Geomechanics and Engineering, *Vol. 34, No. 5 (2023) 547-559*
https://doi.org/10.12989/gae.2023.34.5.547

Mean fragmentation size prediction in an open-pit mine using machine learning techniques and the Kuz-Ram model

Seung-Joong Lee[1a] and Sung-Oong Choi[*2]

[1]*Infra Solution Business Team, Hanwha Corporation / Global, Cheonggyecheon-ro 86, Jung-gu, Seoul, Republic of Korea*
[2]*Department of Energy and Resources Engineering, Kangwon National University, Gangwondaehakgil 1, Chuncheon-si, Gangwon-do, Republic of Korea*

(*Received February 5, 2023, Revised July 18, 2023, Accepted July 31, 2023*)

Abstract. We evaluated the applicability of machine learning techniques and the Kuz–Ram model for predicting the mean fragmentation size in open-pit mines. The characteristics of the in-situ rock considered here were uniaxial compressive strength, tensile strength, rock factor, and mean in-situ block size. Seventy field datasets that included these characteristics were collected to predict the mean fragmentation size. Deep neural network, support vector machine, and extreme gradient boosting (XGBoost) models were trained using the data. The performance was evaluated using the root mean squared error (RMSE) and the coefficient of determination (r^2). The XGBoost model had the smallest RMSE and the highest r^2 value compared with the other models. Additionally, when analyzing the error rate between the measured and predicted values, XGBoost had the lowest error rate. When the Kuz–Ram model was applied, low accuracy was observed owing to the differences in the characteristics of data used for model development. Consequently, the proposed XGBoost model predicted the mean fragmentation size more accurately than other models. If its performance is improved by securing sufficient data in the future, it will be useful for improving the blasting efficiency at the target site.

Keywords: deep neural network; extreme gradient boosting; machine learning; mean fragmentation size; support vector machine

1. Introduction

In blasting, the fragmentation size of the rock is an important factor that affects the overall production process, including loading, hauling, and first crushing. The blasting parameters that affect fragmentation size are divided into controllable and uncontrollable parameters. The controllable parameters include bench height, drilling length, hole spacing, burden, charge length, stemming length, and powder factor. Uncontrollable parameters include the characteristics of the in-situ rock, that is, the engineering characteristics of the rock and the distributional characteristics of the discontinuity. To obtain a fragmentation size suitable for the target site, the optimal blasting pattern must be derived by appropriately adjusting the controllable parameters considering uncontrollable parameters, that is, the characteristics of the in-situ rock. An optimal blasting pattern can be designed using multiple test blasting or empirical blast optimization models. Blasting design using test blasting incurs considerable cost and time. However, empirical blast optimization models enable blasting design only through numerical calculations of measurable blasting parameters.

Therefore, the correlations between blasting parameters and fragmentation size have been analyzed in previous studies, and empirical blast optimization models for controlling the fragmentation size have been developed. Langefors and Kihlström (1978) proposed a simple prediction model that considered only the burden and powder factors. Adams *et al*. (1983) and McHugh (1983) proposed theoretical models that considered all blasting parameters. Various empirical blast optimization models have been developed based on field data (Kuznetsov 1973, Cunningham 1983, Cunningham 1987, Choi *et al*. 2004, Koulli and Rustan 1993, Cunningham 2005); the Kuz–Ram model developed by Cunningham (1983) has been widely employed to date. The Kuz–Ram model comprises a mean fragmentation size equation and uniformity equation. For the prediction model modified and announced by Cunningham in 2005, a field correction factor was introduced to improve the field applicability. In addition, Gheibie *et al*. (2009) and Hekmat *et al*. (2019) proposed a modified Kuz–Ram model applicable to each target site based on the Kuz–Ram model. Researchers have proposed various empirical blast optimization models to control fragmentation size and modify the highly versatile Kuz–Ram model to improve its field applicability.

