applied sciences

MDPI

Article

Several Tree-Based Solutions for Predicting Flyrock Distance Due to Mine Blasting

Mojtaba Yari [1], Danial Jahed Armaghani [2,3], Chrysanthos Maraveas [4], Alireza Nouri Ejlali [1], Edy Tonnizam Mohamad [2] and Panagiotis G. Asteris [5,*]

[1] Department of Mining Engineering, Faculty of Engineering, Malayer University, Malayer 65719-95863, Iran
[2] Centre of Tropical Geoengineering (GEOTROPIK), Institute of Smart Infrastructure and Innovative Engineering (ISIIC), Faculty of Civil Engineering, Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia
[3] Department of Urban Planning, Engineering Networks and Systems, Institute of Architecture and Construction, South Ural State University, 76 Lenin Prospect, 454080 Chelyabinsk, Russia
[4] Department of Natural Resources Management & Agricultural Engineering, Agricultural University of Athens, 11855 Athens, Greece
[5] Computational Mechanics Laboratory, School of Pedagogical and Technological Education, 14121 Athens, Greece
* Correspondence: panagiotisasteris@gmail.com

Abstract: Blasting operations involve some undesirable environmental issues that may cause damage to equipment and surrounding areas. One of them, and probably the most important one, is flyrock induced by blasting, where its accurate estimation before the operation is essential to identify the blasting zone's safety zone. This study introduces several tree-based solutions for an accurate prediction of flyrock. This has been done using four techniques, i.e., decision tree (DT), random forest (RF), extreme gradient boosting (XGBoost), and adaptive boosting (AdaBoost). The modelling of tree-based techniques was conducted with in-depth knowledge and understanding of their most influential factors. The mentioned factors were designed through the use of several parametric investigations, which can also be utilized in other engineering fields. As a result, all four tree-based models are capable enough for blasting-induced flyrock prediction. However, the most accurate predicted flyrock values were obtained using the AdaBoost technique. Observed and forecasted flyrock by AdaBoost for the training and testing phases received coefficients of determination (R^2) of 0.99 and 0.99, respectively, which confirm the power of this technique in estimating flyrock. Additionally, according to the results of the input parameters, the powder factor had the highest influence on flyrock, whereas burden and spacing had the lowest impact on flyrock.

Keywords: mine blasting; environmental issue; flyrock; tree-based techniques; prediction

Citation: Yari, M.; Armaghani, D.J.; Maraveas, C.; Ejlali, A.N.; Mohamad, E.T.; Asteris, P.G. Several Tree-Based Solutions for Predicting Flyrock Distance Due to Mine Blasting. *Appl. Sci.* **2023**, *13*, 1345. https://doi.org/10.3390/app13031345

Academic Editor: Francesco Clementi

Received: 24 November 2022
Revised: 12 January 2023
Accepted: 16 January 2023
Published: 19 January 2023

1. Introduction

Blasting operation is the most conventional method for achieving the desired fragmentation in mining and civil projects [1]. In this operation, just about 20–30% of the energy produced by blasthole explosives is consumed for the breaking of rockmass [2]. The residual part leads to undesirable phenomena such as airblast, ground vibration, and flyrock [3–7]. Considering all the unwanted results of blasting operations, flyrock is reported as the main hazardous consequence of blasting, which causes fatal and non-fatal events in open pit mines [8]. The exact definition of flyrock could be considered as boosting the broken rock particles after blasting operations far from the free face, which could lead to undesirable results [9]. All influencing parameters on flyrock are considered in two categories: controllable and uncontrollable factors, and in order to achieve a reliable attitude, both factors should be assumed [10–12]. Inexact drilling, inappropriate blasthole diameter, incorrect powder factor, insufficient stemming, unsuitable delay time, and inadequate burden are the main controllable sources of flyrock [13,14]. Uncontrollable parameters, on

the other hand, such as inappropriate geological features of the rock mass and placing the loosing rock at the top of the free face, may result in flyrock [15].

In terms of the prediction of flyrock, there are three main groups of methods, i.e., empirical, risk-based/statistical and artificial intelligence (AI). One of the most significant empirical formulations for estimating flyrock was presented by Lundborg et al. [16] on the basis of hole diameter. In another study, Olofsson [17] proposed a practical relationship using stemming to burden ratio for estimating flyrock. In the study conducted by Trivedi et al. [18], rock quality designation (RQD) and stemming-to-burden ratio factors were employed for developing an empirical formulation of flyrock. Inadequate empirical equations result due to practical method limitations, a diversity of influencing factors in the flyrock phenomenon, inherent uncertainty, and a complicated blasting operation [19,20].

As another group of methods in estimating flyrock, the statistical/risk-based approach has been adopted by several scholars. Raina et al. [21] presented a risk-based model for flyrock assessment. In their study, the three main parameters of a factor of safety, threat levels, and flyrock risk were regarded as explaining risk classes for diverse mining situations. According to the presented dynamic danger zone, all factors of safety necessities and production demands are regarded, which makes blasting operations more flexible. Armaghani et al. [22] developed a multiple regression (MR) for flyrock prediction and then simulated this phenomenon using Monte Carlo (MC). The simulated MC results were very close to the measured flyrock. In another investigation, Monjezi et al. [23] presented a statistical model for the assessment of flyrock by utilizing the MR approach. They selected burden, hole spacing, length of stemming, and powder factor as input parameters and obtained acceptable accuracy (i.e., a coefficient of determination, R^2, of 0.86). A nonlinear MR equation with R^2 values of 0.777 and 0.819 for the training and testing phases, respectively, was introduced by Faradonbeh et al. [24]. Risk-based models, probabilistic models, or statistical models are not able to achieve a high level of accuracy in different case studies, as stated by several researchers [24,25].

Considering the limitations of the presented methods, implementing novel approaches such as AI and soft computing (SC) techniques for estimating flyrock seems essential and has been considered by several researchers. Rezaei et al. [14] presented a fuzzy interface system (FIS) model for forecasting flyrock phenomena in the Gol-E-Gohar iron mine, Iran. A Mamdani fuzzy model was implemented in this model considering a database with 490 datasets, and this model was able to receive a high level of accuracy for flyrock estimation. A backpropagation neural network model was proposed by Yari et al. [26] for forecasting flyrock, considering controllable factors of blasting as model inputs. According to Yari et al. [26], the powder factor has the greatest impact on the flyrock. In another study, Ghasemi et al. [27] showed adequate proficiency of both the FIS and neural network in predicting flyrock induced by blasting. Two intelligent flyrock equations, i.e., gene expression programming (GEP) and genetic programming, were proposed by Faradonbeh et al. [28] with a high degree of accuracy. A regression-based technique (i.e., support vector machine) was introduced by Khandelwal and Monjezi [29] to predict flyrock. They reported a close value between the predicted and measured flyrock (R^2 = 0.948). Hybrid AI and SC techniques which are a combination of metaheuristic algorithms and other base models, such as neural networks and extreme learning machines, were also developed for flyrock prediction [4,30–34]. It is obvious that AI and SC techniques are the best categories for flyrock forecasting among the available methodologies in terms of accuracy. It is also important to note that these techniques have been widely employed and suggested in the literature for solving engineering problems [35–55].

Tree-based techniques are one type of hierarchical model that, by regularly separating datasets, provides a technique for assessment and prediction purposes [56]. Diverse classification, variable relationship recognition, and regression problems could be solved by utilizing tree-based models [56]. Compatibility with different assumptions, diverse data distributions, and simple construction make these models one of the applicable methods in geotechnics [57–59]. Pham et al. [60] presented a new model for the classification of

soils using Adaboost and enhanced tree methods. For this purpose, they gathered 440 data samples of soil features and achieved acceptable results. A regression tree technique was used for rock strength assessment in the study conducted by Liang et al. [61]. In another application of tree-based models, the gradient-boosted tree (GBT) technique was used by Huat et al. [62] to forecast pile friction-bearing capacity. The testing phase showed enough efficiency for the GBT model to provide predictions of pile friction-bearing capacity. These techniques were also successful in slope stability classification in an interesting investigation by Asteris et al. [63]. They suggested the AdaBoost approach with an accuracy of 0.93 for slope stability classification.

In light of the above discussion, several tree-based techniques, i.e., DT, RF, XGBoost, and AdaBoost, are used by the authors of this study to evaluate their powers in predicting flyrock induced by blasting. The best model will be chosen after comparing their performance predictions. Below is a plan for the rest of this paper:

The background of tree-based techniques will be described in Section 2. In Section 3, a description of Sungun Copper Mine as the case study will be given. In addition, a statistical overview of the dataset will be proposed in the same section. The modelling procedure will be described in detail in Section 4 with four sub-sections. In Section 5, the performance of tree-based models is assessed and discussed. Finally, Section 6 will discuss the concluding remarks of this study.

2. Model's Background

2.1. DT Model

Supervised-learning DT approaches are powerful tools for performing classification and prediction modelling in machine learning and data mining. Although there are several subsets for DT, e.g., chi-squared automatic interaction detection (CHAID), quick, unbiased, efficient, and statistical trees, C5, and classification and regression trees (CART), out of these algorithms, only CART and CHAID are suitable for predicting continuous variables. Because of its white-box nature and simple interpretability, the CART algorithm makes it easy to understand the relationship between input and output parameters and is outstanding compared to other DT algorithms. Moreover, large-scale datasets do not affect CART results, and this algorithm shows its superiority when dealing with complex samples and a high number of variables [64].

To identify the most influential input parameters, CART inherently employs principal component analysis and eliminates non-significant ones [65]. CART DT can be developed as a classification tree (CT) or regression tree (RT). The performance of a CART can be affected by choosing the right partition for a database based on its main indices. There are three main indices for a CT: the Gini criterion, entropy, and the Twoing criterion. Moreover, due to the non-parametric temperament of CART, the assumption of a distribution for variables is not required. Generally, each CART consists of four parts, including the root, branch, node, and leaf. Each tree starts from the root node (i.e., the first node), which is located at the upper level of each tree and is divided into branches on the left and right sides.

To prevent a CART training algorithm from developing a complicated tree, there are three prevention criteria: (1) a minimum number of observations, (2) maximum tree depth, and (3) reaching the least error value for the estimation of a dependent parameter. The verification dataset will prune the developed tree and allow it to reach the optimal subtree. The pruning procedure also reduces the occurrence of overtraining. As shown in Figure 1, a DT consists of a root node, decision (interior) nodes, and terminal (leaf) nodes. Each sample in a dataset is classified from the root node until it can no longer be divided into decision nodes and reaches a terminal node.

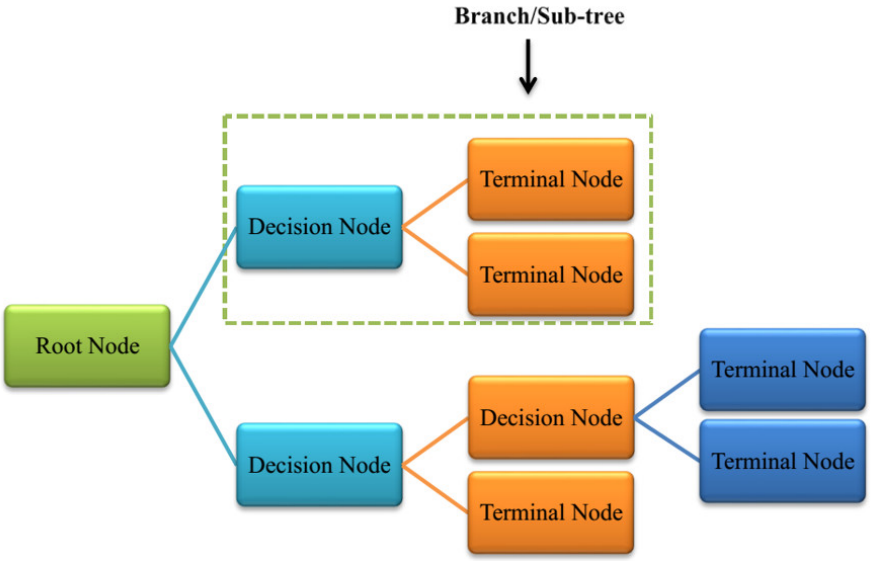


Figure 1. A DT structure consists of root nodes, decision nodes, and terminal nodes.

