Journal of Geomine (2023), 1(3), 123-136
DOI: 10.22077/jgm.2024.7358.1018

# Application of Supervised Machine Learning Inversion in The Estimation of Iron Ore Grade from Geophysical Data: Comparative Investigation of GB, RF and SVM Algorithm

Mohsen Simorgh[1], Andisheh Alimoradi[1], Hamidreza Hemmati Ahooi[1], Mohammad Salsabili[2], Mahdi Fathi[3], Hassanreza Ghasemi Tabar[1], Parisa Rezakhani[1]

Recived: 2024 Feb. 28, Revised: 2024 Apr. 24, Online Published: 2024 Apr. 26

ABSTRACT

Magnetometry is one of the geophysical methods used to explore metal deposits, especially iron deposits and magnetite minerals. The two-dimensional model resulting from the magnetometric operation cannot estimate the grade in the depth of the deposit, so in this article, the attempt is made by using the magnetic outputs obtained after the magnetic survey operation and the two-dimensional model designed with the help of the data extracted from the borehole which is available in the studied area, and combining this information and obtaining relationships between them with the help of artificial intelligence, a three-dimensional numerical model can be obtained that can be generalized to other points that lack depth data. Reduction of human errors and lack of prior knowledge of mine geological structures is one of the capabilities of this method. This method will be a new approach to numerical simulation in the field of investigation of mineral masses. Finally, in the studied area of the Sechahoon deposit in central Iran, high precision was achieved in the ratio of zero iron grade data in the methods of Gradient Boosting and Random Forest. Also, the results of these two algorithms showed that the Maximum Mean Square Error (MSE) and Mean Absolute Error (MAE) in the training data are 0.007 and 0.05, respectively, and in the test data are 0.03 and 0.11, respectively, which these parameters reached the maximum of 0.03 and 0.1 in the inspection of validation boreholes.

KEYWORDS

Magnetometry, Gradient Boosting, Random Forest, Support Vector Machines (SVMs), Three-dimensional Modeling, Iron Deposit

## I. Introduction

One of the basic principles in mineral exploration activities will be the issue of cost reduction in operations. The magnetometry method is one of the cheapest and fastest geophysical methods, which is used in most mineral potentials, especially in iron-rich areas, and in this sense, it is ahead of other geophysical methods (Keary et al., 2002).

With the help of new and practical geophysical methods, it is possible to reduce the operating time, and reduce the burden of false costs, and also by relying on the necessary science and experience and the appropriate adaptation of data, the validity of the results can be increased. Therefore, efforts will be made to design and go through the optimal process using appropriate methods and data analysis through artificial intelligence. The application of artificial intelligence and its sub-branches in geosciences and mineral mass modeling has expanded in recent decades. By using artificial intelligence algorithms, it will be possible to estimate more geomagnetic data. Artificial intelligence is a branch of computer science that examines the practical computing requirements such as perception, reasoning, and learning and provides a system to perform such operations. Artificial intelligence is the study of how to make computers do things that humans can do better, in fact, it is the study of methods to turn a computer into a machine that can perform actions performed by humans. The methods and techniques of artificial intelligence have been created to solve those problems that could not be easily solved by mathematical methods (Kapageridis et al., 1999).

Machine learning, as a subset of artificial intelligence, is the study of algorithms and statistical models used in computer systems that use experimental mathematical patterns and inferences to perform calculations instead of specific instructions (Mitchell, 1997).

Python was designed in the late 1980s by Guido van Rossum, with a dynamic system and an emphasis on readability and rapid prototyping. Python is currently the most preferred programming language for scientific computing, data science, and machine learning, and it increases performance and productivity by using low-level libraries and convenient APIs.

[1] Department of Mining Engineering, Imam Khomeini International University, Iran, [2] Department des Sciences Appliquees, Universite du Quebec a Chicoutimi, [2] Kavoshgaran Consulting Engineers, Iran
✉ A. Alimoradi: alimoradi@eng.ikiu.ac.ir

The most important strengths of this programming language are (Raschka et al., 2020):

- Convenient while ensuring computational efficiency
- Creating efficient libraries with lower-level code than other programming languages
- Parallel processing of operations
- Free and available programming language
- Portability between different operating systems.

A lot of research has been done in recent years to model and optimize various parameters based on geophysical information with the help of artificial intelligence methods. For example, in 2009 Yuan Sanyi and colleagues optimized the inversion of three models of linear geophysical data sensitive to noise, the combination of linear and nonlinear, and nonlinear with the help of Particle Swarm Optimization and Ant Colony Optimization (Yuan et al., 2009). Alimoradi and his colleagues used a backpropagation network with 4 middle layers to model the inversion of the magnetometric data of a dyke (Alimoradi et al., 2011). In another research with the same theme, with the help of a perceptron artificial neural network with two middle layers, the modeling of the electro-seismic data of wells has been done (Ardjmandpour et al., 2011). FitzGerald tried to establish a relationship between airborne geophysical data and identify subsurface structures of the earth by using an artificial intelligence feature extraction technique (FitzGerald, 2019). In another research in 2022, high-quality modeling of subsurface geological structures was achieved using UAV magnetometer data and Deep Learning algorithms (Mukherjee et al., 2022). Bayati and colleagues, using three-dimensional modeling of magnetometric data of an iron deposit, estimated its reserve (Bayati et al., 2022).

Alimoradi evaluated the silver grade of the Zarshuran gold mine using drill spatial data, data from the Induced Polarization (IP) geophysical approach, and the cuckoo search machine learning algorithm. The findings indicate that degree values can be accurately approximated using geophysical data, particularly in locations without drilling data (Alimoradi et al., 2020).

Ghasemi Tabar and his colleagues have performed several algorithms, including Random Forest and Gradient Boosting on the northern anomaly of Choghart under the Python programming language (Ghasemi et al., 2024).

The SVM is a growingly popular learning method based on statistical learning theory developed by Vapnik and his colleagues (Vapnik, 1998). The SVM is based on two key ideas. The first is an optimum margin classifier, which is a linear classifier that creates a separating hyperplane (decision surface) with the maximum distance between positive and negative samples. In the context of assessing seismic liquefaction potential from a database of case records, the problem can be viewed as a binary classification task that distinguishes between positive (liquefied) and negative (non-liquefied) examples. The second major idea is the application of kernel functions. A kernel is a function that computes the dot product between two vectors. By applying a proper nonlinear kernel mapping to the original example data, the data can become linearly separable in a high-dimensional feature space, despite not being separable in the original input space (Cristiani and Shawe-Taylor, 2000; Platt, 1998).

Random Forest or Random Decision Forests is a hybrid learning method for classification, and regression, which is based on a structure consisting of many decision trees, on the training time and the output of classes (classification), or for the average predictions of each tree. They work separately (Fukunaga and Hostetler, 1975). Random Forests are suitable for decision trees that undergo pre-fitting in the training set. Also, this algorithm is very user-friendly and has only two input parameters of the network, which are the number of trees and the number of variables of subsets, which usually the response of the network is not highly sensitive to the value of these parameters (Yizong, 1995).

A decision tree is an algorithm that is easy to understand and interpret, but a single tree may not be enough to learn the features of the model. On the other hand, the random forest algorithm (Fig. 1) is a tree-based algorithm that uses the features of several decision trees to make decisions. Also, the decision tree algorithm is highly vulnerable in terms of overtraining and overfitting, but this problem can be easily overcome by implementing random forest regression. Another important feature of this algorithm in regression is its low variance. In fact, this algorithm uses averaging to improve performance and control overfitting (Dorin and Meer, 2002).

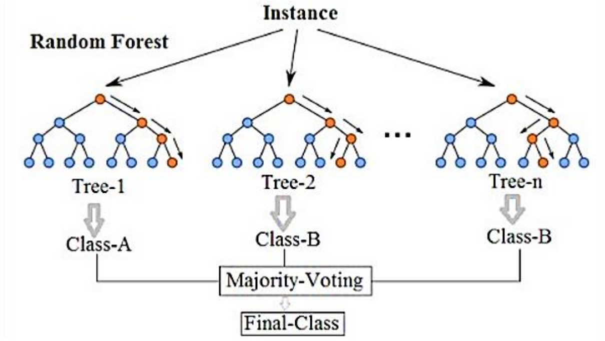

**Fig. 1.** Random Forest performance procedure and merging the results of multiple decision trees

Gradient Boosting algorithm is a machine learning method for regression and classification problems that are generated from a prediction model in the form of a set of weak prediction models, usually a decision tree. It builds the model in a stepwise manner like other booster methods and generalizes the variable function of the decision tree by allowing arbitrary optimization. The

Gradient Boosting algorithm is an integrated algorithm with high performance and stability, and it can handle noisy data well and has a high estimation ability in predicting non-linear data (Ho, 1995).