Recently, research has been actively conducted using machine learning techniques for more accurate fragmentation-size prediction (Bamford *et al*. 2021, Dumakor-Dupey *et al*. 2021, Li *et al*. 2021). Artificial neural networks (ANNs), a representative machine learning technique, have been widely employed in various

∗Corresponding author, Professor
E-mail: choiso@kangwon.ac.kr
[a]Ph.D.
E-mail: sjlee3601@hanwha.com

ISSN: 2005-307X (Print), 2092-6219 (Online)

Table 1 Modified rock factor by Cunningham (1987)

Parameter		Description	Value
RMD		Powdery/friable	10
		Massive	50
		Vertically jointed	JF
JF $(= JPS + JPA)$	JPS	Joint spacing < 0.1 m	10
		Joint spacing = 0.1 m to MS	20
		Joint spacing = MS to DP	50
	JPA	Dip out of face	40
		Strike out of face	30
		Dip into face	20
RDI		$RDI = 25\,SG - 50$	RDI
HF		If E < 50	E / 3
		If E > 50	UCS / 5

$*$ RMD: rock mass description, JF: joint factor, JPS : joint plane spacing, JPA: joint plane angle, MS: oversize (m), DP: drilling pattern size (m), RDI: rock density influence, SG: specific gravity of rock, HF: hardness factor, E: Young's modulus (GPa), UCS: uniaxial compressive strength (MPa)

industries. Ebrahimi *et al*. (2015) conducted research on fragmentation size prediction using this technique by collecting 35 datasets of controllable blasting parameters, such as burden, hole spacing, stemming length, drilling length, and powder factor. Amoako *et al*. (2022) collected 102 datasets of controllable parameters (e.g., the ratio of spacing to burden, ratio of height to burden, ratio of stemming length to burden, ratio of burden to hole diameter, and powder factor) and uncontrollable parameters (e.g., mean in-situ block size and Young's modulus), applied ANN and support vector machine (SVM) techniques, and compared and analyzed the results.

The datasets collected by these studies were obtained from various previous studies, resulting in inconsistent parameter standardization. Additionally, the criteria for parameters selection were ambiguous. They considered different blasting parameters depending on the characteristics of the collected data, and an extremely limited number of uncontrollable parameters, that is, blasting parameters that can consider the characteristics of in-situ rock, were included. As mentioned previously, the characteristics of in-situ rock are important parameters that affect the fragmentation size. Thus, they must be considered for a more accurate prediction of the fragmentation size. However, it is practically difficult to collect all the characteristics of in-situ rock at all blasting sites.

In this study, 70 datasets were collected which included the uniaxial compressive strength and tensile strength, which are rock strength parameters, rock factors that can reflect the engineering characteristics of discontinuity, and mean in-situ block size to predict the fragmentation size. In addition, machine learning techniques such as SVM and deep neural networks (DNNs), which have been utilized in previous studies, as well as the widely employed extreme gradient boosting (XGBoost) technique, were employed.

The prediction results obtained through each technique were compared with those obtained through the empirical blast optimization model, i.e., the Kuz-Ram model.

2. Materials and methods

2.1 Kuz-Ram model

The Kuz–Ram model was first proposed by Cunningham (1983). He announced modified Kuz–Ram models in 1987 and 2005 through continuous research. The Kuz–Ram model comprises a mean fragmentation size equation (X_{50}) and uniformity equation (n) that represents the slope of the Rosin–Rammler particle size distribution curve.

The mean fragmentation size equation proposed by Cunningham was modified from the mean fragmentation size, Eq. (1), proposed by Kuznetsov (1973) to Eq. (2) to reflect the powder factor and power of the explosives (Cunningham 1983).

$$X_{50} = \mathrm{A}\left(\frac{V_o}{Q_{TNT}}\right)^{0.8} Q_{TNT}{}^{\frac{1}{6}} \tag{1}$$

$$X_{50} = \mathrm{A}K^{0.8}Q^{\frac{1}{6}}\left(\frac{115}{RWS}\right)^{\frac{19}{20}} \tag{2}$$

where X_{50} is the mean fragmentation size (cm), A is the rock factor, V_o is the rock volume (m^3), Q_{TNT} is the weight of TNT (kg) equivalent in energy to the explosive charge in one borehole, K is the powder factor (kg/m^3), Q is the mass of explosive per hole (kg), and RWS is the weight strength relative to ANFO (115 is the RWS of TNT).

In 1987, Cunningham introduced the blastability index (BI) proposed by Lilly (1986) to systematically quantify the in-situ rock evaluation parameter, that is, the evaluation method for rock factor (A). The evaluation criteria for the rock factor proposed by Cunningham (1987) are listed in

Table 2 Modified rock factor by Cunningham (2005)

Parameter		Description	Value
RMD		Powdery/friable	10
		Massive formation	50
		Vertically jointed	JF
JF $(= JCF \times JPS + JPA)$	JCF	Tight joints	1.0
		Relaxed joints	1.5
		Gouge = filled joint	2.0
	JPS	Joint spacing < 0.1 m	10
		Joint spacing = 0.1 ~ 0.3 m	20
		Joint spacing = 0.3 m ~ 95% to P	80
		Joint spacing > P	50
		$P = (B \times S)^{0.5}$	P
	JPA	Dip out of face	40
		Strike out of face	30
		Dip into face	20
RDI		$RDI = 25\ SG - 50$	RDI
HF		If E < 50	E / 3
		If E > 50	UCS / 5