2.2. RF Model

RF was developed based on the CART DT algorithm by Breiman [66]. This algorithm combines a large number of DTs and can be used as a classification or regression method without making any prior assumptions about their association with the response variable [67]. At the beginning of the RF algorithm, samples are created using a bootstrap sample selection of the dataset, and each of these bootstrap-created samples constructs one RF tree. In addition, the unused samples from the bootstrap selection procedure, called the Out Of Bag (OOB) samples, are used in the validation process. Figure 2 shows the flowchart of the RF algorithm.

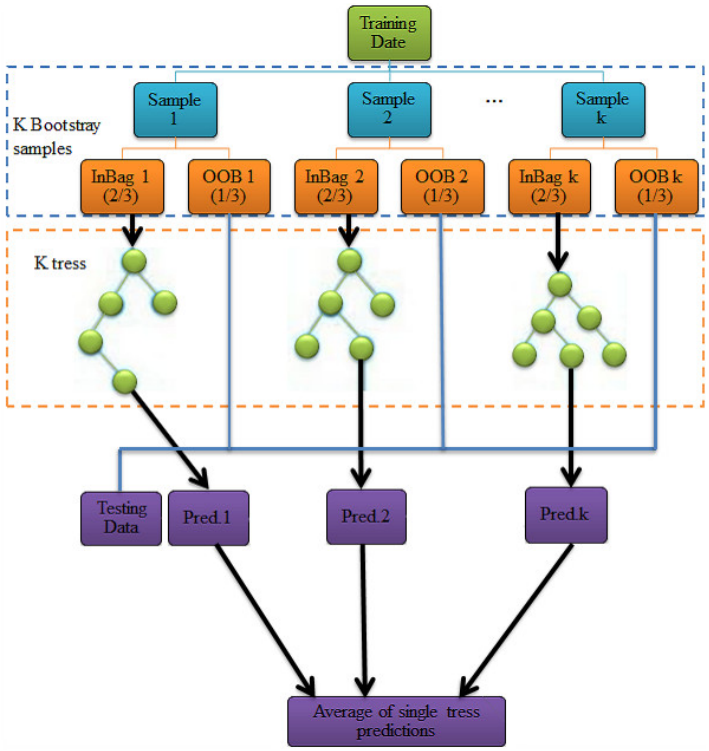


Figure 2. Flowchart of RF algorithm.

In classification problems, the mode of voting of trees, and regression problems, the average value would be selected as the predicted value of RF [66]. The structure of each tree in RF can be controlled by (i) minimum node size, (ii) the number of trees, and (iii) the level of randomness [68]. In comparison with other machine learning techniques, RF has some advantages, e.g., (a) developed trees can be saved and used for future references, (b) is not susceptible to overfitting problems, (c) lower training time and faster prediction, and (d) embedded feature selection makes it easy for RF to rank parameters by importance [69].

Meanwhile, one of the essential features of RF that influences its final performance is its splitting feature. In the most developed and published models using RF, the Gini-index splitting feature was used [70]; however, due to the use of the Scikit-Learn Python library [71] in this study, two splitting features were available, including Gini-index and Entropy. The GINI index uses the impurity of nodes to measure the minimum expected error rate for the set. The entropy criterion measures the information in a data group and splits the tree so that it gives more information at each split. The more extensive the entropy, the better the homogeneity of the subsets [72].

2.3. XGboost Model

XGBoost was developed by Chen and Guestrin [73] as a scalable machine-learning method for tree boosting. This algorithm has shown outstanding performance in different engineering problems compared to other machine learning algorithms due to its effective tree pruning, regularization, and parallel processing [74,75]. The general idea behind the development of boosting algorithms is that weak learners, such as DT, use a random guessing procedure and cannot use errors to improve the final model. However, XGBoost combines weak-based learners with stronger learners, and residuals in each iteration are used to correct the previous predictors, as shown in Figure 3.

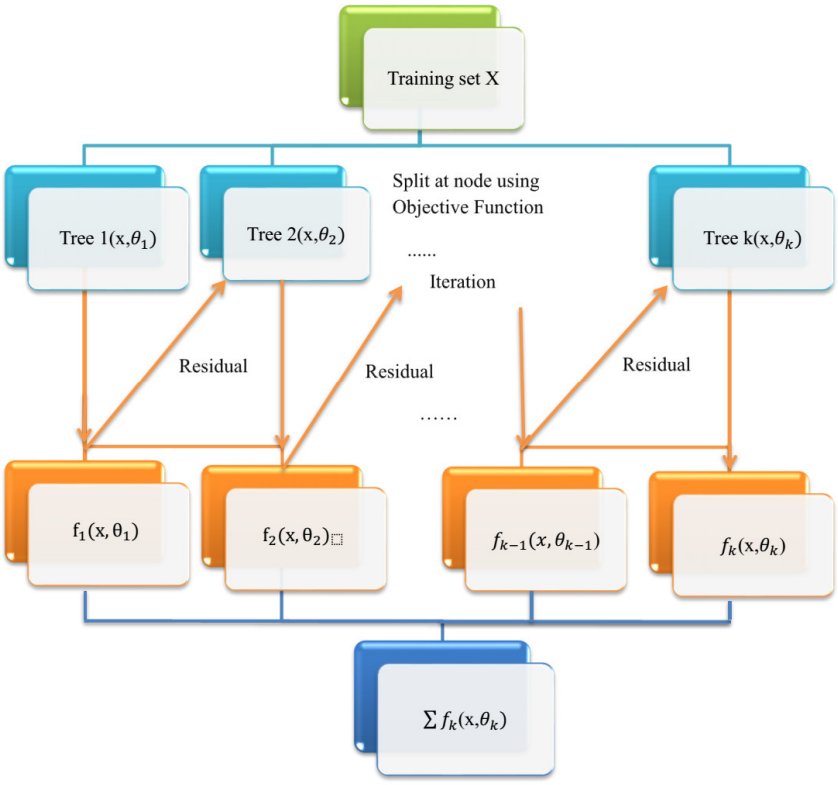


Figure 3. Flowchart of the XGBoost algorithm (X represents a variable, and θ represents the DT features, e.g., TD, splitting number, etc., which can be used to develop a tree for prediction of X, and f is the learned tree).

The main advantages of XGBoost are [76]:

XGBoost is ten times faster than other popular algorithms, and this speed enables it to solve the issues of time consumption in big data and problems where less run time is demanded.

- Parallel processing of XGBoost causes the high scalability and generation of billions of examples using the lowest resource consumption, which allows it to be effective in classification and high-level pre-processing data problems.
- XGBoost can be programmed using a broad scope of languages, e.g., Java, Python, R, and C++.
- XGBoost is less likely to overtrain because it makes strong learners by combining weak learners. This makes XGBoost more accurate at making predictions.
- XGBoost can effectively handle the missing data.
- Cross-validation can be done using the training data, and there is no need for extra cross-validation packages.
- To reach the highest performance of XGBoost, several model choices are needed. To avoid overfitting or to develop too complex models, XGBoost must tune the parameters. XGBoost is susceptible to learning noises or random fluctuations, and overfitting happens when these types of data are considered meaningful to XGBoost.
- Some internal parameters of XGBoost that should be regulated to avoid overfitting include iteration number, which is the number of trees that were fitted in the model; the maximum depth, which shows the maximum number of splits; an increase in this parameter increases the overfitting probability; subsample that shows the percentage of the dataset which are selected for training; learning rate which modifies the weights and impact of each tree to improve the performance of model; colsample_bytree is the ratio of subsample columns in tree construction; lambda and alpha that make regularization on weights and increase in these parameters make the model more conservative [73].

2.4. AdaBoost Model

In the 1990s, researchers wondered whether a strong learner could be obtained if multiple weak learners were combined. Schapire (1990) proposed a solution to this premise 1990 and provided the initial theories for the boosting algorithm. Boosting takes advantage of numerous relatively weak and inaccurate predictors to produce an extremely accurate prediction learner using machine learning concepts. AdaBoost is a boosting algorithm that was introduced by Freund and Schapire [77].

At first, AdaBoost assigns the same weight to every dataset. Then, it repeatedly measures classification errors in iterations and updates the weights of samples. The weights of misclassified samples gradually increase, and the weights of correctly classified samples decrease. Therefore, the subsequent classifier focuses on correctly classifying the previously misclassified sample. Using the new weights in the data set, the lower classifier is trained. Lastly, the final decision classifier is derived by combining the training results of each training session (Figure 4). The general operation flow of the AdaBoost algorithm is listed in Table 1.

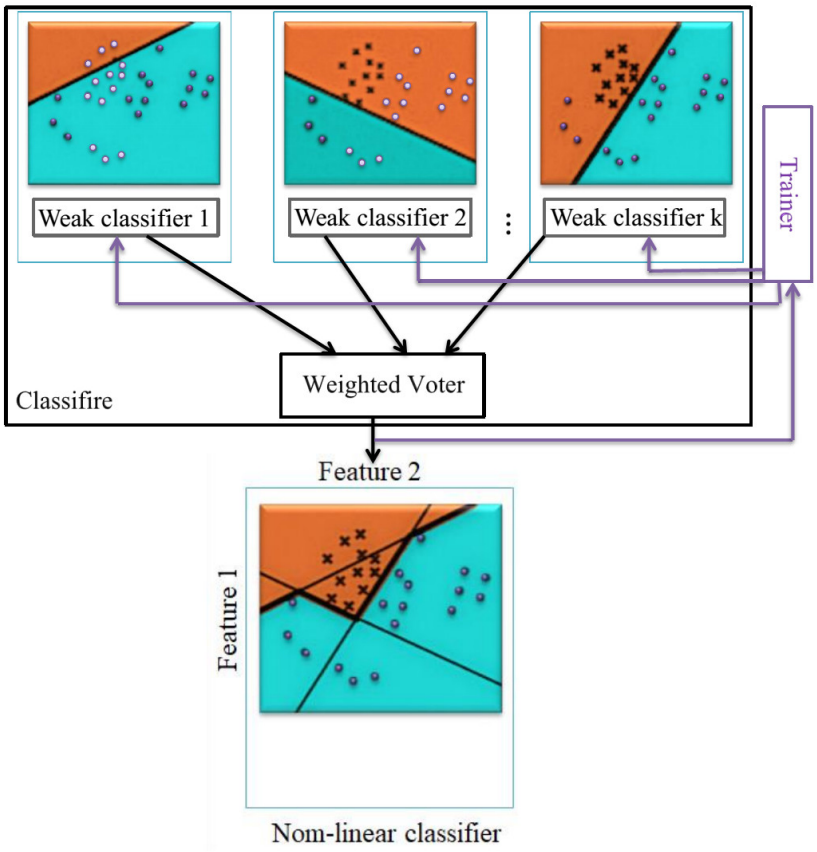


Figure 4. Process of generation of a strong classifier through a combination of weak classifiers with different weights in an AdaBoost algorithm.

Table 1. General outline of AdaBoost algorithm operation [77].