The training process and progress of the enhanced gradient algorithm based on the error function and repetition of the training process are shown in Fig. 2.

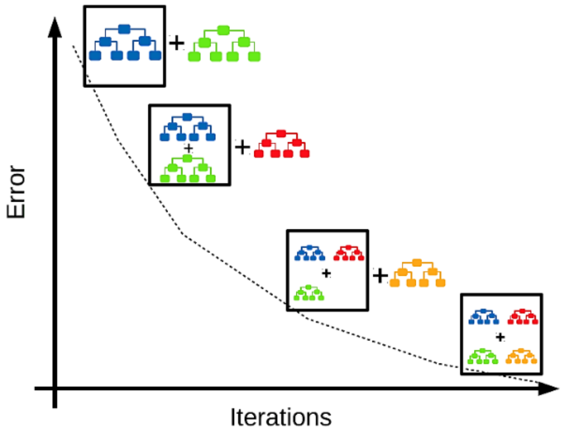

**Fig. 2**. Gradient Boosting training process (Breiman, 2001)

The Gradient Boosting algorithm is a decision tree-based algorithm that was created by developing the Ada Boost method. “Boosting” in these methods means strengthening weak algorithms and turning them into strong algorithms. The correct understanding of the error function depends on the parameter we are trying to optimize. One of the most important features of this method is that it allows the user to determine the error function according to his needs (Dietterich, 2000).

The main difference between this method and the Random Forest method is that in the Random Forest, the trees form a network next to each other, but in this method, the decision trees form a network in line with each other. This is fully visible in Fig. 3 (Boehmke and Greenwell, 2019).

## II. Sechahoon Deposit

Sechahoon Mine is one of the deposits in central Iran, which is located 45 km northeast of Bafq city. The Sechahoon Mine consists of two anomalies with the numbers X and XI, and the X anomaly is located 3 km from the XI anomaly. From the geological point of view, it belongs to the Infracambrian and sedimentary volcanic rocks of the series known as Rizo.

This series of layers is formed of volcanic rocks and basaltic and acid tuffs in the upper parts, and the lower part is mainly limestone of destructive sediments. Mineral masses are located both in volcanic rocks (such as anomaly XI) and at the border of volcanic-sedimentary rocks (such as anomaly X). In Central Iran is located in the Alpine-Himalayan orogeny system that developed during the closing of the Paleo-Tethys Ocean. This area, along with other areas of the Cimmerian block in Iran (Alborz and Sanandaj – Sirjan), is located in the northeastern part of the Zagros – Makran rift connected to the Neo-Tethys Ocean rift. The sub-continents of Central and Eastern Iran can be described separately with fault boundaries, which include three crustal domains of the Lut, Tabas, and Yazd blocks with a north-south orientation, which are adjacent to each other from east to west respectively. The blocks of Tabas and Yazd are separated by an arched and complex structural belt with a long extension as the Kashmar – Kerman tectonic zone, which is also called the Posht-e-Badam block (Fig. 4) (Fathi et al., 2021).

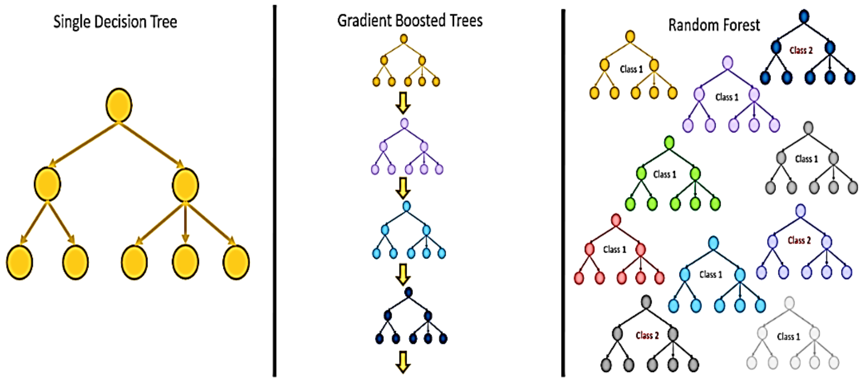

**Fig. 3**.The difference between the Random Forest and Gradient Boosting methods (Boehmke and Greenwell, 2019)

**Table 1.** Types of reserves of Sechahoon deposit – Northern mass

| Class | Volume ($m^3$) | Tonnes (ton) | Density | Fe % | FeO % | P % | Tonnes (%) |
|---|---|---|---|---|---|---|---|
| Measured | 6,315,687 | 21,619,001 | 3.423 | 36.487 | 14.879 | 0.062 | 27% |
| Indicated | 16,461,648 | 54,341,369 | 3.301 | 32.531 | 13.611 | 0.071 | 69% |
| Inferred | 1,003,806 | 3,247,205 | 3.235 | 30.525 | 10.609 | 0.119 | 4% |
| Sum\Avg | 23,781,141 | 79,207,575 | 3.320 | 33.528 | 13.834 | 0.07 | 100% |

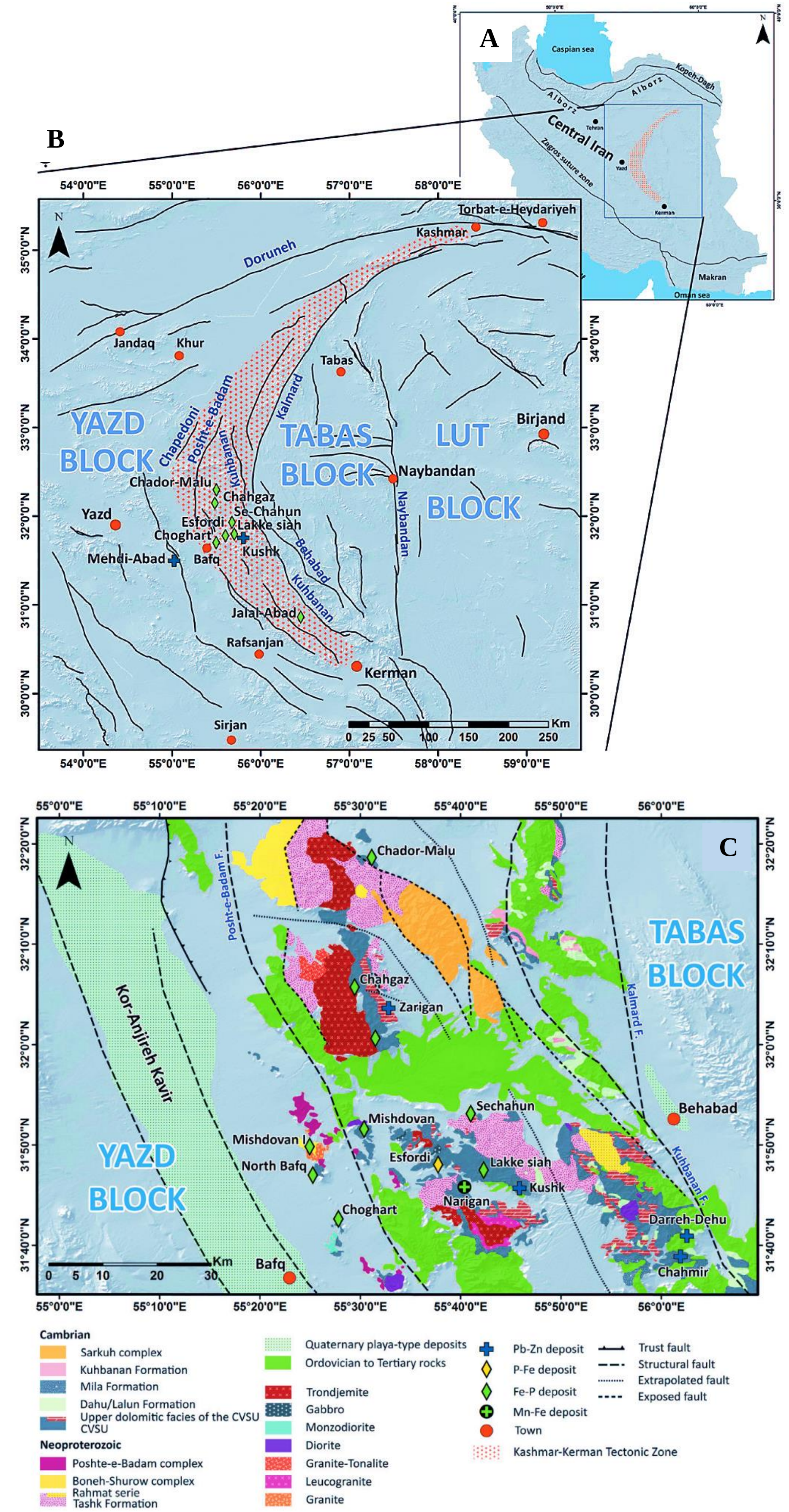

**Fig. 4.** A) The location of central Iran relative to the Zagros and Alborz rifts, B) Tectonics map of Central Iran blocks, C) Geological map of Bafq – Saghand block along with the location of iron oxide-apatite, manganese, lead and zinc deposits (Majidi et al. 2017)

