minerals

MDPI

Article

A Case Study of Rock Type Prediction Using Random Forests: Erdenet Copper Mine, Mongolia

Narmandakh Sarantsatsral [1], Rajive Ganguli [1,*], Rambabu Pothina [1] and Batmunkh Tumen-Ayush [2]

[1] Department of Mining Engineering, University of Utah, Salt Lake City, UT 84112, USA; n.sarantsatsral@utah.edu (N.S.); rambabu.pothina@utah.edu (R.P.)
[2] Erdenet Mining Corporation, Erdenet 61027, Mongolia; tbatmunkh@erdenetmc.mn
* Correspondence: rajive.ganguli@utah.edu

Abstract: In a mine, knowledge of rock types is often desired as they are important indicators of grade, mineral processing complications, or geotechnical attributes. It is common to model the rock types with visual graphics tools using geologist-generated rock type information in exploration drillhole databases. Instead of this manual approach, this paper used random forest (RF), a machine learning (ML) algorithm, to model the rock type at Erdenet Copper Mine, Mongolia. Exploration drillhole data was used to develop the RF models and predict the rock type based on the coordinates of locations. Data selection and model evaluation methods were designed to ensure applicability for real life scenarios. In the scenario where rock type is predicted close to locations where information is available (such as in blocks being blasted), RF did very well with an overall success rate (OSR) of 89%. In the scenario where rock type was predicted for two future benches (i.e., 30 m below known locations), the best OSR was 86%. When an exploration program was simulated, performance was poor with a OSR of 59%. The results indicate that EMC can leverage RF models for short-term and long-term planning by predicting rock types within drilling blocks or future blocks quite accurately.

Keywords: machine learning; random forest; rock type; mining geology

Citation: Sarantsatsral, N.; Ganguli, R.; Pothina, R.; Tumen-Ayush, B. A Case Study of Rock Type Prediction Using Random Forests: Erdenet Copper Mine, Mongolia. *Minerals* **2021**, *11*, 1059. https://doi.org/10.3390/min11101059

Academic Editor: Yosoon Choi

Received: 30 August 2021
Accepted: 24 September 2021
Published: 28 September 2021

1. Introduction

Machine learning (ML) has been applied to mining and geology problems for at least two decades now [1–6]. On the mining geology side, grade estimation has been a major area of focus [7–11]. Machine learning techniques that were commonly applied were neural networks (NN) and support vector machines. Many also tried hybrid approaches [12]. In order to estimate iron ore grades at a mine, researchers [6] used an "extreme learning machine" (a feed forward NN) algorithm in combination with a "particle swarm optimization" approach. To fill the data gaps for geochemical element grades in a porphyry copper deposit, a multi-layer NN was used [13] along with a Gustafson-Kessel clustering algorithm. In a case study to generalize assay values for known and unknown sampled locations of a mineral sand deposit a hybrid NN was deployed. The combination included a trained, tested, and validated feed forward NN along with a geostatistics model [14]. In another instance, a genetic algorithm (GA) was used to train a NN [11] for predicting iron grades.

Researchers investigated methods for generalization, considering the complications typical in earth science data [2,15,16]. Addressing these issues, some researchers have used GA to split datasets properly into training and testing subsets [17,18]. To be method agnostic, recommendations were made on how data should be split to ensure proper evaluation of artificial intelligence models [19].

Some recent examples used ML to identify rock types based on machine operation data from drills (such as drill penetration rate) or other sensor data. Logistic regression, neural networks and gradient boosting were used by [20] to identify rock types based on sensor data in oil well directional drilling. Clustering and other techniques were applied to

"measurement-while-drilling" data to identify rock types in an iron ore mine [21]. Though the nature of the application is different, it is worth mentioning that some have also used machine learning to identify rock types from images [22,23].

Detecting rock types is also a focus of this paper. The large exploration drillhole database of Erdenet Copper Mine (EMC), Mongolia, is utilized in this paper to identify rock types. Traditionally, exploration databases are used primarily for grade estimation. However, rock type modeling is also undertaken in support of grade estimation, geotechnical modeling or mineral processing operations. For example, if rock type is known along with grade, it may be processed a particular way. If rock type can be estimated at depths below current operational depths, it can be used in developing future plans. Currently, rock type modeling is performed manually using visual tools.

Manual modeling performed using 3D visual tools can be difficult and time consuming. Making changes to manual models because of new data is also difficult. ML, on the other hand, not only makes the job easier but also allows incorporation of data from other sources. Therefore, the objective of this paper is to evaluate the effectiveness of using ML in modeling the rock type.

EMC, about 350 km northwest of capital city Ulaanbaatar, mines the Erdenetiin Ovoo copper porphyry deposit, one of the largest copper-molybdenum deposits in Mongolia. The deposit is hosted by an intrusive complex in the Orkhon-Selenge trough [24]. The mine, which started operations in 1978, splits the mining area primarily into four deposits, Central, Northwest, Shand and Oyut. This paper focuses only on the Northwest and Central deposits, as they are the only two deposits being mined currently.

Though the exploration holes were drilled from 1963 to 2018, the drillhole information was only recently entered into a database as part of a relatively new digitization effort at the mine. Therefore, there were several issues with the database, all of which had to be dealt with prior to starting work on this paper. The issues primarily included duplicate holes, irrelevant columns (or fields), terminology issues, missing critical values, and spelling. After cleaning, the database consisted of 2823 exploration drillholes for the Northwest and Central deposits. The total number of lithological "segments" were 90033. Segments are explained later in the paper. Four fields (or columns in tables) in the database were used in this research, three for the coordinates, and one for the rock type. As is common in exploration databases, rock types in the database are geologist's interpretation of the rock.

Figure 1 shows two views of the drillholes. Some holes were drilled from the surface before the start of operations, while other holes were drilled inside the pit. Therefore, hole lengths ranged from 28 m to 1054 m, with a median length of 75 m. About 140 holes were above 485 m in depth (95th percentile). Hole bottom elevations range from 166 m to 1505 m, with the median bottom elevation being 1310 m.

