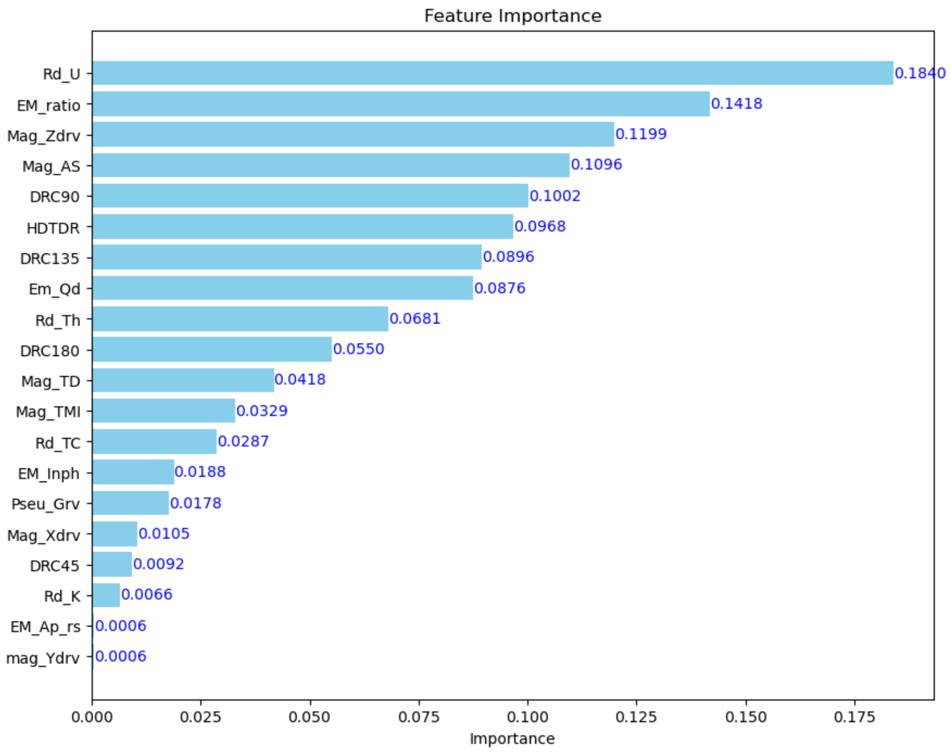

Figure 25: Automatic Relevance Detection (ARD)

Figure 25 illustrated the integration of Automatic Relevance Detection (ARD) within the Bayesian Neural Network (BNN). ARD enables the model to evaluate the significance of individual input features in relation to their impact on predictive outcomes. The BNN uses probabilistic layers to figure out which features are important. It then finds essential features that make the model easier to understand and may let it be simplified without losing its ability to make accurate predictions.

3.4. Model performance evaluation

Performance evaluation is needed for two roles: to tune the hyperparameters of the model to the best possible performance, and to evaluate how useful the model would be in the future, when it faces new situations and new data. Therefore, the project's aim and utility values of correct and incorrect predictions are involved in performance evaluation.

3.4.1 K-fold cross-validation (CV)

K-fold Cross-validation (CV) is a widely used approach in machine learning and statistical modelling for evaluating the performance of predictive models. It provides a reliable estimate of the model's ability to generalize to unseen data by repeatedly splitting the dataset into training and validation subsets (Arlot & Celisse, 2010). K-fold CV incorporates the partitioning data into K equally sized subsets, or "folds." In each iteration, K-1 folds are used for training the model while the remaining fold is used for validation. This process is repeated K times, with each fold used once as the validation set. The final performance metric is estimated by averaging the performance metrics of each iteration, which helps to reduce the risk of overfitting and improve the model's reliability (Kohav RF I, 1995).

The choice of parameter K is a vital factor that impacts the model's performance and the complexity of the process. The value of K is set to 5 or 10, as these values have been found to provide a good balance between bias and variance (James et al., 2013). However, the suitable value of K is somewhat vague; mainly small K’s are for small datasets and large K’s for a large dataset. The ultimate limitation of K is a practical computation time.