* RMD: rock mass description, JF: joint factor, JCF: joint condition factor, JPS : joint plane spacing, JPA: joint plane angle, P: reduced pattern(m), B: burden(m), S: spacing(m), RDI: rock density influence, SG: specific gravity of rock, HF: hardness factor, E: Young's modulus (GPa), UCS: uniaxial compressive strength (MPa)

Table 1, and the parameter calculation formula is given by Eq. (3).

$$A = 0.06(\mathrm{RMD} + \mathrm{JF} + \mathrm{RDI} + \mathrm{HF}) \quad (3)$$

where A is the rock factor, RMD is the rock mass description, JF is the joint factor, RDI is the influence of rock density, and HF is the hardness factor.

In 2005, Cunningham again modified the Kuz–Ram model using research results from over 20 years. For the main modifications, a function related to the delay time was added to the mean fragmentation size equation, and a correction factor ($C(A)$) for the rock factor was added to improve the field applicability of the model. This equation is as follows

$$X_{50} = AA_T K^{0.8} Q^{\frac{1}{6}} \left(\frac{115}{RWS}\right)^{\frac{19}{20}} C(A) \quad (4)$$

where X_{50} is the mean fragmentation size (cm), A is the rock factor, A_T is a timing factor for the effect of inter-hole delay, K is the powder factor (kg/m^3), Q is the mass of the explosive in the hole (kg), RWS is the weight strength relative to the ANFO, and $C(A)$ is a correction factor for the rock factor.

In addition, in the BI evaluation method introduced in 1987 to prepare quantitative criteria for the evaluation of the rock factor, the evaluation criteria for the joint spacing were subdivided, and those for joint aperture conditions were introduced to reconstruct the evaluation items and methods, as shown in Table 2.

2.2 Machine learning for mean fragmentation prediction

Machine learning, an area of artificial intelligence, is a technology that trains computers to learn data, find patterns, and perform appropriate tasks. Recently, research using machine learning has been actively conducted in various research fields (Mahmoodzadeh *et al*. 2022a, b, Kim *et al.* 2022, Kwak and Ko 2022). For mean fragmentation size prediction, this study used the field data collected from limestone open-pit mines in Korea as well as the DNN, SVM, and XGBoost algorithms.

2.2.1 Dataset

In machine learning, if the input parameters cannot appropriately explain the output parameter, model training is not properly performed, regardless of the algorithm. To improve the predictive performance of a machine learning model, a sufficient amount of data must be collected, which must be objective and valid.

Parameters that affect the fragmentation size during the blasting operation can be divided into controllable and uncontrollable parameters, as mentioned previously. Specifically, controllable parameters include those that workers can artificially control, including the bench height, width, hole diameter, hole length, spacing, burden, charge length, stemming length, powder factor, and delay time.

Uncontrollable parameters are the characteristics of in-situ rock and can be divided into the engineering characteristics of rock and engineering characteristics of discontinuity. The engineering characteristics of rocks include the uniaxial compressive strength, tensile strength, Young's modulus, Poisson's ratio, and elastic wave velocity, whereas the engineering characteristics of discontinuity include the spacing of discontinuity, angle between the bench face and discontinuity, and mean in-situ block size.

Generally, at a blasting operation site, a work log is recorded and stored for each blasting operation for on-site operation management, and most of the recorded contents are controllable parameters. However, uncontrollable parameters, that is, the engineering characteristics of discontinuity, require the skill and expertise of the investigator. In the case of the engineering characteristics of rocks, it is practically impossible to acquire the data of each blasting operation because rock sampling and rock property tests must be conducted. Therefore, most data collected from blasting sites are controllable parameters.

Most previous studies on fragmentation size prediction using empirical statistical analysis and machine learning have focused on controllable parameters because it is difficult to collect data related to the characteristics of in-situ rock. Limitations also exist in terms of collecting consistent and massive amounts of data through a literature survey because the blasting parameters collected in the field differ depending on the research purpose.