Fig. 5 shows the magnetic anomaly around the investigated area:

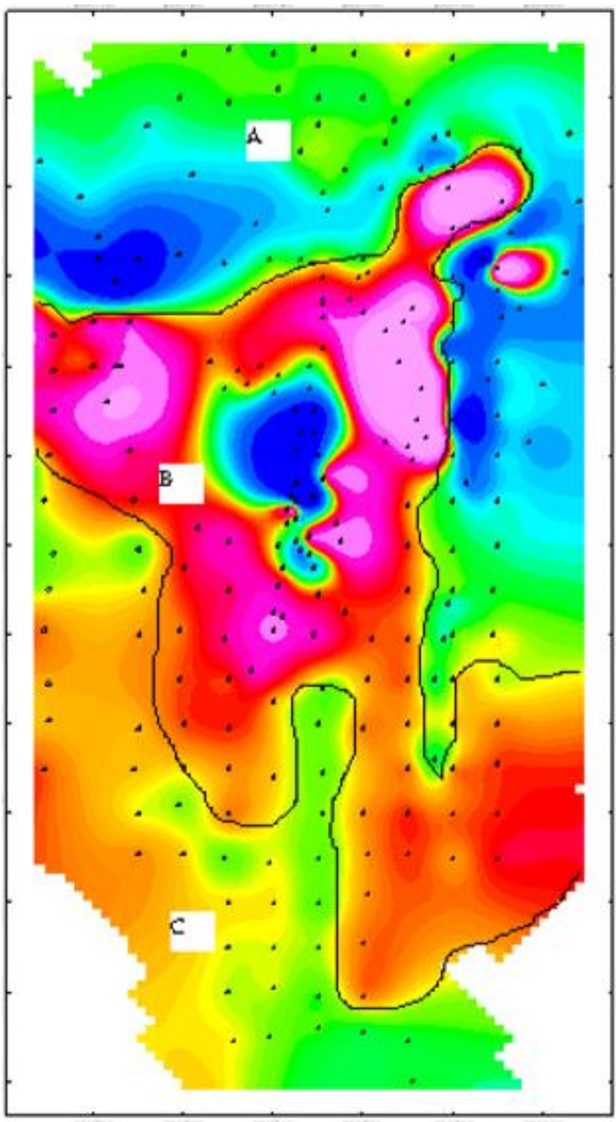

**Fig. 5.** General map of the magnetic anomaly of the studied peripheral area

The above-mentioned magnetic anomaly map shows almost drastic changes that may be caused by changes in the topography or lithology of the area. As can be seen in the picture, three zones have been identified in this area. Zone A has the lowest intensity of the magnetic field, which indicates the absence of ore mass in the area, or its great depth or sedimentary bedrock with east-west extension. Zone B has the highest intensity of the magnetic field, which may be due to the presence of masses with high intensity of magnetization, as well as the low depth of bedrock. Zone C also has moderate magnetic intensity, which probably indicates the presence of acidic igneous rocks, because acidic igneous rocks show moderate to low magnetic resistivity due to low amounts of iron minerals.

## III. Methodology

In this article, according to Fig. **6**, the possibility of grade numerical modeling at different depths has been investigated by applying an upward expansion filter to the magnetometric data and then by entering the data from the upward filter and the grade data at different depths in three algorithms, SVM, Random Forest, and Gradient Boosting.

### A. Surveying and applying an upward expansion filter to magnetometric data

To take the magnetometry of the studied area, a GEM proton magnetometer device made in Canada was used, and the accuracy of this device is equal to 0.01 nanotesla. According to the desired range and required data, 20 x 20 profiles perpendicular to the piles and the drilling network were designed, and finally, after about 13 kilometers of navigation with the device, 459 data points were read and surveyed. They include the latitude and longitude and the number obtained from the magnetometer without depth.

After surveying and applying the desired corrections on the magnetometric data, they were imported from the Geosoft Oasis Montaj software, which led to the preparation of the following two-dimensional magnetometric map of the studied area (Fig. *7*):

In the upward expansion filter stage, according to the final depth of 13 drilled boreholes and the length of the cores received from them, it was decided that the sequence of applying this filter would be 3 meters by 3 meters. The final depth of applying this filter is 282 meters, which is the deepest borehole drilled. As a result, this filter has been applied 94 times for different depths. After applying the filter and obtaining the two-dimensional depth maps, numerical data was extracted from the maps and a numerical value was extracted at each depth in the drilling points, which finally resulted in a complete set of data including longitude and latitude, depth, and magnetic depth number.

The results obtained by applying the upward expansion filter can be seen in Fig. 8 to 11:

After extracting the magnetic depth data, the numerical factor of the boreholes at different depths is added to the category of data obtained from the above operations. Eventually, this data set of 892 numbers will include coordinates, depth, magnetism depth, and finally the true value of the grade at each depth.

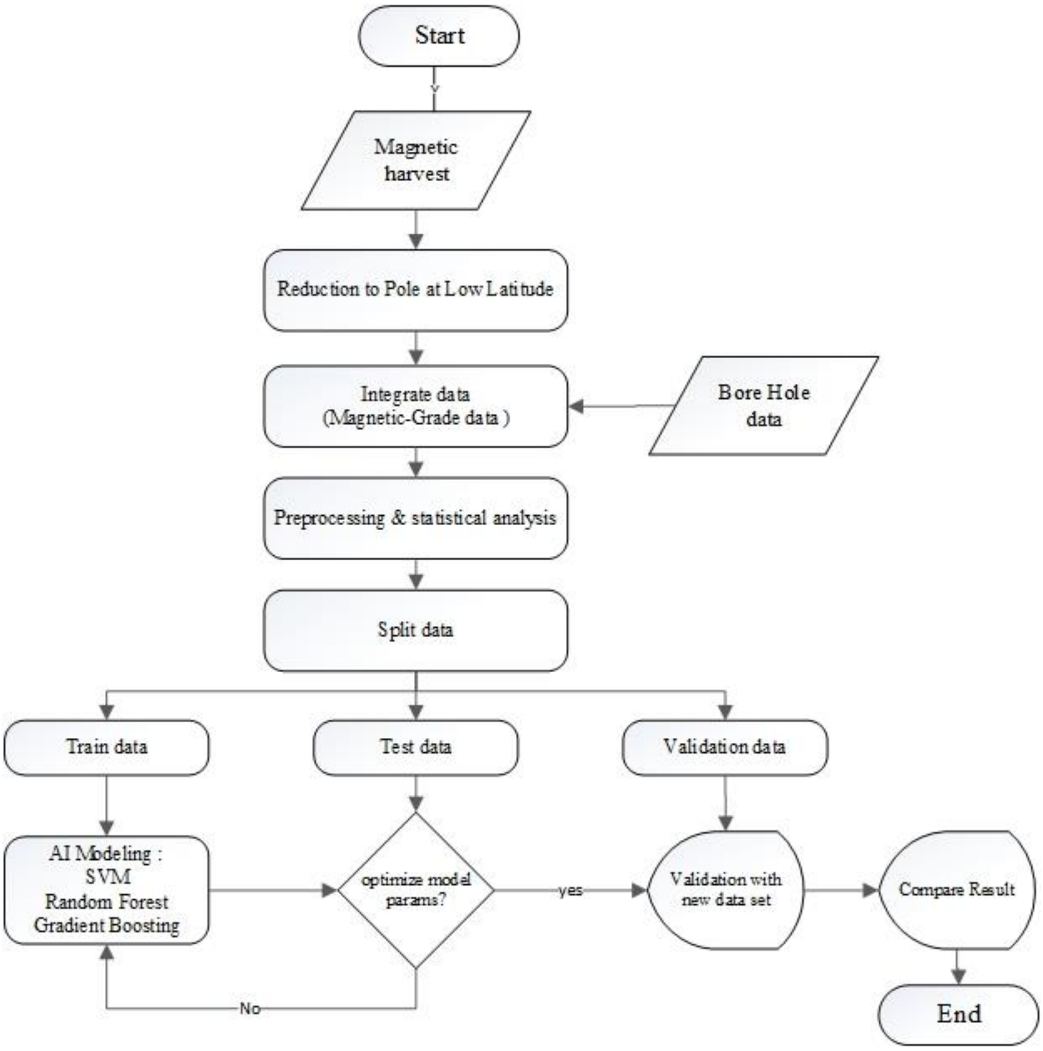
Start
Magnetic harvest
Reduction to Pole at Low Latitude
Integrate data (Magnetic-Grade data )
Bore Hole data
Preprocessing & statistical analysis
Split data
Train data
Test data
Validation data
AI Modeling : SVM Random Forest Gradient Boosting
optimize model params?
yes
No
Validation with new data set
Compare Result
End