1.	Assessing weight coefficient $\left\{W_n = \frac{1}{n}\right\}$ to each dataset.
2.	Choosing weak learner Y (it can be CART, RF, etc) and start to train the weak learner with weight coefficient
3.	For m = 1, . . . , m: (m = number of trials) while $f_m \neq 0$ Accompanied with minimizing the weight of errors, try to fit hm(x) with training data and redistribute the weights using the below equation: $\overline{D}_m(i) = (1 - D_m(i))\overline{Z}_m$ where $\overline{Z}_m$ is a normalizing coefficient Compute probability values: $P_m^{+1}(x) = P_{D_m}(y = +1 \cap \ h_m(x))$ $\overline{P}_m^{+1}(x) = P_{\overline{D}_m}(y = +1 \cap \ h_m(x))$ $P_m^{-1}(x) = P_{D_m}(y = -1 \cap \ h_m(x))$ $\overline{P}_m^{-1}(x) = P_{\overline{D}_m}(y = -1 \cap \ h_m(x))$ the sign of $h_m(x)$ gives the classification, and $\mid h_m(x) \mid$ a measure of classification "confidence". Calculate $f_m(x) = \left(P_m^{+1}\left(1 - \overline{P}_m^{+1}\right) - P_m^{-1}\left(1 - \overline{P}_m^{-1}\right)\right)(x)$ Update the distribution $D_{m+1}(i) = \frac{D_m(i)\exp(-y_i f_m(x_i))}{Z_m}$
4.	Generate new classifier: $F(x) = sign\left[\sum_{m=1}^{m} f_m(h_m(x))\right]$

Furthermore, some unnecessary training data, such as outlier features, would be eliminated by implementing the AdaBoost classifier. Hence, as a result of continuous training, the classification ability would improve by reducing both bias and variance errors. In comparison with XGBoost and RF, AdaBoost has certain benefits and drawbacks. Hyperparameters of AdaBoost are less than XGBoost; therefore, the required time for tuning AdaBoost is less than XGBoost. However, AdaBoost is sensitive to noisy data, which results in poor performance and huge time consumption in irrelevant feature learning in extreme cases, especially in time series analysis [78].

2.5. Performance Indices

This paper develops four models, i.e., DT, RF, XGBoost, and AdaBoost, to forecast flyrock induced by blasting. To conduct the modelling, flyrock datasets were randomly divided into training (75% of the datasets) and testing (25% of the datasets). Several renowned indices were used to analyze the performance of each model, including R2, root means square error (RMSE), variance adjustment factor (VAF), and the A-10 index. These indices, which were used by other researchers [79–84], are presented as follows:

$$R^2 = 1 - \frac{\sum \left(y_{act} - y_{pre}\right)^2}{\sum \left(y_{act} - \overline{y}_{act}\right)^2} \quad (1)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left(y_{pre} - y_{act}\right)^2}{n}} \quad (2)$$

$$VAF = \left[1 - \frac{var\left(y_{act} - y_{pre}\right)}{var\left(y_{act}\right)}\right] \quad (3)$$

$$a10 - index = \frac{m10}{N} \quad (4)$$

where y_{act} and y_{pre} are measured and predicted values, respectively. N is the total number of datasets, and $m10$ is the number of samples with values of the rate measured/predicted value (range between 0.9–1.1).

3. Case Study

3.1. Mine Description

One of the largest porphyry copper mines in Iran is the Sungun Copper Mine, which is located in the northwest, about 125 km northeast of Tabriz, in East Azarbaijan province. The exact location of this mine is shown in Figure 5, which is at 46′43° E longitude and 38′ 42° N latitude. The mine's location is 2000 m above sea level.

The main geological event in this area is a hydrothermal intrusion, and the mineralization of Sungun Copper Mine refers to the Cenozoic Sahand–Bazman orogenic belt hosted by altered quartz–monzonite rocks. Chalcopyrite, pyrite, chalcocite, cuprite, and malachite are the primary minerals that make copper the main product of the Sungun Copper Mine. In addition, the grade trend in this deposit has a direct increasing relationship with depth, especially in the hypo-gene zone. Additionally, gold, silver, and molybdenite are all extractible from this mine. The geological and geometrical features of this mine are indicated in Table 2.

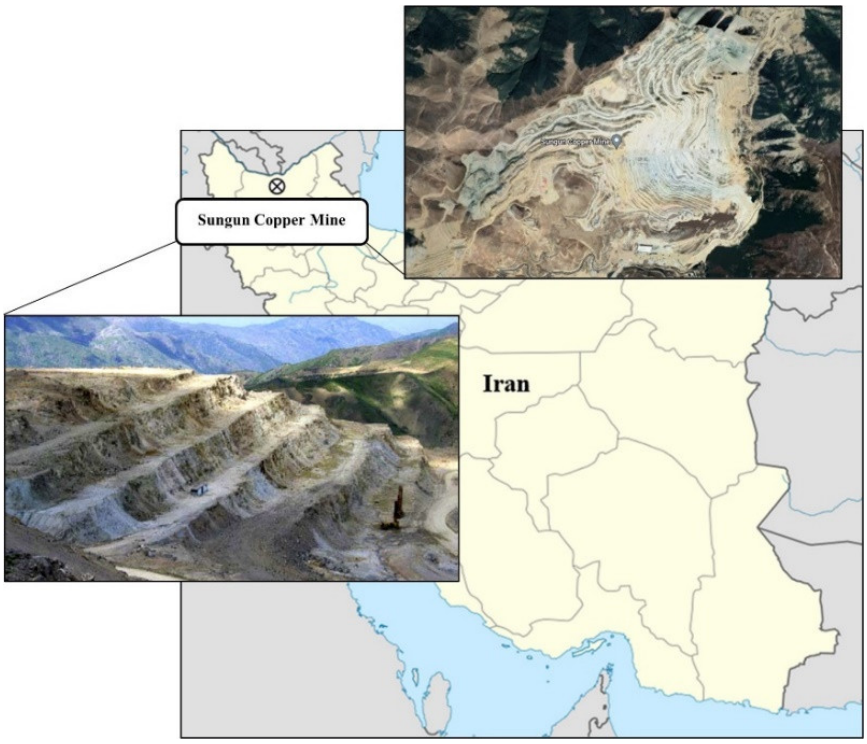


Figure 5. Sungun Copper Mine.

Table 2. Geological and geometrical features of Sungun Copper Mine [85,86].

Geological/Geometrical Properties	Value
Geological reserve of the deposit	796 MT
Proved reserve	410 MT
Average grade	0.67%
Height of the working benches	12.5 m
Slope of the working benches	68′
Angle of the overall pit slope	37′
Width of the ramp	30 m
Slope of the ramp	5′
Age of the mine	about 32 years
Overall stripping ratio (W/O)	1.7

In the Sungun Copper Mine, the extraction process uses bench blasting with one free face. ANFO was used as the main explosive material in blastholes, and drill-cutting particles were used for stemming holes. The initial system of blasting operations was the detonating cord. The main blasting pattern was triangular (staggered), and the ratio of burden to spacing depended on the properties of the blasting block in diverse parts of the mine. The blasting sequence is a flat face method, in which, in this sequence pattern, zero considers as an inter-hole delay time, and a delay event occurs between blasting rows [87]. One of the most undesirable results of the blasting operations in this mine is flyrock. Therefore, this study was planned to be investigated to develop a model for estimating flyrock distance.

3.2. Collected Dataset

To provide a dataset to construct predictive tree-based models, 234 blasting events were considered, and all features, including hole length, spacing, burden, powder factor, specific drilling, and flyrock, were recorded for these events. Figure 6 shows the relationships between the parameters of the database. The upper part of this figure shows the R^2 between the dataset parameters, and the lower part shows the trend between every two parameters

of the dataset. As shown in Figure 6, flyrock shows an inverse relationship with hole depth, spacing, burden, and stemming; however, its relationship with the powder factor is direct. It is important to note that the powder factor has the highest correlation with flyrock, among others. Considering the upper part of Figure 6, it can be observed that the powder factor does not have an appropriate R^2 with other influential parameters on flyrock. This low R^2 value may indicate an inefficient blasting design. Hence, it would be better to study the design procedure for blasting in future studies to reach efficient models from an economic point of view because of blasting powder's high price.

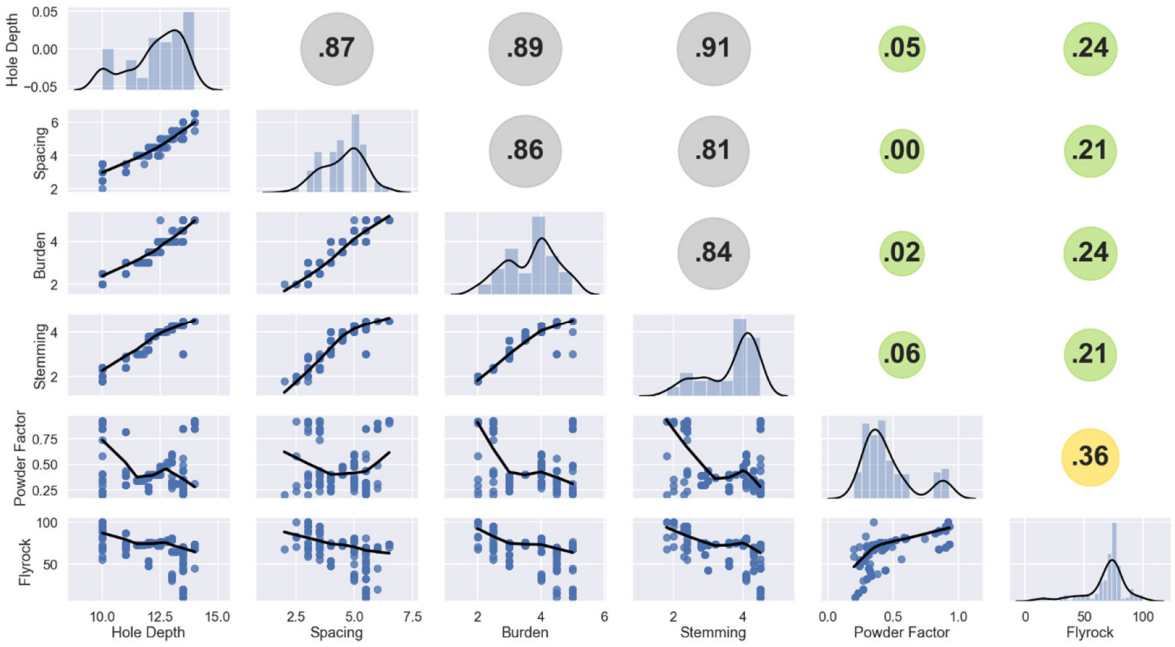


Figure 6. Scatterplot matrix of flyrock parameters' dataset with determination coefficient.

Table 3 presents the basic descriptive statistics of all parameters (input and output). These statistical indices show an acceptable range and distribution for each of the parameters, which are also observable in the histograms of Figure 6. Meanwhile, one parameter that was mentioned as an important parameter in the prediction of flyrock is the ratio of Stemming over Burden [88,89]. However, the R^2 between this parameter and flyrock in these datasets was calculated as 0.005, which does not show any meaningful relationship between this parameter and flyrock. To have a better overview of the used data, a number of ten data samples that were randomly selected are presented in Table 4.

Table 3. Statistical description of parameters used in this study.

Category	Parameter	Unit	Min	Max	Avg	Median	St Deviation
Input	Hole Depth	m	10	14	12.31	12.5	1.18
	Spacing	m	2	6.5	4.53	4.50	0.90
	Burden	m	2	5	3.69	4	0.82
	Stemming	m	1.8	4.5	3.66	4	0.76
	Powder Factor	kg/m^3	0.2	0.93	0.46	0.4	0.20
Output	Flyrock	m	10	100	68.68	73.5	17.42

Table 4. The 10 randomly selected data samples out of the whole database were used in this study.

Number	Hole Depth (m)	Spacing (m)	Burden (m)	Stemming (m)	Powder Factor (kg/m^3)	Flyrock (m)
1	13.5	5.5	5	4.5	0.23	15
2	13	5	4	4.1	0.33	36
3	12	4	3	3.6	0.24	48
4	10	3	2.5	2.3	0.20	59
5	13.4	5.5	4	4.3	0.31	65
6	14	6.5	5	4.5	0.90	70
7	11.5	4	3	3	0.34	73
8	12.1	4.5	3.5	3.8	0.40	75
9	11	3	2.5	2.4	0.82	87
10	10	3.5	2.5	2.4	0.93	95

4. Modeling Procedure

In this section, the modelling procedure for each algorithm in estimating flyrock will be discussed. Each sub-section will be elaborated on the relative parameters, optimal values of the effective parameters, and their importance in reaching the highest performance predictions.