EMC uses the drillhole database to classify the main domains by lithology and fault zones. These zones are then related to mining and mineral processing conditions. Rocks are grouped into five major zones: andesite, granodiorite (GDIR), biotite granodiorite porphyry, dyke and fault zones, and finally, unknown. About 43% of the copper comes from GDIR. Therefore, the goal in this paper is to predict if the rock type in a given location is GDIR or not.

ML, as with most modeling methods, requires data to be split into modeling (or training) subset and testing subset. Usually, data is split into training and testing subsets to ensure that both subsets are similar [15]. However, a model can be developed and evaluated using different strategies to reflect the various ways it can be used in real life. Therefore, a novelty of this paper is in how data is split for modeling and evaluation. This is explained in the next section.

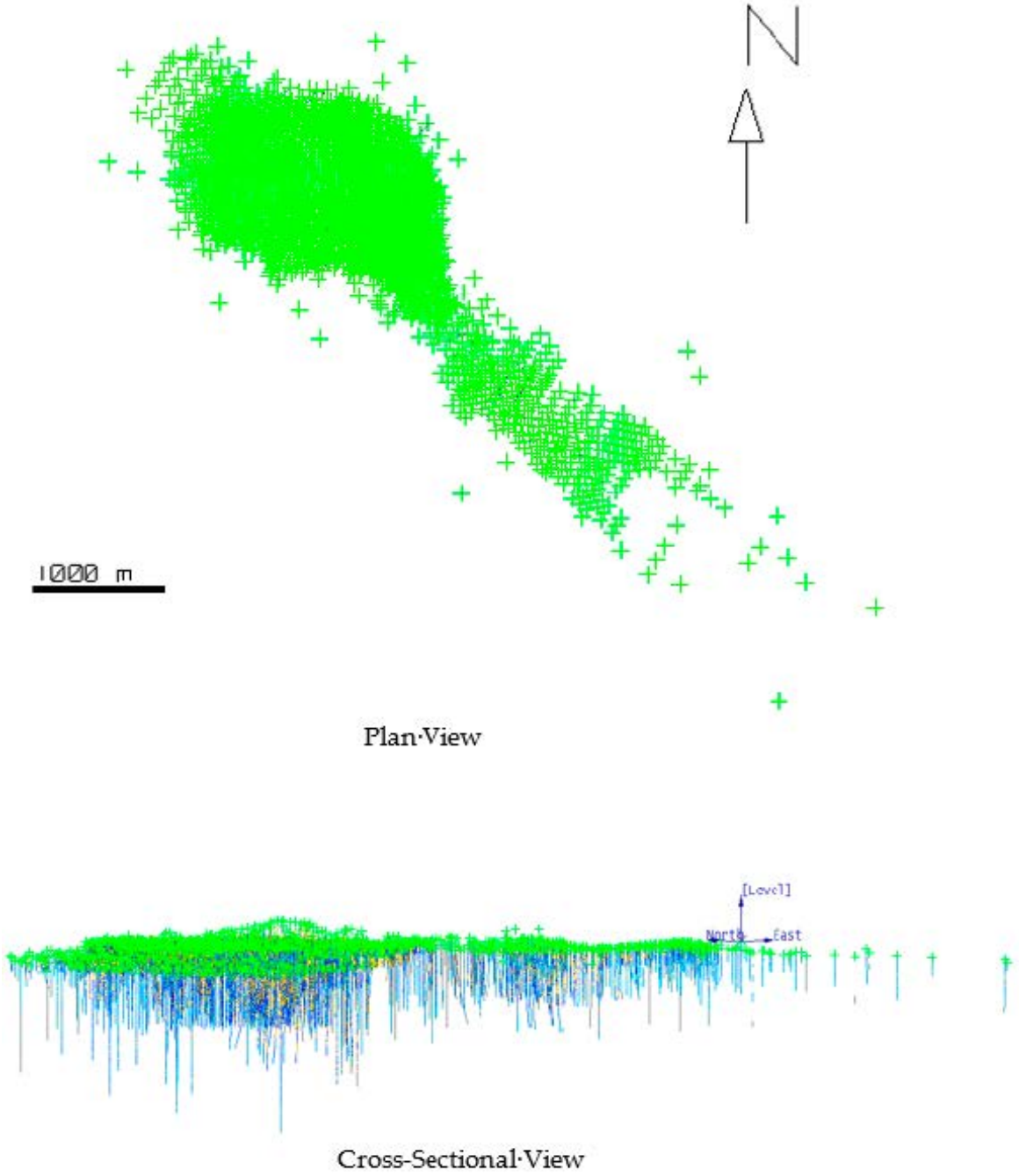

Figure 1. Plan view (top) and cross-sectional view of the 2823 drillholes. Arrow points north. Scale is shown in meters. A total of 90,033 drillhole segments are depicted.

2. Methodology

2.1. Data Selection Approaches

This paper uses two approaches for selecting data for training and testing subsets, segment-based (SB) and hole-based (HB). The reasoning for the two approaches is explained in a subsequent section.

SB and HB approaches are demonstrated using Figure 2. The figure shows a dataset consisting of four holes, H1–H4. Each hole contains several lithological segments. Segments are 5 m in thickness, except when the lithological segment is less than 5 m in thickness or not a perfect multiple of 5 m. For example, consider a granodiorite intersection of 23 m, followed by 3.5 m of diorite. The granodiorite intersection will be split into five segments

of lengths 5 m, 5 m, 5 m, 5 m, 5 m and 3 m. The diorite will be on a separate segment of 3.5 m.