3.4.2 Stratified cross-validation (SCV)

Stratified cross-validation (SCV) is a method used in machine learning and statistical analysis to measure how well a model works and how well it can predict. This method makes sure that the proportion of each class in the dataset is kept when the data is split into training and testing sets, which makes it less likely that the results will be biased (Kohavi, 1995). SCV improves the traditional k-fold cross-validation method, in which the dataset is randomly split into k subsets of equal size. In this method, one-fold is used for testing, while the remaining K-1 folds are used for training the model. It ensures that each fold has the same class distribution as the original dataset. This is particularly important for datasets that are not balanced because it helps avoid overfitting and keeps the model from favouring the majority class (Japkowicz & Shah, 2011). By maintaining the class distribution in each fold, SCV ensures that the model learns to generalize well across all classes.

3.4.3 Leave-one-out cross-validation (LOOCV)

In machine learning and statistics, the Leave-One-Out Cross-Validation (LOOCV) approach is a special kind of cross-validation method that is used to assess the effectiveness of prediction models. It is a special case of K-fold cross-validation, where K is the number of samples in the dataset. Each observation in LOOCV is evaluated exactly once as the validation set, while the remaining datasets are used to train the model (Lachenbruch & Mickey, 1968). LOOCV works by iteratively going through each sample in the dataset, training the model on all samples except the one being validated, and testing the model on the single validation sample. Each time through the loop, the performance metrics, such as accuracy, mean squared error, and others, are calculated. One can estimate the final performance by averaging the performance across all iterations. The main benefit of LOOCV is that it has low bias, since each iteration's training set contains all samples and is a good representation of the original dataset. However, it can have a high variance as the trained models in each iteration are similar, leading to correlated performance estimates. LOOCV can be computationally expensive because it requires training the model N times, especially for large datasets (where N is the number of samples).

4. Limitations

Several functionalities and tools were not included yet in the initial release of the additional algorithms to be implemented into EIS Toolkit. For example, spatial cross-validation (SCV) was left out. SCV indicates the reliability and accuracy of the model over larger distances from the positive samples. It can indicate also possible general applicability in a continent wise scale.

Generative models and probabilistic models covering several environmental features and having an integrated multi-scale functionality are missing. These would require much more research effort, and since the main deliverable is a working EIS system, these were excluded.

An interactive functionality of dividing the shown area by the similarity (using the feature set used in the prediction, and a user-chosen norm) was left out. This kind of tool is easy to implement later to the system, though, and the main issue would be to adapt the usability to typical work cycles of geology experts.

The entire cloud-computing aspect was limited from the project. That means exclusion of a number of big data ML methods and technologies, especially true team co-operation and the large-scale DL. Implementing a single workstation application gives a chance to approach complexities of cloud computing in the future, this especially because data structure definitions, UI workflow design and IR aspect may dominate in the mineral prospectivity mapping field over the technological providence.

In the future, it is possible to include some of the above listed features to the current system or expand the system to the cloud servers, given a suitable funding situation and practical use-cases occur. The most notable of the future inclusions would be the spatial cross-validation, which has been given a special introduction in chapter 2.3.

The final release of the additional algorithms to be implemented into EIS Toolkit with be submitted in April 2024 with the Deliverable D3.6.

5. Conclusion

The presented functionalities provide an overview of how to use the developed additional ML algorithms potentially for mineral prospectivity analysis and different results visualizations. Some of the functionalities may be useful, and some less so, but this remains to be judged by the practical use of the system. There would have been many other possibilities, but the focus is on these functions. It is important that the functionalities described in this document are accessible and integrated within EIS Toolkit and will be made available in the EIS QGIS Plugin in April 2024.

Future development may use the integration interfaces and the GIS framework to introduce better and more useful services later. The next possible steps would be weighting in data availability and data costs within feature selection. One should also evaluate the usefulness of global or near global data sources, and possible benefits of adding open data features and cloud computing level to the system. Also, one needs to develop a wider selection of models that can handle complex geolocation data and improve the accuracy of mineral prospectivity mapping, especially considering the underrepresentation of positive samples and challenges of uncertainty estimation created by this sparsity.

Each ML method added has a specific demand on implementation. This means it is not a routine task to define the required work effort, but some margin has to be added, this especially when a data amount increases and need for several optimizations (better feature selection, data compression) arise.

6. References

Abdi, H., & Williams, L. J. (2010). Principal component analysis. Wiley interdisciplinary reviews: computational statistics, 2(4), 433-459.

Agresti, Alan. "Categorical data analysis. Hoboken." (2002).