4.1. DT

The learning results with the DT algorithm depend on the algorithm's parameters chosen to optimize the learning process. DT was developed using the SciKit library in Python, and the Graphviz library was used for visualization. There were three splitting criteria available to measure the quality of splits (i) mean square error (MSE), which is the same as variance reduction as a criterion for feature selection, (ii) Friedman-MSE, to assess potential splits, Friedman's improvement score is used with the mean squared error [90,91], and (iii) Poisson, a process that finds splits by reducing Poisson deviance [92]. Because there is no specific rule to choose between the splitting criteria, a trial and error procedure is needed for every problem to select the best one. Hence, after a trial and error process, the MSE criterion was chosen as the best splitting criterion.

In terms of splitting internal nodes, the minimum sample split represents the minimum number of samples required to break an internal node. Because of the importance of this parameter, we specified it as equal to two and allowed the tree to grow and reach the highest accuracy. There must be a minimum number of samples at the terminal node, as specified in the minimum sample leaf. To allow the tree to grow as accurately as possible, this parameter was set to one. To prevent a tree from overgrowing and overtraining, tree depth (TD) plays an important role. In most studies, this parameter is chosen as one of the main parameters to select the best tree among developed trees [93]. By increasing the TD, the danger of the algorithm's memorization of data will increase; therefore, it would be better to use a lower depth. Using the mentioned parameters, a DT was developed for flyrock prediction. The optimal values of DT parameters for the generation of flyrock induced by blasting are shown in Table 5. In addition, Figure 7 shows the visualization of the trained DT for flyrock prediction.

Table 5. The optimal values of DT parameters for generation flyrock.

Parameter	Range	Value
criterion	[MSE, friedman_mse, poisson]	MSE
Minimum sample split	[2, 3, 4, 5, 6]	2
max depth	[2, 3, 4, 5, 6, 7, 8, 9, 10]	4
Minimum samples leaf	[1–∞]	1

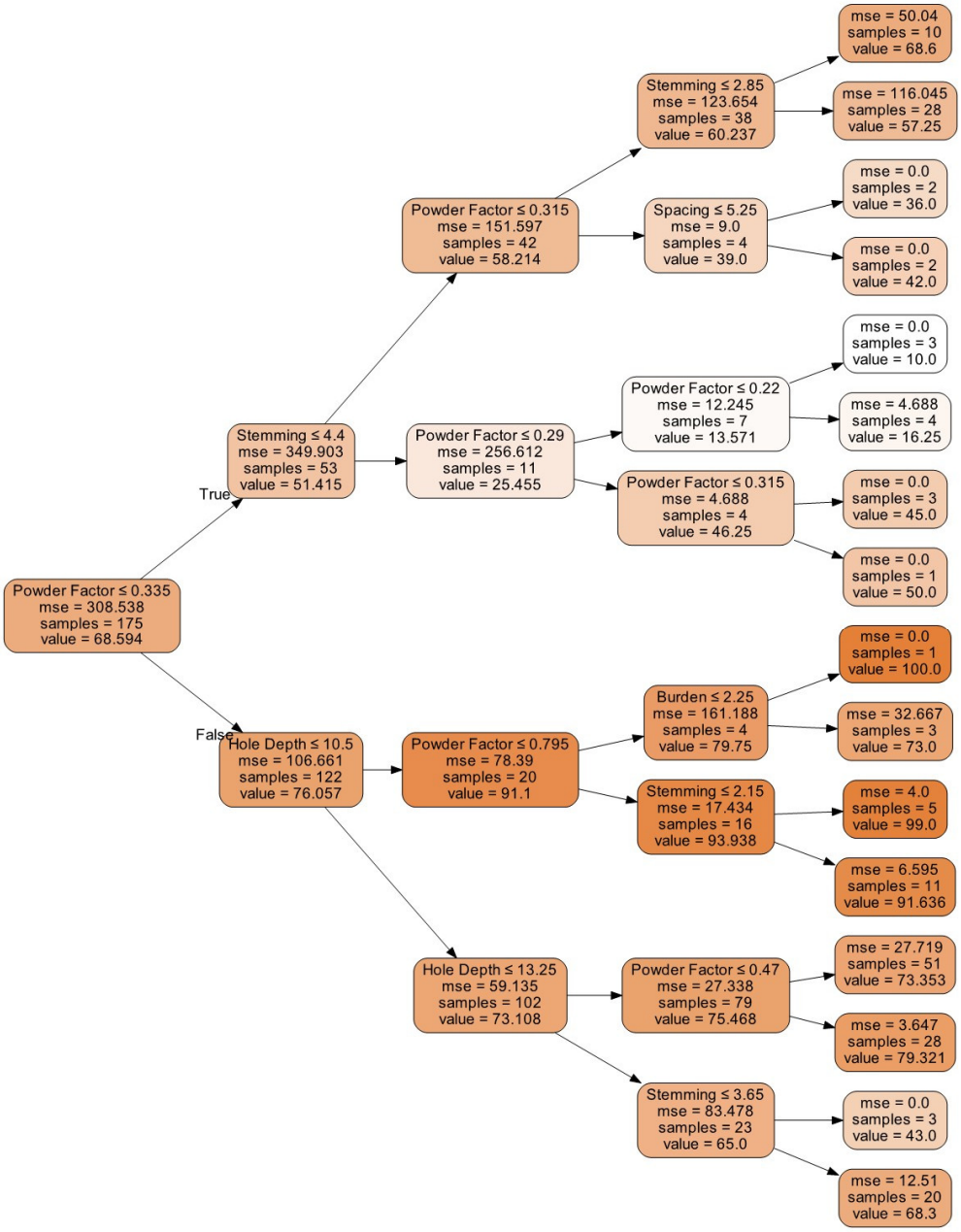


Figure 7. The tree structure of the DT model for flyrock prediction.

Using an ad-hoc ranking, DT can evaluate the input variables based on their importance. Using this ranking system, more important variables become closer to the root node and play an important role in tree splitting and predicting target values.

4.2. RF

To develop an RF model for flyrock prediction, the Bootstrap sampling system was first performed on the training data samples. As mentioned before, the RF algorithm generates a wide scope of DT for prediction and then, using an averaging system, tries to reach the best prediction for an observation. To assess the performance of the trained model, the OOB evaluation of training datasets was set to be true in the Python SciKit package code to use the samples that were not included in the bootstrapping process to construct the trees as evaluation samples.

The number of estimators represents the number of DTs generated during the RF run. The more estimators, the more variable importance evaluations there are. Although higher

estimator numbers increase the algorithm's run time, they lead to more stable results [94]. The number of estimators is chosen to be equal to 200. A comprehensive trial-and-error process was performed to reach a number for TD. Although deeper trees result in better performance, they can cause overfitting of the model. For information gain, the entropy criterion displayed better results than Gini. Other influential features and their optimum values are presented in Table 6.

Table 6. RF Hyperparameters for estimating flyrock.

Parameter	Range	Value
Criterion	[Entropy, Gini]	Entropy
Estimators number	[50, 100, 200, 300]	200
Bootstrap	[True, False]	True
Max depth	[3, 4, 5, 6, 7]	4
Max features	[sqrt, log2]	Sqrt
OOB score	[True, False]	True

4.3. XGBoost

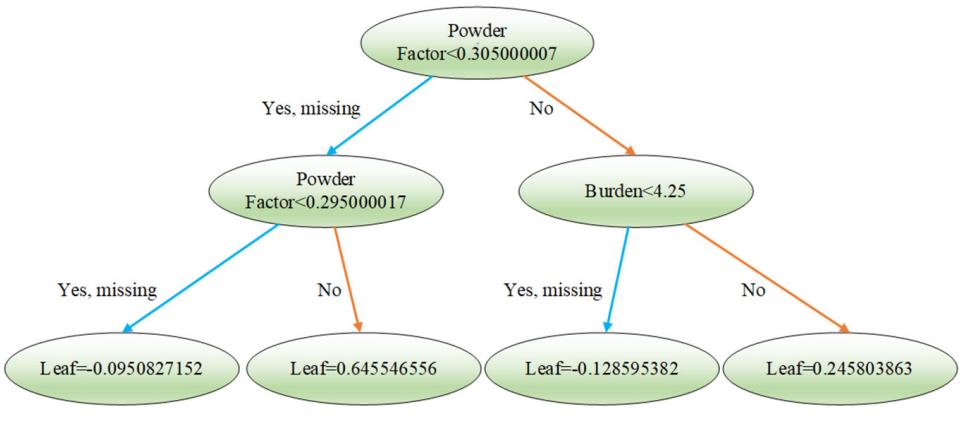
Although RF is renowned as a powerful prediction tool, it has some drawbacks. The main defect of RF is that if it initially makes a mistake in all trees, the algorithm will repeat this mistake in its next iterations. To this end, ensemble methods such as XGBoost and AdaBoost are developed to address the problems by their nature and to learn from the mistakes of trees to overcome the shortcomings and improve the results considerably.

In this section, the XGBoost model was developed in order to determine flyrock distance. For learning a full tree, Scikit-learn and XGBoost Python packages were used. To develop an XGBoost model, hyperparameters should be considered as the main factors during model development. These parameters are the number of features, the number of trees, the maximum depth of trees, whether to bootstrap, how many samples should be left in a node before the split, and how many samples should remain in the leaf node at the end. Generally, to reduce the probability of overfitting, it is necessary to avoid complexity by increasing the values of hyperparameters. Moreover, accompanied by performance maximization, it is important to consider parameter tuning to prevent overtraining and generating too complex models. After a trial and error procedure, the optimal values for XGBoost were acquired. Table 7 displays the optimum XGBoost parameters for predicting flyrock caused by mine blasting.

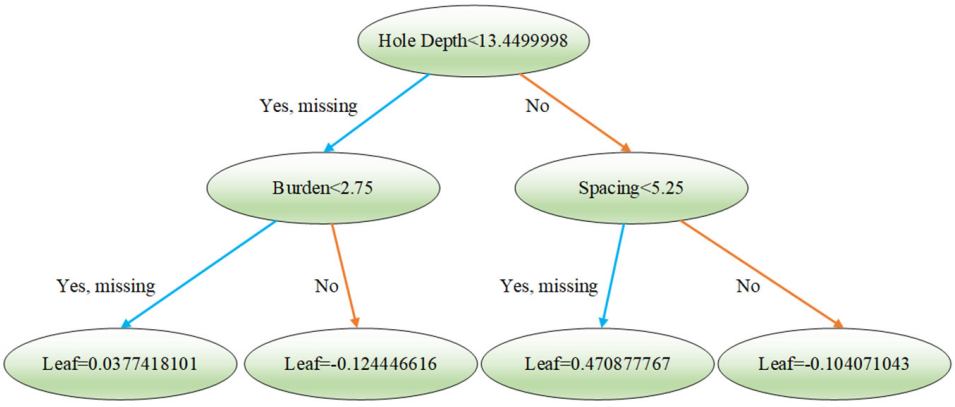
Table 7. Optimum XGBoost algorithm parameters for flyrock prediction.