In this study, data from the doctoral dissertations of Choi (2005) and Lee (2016) were used to collect data that included parameters related to in-situ rock. They conducted test blasting at a limestone open-pit mine located in the Donghae mine area, Korea, to develop a statistical prediction model and investigated the characteristics of in-situ rock, blasting pattern investigation, and fragmentation size analysis. Choi (2005) evaluated the characteristics of in-situ rock for a blasting target bench using the geological strength index (GSI) method and presented a modified Kuz–Ram model by introducing a mean in-situ block size evaluation parameter to the Kuz–Ram model previously proposed by Cunningham (1983). Lee (2016) evaluated rocks using the RMR method and proposed a statistical prediction model by introducing a delay time parameter.

These two studies were used as basic data of this study because they summarized parameters related to the characteristics of in-situ rock relatively well, including the uniaxial compressive strength, tensile strength, mean in-situ block size, and rock factor. Twenty-six datasets from Choi (2005) and 44 from Lee (2016) were used in this study (Table 3). Ten percent (7 datasets) of the data were randomly extracted and used as the evaluation set for the prediction model (Table 4), and the remaining 90% (63 sets) were utilized for training the model. Typically, machine learning models are trained on thousands of data points, but in this study, a relatively small dataset consisting of 70 data points was used. Insufficient training data can lead to a decrease in the performance of machine learning models. Therefore, the ratio of train, validation, and test data should be appropriately distributed, considering the research objectives and characteristics of the data. Similarly, in the study on rock slope analysis using deep learning by Lee *et al*. (2019), approximately 10% (69 out of 681 data points) were used as validation data, resulting in meaningful outcomes.

Among the datasets used in this study, controllable parameters included spacing, burden, bench height, hole length, hole diameter, stemming length, charge length, delay time, mass of explosive per hole, and powder factor; uncontrollable parameters included mean in-situ block size, rock factor, uniaxial compressive strength, and tensile strength.

In blasting operations, the fragmentation size is affected by various blasting parameters that are related to each other in an organic manner from an engineering perspective. In this study, for some controllable parameters, the ratios of organically related parameters were used by referring to previous research. The ratio of spacing to burden (S/B) was the reference parameter used in the wide-spacing blasting method. It was controlled at each site to reduce the fragmentation size or make it relatively homogeneous. The ratio of the bench height to burden (H/B) is a parameter used in bench blasting design. Depending on its value, blasting can be divided into high- and low-bench blasting. It is used in the calculation of delay time conditions and the subsequent hole spacing, depending on the bench scale (Langefors and Kihlström 1978). The burden-to-hole diameter (B/D) ratio is the most important parameter in blasting operation. Ash (1963) proposed 30 as the B/D ratio under general blasting conditions and 25 when low-density explosives such as ANFO were used. The ratio of stemming length to burden (SL/B) is a parameter that is controlled according to the joint or natural crack condition of the rock around the blasting hole. Through a case study, Singh *et al.* (2016) reported that the mean fragmentation size increases as the SL/B ratio increases. The ratio of charge length to hole length (CL/HL) and the ratio of stemming length to charge length (SL/CL) vary depending on the proportion of the charge length in the entire hole length. Lee (2016) analyzed the correlation between the charge length and fragmentation size according to the proportion of the charge length. The ratio of delay time to burden (T/B) expresses the influence of the delay time and burden on the fragmentation size. Bergmann *et al*. (1974) and Katsabanis *et al*. (2006) derived the optimal delay time that could control the fragmentation size through a scaled model test.

Consequently, the input parameters used in this study were combination parameters, including S/B, H/B, B/D, CL/HL, SL/CL, and T/B, and single parameters, including the mass of explosives per hole, powder factor, mean in-situ block size, rock factor, uniaxial compressive strength, and tensile strength. The output parameter was the mean fragmentation size (X50), which ranged from 18.10 to 65.4 cm.

Figs. 1 and 2 show the linear correlations between input parameters and output parameter. BS had the highest positive linear correlation. It was estimated that the separation effect between rock blocks had a greater influence on the mean fragmentation size compared to the direct fracturing effect on the rock during blasting. On the other hand, the parameter with the highest negative correlation was UCS. Generally, under the same blasting conditions, fragmentation size tends to increase as UCS increases. However, in this study, the negative correlation between UCS and X50 is attributed to the interaction of controllable blasting parameters such as spacing, burden, and powder factor.