**Fig. 6.** General flowchart of the article’s methodology

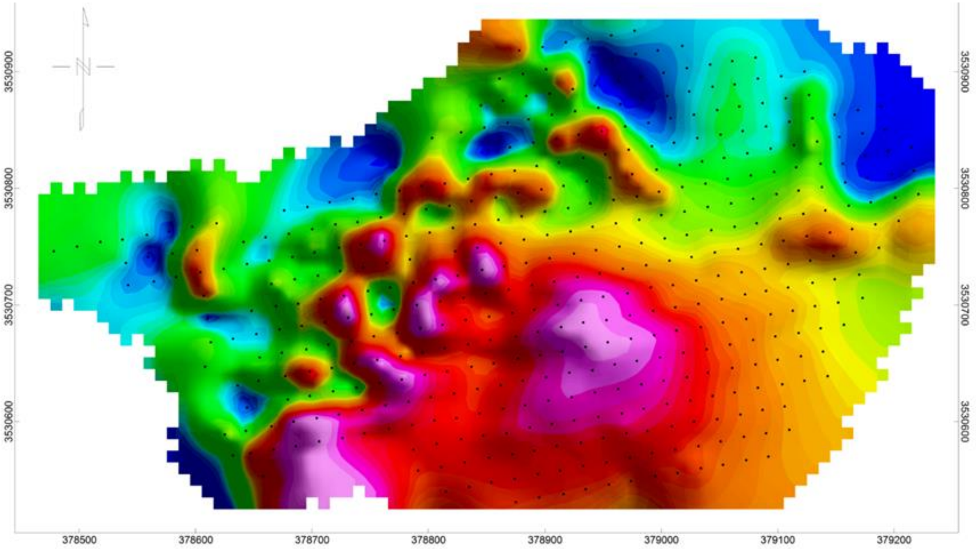

**Fig. 7.** The total map shows the regional distribution of the magnetic property of the ore

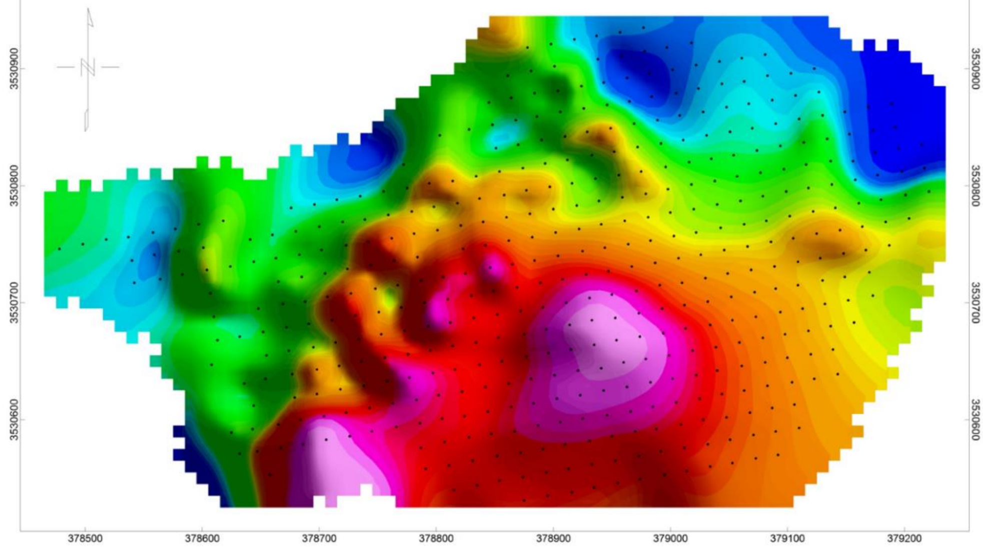

**Fig. 8.** The map of applying the upward expansion filter of magnetometry at a depth of 9 meters

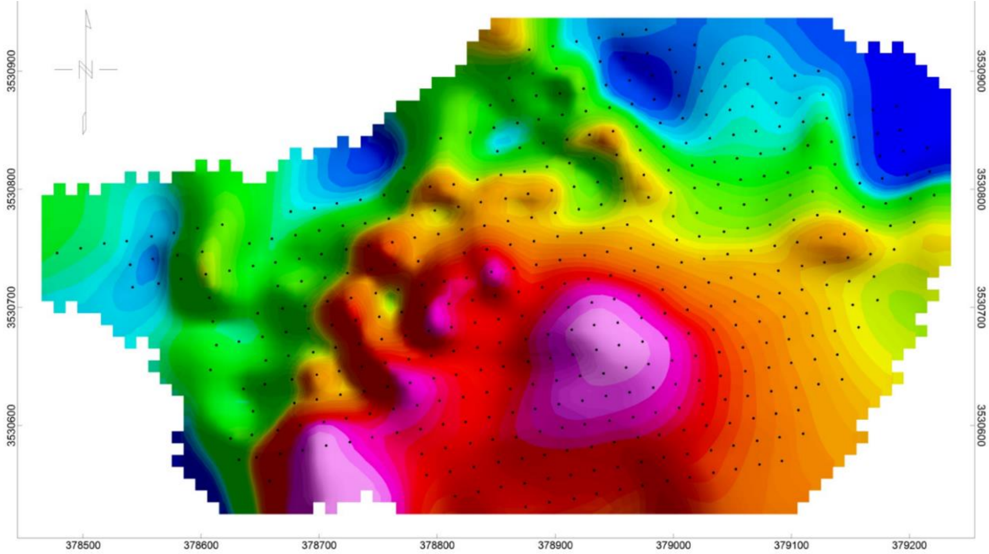

**Fig. 9.** The map of applying the upward expansion filter of magnetometry at a depth of 21 meters

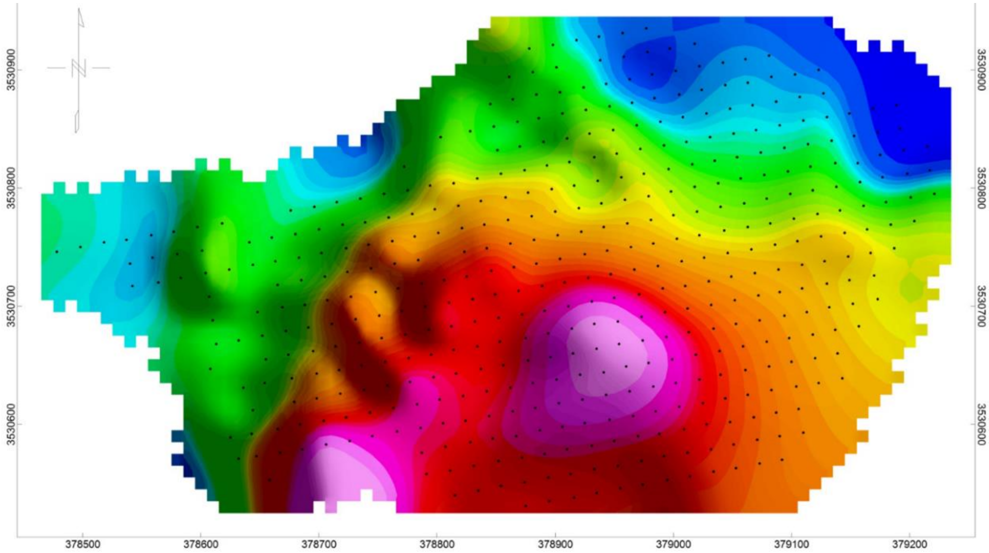

**Fig. 10.** The map of applying the upward expansion filter of magnetometry at a depth of 39 meter

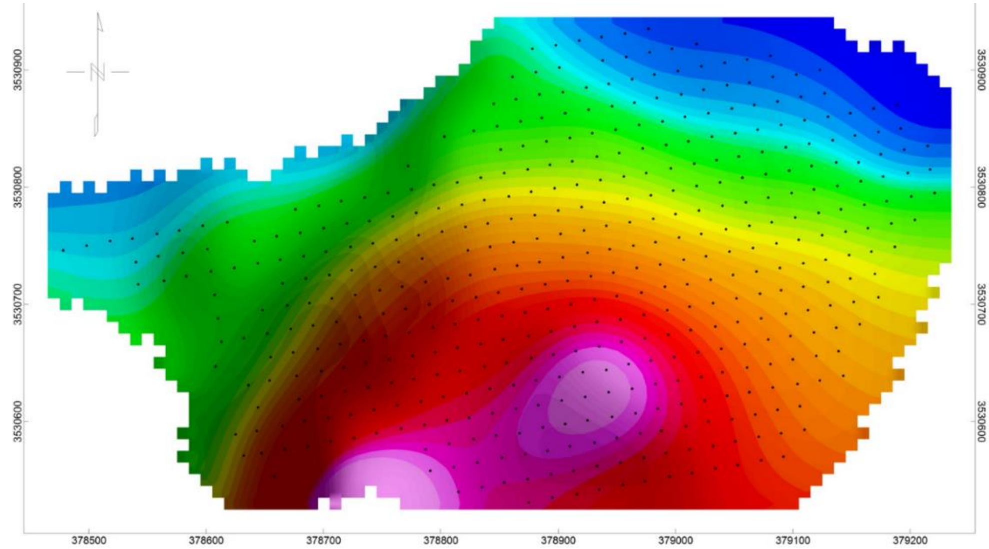

**Fig. 11.** The map of applying the upward expansion filter of magnetometry at a depth of 60 meters

### B. Statistical analysis

The statistical results of the entire data set were extracted using the Pandas library in the Python environment as follows:

**Table 2.** Statistical parameters of the studied data

| | X | Y | Mag. | Fe % |
|---|---|---|---|---|
| count | 892 | 892 | 892 | 892 |
| mean | 379063.1 | 3530599 | -1206.22 | 11.85294 |
| std | 119.4184 | 106.0084 | 1841.98 | 17.24815 |
| min | 378867.7 | 3530406 | -5586 | 0 |
| 25% | 378930.7 | 3530533 | -2368.75 | 0 |
| 50% | 379064.7 | 3530580 | -616.5 | 0 |
| 75% | 379178.7 | 3530709 | 210.25 | 22.305 |
| max | 379245.7 | 3530762 | 1852 | 63.79 |

From the points obtained from Table 2, the first one can be the large amount of zero data in the measured values of grade, which can be related to the low accuracy of sampling and analysis of boreholes, as well as human error; this issue itself can inevitably cause errors. The second case can be mentioned as the low changes in the location coordinates of the boreholes and the high changes in the value of the magnetometric parameter, which indicates the proximity of the boreholes to the existing ore in the area.

**Table 3.** Correlations (Percent) in Inputs and Output of the studied data

| | X | Y | Mag. | Fe |
|---|---|---|---|---|
| X | 100 | 15.89 | 59.23 | -6.5 |
| Y | 15.89 | 100 | 29.49 | 13.52 |
| Mag. | 59.23 | 29.49 | 100 | -4.15 |
| Fe | -6.5 | 13.52 | -4.15 | 100 |

The only interactive point Table 3 is the higher correlation between the magnetometric parameter with the longitude coordinates of the boreholes than the correlation between the same parameter with the latitude coordinates of the boreholes, which indicates the greater dispersion of the magnetometric parameter along the longitude coordinates.

Fig. 12 and Fig. **13** show the distribution of two key parameters relative to the geographic coordinates of the boreholes in this data set.

According to the Fig. 14 and 15 it can be easily understood that the data used in artificial intelligence algorithms needs normalization, which will be discussed in the following sections.

### C. Data Preparation

In order not to interfere with the size and amount of data and to apply the effect of the deviation of numbers from the mean, mode and median values in the learning process of the algorithm, there is a need to normalize the input data. To prepare the data for use in neural networks, it is necessary to pre-process the data first. In this stage of the work, after checking the data for validity and usability and removing invalid data, data normalization is done.

In this article, the Min-Max model is used to normalize the data, and the data are defined in the range of zero and one. In the Python programming language, the preprocessing library is used to do this. In this simple method, each set of data is mapped to arbitrary interval whose minimum and maximum values are already known. In this method, any arbitrary interval can be converted into a new interval with just a simple conversion. Suppose feature A, from the data set that is in the interval between min A to max A, is to be mapped to the new interval new Min to new Max. For this purpose, any initial value such as v in the initial interval will be converted to the new value v' in the new interval according to the following relation:

$$v' = (v - min_A)\frac{newMax - newMin}{maxA - minA} + (newMin) \qquad (1)$$

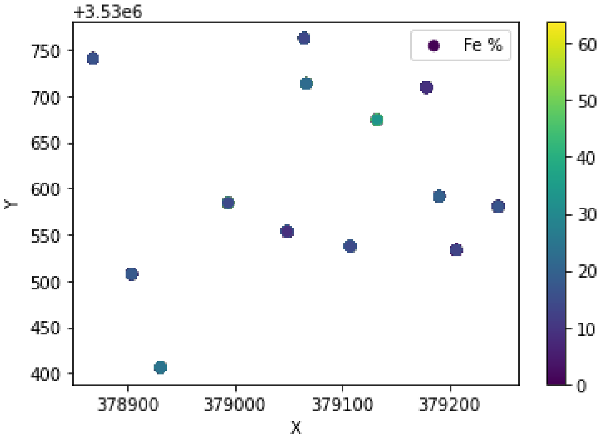

**Fig. 12.** Dispersion of iron grade values in magnetic longitude and latitude coordinates

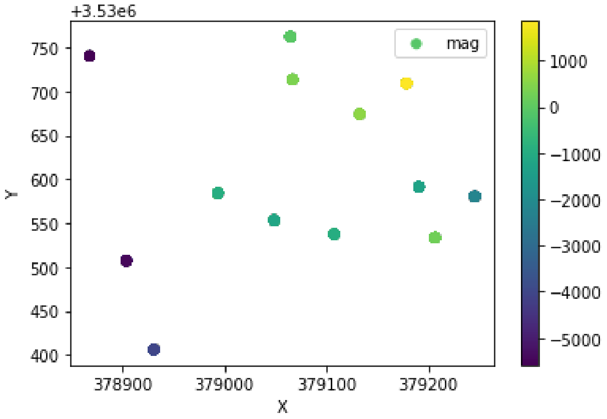

**Fig. 13.** Dispersion of magnetism values in magnetic longitude and latitude coordinates

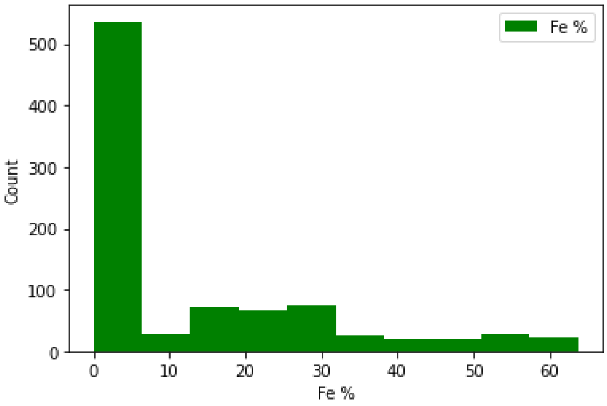

**Fig. 14.** Histogram of percentage of iron in the data set

## D. Support Vector Machine Algorithm

Vapnik and Lerner in 1963, and Vapnik and Chervonenkis in 1964 developed the SV algorithm as a nonlinear generalization of the Generalized Portrait algorithm (Smola and Bernhard Schoellkopf, 2004). It relies heavily on Vapnik and Chervonenkis (1974), and Vapnik (1982, 1995)'s VC theory, which developed over the past three decades. In a nutshell, VC theory describes properties of learning machines that enable them to generalize well to new data sets (Vapnik et al., 1997).

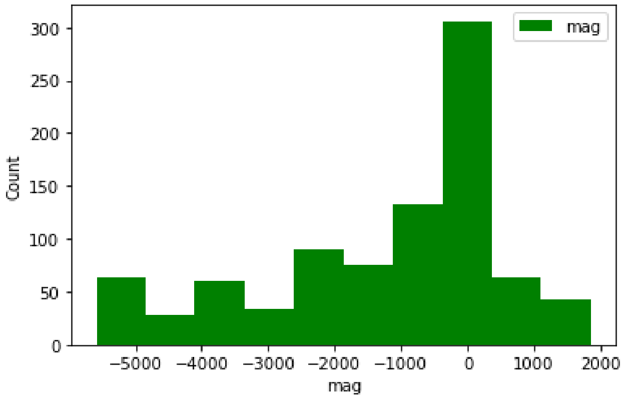

**Fig. 15.** Histogram of percentage of Magnetization in the data set

Suppose we are given training data $\{(x_1, y_1), \ldots, (x_L, y_L)\} \subset X \times R$, where X represents the input pattern space (e.g. X = Rd). An example would be the exchange rate for a given currency measured the following day along with the corresponding economic indicators. In ε-SV regression (Smola and Bernhard, 2004), our goal is to discover a function f (x) that has at most ε deviation from the actual attained targets $y_i$ for all training data, while remaining as flat as possible. In other words, errors are not concerned as long as they are less than ε, but will not be accepted deviations more than this. When dealing with exchange rates, it is crucial to ensure you don't lose more than ε.