Parameter	Range	Value
learning rate (eta)	[0.1, 0.2, 0.3, 0.5, 1, 1.5, 2, 3]	0.3
number of estimators	[50, 100, 200, 300]	100
max depth	[2, 3, 4, 5, 6]	3
gamma	[0.0001, 0.001, 0.01, 0.1,0.5, 1]	0.001
min child weight	[0.2, 0.5, 0.8, 0.9, 1, 1.5, 2]	0.9
max delta step	[0.2, 0.5, 0.8, 0.9, 1, 1.5, 2]	0.9
booster	[gbtree, gblinear, DART]	DART

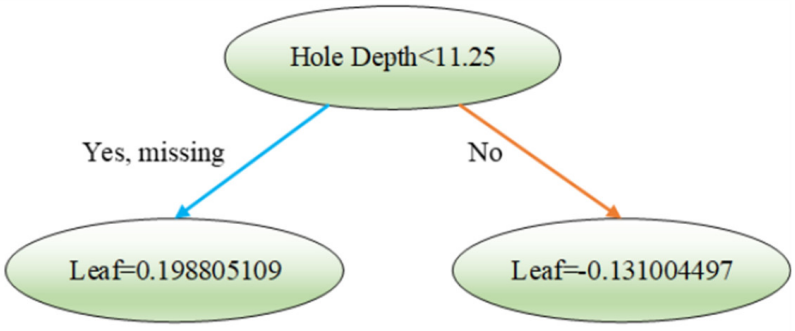
One of the drawbacks of XGBoost is the inconsistency between the speed of trees' generation and the learning from them. This inconsistency causes newly generated trees to receive less importance than older ones. To overcome this drawback, Vinayak and Gilad-Bachrach [95] introduced the DART booster for XGBoost. In this study, the DART booster was applied, and it was found that, compared to other boosting algorithms, its performance is better. Figure 8 shows some of the estimators that were developed by XGBoost for flyrock prediction.



(**a**) estimator 0

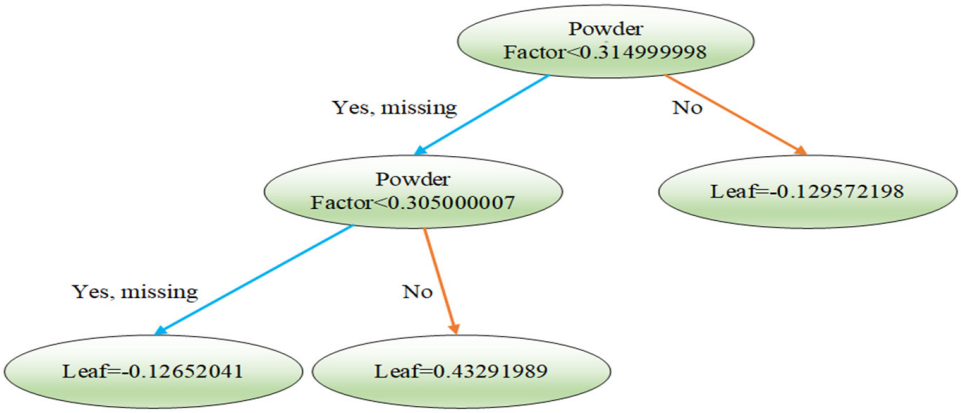


(**b**) estimator 1

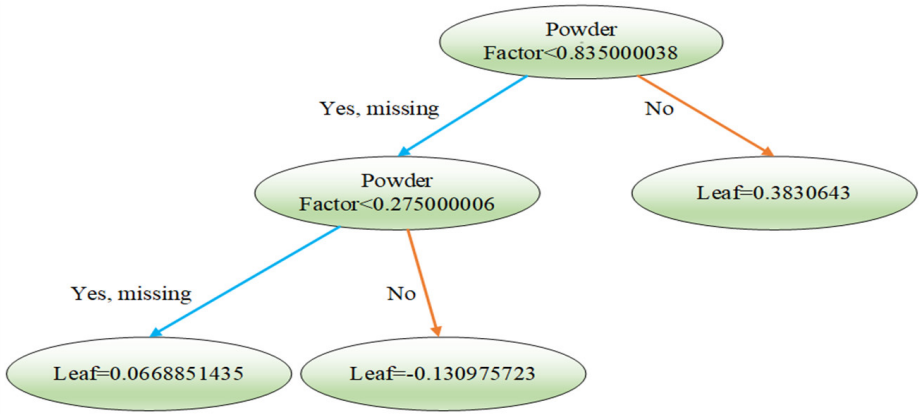


(**c**) estimator 2

Figure 8. *Cont.*



(**d**) estimator 3



(**e**) estimator 4

Figure 8. First five tree estimators developed by XGBoost in predicting flyrock.

4.4. AdaBoost

To develop an AdaBoost model, several features were regulated using a trial and error procedure. First, both CART and RF learners were examined using the AdaBoost algorithm, and it was found that the RF classifier is able to provide more accurate results than the CART model. As previously mentioned, among the two splitting criteria, i.e., entropy and Gini-index, entropy gives more precise results than Gini-index on this database; hence, the entropy criterion is selected for splitting purposes. Another parametric study was performed to acquire the appropriate TD. The lower the TD, the simpler the model; therefore, between sets of [2, 3, 4, 5] TD, it tried to, by modifying other features, reach the lowest TD number. Finally, TD = 3 was selected for model development. Table 8 includes a list of other factors that influenced the development of the AdaBoost model.

One crucial factor in the configuration of an AdaBoost model is the type of algorithm. There are two types of algorithms available for AdaBoost development, namely SAMME and SAMME.R. These two algorithms were introduced by Hastie et al. [96] to overcome the deficiencies of AdaBoost in modelling multi-class problems. They tested these two algorithms using CART weak learners and concluded that the final results were similar. However, as this study used an RF-weak learner, the efficiency of each algorithm was investigated using flyrock datasets. Finally, as shown in Figure 9, the SAMME algorithm gives more accurate models in the boosting process than SAMME.R; therefore, SAMME was selected as the main algorithm in this research.

Table 8. The best AdaBoost model's features in predicting flyrock.

Feature	Range	Value
Weak learner algorithm	Random Forest, Decision Tree	Random Forest Classifier
Tree depth	[2, 3, 4, 5, 6]	3
Algorithm	SAMME, SAMME.R	SAMME
criterion	Entropy, Gini	Entropy
Number of estimators	[50, 100, 200, 300]	100
Learning rate	[0.5, 1, 1.5, 2]	1.0

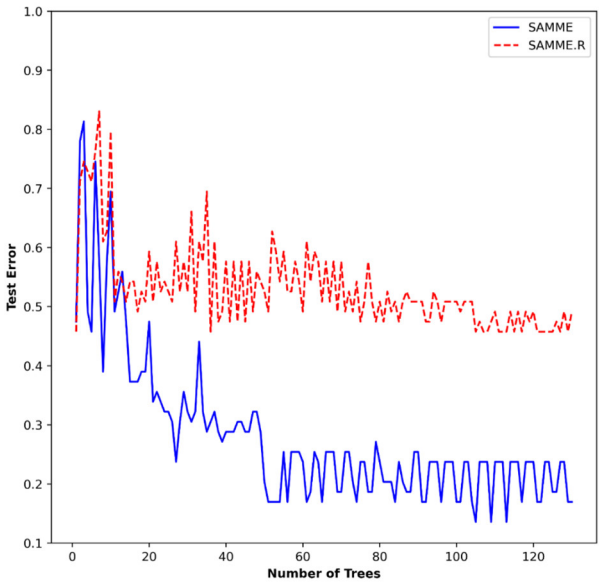


Figure 9. Performance of SAMME and SAMME.R algorithms in AdaBoost model.

5. Results and Discussion

Four tree-based methods were developed in this study, i.e., DT, RF, XGBoost, and AdaBoost, to predict flyrock due to blasting operations. The performance of each model was evaluated by four popular performance indices (i.e., R^2, RMSE, VAF, and the A-10 index). Table 9 shows the results of these tree-based models based on the training and testing phases. The accuracy of the developed models was compared using a ranking system introduced by Zorlu et al. [97]. It is observed from Table 9 that although RF and DT achieved acceptable prediction accuracy, the AdaBoost model, which used the RF classifier in its core algorithm, yielded the best results, with a final rank of 32. Furthermore, XGBoost performed better in prediction than RF and DT; however, its performance compared to the AdaBoost model is poor.

Table 9. Performance indices of the developed models for flyrock prediction.

		R^2		RMSE		VAF		A10		Final Rank
Model Name		**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	
AdaBoost		0.99	0.98	1.47	2.66	99.56	98.87	0.98	0.98	
	Rank score	4	4	4	4	4	4	4	4	32
XGBoost		0.97	0.92	2.91	4.85	97.33	91.99	0.97	0.94	
	Rank score	3	3	3	3	3	3	3	3	24
RF		0.93	0.91	4.32	4.86	92.96	92.12	0.96	0.91	
	Rank score	2	2	2	2	2	2	2	2	16
DT		0.89	0.84	5.72	6.58	89.39	84.81	0.89	0.84	
	Rank score	1	1	1	1	1	1	1	1	8

Although the datasets used in the current study have not been used in any other studies, the authors tried to compare the results with similar studies in the literature.

Monjezi et al. [23] modelled flyrock by GEP using the same input parameters used in the current study. The final results for the best GEP model were R2train = 0.83 and R2test = 0.91. Dehghani and Pourzafar [98] developed another GEP model for flyrock prediction using the same inputs. They received R2train = 0.91 and R2test = 0.84 for their GEP model. In another study, Jamei et al. [99] developed a boosted regression tree (BRT) for flyrock estimation. The highest results for BRT are R2train = 0.96 and R2test = 0.84. In addition, by probing into some other studies that have been conducted in recent years [100–106]. It can be seen that the results of the AdaBoost and XGBoost models developed in the current study to predict flyrock distance are more accurate compared to similar investigations in the literature.

Figure 10 presents the correlation between predicted and measured flyrock for the AdaBoost model's training and testing sets. A coloured shadow was applied to the plots to understand data concentration and distribution better. It is clear that AdaBoost is a powerful technique that is able to predict flyrock values that are very close to the measured ones.

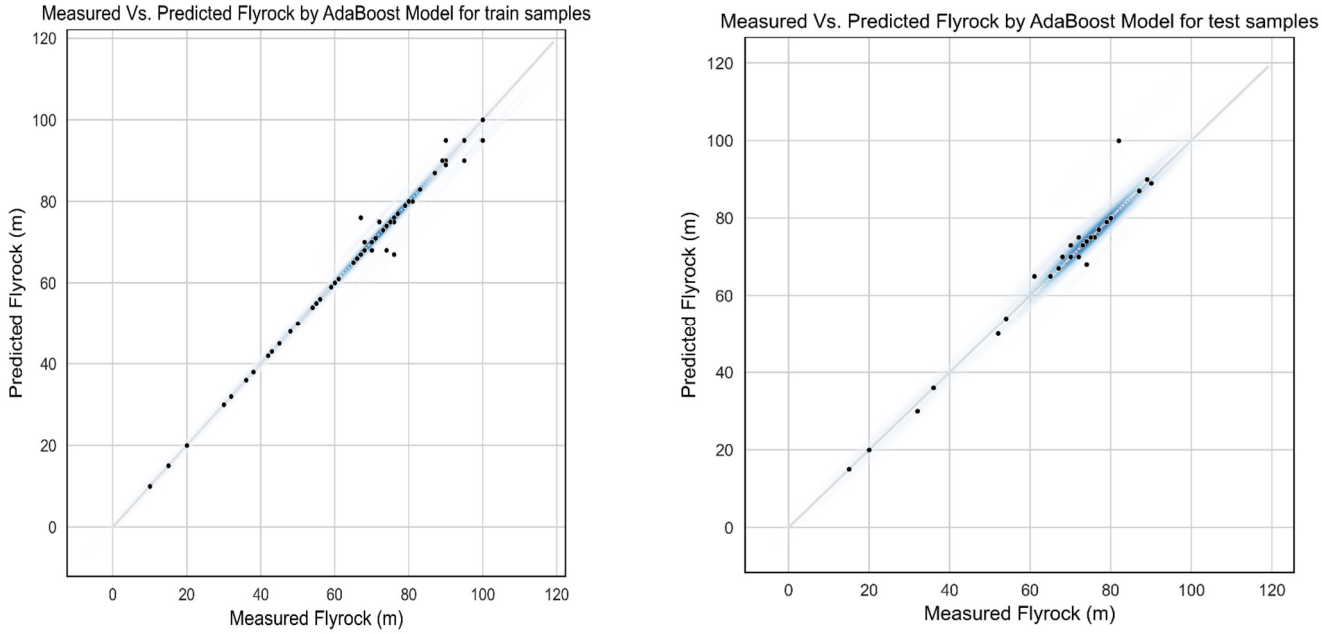


Figure 10. The graphs of measured and predicted flyrock for the training (R^2 = 0.99) and testing (R^2 = 0.97) datasets.