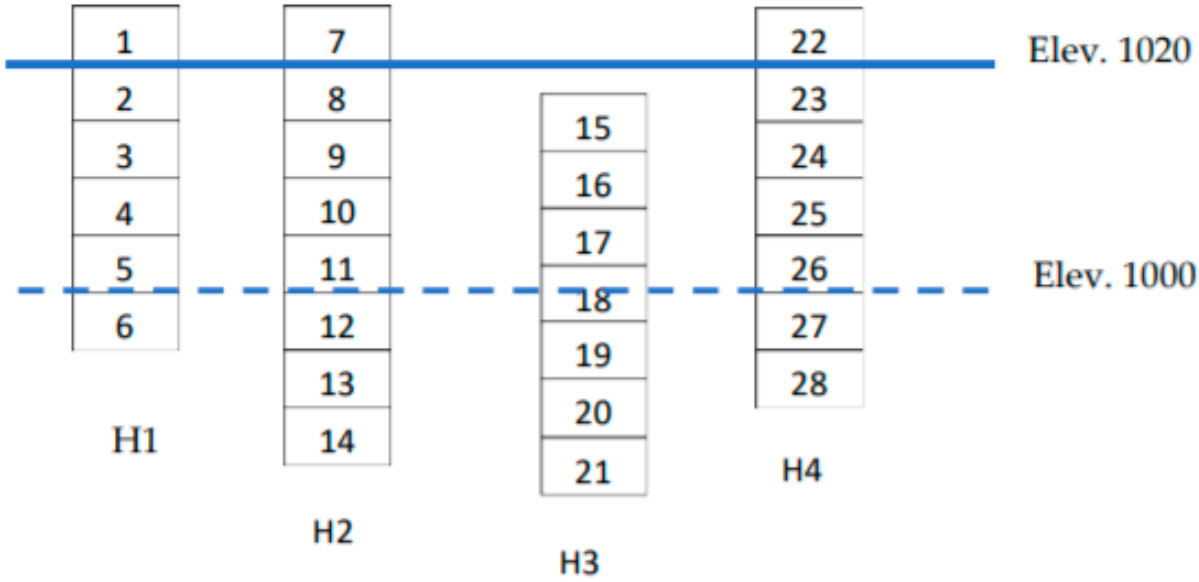

Figure 2. Cross section view of the example holes (four) containing a total of 28 lithological segments. Two lines show elevations ("Elev.") of 1020 and 1000.

In Figure 2, there are a total of 28 segments between the four holes. The figure also shows two lines that indicate two arbitrary elevations (1020 and 1000). These lines will be used later to explain additional concepts.

Assume that it is determined that 75% of the data will be selected for training. In the SB method, 21 segments are selected for training. Of course, segments are selected so that the training and testing subsets are similar in their distribution of rock types [19] or meet the real life considerations. In the SB method, each hole will likely contribute to both training and testing subsets. In the HB method, selection is made by holes and not by segments. Therefore, 75% of the holes are selected for the training subset. Each segment in the selected hole contributes only to the training subset. Segments in the other holes are all in the testing subset.

Note that regardless of method, there would be exactly 28 rows of total data in the data set. However, while the number of rows in the training subset will be 21 in the SB approach, this will be different for the HB approach. It depends on which holes are selected for training and testing subsets. For example, if H1 is sent to the testing set, the training subset would have 22 rows.

2.2. Operational Situations and Their Relationship to Evaluation Methods

In a mine, there is information about rock type in areas that are drilled. However, information is often preferred at a more granular level for operational reasons. Many times, in this scenario, there is information available close to and surrounding the non-drilled location. This operational situation is reflected in the SB strategy, where rock types are predicted at locations close to where information is available. For example, if segments 3 and 5 in Figure 2 are in the test set, they are locations close to where information is available (segments 1, 2, 4, 6). Segments are about 5 m apart. Therefore, this is similar to desiring to know the rock type in a particular production blast, since drillhole spacing in a typical blast is 5-by-5 m at EMC. Knowing the rock type has immediate operational value as it can help predict grades or mineral processing complexities.

Another situation that occurs at a mine is when information is needed for areas where hole density is sparse. This scenario is captured by the HB method. Since the holes in the test set are not known to the model, this method simulates predicting an entire drillhole

between known drillholes. The difference with SB is that the distance of testing segments from training segments is much larger in HB. In HB, when a prediction is made for a test segment, it is made based on segments (training data) that are in other holes. Since holes are 50 m or more apart, predictions are essentially for locations 50 m or more away from known data. In SB, however, predictions are made based on segments, some of which are in the same hole (perhaps as close as 5 m away). SB is thus a scenario where predictions are for locations that are near to locations with known data. Hence, SB-versus-HB is also a near-versus-far comparison.

A variant of the above scenario is when information is required at depths beyond the current drilling depth. In this situation, named "SB specific to elevation" (SBE), information is available up to a given elevation, while there is interest in knowing the rock types below this elevation. Therefore, using information up to this elevation, rock type has to be predicted for deeper locations (future benches) for short-term or long-term planning purposes. In this method, all segments above the specific elevation are in training subset, while locations deeper than that are in the test subset. To define terminology, SBE-1600-1300-30 indicates the SB evaluation method where segments between 1600 m and 1300 m elevations are part of the training subset. The "30" refers to the segments in the next 30 m of depth (1270–1300 m elevation). This 30 m forms the test set. Thus, the evaluation is occurring at 1300 m elevation, with 1600–1300 m being the training set and 1270–1300 being the test set.

In the label SBE-1600-1300-30, 1600–1300 is referred to as the training interval (TI) with a training width (TW) of 300 (1600–1300 = 300), while 30 is the evaluation width. Incidentally, the highest collar elevation is 1600 m and, therefore, when the training interval starts at 1600 m, it implies all segments up to a certain depth are included in the training subset.

One may also use Figure 2 to understand this method. When applied to Figure 2, SBE-1020-1000-5 would imply that all segments of the dataset between the thick blue line and the dashed blue line would be used in the training set. Predictions will be made for 5 m below this line, i.e., one segment below the dashed line. Note that in the dataset a segment is represented by the coordinates of its centroid. Therefore, unlike Figure 2, it is always clear whether a segment is above or below a line.

In the SB and HB strategies, training and testing subsets are selected by randomly splitting the datasets [25]. In the results section, it is shown that despite the random shuffling, the characterization of the subsets is almost identical in both strategies. In the SBE strategy, training data is everything within a particular training interval, while testing data is everything within a particular evaluation width that is just outside the training interval. Since the two subsets represent different 3D spaces, there is no reason for them to be similarly characterized. Normally, this would be an improper modeling approach. However, that concern does not apply here as the intention is to test if ML can predict just outside its training area.

The ML method used in the paper is random forest (RF). RF were used for two major reasons [26]. One, unlike geostatistics, RF do not require any assumptions on the distribution of data. Two, as explained in the section below, RF tend to generalize well. RF are not new to mining geology [27,28], but since they are not a common technique in mining they are briefly presented next.