Table 3 Descriptive statistics of input and output parameters (Choi 2005, Lee 2016)

	Parameters	Minimum	Maximum	Mean	Std. deviation
Input	S/B	0.86	1.61	1.19	0.18
	H/B	2.44	6.40	3.85	0.81
	B/D	24.51	44.90	36.67	4.38
	SL/B	0.71	1.60	1.23	0.23
	CL/HL	0.58	0.85	0.72	0.05
	SL/CL	0.17	0.71	0.39	0.10
	T/B	0.98	10.81	5.55	2.54
	Q	48.40	148.10	101.07	30.34
	PF	0.31	0.65	0.48	0.09
	BS	0.09	1.50	0.70	0.33
	A	7.13	15.75	10.37	1.90
	UCS	51.00	178.40	90.58	33.10
	TS	4.40	12.70	7.04	2.65
Output	X50	18.10	65.40	38.74	9.55

* S/B: ratio of spacing to burden, H/B: ratio of bench height to burden, B/D: ratio of burden to hole diameter, SL/B: ratio of stemming length to burden, CL/HL: ratio of charge length to hole length, SL/CL: ratio of stemming length to charge length, T/B: ratio of delay time to burden, Q: mass of explosive per the hole (kg), PF: powder factor (kg/m^3), BS: mean in-situ block size (m), A: rock factor, UCS: uniaxial compressive strength (MPa), TS: tensile strength (MPa), X50: mean fragmentation size (cm)

Table 4 Test datasets used for the evaluation of each machine learning model (Choi 2005, Lee 2016)

Source	S/B	H/B	B/D	St/B	CL/HL	St/CL	T/B	Q	PF	BS	A	UCS	TS	X50
Choi (2005)	1.40	4.40	24.51	1.60	0.68	0.48	8.00	54.90	0.57	1.00	8.15	85.30	6.60	59.30
	1.36	6.00	24.51	1.60	0.75	0.33	8.00	79.80	0.63	0.10	7.91	115.60	4.70	18.10
	1.33	4.67	29.41	1.33	0.76	0.31	6.67	85.00	0.51	0.90	7.46	93.10	9.40	51.30
Lee (2016)	1.53	2.50	39.22	0.88	0.72	0.39	10.00	91.93	0.43	0.20	13.32	69.07	4.93	31.95
	1.18	3.00	39.22	1.38	0.63	0.58	7.00	97.03	0.64	0.80	11.49	82.00	6.42	39.12
	0.95	3.66	40.20	1.22	0.72	0.38	4.88	132.78	0.53	0.60	12.02	73.49	5.65	39.81
	1.21	3.06	40.78	0.96	0.74	0.36	7.69	114.40	0.40	0.60	10.39	69.07	4.93	34.03