Agterberg, F. P. (1989). Computer programs for mineral exploration. Science, 245, 76–81.

Agterberg, F. P., & Bonham-Carter, G. F. (1990). Deriving weights-of-evidence from geoscience contour maps for prediction of discrete events. In Proceedings of the 22nd APCOM symposium, Berlin (Vol. 2, pp. 381–395). Bonham-Carter, G. F. (1994). Geographic information systems for geoscientists: Modelling with GIS (p. 398). Oxford: Pergamon Press.

Airola, A., Pohjankukka, J., Torppa, J. et al. The spatial leave-pair-out cross-validation method for reliable AUC estimation of spatial classifiers. Data Min Knowl Disc 33, 730–747 (2019). https://doi.org/10.1007/s10618-018-00607-x

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. Statistics Surveys, 4, 40-79.

Beigaitė, R., Mechenich, M., Žliobaitė, I. (2022). Spatial Cross-Validation for Globally Distributed Data. In: Pascal, P., Ienco, D. (eds) Discovery Science. DS 2022. Lecture Notes in Computer Science(), vol 13601. Springer, Cham. https://doi.org/10.1007/978-3-031-18840-4_10

Bishop C. 1995. Neural networks for pattern recognition. New York: Oxford University Press.

Bishop, Christopher M., and Nasser M. Nasrabadi. Pattern recognition and machine learning. Vol. 4. No. 4. New York: springer, 2006.

Bonham-Carter, G. F. (1994). Geographic information systems for Porwal, A., & Carranza, E. M. J. (2015). Introduction to the special issue: GIS-based mineral potential modelling and geological data analyses for mineral exploration. Ore Geology Reviews, 71, 477–483.

Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.

Burden, Frank R., et al. "Use of automatic relevance determination in QSAR studies using Bayesian neural networks." Journal of Chemical Information and Computer Sciences 40.6 (2000): 1423-1430.

Carranza, E. J. M. (2008). Geochemical Anomaly and Mineral Prospectivity Mapping in GIS Handbook of Exploration and Environmental Geochemistry, 11, 1-425

Carranza, E. J. M. (2011). Analysis and mapping of geochemical anomalies using logratio-transformed stream sediment data with censored values. Journal of Geochemical Exploration, 110(2), 167-185.

Carranza, E.J.M., and Hale, M. (2001) Logistic regression for geologically constrained mapping of gold mineralization potential, Baguio district, Philippines. Applied Earth Science, 110, 59-70.

Chang, Daniel T. "Hybrid Bayesian Neural Networks with Functional Probabilistic Layers." arXiv preprint arXiv: 2107.07014 (2021).

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321-357.

Chen, Y., Li, J., Zhu, Q., & Zhou, K. (2020). A deep learning-based approach for automated mineral prospectivity mapping using remote sensing and geochemical data. Remote Sensing, 12(3), 375.

Chung, C. F., & Agterberg, F. P. (1980). Regression models for estimating mineral resources from geological map data. Mathematical Geology, 12, 472–488.

Cracknell, M. J., & Reading, A. M. (2014). Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information. Computers & Geosciences, 63, 22-33.

Deville, P., Linard, C., Martin, S., Gilbert, M., Stevens, F. R., Gaughan, A. E., ... & Tatem, A. J. (2014). Dynamic population mapping using mobile phone data. Proceedings of the National Academy of Sciences, 111(45), 15888-15893.

Fisher, A., Rudin, C., & Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. Journal of Machine Learning Research, 20(177), 1-81.

Granek, J. (2016). Application of machine learning algorithms to mineral prospectivity mapping (Doctoral dissertation, University of British Columbia).

Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In Proceedings of the 2005 International Conference on Advances in Intelligent Computing (pp. 878-887).

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263-1284.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (pp. 1322-1328).

Hosmer Jr, David W., Stanley Lemeshow, and Rodney X. Sturdivant. Applied logistic regression. Vol. 398. John Wiley & Sons, 2013.

Inselberg, A. (1985). The plane with parallel coordinates. The Visual Computer, 1(2), 69-91.

Inselberg, A., & Dimsdale, B. (1990, October). Parallel coordinates: a tool for visualizing multi-dimensional geometry. In Proceedings of the first IEEE conference on visualization: visualization90 (pp. 361-378). IEEE.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: with Applications in R. Springer.