2.3. Random Forest: Background

This paper is not intended to be a manual on random forest (RF). Those seeking a deeper understanding are referred to [29], the source for this introduction. First, a note on terminology. In machine learning terminology, 'feature' refers to a database field. A drillhole database that contains the coordinates (northing, easting, elevation) and the rock type code has four features. A RF developed to determine the rock type will then be based on three features (northing, easting, elevation).

To understand random forests, one must first understand decision trees. A decision tree is a series of yes/no questions that are used to sub-divide the samples in the training

set. A question applied to a group of data acts like a boundary, as it splits the parent group into two. The child groups can then be further split using boundaries of their own. The application of decision trees is explained through an example.

Consider the training set in Figure 3 where each sample consists of x-coordinates, y-coordinates, and a binary class indicator (1 or 0). In this example, the goal of the decision tree is to determine the class for a given (x, y) location.

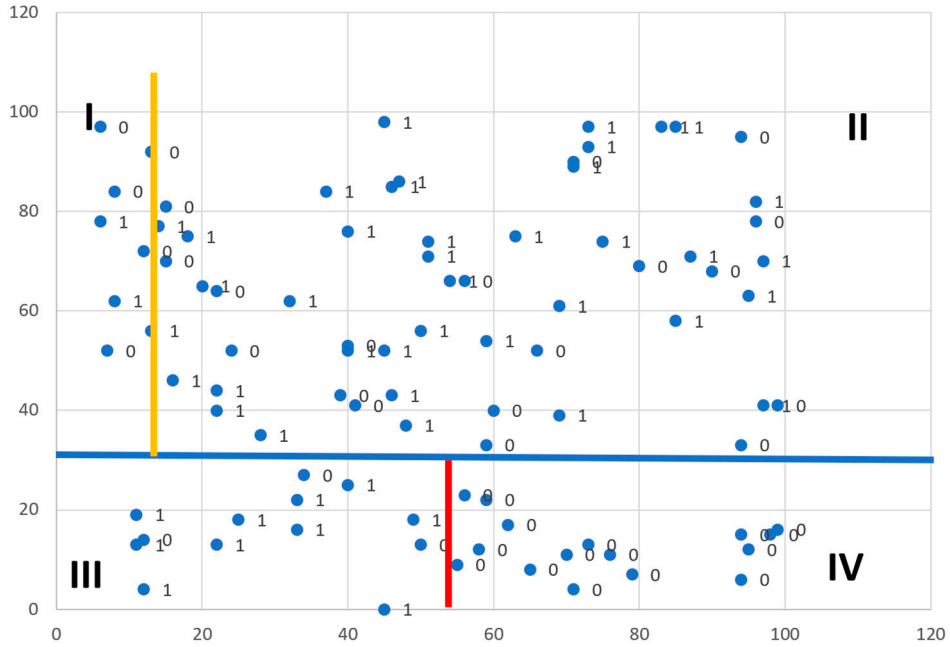

Figure 3. Example training data set showing the two classes (1 and 0) and their coordinates, x (horizontal axis) and y. The three lines shows three boundaries.

Assume that the tree starts with the blue boundary (Y > 36), splitting the data into two. The two resultant groups are further split using the red (bottom group) and yellow (top group) boundaries. The four subgroups are numbered I-IV to assist in the description. Assume that the above was the extent of the tree, and the modeler wishes to know the class for the test point (20,5). When the decision tree is applied to the point, it lands in Group III. Therefore, the class assigned to (20,5) is the class implied by the samples in Group III. Since 1's form the majority in Group III, the class assigned to (20,5) is 1. In a regression decision tree, the assigned value can be the mean or median (or any other appropriate statistic) of the group into which the point lands. In this example, any point being evaluated will face at most two boundaries. Therefore, the depth of the tree is 2. Figure 4 shows a representation of the decision tree, with the "yes" branch progressing to the left. The location at which a boundary exists is called a node, i.e., a group of data points is a node. The final nodes are also shown (I, II, III, and IV).

When a node is to be divided, one must first decide which feature to use for the boundary. In this example, two features are available to be used as a basis for dividing the boundary. The first boundary in the above example could have been on the X-axis instead of the Y-axis. The next design choice is to identify where to locate the boundary on the selected feature. In this example, the choice was to locate the first boundary at 36 (i.e., Y > 36). Most decision tree algorithms make both choices at once. If the number of features is low, one could systematically apply boundaries in all the features, and then pick the one where the resultant child groups have the least error (i.e., each node is homogenous and contains only or mostly samples from the same category). Notice how group IV contains only 0. This node can no longer be divided as it is fully homogeneous. The process of dividing nodes can continue till the final nodes are all homogenous or have at least one sample. One may also choose to limit the depth of the tree. Usually, a tree that is too deep

may not be generalized. When the number of features is large, to reduce computations, the algorithm may randomly choose a set of features to be used a basis for the boundary. Different features are then considered for different boundaries.

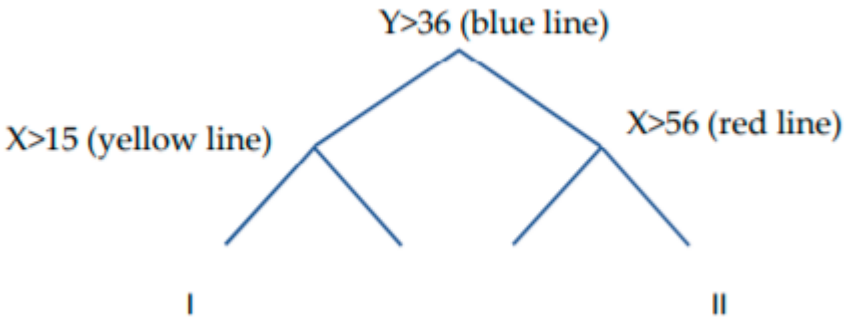

Figure 4. The example decision tree showing a tree depth of two. The labels (such as Y > 36) describe the decision boundary at a node.