For pedagogical reasons, it begins by describing the case of linear functions f, taking the form:

$$f\ (x) = [w, x] +\ b\ \ with\ \ w\ \in\ X, b\ \in\ R \qquad (1)$$

[·,·] represents the dot product in X. In the prior equation, flatness implies a small w. One technique to achieve this is to minimize the norm, i.e. $\|w\|^2 = [w, w]$. This problem can be expressed as a convex optimization problem:

$$\text{Minimize}\quad \frac{1}{2}\|w\|^2$$
$$\text{subject to}\quad \begin{cases} y_i - [w, x_i] - b \leq \varepsilon \\ [w, x_i] + b - y_i \leq \varepsilon \end{cases} \qquad (2)$$

The equation above assumes the existence of a function *f* that approximates all pairings $(x_i, y_i)$ with ε precision, implying that the convex optimization problem is possible.

However, this is not always the case, and it may be needed to allow for certain faults. Similar to Cortes and Vapnik's (1995) "soft margin" loss function for SV machines, slack variables $(\xi_i, \xi_i^*)$ can be employed to address infeasible restrictions in optimization problems.

Hence, the formulation stated in Vapnik (1995) is as follows:

$$\text{minimize } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{L}(\xi_i + \xi_i^*)$$

$$\text{subject to} \begin{cases} y_i - [w, x_i] - b \le \varepsilon + \xi_i \\ [w, x_i] + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \quad (3)$$

The constant C > 0 sets the trade-off between f's flatness and the maximum tolerance for deviations higher than ε. This refers to the ε-insensitive loss function |ξ|ε, as described by (Herbrich, 2002):

$$|\xi|_\varepsilon = \begin{cases} 0 & if\ |\xi| \le \varepsilon \\ |\xi| - \varepsilon & otherwise \end{cases} \quad (4)$$

## E. Random Forests Algorithm

### 1) *Tree Learning*

Decision trees are a popular approach to numerous machine learning challenges. Tree learning comes closest to relating the criteria for being used as an off-the-shelf data mining process since it is invariant under scaling and other transformations of feature values, is resilient to the insertion of irrelevant information, and creates inspectable models. However, they are rarely correct (Aliyari Ghassabeh, 2013).

Deeply developed trees, in particular, have a tendency to acquire highly irregular patterns: they overfit their training sets, resulting in low bias but large variation. Random forests are a method of averaging numerous deep decision trees trained on different regions of the same training set, with the purpose of reducing variation (Aliyari Ghassabeh, 2013).

This results in a modest increase in bias and some loss of interpretability, but it significantly improves the final model's performance. Forests are like the culmination of decision tree algorithm work. Using the combined efforts of several trees to improve the performance of a single random tree. Though not quite equivalent, forests provide the effects of a K-fold cross-validation (Carreira-Perpinan et al., 2007).

### 2) *Bagging*

The random forest training algorithm relies on the common approach of bootstrap aggregation, sometimes known as bagging, to tree learners. Given a training set $X = x_1, ..., x_n$ and responses $Y = y_1, ..., y_n$, bagging chooses a random sample (B times) with replacement of the training set and fits trees to these samples.

For $b = 1, ..., B$:

I. Sample, with replacement, n training examples from X, Y; call these $X_b$, $Y_b$.

II. Train a classification or regression tree fb on $X_b$, $Y_b$.

After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x':

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x') \quad (5)$$

or by taking the majority vote in the case of classification trees.

This bootstrapping approach improves model performance because it reduces variance without raising bias. This means that, whereas a single tree's predictions are highly sensitive to noise in its training set, the average of multiple trees is not, as long as the trees are uncorrelated. Simply training multiple trees on a single training set would result in heavily correlated trees (or the same tree many times, if the training technique is deterministic); bootstrap sampling is a method of de-correlating the trees by exposing them to different training sets (Aliyari Ghassabeh, 2015).

### 3) *From Bagging to Random Forests*

The approach outlined above describes the original tree-bagging algorithm. Random forests differ from this broad scheme in only one respect: they employ a modified tree learning algorithm that selects a random subset of the features at each candidate split in the learning process. This is known as "feature bagging" at times. The reason for this is the correlation of the trees in an ordinary bootstrap sample: if one or a few features are particularly strong predictors of the response variable (target output), these features will be chosen in many of the B trees, causing them to become correlated. Ho provides an examination of how bagging and random subspace projection lead to increased accuracy under various scenarios (Li et al., 2007).

Typically, for a classification problem with p characteristics, each split uses $\sqrt{p}$ (rounded down) features. For regression problems, the inventors recommend $\frac{p}{3}$ (rounded down) with a minimum node size of 5 as the default. In practice, the ideal values for these factors will vary depending on the problem, and they should be considered tuning parameters (Aliyari Ghassabeh, 2013).

## F. Gradient Boosting Algorithm

Many supervised learning tasks have an output variable $y$ and a vector of input variables x that are related to one another using a probabilistic distribution. The goal is to discover the function $\hat{F}(x)$ that best approximates the output variable based on the values of input variables. This is formalized by defining a loss function $L$ (y, F(x)) and minimizing it:

$$\hat{F} = arg\min_{F} E_{x,y}\,[L(y, F(x))].\sqrt{p} \quad (6)$$

The gradient boosting approach uses a real-valued y to obtain an approximation Fˆ(x) as a weighted sum of functions hi(x) from the base (or weak) learners:

$$\hat{F}(x) = \sum_{i=1}^{M} \gamma_i \, h_i(x) + const. \qquad (7)$$

Typically, we are given a training set {(x₁, y₁), (x₂, y₂), ..., (xₙ, yₙ)} with known x and y values. The method follows the empirical risk minimization principle, aiming to discover an approximation $\hat{F}(x)$ that minimizes the average value of the loss function on the training set, hence minimizing empirical risk. It accomplishes this by starting with a model composed of a constant function *F₀(x)* and incrementally expanding it greedily:

$$F_0(x) = arg \min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma) \qquad (8)$$

$$F_m(x) = F_{m-1}(x) + arg \min_{h_m \epsilon H} [\sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + h_m(x_i)] \qquad (9)$$

where $h_m \epsilon$H is a base learner function.

Unfortunately, selecting the optimum function h at each step given an arbitrary loss function L is a computationally impractical optimization task in general. As a result, we restrict our approach to a simplified form of the problem. The goal is to apply the steepest descent step to this minimization issue (functional gradient descent). To update the model for the continuous case, where H represents the set of arbitrary differentiable functions on R, use the following equations:

$$F_m(x) = F_{m-1}(x) - \gamma_m \sum_{i=1}^{n} \nabla_{F_{m-1}} L(y_i, F_{m-1}(x_i)) \qquad (10)$$

$$\gamma_m = arg \min_{\gamma} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) - \gamma \nabla_{F_{m-1}} L(y_i, F_{m-1}(x_i))\right) \qquad (11)$$

The derivatives are taken with regard to the functions *Fᵢ* for *i ϵ {1, 2, ..., m}*, and the step length is $\gamma_m$. In the discrete situation when the set H is finite, we choose the candidate function h closest to the gradient of L. The coefficient γ can then be computed using a line search on the equations above. Note that this strategy is a heuristic, so it does not produce an exact solution to the given problem, but rather an estimate (Painsky and Rosset, 2017).