Japkowicz, N., & Shah, M. (2011). Evaluating Learning Algorithms: A Classification Perspective. Cambridge University Press.

Jiang, Z., Chen, T., Chen, T., & Wang, Z. (2021). Improving contrastive learning on imbalanced data via open-world sampling. Advances in Neural Information Processing Systems, 34, 5997-6009.

Joachims, Per. Uncertainty Quantification with Bayesian Neural Networks. Diss. Humboldt-Universität zu Berlin, 2021.

Johnson, J., Alahi, A., Li, F. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. arXiv: 1603.08155

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150202.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the 14th International Joint Conference on Artificial Intelligence (Vol. 2, pp. 1137-1143).

Kohonen, T. (1982) Self-organized formation of topologically correct feature maps. Biological Cybernetics, 43, 56–69.

Lachenbruch, P. A., & Mickey, M. R. (1968). Estimation of error rates in discriminant analysis. Technometrics, 10(1), 1-11.

Luo, Z., Xiong, Y., & Zuo, R. (2020). Recognition of geochemical anomalies using a deep variational autoencoder network. Applied Geochemistry, 122, 104710.

McCullough, P., and J. A. Nelder. "Generalized linear models chapman and hall." New York (1989).

Oliveira, S., Ercan, I., & Filzmoser, P. (2019). Mineral prospectivity mapping using ensemble models based on Random Forest, Boosted Regression Trees, and Rotation Forest algorithms. Computers & Geosciences, 124, 14-24.

Olivier, Audrey, Michael D. Shields, and Lori Graham-Brady. "Bayesian neural networks for uncertainty quantification in data-driven materials modeling." Computer methods in applied mechanics and engineering 386 (2021): 114079.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

Ploton, P., Mortier, F., Réjou-Méchain, M. et al. Spatial validation reveals poor predictive performance of large-scale ecological mapping models. Nat Commun 11, 4540 (2020). https://doi.org/10.1038/s41467-020-18321-y

Rodriguez-Galiano, V., Sanchez-Castillo, M., Chica-Olmo, M., & Chica-Rivas, M. J. O. G. R. (2015). Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. Ore Geology Reviews, 71, 804-818.

Sansen, J., Richer, G., Jourde, T., Lalanne, F., Auber, D., & Bourqui, R. (2017, July). Visual exploration of large multidimensional data using parallel coordinates on big data infrastructure. In Informatics (Vol. 4, No. 3, p. 21). MDPI.

Singer, D. A., & Kouda, R. (1996). Application of a feedforward neural network in the search for Kuroko deposits in the Hokuroku district, Japan. Mathematical Geology, 28, 1017-1023.

Strobl, C., Boulesteix, A. L., Kneib, T., Augustin, T., & Zeileis, A. (2008). Conditional variable importance for random forests. BMC Bioinformatics, 9, 307.

Torppa, J. Chudsama, B., Hautala, S. & Kim, Y. (2021). GiSSOM for clustering multivariate data. GTK Open file Research Report 52/2021.

Towards Data Science. (2024). Spatial Cross-Validation Using Scikit-Learn. Retrieved from https://towardsdatascience.com/spatial-cross-validation-using-scikit-learn-74cb8ffe0ab9

Van Der Maaten, L., Postma, E., & Van den Herik, J. (2009). Dimensionality reduction: a comparative. J Mach Learn Res, 10(66-71), 13.

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3), 37-52.

Yin, B., Zuo, R., & Xiong, Y. (2022). Mineral prospectivity mapping via gated recurrent unit model. Natural Resources Research, 1-15.

Zhang, W., Luo, X., Hu, Y., & Liu, X. (2019). Deep learning-based remote sensing images of porphyry copper deposits: A case study in the East Kunlun Mountains, China. Journal of Applied Remote Sensing, 13(3), 034522.

Zhao, H., Deng, K., Li, N., Wang, Z., & Wei, W. (2020). Hierarchical spatial-spectral feature extraction with long short-term memory (LSTM) for mineral identification using hyperspectral imagery. Sensors, 20(23), 6854.

Zuo, R., Zhang, Z., Zhang, D., & Carranza, E. J. M. (2017). Evaluation of uncertainty in mineral prospectivity mapping due to missing evidence: A case study with skarn-type Fe deposits in Southwestern Fujian Province, China. Ore Geology Reviews, 81, 29-47.