In a decision tree, algorithms will generally yield the same set of boundaries for a given training set if all the features are considered for every boundary. In a random forest with N training data points, decision trees are formed by randomly selecting (with replacement) N of the training data points. Thus, the same data point may be selected many times for modeling a tree, at the cost of other data points that are not selected. Multiple trees are formed this way to make the forest. When the forest is applied to determine the category for a given test point, the decisions of the various trees in the forest are combined to form the final decision. One may use different strategies to combine the decisions. Random forests have been found to be superior to a single decision tree, with generalization not being an issue [26].

3. RF Modeling and Results

RF models were developed using the RandomClassifier() tool in scikit [30]. Only one hyper parameter was set: maximum tree depth (MTD). It was set using trial and error runs. Tree depth was increased until performance did not increase. In other words, the shortest tree depth for the highest performance was used as the setting. The task of the RF was to predict the rock class, GDIR (1) or not (0). Table 1 shows the distribution of GDIR rock type in the training and testing subsets for the various strategies. Table 2 shows the performance of the RF models for the various strategies.

Table 1. Data characterization for various evaluation strategies.

Strategy	MTD	NTrain	GDIR_Train	GDIR_Train_Prop	NTest	GDIR_Test	GDIR_Test_Prop	nonGDIR_Test
SB	20	45,016	18,696	42%	45,017	18,404	41%	26,613
SBE-1600-1300-30	25	45,603	20,872	46%	5473	2198	40%	3275
SBE-1600-1300-45	25	45,603	20,872	46%	7995	3216	40%	4779
SBE-1600-1300-60	25	45,603	20,872	46%	10,468	4230	40%	6238
SBE-1400-1300-30	25	28,531	12,744	45%	5473	2198	40%	3275
SBE-1400-1300-45	25	28,531	12,744	45%	7995	3216	40%	4779
SBE-1400-1300-60	25	28,531	12,744	45%	10,468	4230	40%	6238
SBE-1500-1200-30	25	61,589	27,411	45%	4093	1490	36%	2603

Table 1. *Cont.*

Strategy	MTD	NTrain	GDIR_Train	GDIR_Train_Prop	NTest	GDIR_Test	GDIR_Test_Prop	nonGDIR_Test
SBE-1500-1200-45	25	61,589	27,411	45%	6008	2171	36%	3837
SBE-1500-1200-60	25	61,589	27,411	45%	7786	2804	36%	4982
SBE-1300-1200-30	25	16,632	6590	40%	4093	1490	36%	2603
SBE-1300-1200-45	25	16,632	6590	40%	6008	2171	36%	3837
SBE-1300-1200-60	25	16,632	6590	40%	7786	2804	36%	4982
HB	25	45,154	18,467	41%	44,879	18,632	42%	26,247

MTD = Maximum Tree Depth; NTrain = Total rows in training subset; GDIR_Train = Number of rows in training set with GDIR; GDIR_Train_Prop = Proportion of GDIR in training subset; NTest = Total rows in testing subset; GDIR_Test = Number of rows in testing set with GDIR; GDIR_Test_Prop = Proportion of GDIR in testing subset; nonGDIR_Test = Number of rows in testing set with rocks other than GDIR

Table 2. Performance of RF models for various evaluation strategies.

Strategy	GDIR_success_num	GDIR_success_prop	GDIR False Positive	nonGDIR_success_num	nonGDIR_success_prop	OSR
SB	15,760	86%	9%	24,246	91%	89%
SBE-1600-1300-30	1584	72%	13%	2865	87%	81%
SBE-1600-1300-45	2179	68%	14%	4115	86%	79%
SBE-1600-1300-60	2758	65%	15%	5301	85%	77%
SBE-1400-1300-30	1414	64%	11%	2909	89%	79%
SBE-1400-1300-45	1939	60%	13%	4175	87%	76%
SBE-1400-1300-60	2444	58%	14%	5376	86%	75%
SBE-1500-1200-30	1209	81%	12%	2302	88%	86%
SBE-1500-1200-45	1704	78%	13%	3353	87%	84%
SBE-1500-1200-60	2146	77%	14%	4304	86%	83%
SBE-1300-1200-30	1415	95%	71%	763	30%	53%
SBE-1300-1200-45	2053	95%	71%	1100	29%	52%
SBE-1300-1200-60	2656	95%	71%	1424	29%	52%
HB	7756	42%	29%	18727	71%	59%

GDIR_success_num = Number of GDIR test rows successfully classified; GDIR_success_prop = Proportion of GDIR test rows successfully classified (100 × GDIR_success_num/ GDIR_Test); nonGDIR_success_num = Number of non-GDIR test rows successfully classified; nonGDIR_success_prop = Proportion of non-GDIR test rows successfully classified (100 × nonGDIR_success_num/non-GDIR_Test);

The results demonstrate the following:

- The proportion of GDIR in the training and testing subsets depend on the evaluation strategy.
    - In SB and HB, despite random shuffling, GDIR is split about evenly between training and testing subsets. This similarity between training and testing subsets is appropriate as both represent the same 3D space.
    - In the SBE strategies, the training subsets are much larger than the testing subsets, since the training interval (e.g. 1600–1300 implies a 300 m training interval) is much larger than the evaluation widths (e.g. 30 m). Since the two

subsets represent completely different 3D spaces, the proportion of GDIR and non-GDIR in the two subsets can be quite different.

- SBE models were developed for elevations of 1300 and 1200 m, as the mine is currently operating approximately between those levels.
- RF performs quite well in the SB strategy. 81% of GDIR in the test subset is detected, while 90% of non-GDIR is detected. The overall success rate (OSR) was 87%, i.e., 87% of the rocks are recognized correctly as GDIR or non-GDIR.
- In the SBE strategy (also see Figure 5):
  - Notice how the performance lines in Figure 5 are inclined downwards to the right. In each scenario, the performance falls as the evaluation width increases from 30 m to 60 m. This is not surprising, as a larger evaluation width tests space farther away from the modeling space.
  - The overall accuracy is higher for higher training intervals (Figure 6). Thus, at 1300 m, 1600–1300 (training interval = 300) outperforms 1400–1300 (training interval = 100). Similarly, at 1200 m, 1500–1200 outperforms 1300–1200. The effect is more pronounced at 1200 m elevation.
  - The seemingly flawless performance for SBE-1300-1200 is misleading (Table 2, column GDIR_success_prop). The ability to classify 95% of the GDIR rock type as GDIR is paired with a 71% false positive rate. In other words, the classification of rock as GDIR is unreliable. This strategy classifies most segments as GDIR. Though that results in capturing all the GDIR, it also ends up classifying non-GDIR as GDIR. This is seen in the low success rate for classifying non-GDIR.
- The false positive rate of 9–15% (for most cases) is decent. This means that when a rock is classified as GDIR, it is most likely to be GDIR.
- HB strategy showed that predicting entire holes is difficult. When a hole is hidden in its entirety, only 42% of the GDIR rock segments in the hole are classified accurately. This is accompanied by a 29% false positive rate, which is not good.