Figure 11 shows the model variables' importance results of four tree-based techniques, i.e., AdaBoost, XGBoost, RF, and DT. It can be seen that it is critical to determine the powder factor for all tree-based models. However, different results were obtained for the rest of the parameters using the linear regression analysis (see Figure 6). In this analysis, stemming, hole length, spacing, and burden showed the highest to lowest importance in model development, respectively. As shown in Figure 11d, spacing and burden are detected as the least important parameters by the DT technique. It is essential to note that these two parameters were repeated only once in the tree structure by DT (see Figure 7). Accordingly, powder factor, stemming, and hole depth have the highest repetition rate in the tree structure, respectively. Burden received an average level of importance considering all tree-based models. However, the lowest importance was reported by the DT model. Since DT is considered a base tree model (benchmark tree-based model), it was not able to receive a high-performance prediction during the training and testing phases. Therefore, the importance of input parameters seems to be different from the other three techniques. The results obtained from this part can be further supported by previous studies in the field [11,107].

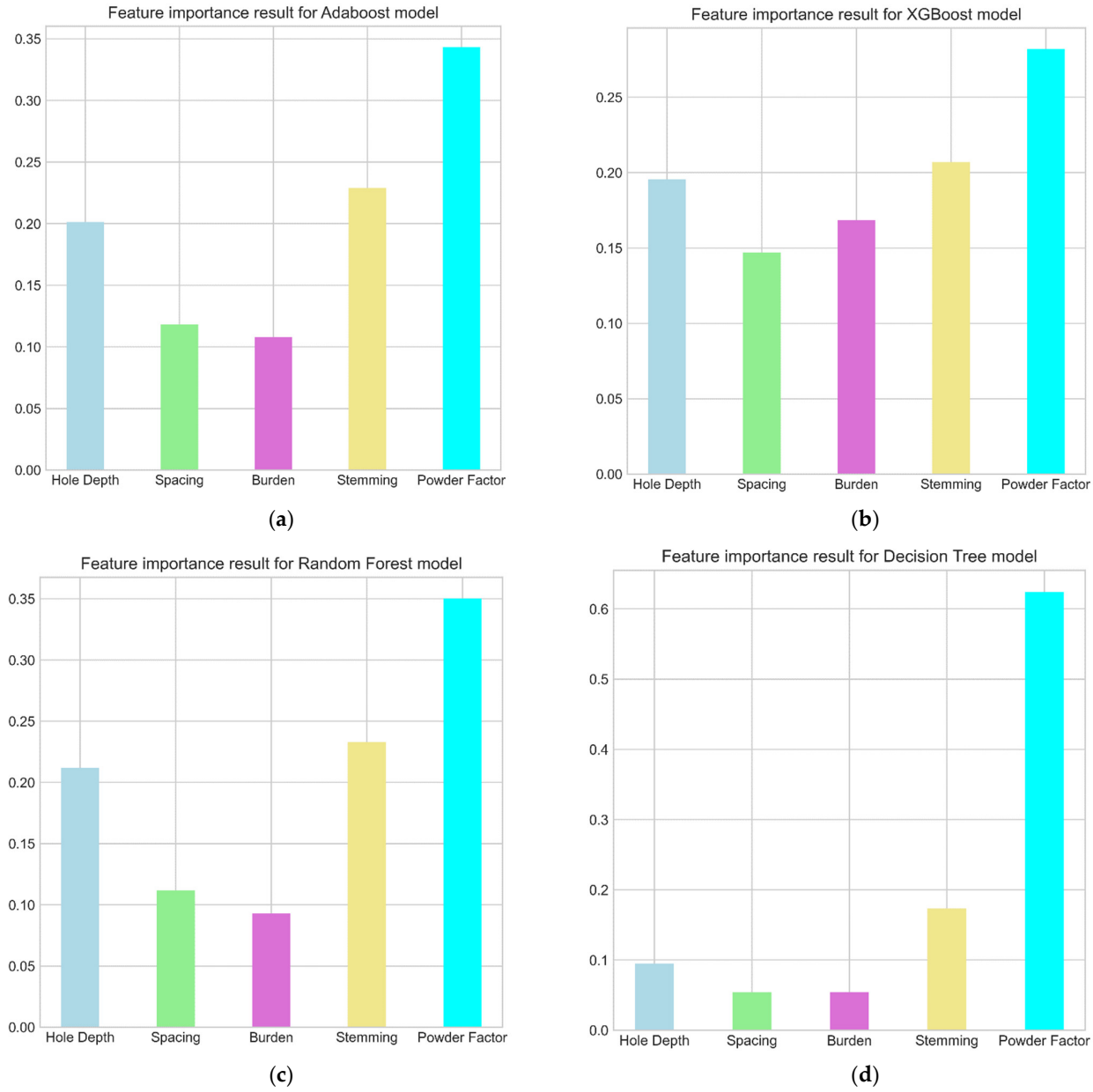


Figure 11. Feature importance analysis (**a**) AdaBoost, (**b**) XGBoost, (**c**) RF, and (**d**) DT.

6. Conclusions

This study aimed to develop tree-based predictive models for flyrock due to mine blasting. Four models, i.e., DT, RF, XGBoost, and AdaBoost, was developed to predict flyrock considering several parametric studies. These parametric studies were carried out to determine the best hyperparameters for each tree-based model. It was found that although DT and RF are among the most powerful and popular predictive models, the final results of booster algorithms (i.e., XGBoost and AdaBoost) are far better than theirs. Their robustness is because of their nature to learn from previous mistakes compared to the semi-stochastic nature of DT and RF. However, when considering results obtained by the AdaBoost and XGBoost models, the AdaBoost model with R^2train = 0.99 and R^2test = 0.98 showed better performance. In addition, the overall performance of each model was examined using three other performance indices (RMSE, VAF, and A10-index). As a result, the AdaBoost model's high accuracy in prediction is proved using the other indices. XGBoost used DT as a weak learner in an incremental learning process. There are similarities between these trees and DT; however, XGBoost combined the errors of the trees and attempted

to reduce the overall error. This procedure allows XGBoost to grow the tree and learn from the previous iterations without increasing the error rate. Finally, by looking at the previously developed model using AdaBoost and XGBoost, it was concluded that AdaBoost models perform better in low-noise datasets. This technique could be chosen when it is not important whether the results are timely or computationally complex and knowledgeable users and enough time for tuning the hyperparameters are not available. However, the lack of visualization, tuning multiple hyperparameters, and sensitivity to noisy data and outliers make XGBoost and AdaBoost complex models.

It is crucial to note that the models developed in this study are based on the blasting operations collected from one case study with limited geological conditions. Therefore, the results are only valid for further prediction if the geological conditions, blasting pattern parameters, and other important parameters are the same or very similar to those presented in this study. In addition, if inputs beyond the ranges mentioned in this research are available, the results will have larger errors. To improve performance prediction, theory-guided machine learning, which uses theories and empirical equations in the data preparation stage, can be used. These techniques may include a better understanding of the use of machine learning in mining environmental issues for a mining or civil engineer. Of course, these models can help with model generalization, which is one of the drawbacks of AI models, at least in engineering fields.

Author Contributions: Conceptualization, M.Y., P.G.A., D.J.A. and E.T.M., methodology, M.Y., P.G.A. and D.J.A., software, M.Y., P.G.A. and D.J.A., formal analysis, C.M., M.Y., P.G.A. and D.J.A., data curation, M.Y., writing—original draft preparation, C.M., M.Y., P.G.A., D.J.A., E.T.M. and A.N.E., writing—review and editing, C.M., M.Y., P.G.A., D.J.A., E.T.M. and A.N.E., supervision, E.T.M. and P.G.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data is available upon request.

Acknowledgments: The authors of this study want to thank the Sungun Copper Mine's management and mining engineers for their help in conducting this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bhandari, S. *Engineering Rock Blasting Operations*; A.A. Balkema: Rotterdam, The Netherlands, 1997; 388p.
2. Yari, M.; Bagherpour, R.; Jamali, S. Development of an evaluation system for blasting patterns to provide efficient production. *J. Intell. Manuf.* **2017**, *28*, 975–984. [CrossRef]
3. Harandizadeh, H.; Armaghani, D.J. Prediction of air-overpressure induced by blasting using an ANFIS-PNN model optimized by GA. *Appl. Soft Comput.* **2020**, *99*, 106904. [CrossRef]
4. Murlidhar, B.R.; Nguyen, H.; Rostami, J.; Bui, X.; Armaghani, D.J.; Ragam, P.; Mohamad, E.T. Prediction of flyrock distance induced by mine blasting using a novel Harris Hawks optimization-based multi-layer perceptron neural network. *J. Rock Mech. Geotech. Eng.* **2021**, *13*, 1413–1427. [CrossRef]
5. Monjezi, M.; Amiri, H.; Farrokhi, A.; Goshtasbi, K. Prediction of rock fragmentation due to blasting in Sarcheshmeh copper mine using artificial neural networks. *Geotech. Geol. Eng.* **2010**, *28*, 423–430. [CrossRef]
6. Monjezi, M.; Amini Khoshalan, H.; Yazdian Varjani, A. Optimization of open pit blast parameters using genetic algorithm. *Int. J. Rock Mech. Min. Sci.* **2011**, *48*, 864–869. [CrossRef]
7. Chen, L.; Asteris, P.G.; Tsoukalas, M.Z.; Armaghani, D.J.; Ulrikh, D.V.; Yari, M. Forecast of Airblast Vibrations Induced by Blasting Using Support Vector Regression Optimized by the Grasshopper Optimization (SVR-GO) Technique. *Appl. Sci.* **2022**, *12*, 9805. [CrossRef]
8. Bajpayee, T.; Verakis, H.; Lobb, T. An analysis and prevention of flyrock accidents in surface blasting operations. In Proceedings of the Annual Conference on Explosives and Blasting Technique, New Orleans, LA, USA, 1–4 February 2004; ISEE: Herndon, VA, USA, 2004; Volume 2, pp. 401–410.
9. Yari, M.; Monjezi, M.; Bagherpour, R.; Jamali, S. Developing a mathematical assessment model for blasting patterns management: Sungun copper mine. *J. Central South Univ.* **2014**, *21*, 4344–4351. [CrossRef]