## IV. Magnetometric – Grade Modeling

### A. Modeling with Support Vector Regression Algorithm

After applying the normalization function on the data set, at the beginning and before entering the data into the algorithm, 10% of the data is randomly separated for validation and then the rest of the data is 80 to 20 for regression learning, (training data) and (testing) was divided. The selection of regression parameters in each part of learning is based on the selection of the best result in terms of the lowest average learning error (MSE). The results of learning for magnetism-grade data can be seen in Fig. 16:

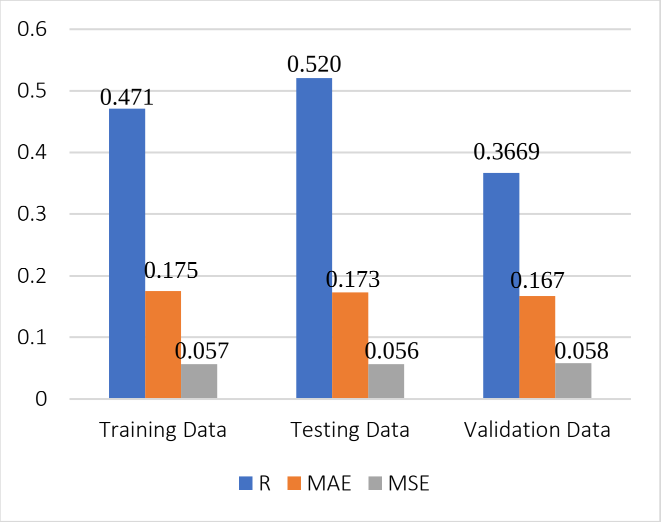

**Fig. 16.** Comparison of correlation coefficient values (R) and two error parameters (MAE, MSE) in SVR algorithm

In Fig. 17, the results of the validation of the model with the SVR algorithm can be seen:

### B. Modeling with Gradient Boosting Algorithm

The most important parameter in the GB algorithm is determining the learning rate. Like the previous algorithm, data segmentation is done for algorithm processes. The selection of regression parameters in each part of learning is based on the selection of the best result in terms of the lowest average learning error (MSE). The results of learning for magnetism-grade data can be seen in Fig. 18:

In Fig. 19, the results of the validation of the model with the GB algorithm can be seen:

### C. Modeling with Random Forest Algorithm

Like the previous two algorithms, data segmentation is done for algorithm processes. The selection of regression parameters in each part of learning is based on the selection of the best result in terms of the lowest average learning error (MSE). The results of learning for magnetism-grade data can be seen in Fig. 20:

In Fig. 21, the results of the validation of the model with the RF algorithm can be seen:

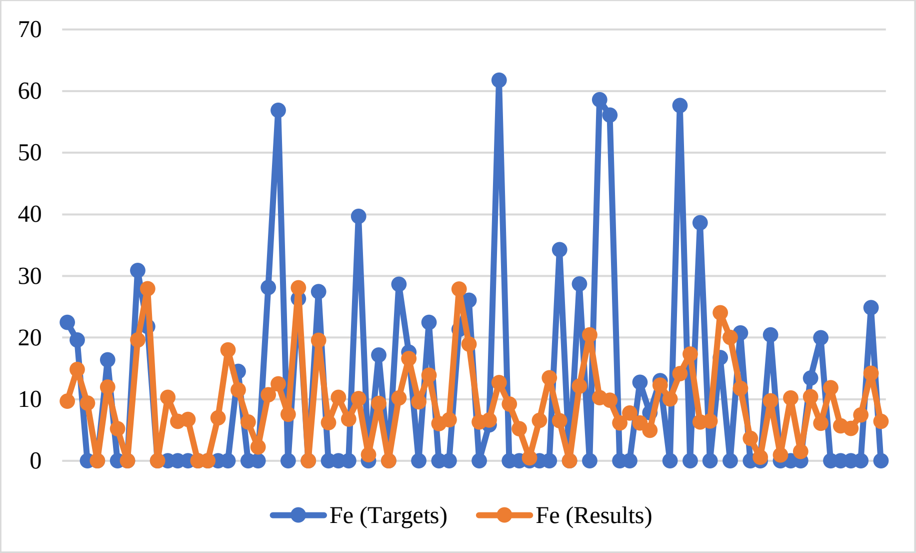

**Fig. 17.** Comparison of real iron grade values and results of the SVR algorithm in the validation stage

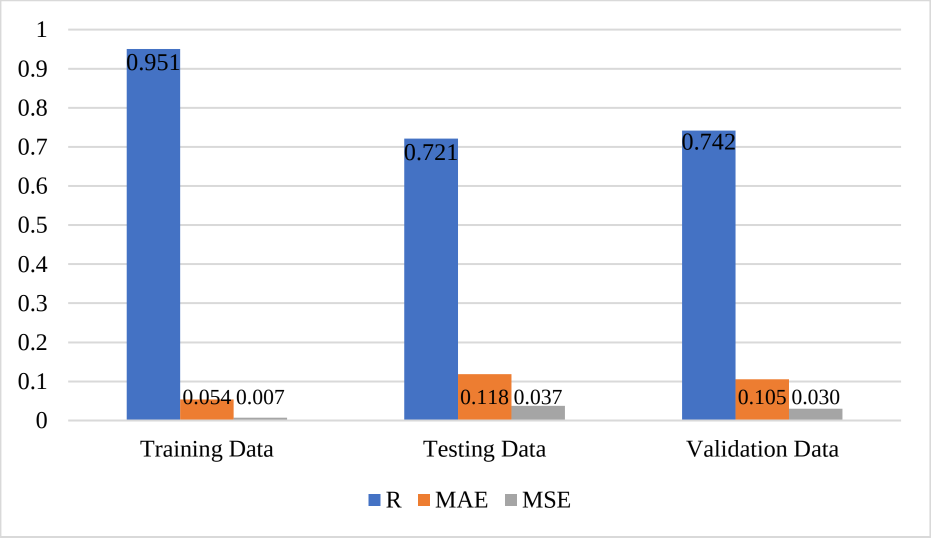

**Fig. 18.** Comparison of correlation coefficient values (R) and two error parameters (MAE, MSE) in GB algorithm

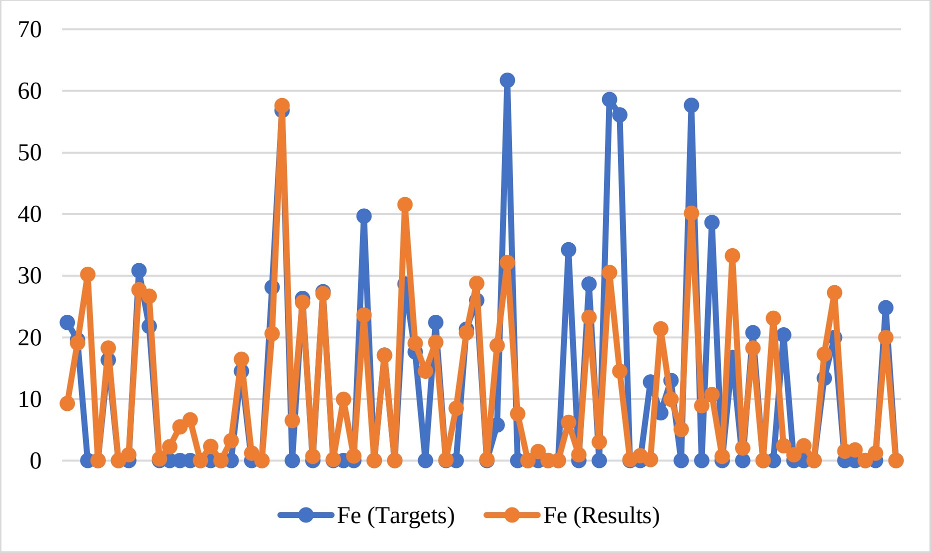

**Fig. 19.** Comparison of real iron grade values and results from the GB algorithm in the validation stage

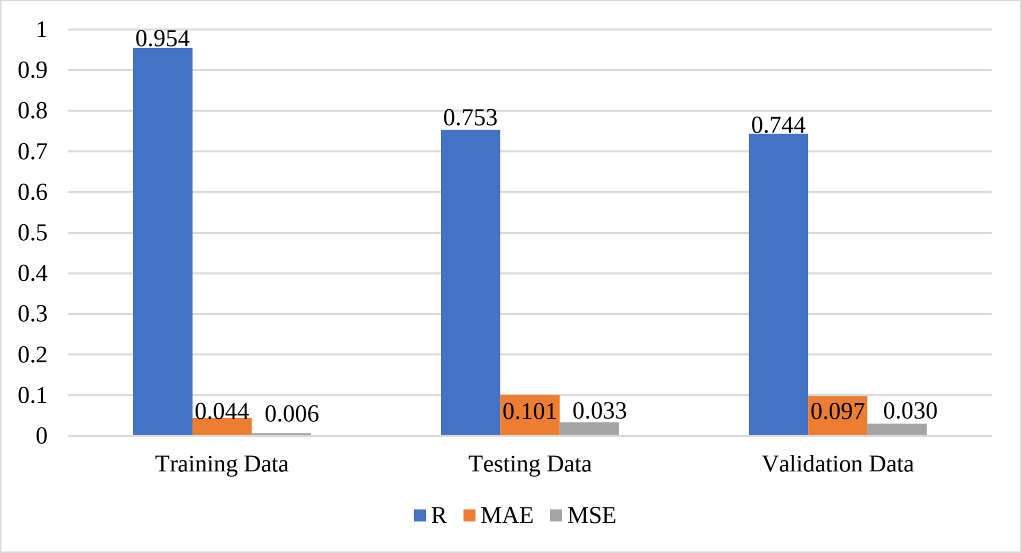

**Fig. 20.** Comparison of correlation coefficient values (R) and two error parameters (MAE, MSE) in RF algorithm

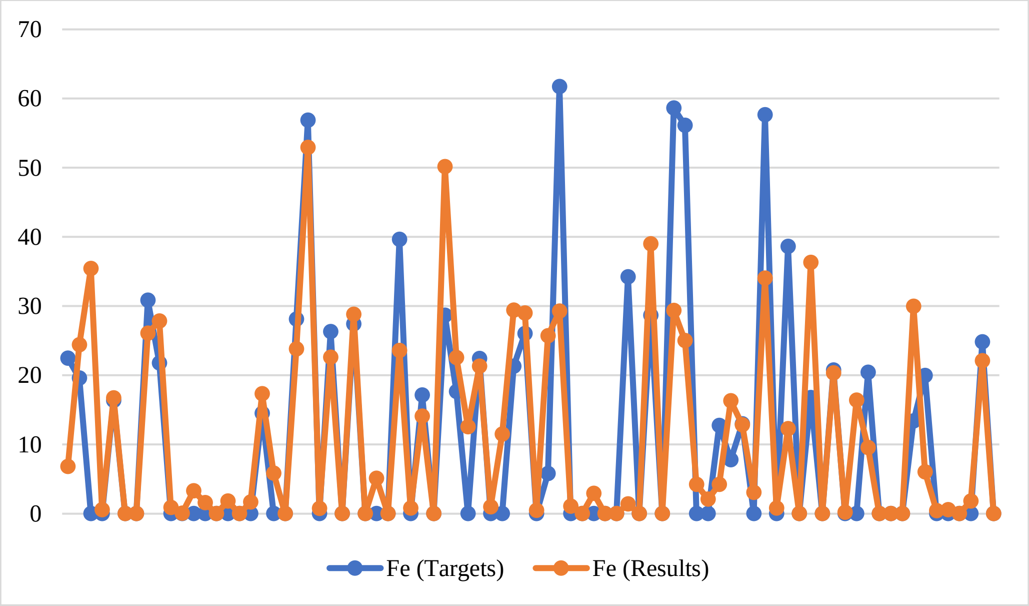

**Fig. 21.** Comparison of real iron grade values and results from the RF algorithm in the validation stage

## V. Conclusion

The main problem in the use of artificial intelligence and machine learning in various sciences is the existence of correct and appropriate data in terms of quantity and quality. In this article, the most limiting parameter is the data parameter and its accuracy and dispersion. As it was observed in the statistical analysis section, the number of dispersions of magnetometric data was very high in the amount of iron grade equal to zero, which reduces the correct relationship between iron grade and magnetometric number. As a result, it disrupts the learning process of the network and reduces the accuracy of the network.

The correct distribution of boreholes in the magnetometer range with certain intervals can help to increase the accuracy of the model. According to the results, it is possible to reduce the cost of mineral exploration by expanding this method and take a big step towards optimizing the exploratory drilling network as the most expensive part of mining exploration.

## Acknowledgments

The authors would like to express their gratitude to the Central Iran mines industries and the Kavoshgaran Consulting Engineers Company for their collaboration and support in permitting us to use the data that made this research work possible.

## References

Alimoradi A., Angorani S., Ebrahimzadeh M. and Shariat Panahi M. (2011). Magnetic Inverse Modelling of a Dike Using the Artificial Neural Network Approach, Near Surface Geophysics 9. 339-347.

Alimoradi A., Maleki B., Karimi A., Sahafzadeh M. and Abbasi S. (2020). Integrating Geophysical Attributes with New Cuckoo Search Machine-Learning Algorithm to Estimate Silver Grade Values–Case

Study: Zarshouran Gold Mine, Journal of Mining and Environment, Volume 11, Issue 3, Summer 2020, Pages 865-879.

Aliyari Ghassabeh, Youness (2013). On The Convergence of the Mean Shift Algorithm in the One-Dimensional Space, Pattern Recognition Letters, 34 (12). 1423–1427.

Aliyari Ghassabeh, Youness (2015). A Sufficient Condition for the Convergence of the Mean Shift Algorithm with Gaussian Kernel, Journal of Multivariate Analysis, 135, 1–10.

Ardjmandpour N., Pain C., Singer J., Saunders J., Aristodemou E. and Carter J. (2011). Artificial Neural Network Forward Modelling and Inversion of Electrokinetic Logging Data, Geophysical Prospecting, 59. 721–748.

Bayati A., Alimoradi A., Sarkheil H. (2022). Reserve Estimation of Iron Ore Deposit Using Ground Magnetic Data and 3D Modeling, 6rd Seminar Petroleum Geophysical Exploration, 94-101.

Boehmke, B., Greenwell, B. (2019). Gradient Boosting, Hands-On Machine Learning with R, Chapman & Hall, PP. 221–245.

Breiman L. (2001). Random Forests, Machine Learning, 45 (1). 5–32.

Carreira-Perpinan, Miguel A. (2007). Gaussian Mean-Shift Is an EM Algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence, 29 (5). 767–776.

Cristiani N., Shawe-Taylor J., (2000), An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press.

Dietterich T. (2000). An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization, Machine Learning, 40 (2). 139–157.

Dorin C., Meer P. (2002). Mean Shift: A Robust Approach Toward Feature Space Analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24 (5). 603–619.

Fathi M., Alimoradi A. and Hemati Ahooi H., (2021). Optimizing The Extreme Learning Machine Algorithm using Particle Swarm Optimization to Estimate Iron Ore Grade, Journal of Mining and Environment, Journal of Mining and Environment, Vol.12, No.2, 397-411.

FitzGerald D. (2019). Artificial Intelligence Techniques to The Interpretation of Geophysical Measurements, ASEG Extended Abstracts, 1, 1-5.

Fukunaga K., Hostetler L.D. (1975). The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition, IEEE Transactions on Information Theory, 21 (1). 32–40.

Ghasemi Tabar H.R., Alimoradi A., Hemmati Ahooi H.R., Fathi M., and Sarookhani M. (2024). Intelligent Borehole simulation with python Programming, Journal of Mining and Environment, Journal of Mining and Environment, Vol.15, No.2, 707-730.

Herbrich R. 2002. Learning Kernel Classifiers: Theory and Algorithms, MIT Press.

Ho, Tin Kam (1995). Random Decision Forests, Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, PP. 278–282.

Kapageridis I.K., et al. (1999). Application of Artificial Neural Network Systems to Grade Estimation from Exploration Data, University of Nottingham Publications.

Keary P., Brooks M., and Hill I. (2002). An Introduction to Geophysical Exploration, 3rd edition, Blackwell.

Li, Xiangru; Hu, Zhanyi; Wu, Fuchao (2007). A Note on The Convergence of the Mean Shift, Pattern Recognition, 40 (6). 1756–1762.

Majidi S. A., Lotfi M., Emami M. H., and Nezafati N. (2017). The Genesis of Iron Oxide-Apatite (IOA) Deposits: Evidence from the Geochemistry of Apatite in Bafq-Saghand District, Central Iran, Scientific Quarterly Journal, Geosciences, Vol. 27, No. 105, 233-244.

Mitchell T.M. (1997). Machine Learning, McGraw-Hill Companies, Inc.

Mukherjee S., Bell R.S., Barkhouse W.N., Adavani S., Lelièvre P.G., and Farquharson C.G. (2022). High-Resolution Imaging of Subsurface Infrastructure Using Deep Learning Artificial Intelligence on Drone Magnetometry, The Leading Edge, 462-471.

Painsky A., Rosset S. (2017). Cross-Validated Variable Selection in Tree-Based Methods Improves Predictive Performance, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39 (11). 2142–2153.

Platt J. (1998), Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, Technology rep.

Raschka S., Patterson J., Nolet C. (2020). Machine Learning in Python: Main Developments & Technology Trends in Data Science Machine Learning & Artificial Intelligence, arxiv .45.

Smola A.J., Bernhard Schoellkopf B. (2004), A Tutorial on Support Vector Regression, Statistics and Computing, 14, 199–222.

Vapnik V., Golowich S., and Smola A. (1997). Support vector method for function approximation, regression estimation, and signal processing. In: Mozer M.C., Jordan M.I., and Petsche T. (Eds.) Advances in Neural Information Processing Systems 9, MA, MIT Press, Cambridge. pp. 281–287.

Vapnik V.N., (1998). The Nature of Statistical Learning Theory, Springer.

Yizong C. (1995). Mean Shift, Mode Seeking, and Clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence, 17 (8). 790–799.

Yuan S., Wang S., and Tian N., (2009). Swarm Intelligence Optimization and Its Application in Geophysical Data Inversion, Applied Geophysics, Vol.6, No.2, P. 166 – 174.