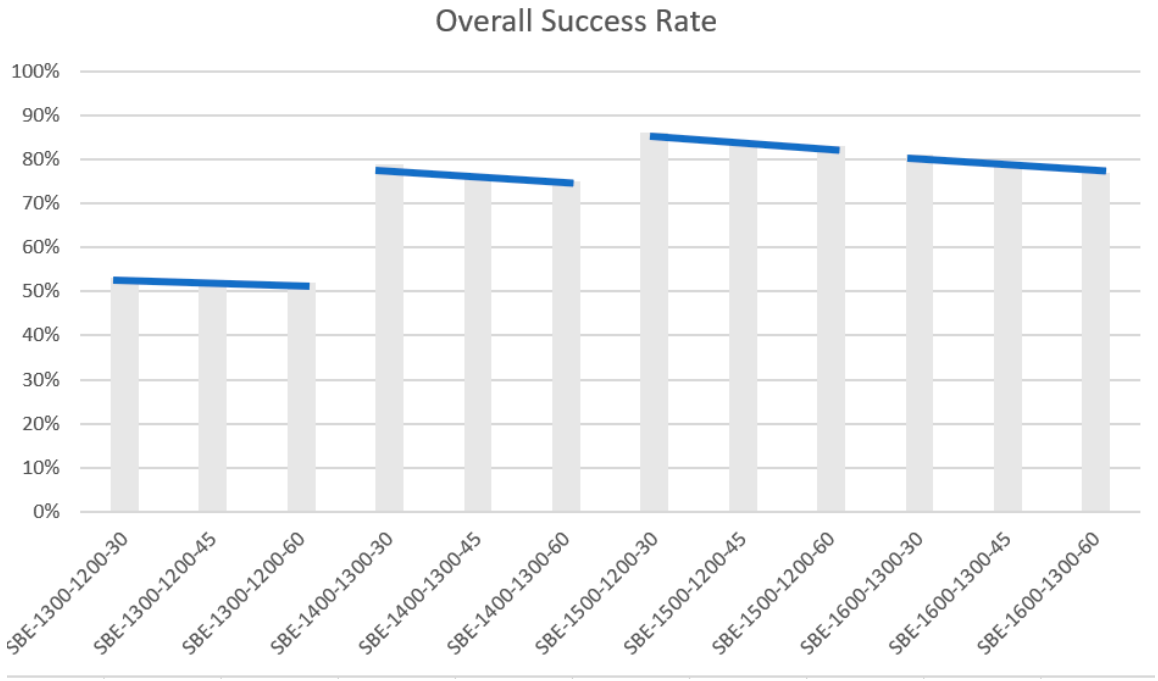

Figure 5. The overall success rate for the SBE strategy at each of the elevations, for different evaluation widths. Each blue line represents performance at a particular elevation. At each depth, performance falls as evaluation width increases. OSR = Overall success rate =100 × (GDIR_success_num + nonGDIR_success_num)/NTest.

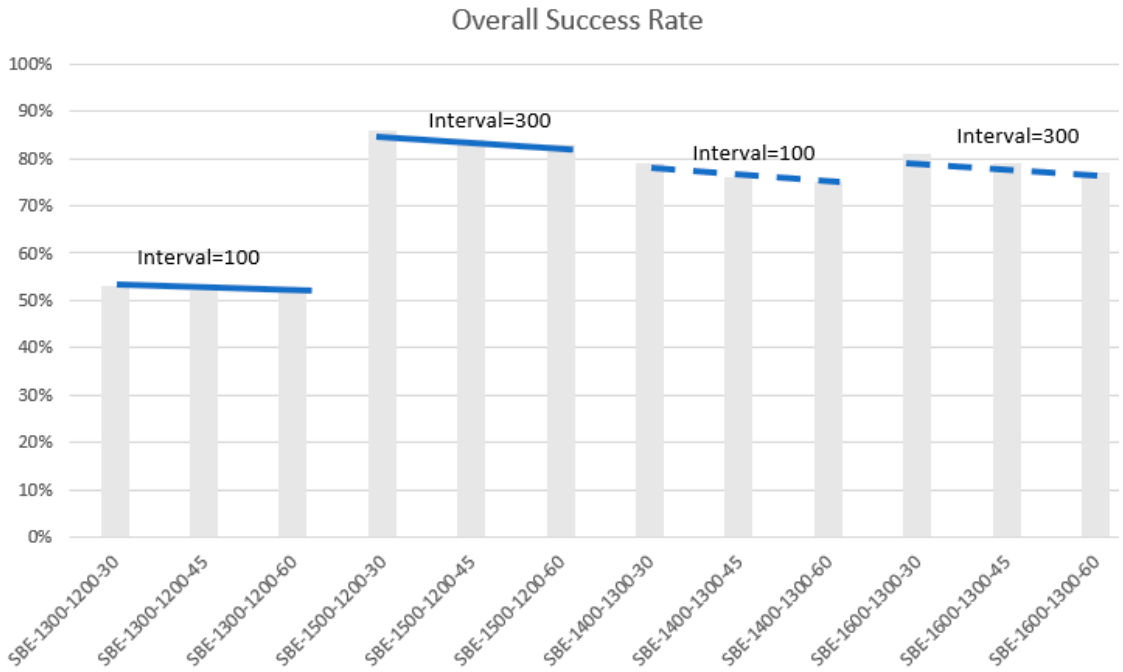

Figure 6. The overall success rate for the SBE strategy arranged by training width. Each line represents a particular elevation, with dashed lines representing 1300 m, while thick line representing 1200 m elevation.