10. Yari, M.; Monjezi, M.; Bagherpour, R. Selecting the most suitable blasting pattern using AHP-TOPSIS method: Sungun copper mine. *J. Min. Sci.* **2013**, *49*, 967–975. [CrossRef]
11. Armaghani, D.; Mohamad, E.; Hajihassani, M. Evaluation and prediction of flyrock resulting from blasting operations using empirical and computational methods. *Eng. Comput.* **2016**, *32*, 109–121. [CrossRef]
12. Langefors, U.; Kihlström, B. *The Modern Technique of Rock Blasting*; Wiley: New York, NY, UDA, 1963.
13. Monjezi, M.; Bahrami, A.; Varjani, A.Y.; Sayadi, A.R. Prediction and controlling of flyrock in blasting operation using artificial neural network. *Arab. J. Geosci.* **2011**, *4*, 421–425. [CrossRef]
14. Rezaei, M.; Monjezi, M.; Varjani, A. Development of a fuzzy model to predict flyrock in surface mining. *Saf. Sci.* **2011**, *49*, 298–305. [CrossRef]
15. Raina, A.K.; Chakraborty, A.K.; Ramulu, M.; Sahu, P.B.; Haldar, A.; Choudhury, P.B. Flyrock prediction and control in opencast mines: A critical appraisal. *Min. Eng. J.* **2004**, *6*, 10–20.
16. Lundborg, N.; Persson, N.; Ladegaard-Pedersen, A. Keeping the lid on flyrock in open-pit blasting. *Eng. Min. J.* **1975**, *176*, 95–100.
17. Olofsson, S. *Applied Explosives Technology for Construction and Mining*; Applex Publisher: Arla, France, 1990.
18. Trivedi, R.; Singh, T.; Raina, A. Prediction of blast-induced flyrock in Indian limestone mines using neural networks. *J. Rock Mech. Geotech. Eng.* **2014**, *6*, 447–454. [CrossRef]
19. Hustrulid, W.A. *Blasting Principles for Open Pit Mining: General Design Concepts*; Balkema: Boca Raton, FL, USA, 1999; ISBN 9054104589.
20. Monjezi, M.; Dehghan, J.A.H.; Samimi, N.F. Application of TOPSIS method in controlling fly rock in blasting operations. In Proceedings of the 7-th International Multidisciplinary Scientific GeoConference SGEM 2007, Albena, Bulgaria, 11–15 June 2007; p. 41.
21. Raina, A.K.; Chakraborty, A.K.; Choudhury, P.B.; Sinha, A. Flyrock danger zone demarcation in opencast mines: A risk based approach. *Bull. Eng. Geol. Environ.* **2011**, *70*, 163–172. [CrossRef]
22. Armaghani, D.J.; Mahdiyar, A.; Hasanipanah, M.; Faradonbeh, R.S.; Khandelwal, M.; Amnieh, H.B. Risk Assessment and Prediction of Flyrock Distance by Combined Multiple Regression Analysis and Monte Carlo Simulation of Quarry Blasting. *Rock Mech. Rock Eng.* **2016**, *49*, 1–11. [CrossRef]
23. Monjezi, M.; Dehghani, H.; Shakeri, J.; Mehrdanesh, A. Optimization of prediction of flyrock using linear multivariate regression (LMR) and gene expression programming (GEP)—Topal Novin mine, Iran. *Arab. J. Geosci.* **2021**, *14*, 1–12. [CrossRef]
24. Faradonbeh, R.S.; Armaghani, D.J.; Monjezi, M. Development of a new model for predicting flyrock distance in quarry blasting: A genetic programming technique. *Bull. Eng. Geol. Environ.* **2016**, *75*, 993–1006. [CrossRef]
25. Jahed Armaghani, D.; Hajihassani, M.; Monjezi, M.; Mohamad, E.T.; Marto, A.; Moghaddam, M.R. Application of two intelligent systems in predicting environmental impacts of quarry blasting. *Arab. J. Geosci.* **2015**, *8*, 9647–9665. [CrossRef]
26. Yari, M.; Bagherpour, R.; Jamali, S.; Shamsi, R. Development of a novel flyrock distance prediction model using BPNN for providing blasting operation safety. *Neural Comput. Appl.* **2016**, *27*, 699–706. [CrossRef]
27. Ghasemi, E.; Amini, H.; Ataei, M.; Khalokakaei, R. Application of artificial intelligence techniques for predicting the flyrock distance caused by blasting operation. *Arab. J. Geosci.* **2014**, *7*, 193–202. [CrossRef]
28. Faradonbeh, R.S.; Armaghani, D.J.; Monjezi, M.; Mohamad, E.T. Genetic programming and gene expression programming for flyrock assessment due to mine blasting. *Int. J. Rock Mech. Min. Sci.* **2016**, *88*, 254–264. [CrossRef]
29. Manoj, K.; Monjezi, M. Prediction of flyrock in open pit blasting operation using machine learning method. *Int. J. Min. Sci. Technol.* **2013**, *23*, 313–316. [CrossRef]
30. Murlidhar, B.R.; Kumar, D.; Jahed Armaghani, D.; Mohamad, E.T.; Roy, B.; Pham, B.T. A Novel Intelligent ELM-BBO Technique for Predicting Distance of Mine Blasting-Induced Flyrock. *Nat. Resour. Res.* **2020**, *29*, 4103–4120. [CrossRef]
31. Monjezi, M.; Khoshalan, H.A.; Varjani, A.Y. Prediction of flyrock and backbreak in open pit blasting operation: A neuro-genetic approach. *Arab. J. Geosci.* **2012**, *5*, 441–448. [CrossRef]
32. Saghatforoush, A.; Monjezi, M.; Faradonbeh, R.S.; Armaghani, D.J. Combination of neural network and ant colony optimization algorithms for prediction and optimization of flyrock and back-break induced by blasting. *Eng. Comput.* **2016**, *32*, 255–266. [CrossRef]
33. Armaghani, D.J.; Koopialipoor, M.; Bahri, M.; Hasanipanah, M.; Tahir, M.M. A SVR-GWO technique to minimize flyrock distance resulting from blasting. *Bull. Eng. Geol. Environ.* **2020**, *79*, 643715. [CrossRef]
34. Marto, A.; Hajihassani, M.; Jahed Armaghani, D.; Tonnizam Mohamad, E.; Makhtar, A.M. A novel approach for blast-induced flyrock prediction based on imperialist competitive algorithm and artificial neural network. *Sci. World J.* **2014**, *2014*, 1–11. [CrossRef]
35. Asteris, P.G.; Lourenço, P.B.; Roussis, P.C.; Adami, C.E.; Armaghani, D.J.; Cavaleri, L.; Chalioris, C.E.; Hajihassani, M.; Lemonis, M.E.; Mohammed, A.S. Revealing the nature of metakaolin-based concrete materials using artificial intelligence techniques. *Constr. Build. Mater.* **2022**, *322*, 126500. [CrossRef]
36. Mahmood, W.; Mohammed, A.S.; Asteris, P.G.; Kurda, R.; Armaghani, D.J. Modeling Flexural and Compressive Strengths Behaviour of Cement-Grouted Sands Modified with Water Reducer Polymer. *Appl. Sci.* **2022**, *12*, 1016. [CrossRef]
37. Yang, H.; Song, K.; Zhou, J. Automated Recognition Model of Geomechanical Information Based on Operational Data of Tunneling Boring Machines. *Rock Mech. Rock Eng.* **2022**, *55*, 1499–1516. [CrossRef]

38. Yang, H.; Wang, Z.; Song, K. A new hybrid grey wolf optimizer-feature weighted-multiple kernel-support vector regression technique to predict TBM performance. *Eng. Comput.* **2020**, *38*, 2469–2485. [CrossRef]
39. Asteris, P.G.; Mamou, A.; Ferentinou, M.; Tran, T.-T.; Zhou, J. Predicting clay compressibility using a novel Manta ray foraging optimization-based extreme learning machine model. *Transp. Geotech.* **2022**, *37*, 100861. [CrossRef]
40. Hamzehkolaei, N.S.; Alizamir, M. Performance Evaluation of Machine Learning Algorithms for Seismic Retrofit Cost Estimation using Structural Parameters. *J. Soft Comput. Civ. Eng.* **2021**, *5*, 32–57. [CrossRef]
41. Ghanizadeh, A.R.; Heidarabadizadeh, N.; Heravi, F. Gaussian Process Regression (GPR) for Auto-Estimation of Resilient Modulus of Stabilized Base Materials. *J. Soft Comput. Civ. Eng.* **2021**, *5*, 80–94. [CrossRef]
42. Armaghani, D.J.; Asteris, P.G.; Fatemi, S.A.; Hasanipanah, M.; Tarinejad, R.; Rashid, A.S.A.; Huynh, V. Van On the Use of Neuro-Swarm System to Forecast the Pile Settlement. *Appl. Sci.* **2020**, *10*, 1904. [CrossRef]
43. Ikram, R.M.A.; Ewees, A.A.; Parmar, K.S.; Yaseen, Z.M.; Shahid, S.; Kisi, O. The viability of extended marine predators algorithm-based artificial neural networks for streamflow prediction. *Appl. Soft Comput.* **2022**, *131*, 109739. [CrossRef]
44. Ikram, R.M.A.; Dai, H.-L.; Chargari, M.M.; Al-Bahrani, M.; Mamlooki, M. Prediction of the FRP reinforced concrete beam shear capacity by using ELM-CRFOA. *Measurement* **2022**, *205*, 112230. [CrossRef]
45. Ikram, R.M.A.; Jaafari, A.; Milan, S.G.; Kisi, O.; Heddam, S.; Zounemat-Kermani, M. Hybridized Adaptive Neuro-Fuzzy Inference System with Metaheuristic Algorithms for Modeling Monthly Pan Evaporation. *Water* **2022**, *14*, 3549. [CrossRef]
46. Adnan, R.M.; Dai, H.-L.; Mostafa, R.R.; Islam, A.R.M.T.; Kisi, O.; Heddam, S.; Zounemat-Kermani, M. Modelling groundwater level fluctuations by ELM merged advanced metaheuristic algorithms using hydroclimatic data. *Geocarto Int.* **2022**, *38*, 1–39. [CrossRef]
47. Asteris, P.G.; Mamou, A.; Hajihassani, M.; Hasanipanah, M.; Koopialipoor, M.; Le, T.-T.; Kardani, N.; Armaghani, D.J. Soft computing based closed form equations correlating L and N-type Schmidt hammer rebound numbers of rocks. *Transp. Geotech.* **2021**, *29*, 100588. [CrossRef]
48. Ikram, R.M.A.; Goliatt, L.; Kisi, O.; Trajkovic, S.; Shahid, S. Covariance Matrix Adaptation Evolution Strategy for Improving Machine Learning Approaches in Streamflow Prediction. *Mathematics* **2022**, *10*, 2971. [CrossRef]
49. Armaghani, D.J.; Harandizadeh, H.; Momeni, E.; Maizir, H.; Zhou, J. An optimized system of GMDH-ANFIS predictive model by ICA for estimating pile bearing capacity. *Artif. Intell. Rev.* **2021**, *55*, 2313–2350. [CrossRef]
50. Parsajoo, M.; Armaghani, D.J.; Mohammed, A.S.; Khari, M.; Jahandari, S. Tensile strength prediction of rock material using non-destructive tests: A comparative intelligent study. *Transp. Geotech.* **2021**, *31*, 100652. [CrossRef]
51. Cavaleri, L.; Barkhordari, M.S.; Repapis, C.C.; Armaghani, D.J.; Ulrikh, D.V.; Asteris, P.G. Convolution-based ensemble learning algorithms to estimate the bond strength of the corroded reinforced concrete. *Constr. Build. Mater.* **2022**, *359*, 129504. [CrossRef]
52. Barkhordari, M.S.; Armaghani, D.J.; Asteris, P.G. Structural Damage Identification Using Ensemble Deep Convolutional Neural Network Models. *Comput. Model. Eng. Sci.* **2022**, *134*, 835–855. [CrossRef]
53. Skentou, A.D.; Bardhan, A.; Mamou, A.; Lemonis, M.E.; Kumar, G.; Samui, P.; Armaghani, D.J.; Asteris, P.G. Closed-Form Equation for Estimating Unconfined Compressive Strength of Granite from Three Non-destructive Tests Using Soft Computing Models. *Rock Mech. Rock Eng.* **2022**. [CrossRef]
54. Koopialipoor, M.; Asteris, P.G.; Mohammed, A.S.; Alexakis, D.E.; Mamou, A.; Armaghani, D.J. Introducing stacking machine learning approaches for the prediction of rock deformation. *Transp. Geotech.* **2022**, *34*, 100756. [CrossRef]
55. Ikram, R.M.A.; Dai, H.-L.; Ewees, A.A.; Shiri, J.; Kisi, O.; Zounemat-Kermani, M. Application of improved version of multi verse optimizer algorithm for modeling solar radiation. *Energy Rep.* **2022**, *8*, 12063–12080. [CrossRef]
56. Wang, H.; Zhang, L.; Yin, K.; Luo, H.; Li, J. Landslide identification using machine learning. *Geosci. Front.* **2021**, *12*, 351–364. [CrossRef]
57. Liu, Z.; Armaghani, D.J.; Fakharian, P.; Li, D.; Ulrikh, D.V.; Orekhova, N.N.; Khedher, K.M. Rock Strength Estimation Using Several Tree-Based ML Techniques. *C. Model. Eng. Sci.* **2022**, *1–26*, 021165. [CrossRef]
58. He, B.; Armaghani, D.J.; Lai, S.H. Assessment of tunnel blasting-induced overbreak: A novel metaheuristic-based random forest approach. *Tunn. Undergr. Space Technol.* **2023**, *133*, 104979. [CrossRef]
59. Tan, W.Y.; Lai, S.H.; Teo, F.Y.; Armaghani, D.J.; Pavitra, K.; El-Shafie, A. Three Steps towards Better Forecasting for Streamflow Deep Learning. *Appl. Sci.* **2022**, *12*, 12567. [CrossRef]
60. Pham, B.T.; Nguyen, M.D.; Nguyen-Thoi, T.; Ho, L.S.; Koopialipoor, M.; Quoc, N.K.; Armaghani, D.J.; Van Le, H. A novel approach for classification of soils based on laboratory tests using Adaboost, Tree and ANN modeling. *Transp. Geotech.* **2020**, *27*, 100508. [CrossRef]
61. Liang, M.; Mohamad, E.T.; Faradonbeh, R.S.; Jahed Armaghani, D.; Ghoraba, S. Rock strength assessment based on regression tree technique. *Eng. Comput.* **2016**, *32*, 343–354. [CrossRef]
62. Huat, C.Y.; Moosavi, S.M.H.; Mohammed, A.S.; Armaghani, D.J.; Ulrikh, D.V.; Monjezi, M.; Hin Lai, S. Factors Influencing Pile Friction Bearing Capacity: Proposing a Novel Procedure Based on Gradient Boosted Tree Technique. *Sustainability* **2021**, *13*, 11862. [CrossRef]
63. Asteris, P.G.; Rizal, F.I.M.; Koopialipoor, M.; Roussis, P.C.; Ferentinou, M.; Armaghani, D.J.; Gordan, B. Slope Stability Classification under Seismic Conditions Using Several Tree-Based Intelligent Techniques. *Appl. Sci.* **2022**, *12*, 1753. [CrossRef]
64. Hasanipanah, M.; Faradonbeh, R.S.; Amnieh, H.B.; Armaghani, D.J.; Monjezi, M. Forecasting blast-induced ground vibration developing a CART model. *Eng. Comput.* **2017**, *33*, 307–316. [CrossRef]