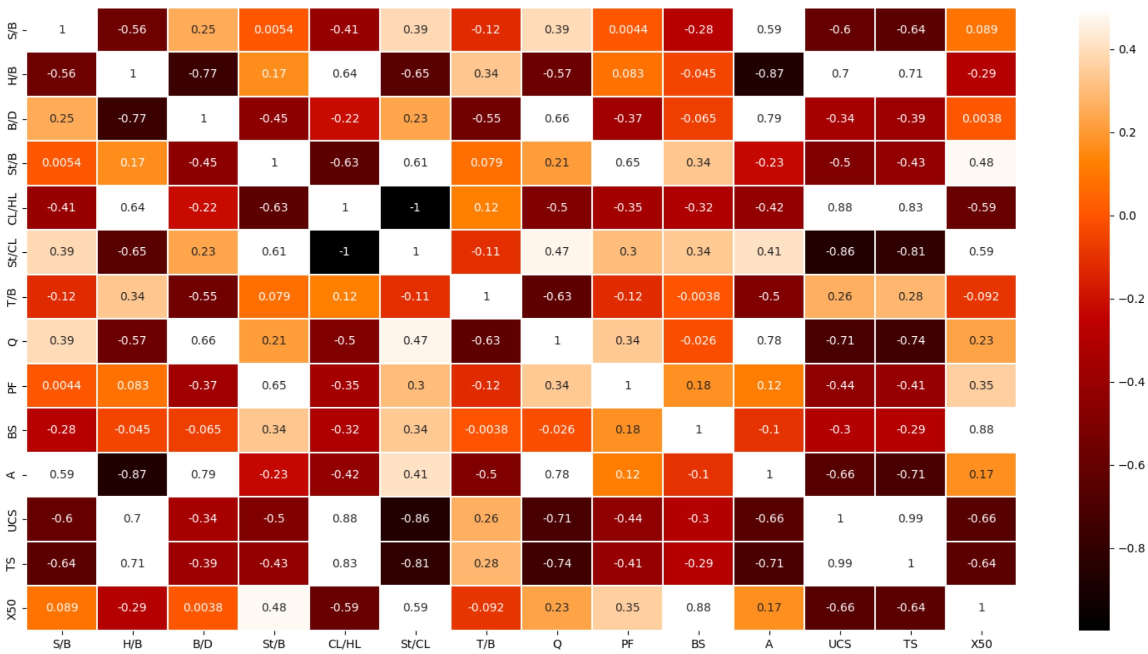
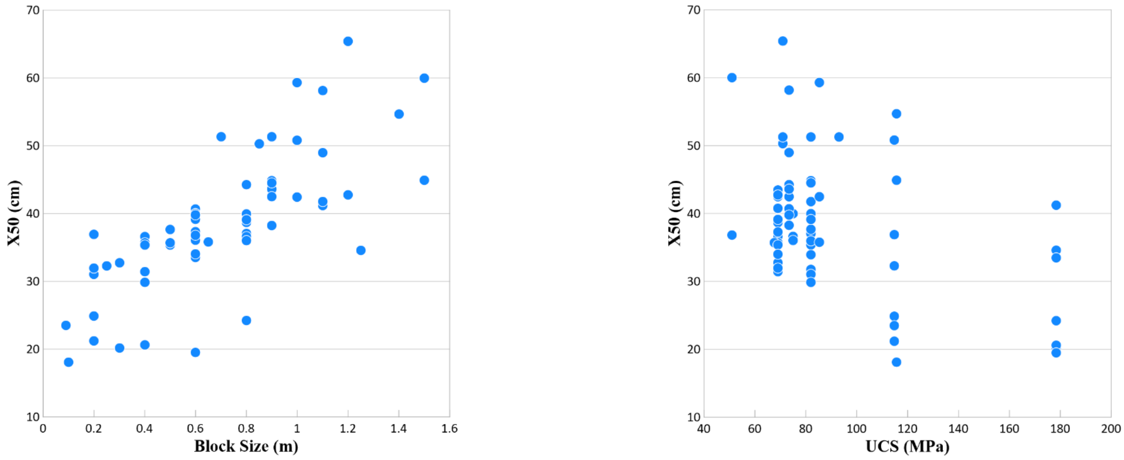


Fig. 1 Correlation analysis results with blasting parameters and X50



(a) Block size and X50 (b) UCS and X50

Fig. 2 Relationship between blasting parameters and X50

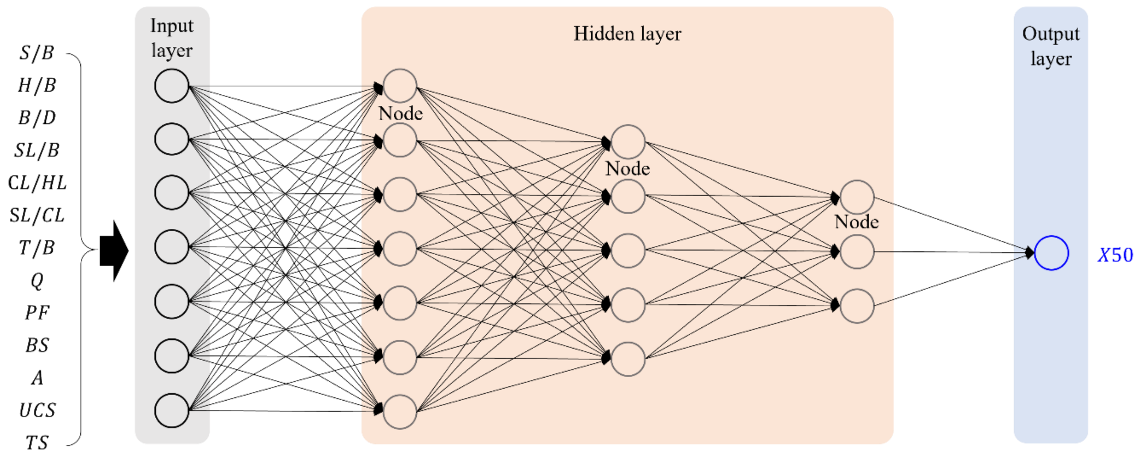


Fig. 3 DNN architecture for the prediction of the mean fragmentation size

2.2.2 Machine learning based regression analysis

Deep neural network

A DNN is an ANN with two or more hidden layers between the input and output layers (Bengio *et al*. 2013). As a series of algorithms for effectively training ANNs comprising several layers, a DNN can model complex nonlinear relationships and exhibits excellent performance in executing complex tasks that require high-level features as the number of layers increases because high-dimensional features can be extracted (Dahl *et al*. 2013). To apply a DNN model, an analyst determines model complexity according to the characteristics of the data and training is repeated several times to reduce the model's errors and derive optimal results. Because problems such as overfitting and high time complexity may occur as the number of hidden layers and the number of nodes constituting them increases, training a DNN model after determining the target level is important.