4. Discussion

Most mining operations either use the manually developed rock type models or sensor technologies to make assumptions on the rock types contained within a drill block, or in future benches/blocks. This paper tested ML algorithms as an alternative to both approaches.

The SB strategy demonstrated that given a good density of information, the gaps can be predicted with high accuracy. This would suggest that ML of existing information may be a good substitute for using technologies to detect rock types, when information is available for nearby locations.

The SBE strategies demonstrated that mine planning can benefit from ML. Erdenet Copper Mine, with a bench height of 15 m, can predict rock type two to three benches below the current depth with significant reliability.

The HB strategy demonstrated that RF machine learning cannot yet replace a drilling campaign. The HB strategy simulated data sparsity. Without data density, ML can have problems. A research team [31] cited inadequate data as the reason for overfitting when applying neural networks to estimate grades based on sample locations, lithological features and alteration levels. Another team [28] cited data density as a concern when applying RF for mineral prospectivity mapping.

Despite the mixed results, there are advantages to using RF. Unlike geostatistics, no assumptions are made about the statistical characterization of drillhole data. However, RF performs about as well as geostatistics [32]. Performance aside, geostatistical methods take advantage of spatial relationships as defined by variograms. RF does not explicitly take advantage of spatial relationships. The K-nearest neighbor machine learning technique [33], which is a version of the common inverse distance squared technique in geostatistics, does take distances into consideration. However, it is not a sophisticated algorithm. It is possible that by incorporating spatial relationships such as variograms, RF or other machine learning techniques may perform better. This would be an excellent topic for future research, would be along approaches being attempted in recent times [18].

5. Conclusions

The machine learning technique random forest was applied to the exploration drillhole database at Erdenet Copper Mine in Mongolia to predict the presence of rock type

granodiorite. Granodiorite is an important rock type at the mine as it contains 43% of the copper. The data consisted of 90,033 drillhole segments from 2823 drillholes. Most segments were 5 m in thickness. Two data selection approaches, segment-based and hole-based, were utilized to ensure that models could be tested to align with real life needs. Models were developed to test for three operational scenarios. The base SB method tested for the scenario when rock type is predicted at locations close to where rock types are known. This simulates the typical block that is blasted as part of day-to-day operation, where rock type is known in a relatively dense grid. The base HB method tested for the scenario where rock type is unknown for the entire length of a drillhole in between other drill holes. The SBE method tested for the scenario where rock type is known up to a given elevation but is unknown beyond that elevation. In the SBE method, rock types were predicted for 30, 45 and 60 m (evaluation width) beyond a specific elevation. The information made available to the models in the SBE method, or the training interval, varied from 100 m to 300 m. Given the 15 m benches at the mine, the 30, 45 and 60 m evaluation widths implied predictions to 2, 3 and 4 benches below where rock types were known.

The models performed very well in the SB scenario, with 86% of granodiorite being predicted accurately, with a false positive rate of 9%, resulting in an overall accuracy level of 89%. In the SBE method, the overall accuracy varied from 52% to 86%. Performance was better for higher training intervals, and for shorter evaluation widths. Performance was best in the SBE method at 1200 m, i.e., rock type was predicted better at 1200 m than at other elevations. The highest performance was achieved at 1200 m elevation with a training interval of 300 m and evaluation width of 30 m. The performance in the HB method was not encouraging, with an overall success rate of 59%.

This paper demonstrated that random forest-based machine learning can be very effective for predicting rock types in near distances. Predicting the entire length of a missing drillhole is, however, another story. The good performance of near-distance predictions should prompt mines to perhaps switch to machine learning over traditional manual modeling (or imperfect sensor technologies) to predict rock types in ore blocks blasted for production.

Author Contributions: Conceptualization, R.G.; data curation, N.S. and B.T.-A.; formal analysis, N.S, R.G.; funding acquisition, R.G.; investigation, N.S, R.G.; methodology, N.S, R.G, R.P; validation, N.S., R.G., R.P., B.T.-A.; visualization, N.S., R.G., R.P. and B.T.-A.; writing—original draft, N.S., R.G., R.P., and B.T.-A.; writing—review & editing, N.S., R.G., R.P. and B.T.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was performed as part of an agreement between the University of Utah and Erdenet Mining Corporation.

Data Availability Statement: Not applicable.

Acknowledgments: The help of the geologists at EMC is gratefully acknowledged.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Dutta, S.; Ganguli, R.; Samanta, B. Investigation of Two Neural Network Methods in an Automatic Mapping Exercise. *Appl. GIS* **2005**, *1*, 1–19. [CrossRef]
2. Dutta, S.; Bandopadhyay, S.; Ganguli, R.; Misra, D. Machine Learning Algorithms and Their Application to Ore Reserve Estimation of Sparse and Imprecise Data. *J. Intell. Learn. Syst. Appl.* **2010**, *2*, 86–96. [CrossRef]
3. Yu, S.; Ganguli, R.; Bandopadhyay, S.; Patil, S.L.; Walsh, D.E. Calibration of online ash analyzers using neural networks. *Min. Eng.* **2004**, *56*, 99–102.
4. LaBelle, D. *Lithological Classification by Drilling*; Carnegie Mellon University: Pittsburgh, PA, USA, 2001.
5. Wu, X.; Zhou, Y. Reserve estimation using neural network techniques. *Comput. Geosci.* **1993**, *19*, 567–575. [CrossRef]
6. Fathi, M.; Alimoradi, A.; Hemati Ahooi, H.R. Optimizing Extreme Learning Machine Algorithm using Particle Swarm Optimization to Estimate Iron Ore Grade. *J. Min. Environ.* **2021**, *12*, 397–411. [CrossRef]