65. Samaie, M.; Ranjbarnia, M.; Zare Naghadehi, M. Prediction of the Rock Brittleness Index Using Nonlinear Multivariable Regression and the CART Regression Tree. *J. Civ. Environ. Eng.* **2018**, *48*, 33–40.
66. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
67. Zhou, J.; Asteris, P.G.; Armaghani, D.J.; Pham, B.T. Prediction of ground vibration induced by blasting operations through the use of the Bayesian Network and random forest models. *Soil Dyn. Earthq. Eng.* **2020**, *139*, 106390. [CrossRef]
68. Probst, P.; Wright, M.N.; Boulesteix, A. Hyperparameters and tuning strategies for random forest. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2019**, *9*, e1301. [CrossRef]
69. Witten, I.H.; Frank, E. Data mining: Practical machine learning tools and techniques with Java implementations. *ACM SIGMOD Rec.* **2002**, *31*, 76–77. [CrossRef]
70. Resende, P.A.A.; Drummond, A.C. A survey of random forest based methods for intrusion detection systems. *ACM Comput. Surv.* **2018**, *51*, 1–36. [CrossRef]
71. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
72. Kuhn, M.; Johnson, K. *Applied Predictive Modeling*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 26.
73. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; ACM: New York, NY, USA, 2016; Volume 10, pp. 785–794.
74. Zhang, W.; Zhang, R.; Wu, C.; Goh, A.T.C.; Lacasse, S.; Liu, Z.; Liu, H. State-of-the-art review of soft computing applications in underground excavations. *Geosci. Front.* **2020**, *11*, 1095–1106. [CrossRef]
75. Zhou, J.; Qiu, Y.; Armaghani, D.J.; Zhang, W.; Li, C.; Zhu, S.; Tarinejad, R. Predicting TBM penetration rate in hard rock condition: A comparative study among six XGB-based metaheuristic techniques. *Geosci. Front.* **2020**, *12*, 101091. [CrossRef]
76. Dhaliwal, S.S.; Nahid, A.-A.; Abbas, R. Effective intrusion detection system using XGBoost. *Information* **2018**, *9*, 149. [CrossRef]
77. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
78. Rätsch, G.; Onoda, T.; Müller, K.-R. Soft margins for AdaBoost. *Mach. Learn.* **2001**, *42*, 287–320. [CrossRef]
79. Naghadehi, M.Z.; Samaei, M.; Ranjbarnia, M.; Nourani, V. State-of-the-art predictive modeling of TBM performance in changing geological conditions through gene expression programming. *Measurement* **2018**, *126*, 46–57. [CrossRef]
80. Armaghani, D.J.; Yagiz, S.; Mohamad, E.T.; Zhou, J. Prediction of TBM performance in fresh through weathered granite using empirical and statistical approaches. *Tunn. Undergr. Space Technol.* **2021**, *118*, 104183. [CrossRef]
81. Ghanizadeh, A.R.; Ghanizadeh, A.; Asteris, P.G.; Fakharian, P.; Armaghani, D.J. Developing Bearing Capacity Model for Geogrid-Reinforced Stone Columns Improved Soft Clay utilizing MARS-EBS Hybrid Method. *Transp. Geotech.* **2022**, *38*, 100906. [CrossRef]
82. Ghanizadeh, A.R.; Delaram, A.; Fakharian, P.; Armaghani, D.J. Developing Predictive Models of Collapse Settlement and Coefficient of Stress Release of Sandy-Gravel Soil via Evolutionary Polynomial Regression. *Appl. Sci.* **2022**, *12*, 9986. [CrossRef]
83. Naderpour, H.; Rafiean, A.H.; Fakharian, P. Compressive strength prediction of environmentally friendly concrete using artificial neural networks. *J. Build. Eng.* **2018**, *16*, 213–219. [CrossRef]
84. Rezazadeh Eidgahee, D.; Jahangir, H.; Solatifar, N.; Fakharian, P.; Rezaeemanesh, M. Data-driven estimation models of asphalt mixtures dynamic modulus using ANN, GP and combinatorial GMDH approaches. *Neural Comput. Appl.* **2022**, *34*, 17289–17314. [CrossRef]
85. Ghasemi, E.; Sari, M.; Ataei, M. Development of an empirical model for predicting the effects of controllable blasting parameters on flyrock distance in surface mines. *Int. J. Rock Mech. Min. Sci.* **2012**, *52*, 163–170. [CrossRef]
86. Sari, M.; Ghasemi, E.; Ataei, M. Stochastic modeling approach for the evaluation of backbreak due to blasting operations in open pit mines. *Rock Mech. Rock Eng.* **2014**, *47*, 771–783. [CrossRef]
87. Gokhale, B.V. *Rotary Drilling and Blasting in Large Surface Mines*; CRC Press: Boca Raton, FL, USA, 2010; ISBN 0203841395.
88. Kecojevic, V.; Radomsky, M. Flyrock phenomena and area security in blasting-related accidents. *Saf. Sci.* **2005**, *43*, 739–750. [CrossRef]
89. Konya CJ, W.E. *Surface Blast Design*; Prentice Hall: Englewood Cliffs, NJ, USA, 1990.
90. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *2*, 1189–1232. [CrossRef]
91. Friedman, J.H. Stochastic gradient boosting. *Comput. Stat. Data Anal.* **2002**, *38*, 367–378. [CrossRef]
92. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, inference, and Prediction*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009; ISBN 0387848584.
93. Samaei, M.; Ranjbarnia, M.; Nourani, V.; Zare Naghadehi, M. Performance prediction of tunnel boring machine through developing high accuracy equations: A case study in adverse geological condition. *Meas. J. Int. Meas. Confed.* **2020**, *152*, 107244. [CrossRef]
94. Lunetta, K.L.; Hayward, L.B.; Segal, J.; Van Eerdewegh, P. Screening large-scale association study data: Exploiting interactions using random forests. *BMC Genet.* **2004**, *5*, 32. [CrossRef] [PubMed]
95. Vinayak, R.K.; Gilad-Bachrach, R. Dart: Dropouts meet multiple additive regression trees. In Proceedings of the Artificial Intelligence and Statistics, PMLR, San Diego, CA, USA, 9–12 May 2015; pp. 489–497.
96. Hastie, T.; Rosset, S.; Zhu, J.; Zou, H. Multi-class adaboost. *Stat. Interface* **2009**, *2*, 349–360. [CrossRef]

97. Zorlu, K.; Gokceoglu, C.; Ocakoglu, F.; Nefeslioglu, H.A.; Acikalin, S. Prediction of uniaxial compressive strength of sandstones using petrography-based models. *Eng. Geol.* **2008**, *96*, 141–158. [CrossRef]
98. Dehghani, H.; Pourzafar, M.; Zadeh, M. Prediction and minimization of blast-induced flyrock using gene expression programming and cuckoo optimization algorithm. *Environ. Earth Sci.* **2021**, *80*, 12. [CrossRef]
99. Jamei, M.; Hasanipanah, M.; Karbasi, M.; Ahmadianfar, I.; Taherifar, S. Prediction of flyrock induced by mine blasting using a novel kernel-based extreme learning machine. *J. Rock Mech. Geotech. Eng.* **2021**, *13*, 1438–1451. [CrossRef]
100. Fattahi, H.; Hasanipanah, M. An integrated approach of ANFIS-grasshopper optimization algorithm to approximate flyrock distance in mine blasting. *Eng. Comput.* **2021**, *1*, 2619–2631. [CrossRef]
101. Guo, H.; Zhou, J.; Koopialipoor, M.; Armaghani, D.J.; Tahir, M. Deep neural network and whale optimization algorithm to assess flyrock induced by blasting. *Eng. Comput.* **2021**, *37*, 173–186. [CrossRef]
102. Hasanipanah, M.; Amnieh, H.B. A fuzzy rule-based approach to address uncertainty in risk assessment and prediction of blast-induced Flyrock in a quarry. *Nat. Resour. Res.* **2020**, *29*, 669–689. [CrossRef]
103. Lu, X.; Hasanipanah, M.; Brindhadevi, K.; Amnieh, H.B.; Khalafi, S. ORELM: A novel machine learning approach for prediction of flyrock in mine blasting. *Nat. Resour. Res.* **2020**, *29*, 641–654. [CrossRef]
104. Nikafshan Rad, H.; Bakhshayeshi, I.; Wan Jusoh, W.A.; Tahir, M.M.; Foong, L.K. Prediction of Flyrock in Mine Blasting: A New Computational Intelligence Approach. *Nat. Resour. Res.* **2020**, *29*, 609–623. [CrossRef]
105. Rad, H.; Hasanipanah, M.; Rezaei, M.; Eghlim, A. Developing a least squares support vector machine for estimating the blast-induced flyrock. *Eng. Comput.* **2018**, *34*, 709–717. [CrossRef]
106. Murlidhar, B.R.; Armaghani, D.J.; Mohamad, E.T. Intelligence Prediction of Some Selected Environmental Issues of Blasting: A Review. *Open Constr. Build. Technol. J.* **2020**, *14*, 298–308. [CrossRef]
107. Armaghani, D.J.; Hajihassani, M.; Mohamad, E.T.; Marto, A.; Noorani, S.A. Blasting-induced flyrock and ground vibration prediction through an expert artificial neural network based on particle swarm optimization. *Arab. J. Geosci.* **2014**, *7*, 5383–5396. [CrossRef]