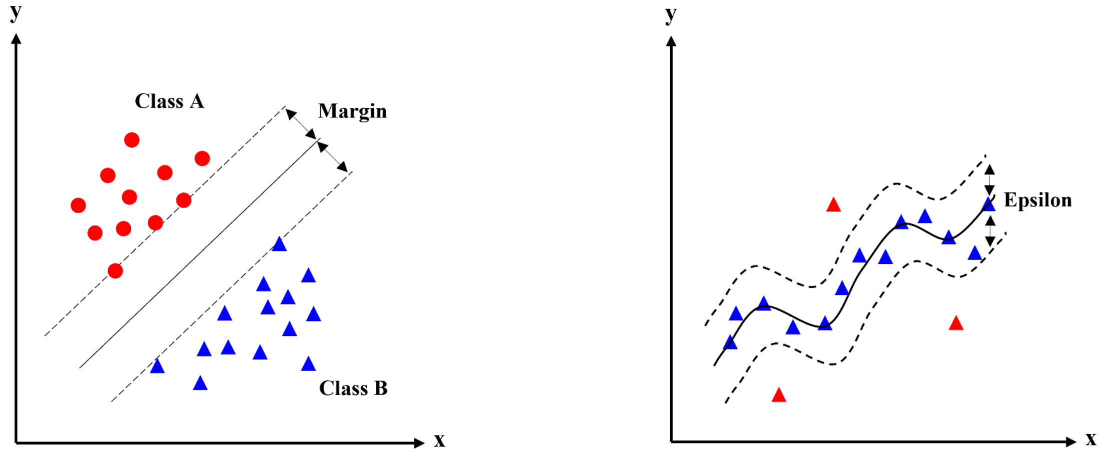
In this study, a Python-based DNN model was implemented using TensorFlow 2.6.1, which is an open-source library. Fig. 2 indicates the basic DNN architecture. It has the 13 input parameters described in Table 3 and 100 nodes in the input layer. Three hidden layers, each with 300, 100, and 50 nodes, are included. The output layer has one node to derive X50. The DNN architecture was determined using a trial-and-error method.

ReLU (Fukushima 1975) is an activation function used for training a DNN model, and the mean squared error, which is commonly used in regression problems, was used as the loss function. The error calculated from the loss function was used to modify the weight of the model during the backpropagation step. Adam, a commonly used model, was used as the optimizer. The weight of the model was updated by setting the batch size to five.

For the performance evaluation and validation of the model, 10% of the 63 training datasets were used as the validation datasets. The early stopping function was used to prevent overfitting, and the training was set to terminate early if the loss value for the validation data did not improve after the initial ten epochs. Table 5 summarizes the DNN structure and training methods.

Support vector machine

SVM, a method based on statistical learning theory proposed by Vapnik (1999), has attracted considerable attention because of its excellent generalization capability in the fields of pattern recognition and learning theories. The performance of SVM is superior to that of conventional methods because it can minimize the upper limit of the



(a) SVC (b) SVR

Fig. 4 Support vector machine

Table 5 Summary of DNN training method

Items	Details
Input parameter	13(S/B, H/B, B/D, St/B, CL/HL, St/CL, T/B, Q, PF, RS, A, UCS, TS)
Output parameter	1(X50)
Input/Output node	100 nodes / 1 nodes
Hidden layer and node	3 layers (300, 100, 50 nodes)
Batch size	5
Activation function	ReLU
Loss function	Mean squared error
Optimizer	Adam

generalization error based on structural risk minimization rather than conventional empirical risk minimization (Burges 1998).

SVM is divided into support vector classification (SVC), which is used for classification problems, and support vector regression (SVR), which is used for regression problems. SVC can determine the decision boundary, that is, a baseline for classification. Upon inputting new unclassified data, SVC solves the classification problem by identifying the side of the boundary to which the data belong. Fig. 4(a) shows the basic concept of SVC. The solid line in the middle represents the decision boundary, and the support vector represents the data close to the decision boundary. The margin indicates the distance between the decision boundary and support vector. In the SVC method, it is important to find an optimal decision boundary that can maximize the margin. By contrast, SVR derives a line that represents all data. Unlike SVC, its objective is to perform training so that the data can be included in the epsilon range (Fig. 4(b)). In this study, SVR was used to predict the mean fragmentation size, which is an arbitrary real number.