7. Samanta, B.; Bandopadhyay, S.; Ganguli, R.; Dutta, S. A comparative study of the performance of single neural network vs. Adaboost algorithm based combination of multiple neural networks for mineral resource estimation. *J. S. Afr. Inst. Min. Metall.* **2005**, *105*, 237–246.
8. Samanta, B.; Bandopadhyay, S.; Ganguli, R.; Dutta, S. An Application of Neural Networks to Gold Grade Estimation in Nome Placer Deposit. *J. S. Afr. Inst. Min. Met.* **2005**, *105*, 237–246.
9. Chatterjee, S.; Bandopadhyay, S.; Ganguli, R.; Bhattacherjee, A.; Samanta, B.; Pal, S.K. General regression neural network residual estimation for ore grade prediction of limestone deposit. *Min. Technol.* **2007**, *116*, 89–99. [CrossRef]
10. Tahmasebi, P.; Hezarkhani, A. A hybrid neural networks-fuzzy logic-genetic algorithm for grade estimation. *Comput. Geosci.* **2012**, *42*, 18–27. [CrossRef]
11. Mahmoudabadi, H.; Izadi, M.; Menhaj, M.B. A hybrid method for grade estimation using genetic algorithm and neural networks. *Comput. Geosci.* **2009**, *13*, 91–101. [CrossRef]
12. Jafrasteh, B.; Fathianpour, N. A hybrid simultaneous perturbation artificial bee colony and back-propagation algorithm for training a local linear radial basis neural network on ore grade estimation. *Neurocomputing* **2017**, *235*, 217–227. [CrossRef]
13. Jahangiri, M.; Ghavami Riabi, S.R.; Tokhmechi, B. Estimation of geochemical elements using a hybrid neural network-Gustafson-Kessel algorithm. *J. Min. Environ.* **2018**, *9*, 499–511. [CrossRef]
14. Jalloh, A.B.; Kyuro, S.; Jalloh, Y.; Barrie, A.K. Integrating artificial neural networks and geostatistics for optimum 3D geological block modeling in mineral reserve estimation: A case study. *Int. J. Min. Sci. Technol.* **2016**, *26*, 581–585. [CrossRef]
15. Dutta, S.; Misra, D.; Ganguli, R.; Samanta, B.; Bandopadhyay, S. A hybrid ensemble model of kriging and neural network for ore grade estimation. *Int. J. Min. Reclam. Environ.* **2006**, *20*, 33–45. [CrossRef]
16. Ganguli, R. A critical review of on-line quality analyzers. *Miner. Resour. Eng.* **2001**, *10*, 435–444. [CrossRef]
17. Samanta, B.; Ganguli, R.; Bandopadhyay, S. Comparing the predictive performance of neural networks with ordinary kriging in a bauxite deposit. *Min. Technol.* **2005**, *114*, 129–139. [CrossRef]
18. Samanta, B.; Bhattacherjee, A.; Ganguli, R. A genetic algorithms approach for grade control planning in a bauxite deposit. In *Proceedings of the 32nd International Symposium on the Application of Computers and Operations Research in the Mineral Industry*; APCOM: Tucson, AZ, USA, 2005.
19. Ganguli, R.; Dagdelen, K.; Grygiel, E. Systems engineering. In *Mining Engineering Handbook*; Darling, P., Ed.; Society for Mining, Metallurgy and Exploration, Inc.: Littleton, CO, USA, 2011.
20. Klyuchnikov, N.; Zaytsev, A.; Gruzdev, A.; Ovchinnikov, G.; Antipova, K.; Ismailova, L.; Muravleva, E.; Burnaev, E.; Semenikhin, A.; Cherepanov, A.; et al. Data-driven model for the identification of the rock type at a drilling bit. *J. Pet. Sci. Eng.* **2019**, *178*, 506–516. [CrossRef]
21. Zhou, H.; Hatherly, P.; Monteiro, S.T.; Ramos, F.; Oppolzer, F.; Nettleton, E.; Scheding, S. Automatic rock recognition from drilling performance data. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA; 2012; pp. 3407–3412.
22. Koch, P.-H.; Lund, C.; Rosenkranz, J. Automated drill core mineralogical characterization method for texture classification and modal mineralogy estimation for geometallurgy. *Miner. Eng.* **2019**, *136*, 99–109. [CrossRef]
23. Sinaice, B.; Owada, N.; Saadat, M.; Toriya, H.; Inagaki, F.; Bagai, Z.; Kawamura, Y. Coupling NCA Dimensionality Reduction with Machine Learning in Multispectral Rock Classification Problems. *Minerals* **2021**, *11*, 846. [CrossRef]
24. Gerel, O.; Munkhtsengel, B. Erdenetiin Ovoo Porphyry Copper-Molybdenum Deposit in Central Mongolia. In *Super Porphyry Copper & Gold Deposits: A Global Perspective*; Porter, T.M., Ed.; PGC Publishing: Adelaide, Australia, 2004.
25. Humphries, G.W.; Magness, D.R.; Huettmann, F. *Machine Learning for Ecology and Sustainable Natural Resource Management*; Springer: New York, NY, USA, 2018; ISBN 9783319969763.
26. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
27. Jafrasteh, B.; Fathianpour, N.; Suárez, A. Comparison of machine learning methods for copper ore grade estimation. *Comput. Geosci.* **2018**, *22*, 1371–1388. [CrossRef]
28. McKay, G.; Harris, J.R. Comparison of the Data-Driven Random Forests Model and a Knowledge-Driven Method for Mineral Prospectivity Mapping: A Case Study for Gold Deposits Around the Huritz Group and Nueltin Suite, Nunavut, Canada. *Nat. Resour. Res.* **2016**, *25*, 125–143. [CrossRef]
29. Mitchell, T.M. *Machine Learning*; McGraw-Hill: New York, NY, USA, 1997; Volume 45, ISBN 0070428077.
30. Scikit-Learn. Scikit-Learn: Machine Learning in Python. *Scikit-Learn*. 2020. Available online: https://scikit-learn.org/stable/ (accessed on 15 July 2021).
31. Kaplan, U.; Topal, E. A New Ore Grade Estimation Using Combine Machine Learning Algorithms. *Minerals* **2020**, *10*, 847. [CrossRef]
32. Hengl, T.; Nussbaum, M.; Wright, M.; Heuvelink, G.; Gräler, B. Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables. *PeerJ* **2018**, *6*, e5518. [CrossRef] [PubMed]
33. Scikit-Learn. Nearest Neighbors. Available online: https://scikit-learn.org/stable/modules/neighbors.html#neighbors (accessed on 15 September 2021).