In this study, Scikit-learn API's SVR library was used to train the SVM model. In general, the user must directly set the hyperparameters required for training, considering the characteristics of the training data for SVM model training. The hyperparameters generally used in the SVM model include the Kernel, C, Gamma, and epsilon. The kernel

Table 6 Hyperparameters used in SVM training method

Parameter	Range of input values	Optimal values
Kernel	RBF, ploy	Poly
C	1 ~ 15	0.500
Epsilon	0.1 ~ 1	0.097
Gamma	0.01 ~ 15	2.429

designates the type of function and includes liner, poly, rbf, and sigmoid. C denotes the degree of regularization and determines the margin of the decision boundary. Gamma is a hyperparameter that designates the coefficient of the kernel and is valid when the kernel is RBF, poly, or sigmoid. Epsilon determines the acceptable noise level based on the decision boundary. The process of determining the optimal values of these hyperparameters is referred to as hyperparameter tuning. Overfitting can be prevented through hyperparameter tuning, and the predictive performance of the model can be improved. The simplest method is to manually adjust each hyperparameter until a satisfactory combination of hyperparameters is obtained. However, this method requires considerable time because several cases need to be considered. The Scikit-learn library provides GridSearchCV and RandomSearchCV classes so that hyperparameter tuning can be automatically performed. GridSearchCV finds the combination with the best performance after cross-validation for all combinations once arbitrary values of each hyperparameter have been specified. However, much time is required when the hyperparameter values increase. By contrast, when the range of each hyperparameter value is determined and the number of hyperparameter combinations to be cross-validated is designated, RandomSearchCV finds the optimal values of the hyperparameters by randomly performing combinations and repetitions by the designated number. In this study, the optimal values for each hyperparameter were derived using RandomSearchCV, as shown in Table 6.

Extreme gradient boosting

XGBoost supports both the regression and classification problems. This method is widely known because it has been

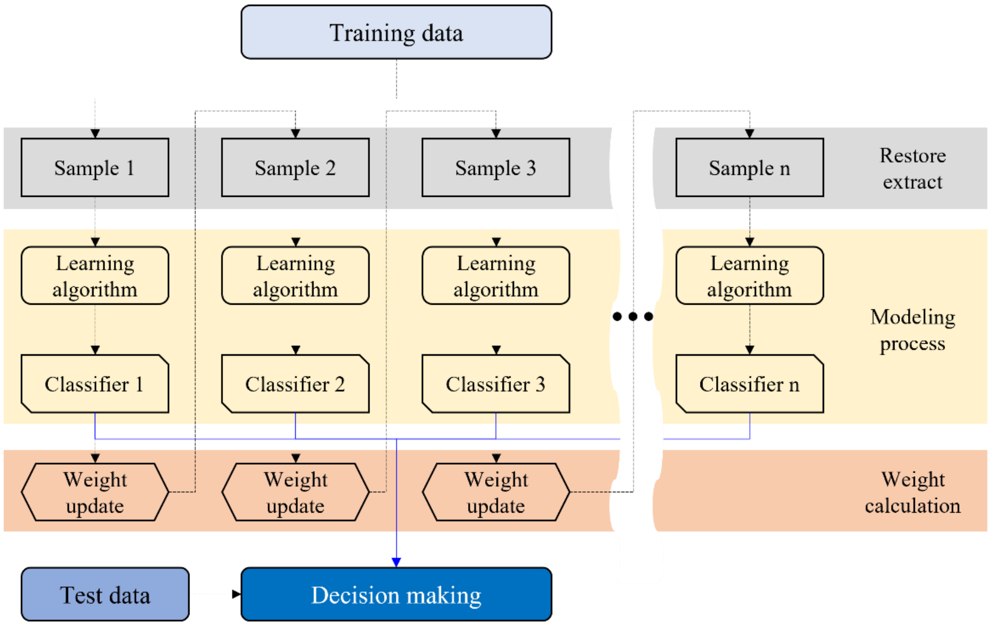


Fig. 5 Boosting architecture

used by numerous top-ranking participants in Kaggle competitions. It requires less training time than a gradient boosting machine (GBM) and contains functions required for performance improvement, such as regularization, tree pruning, and early stopping.

XGBoost is an ensemble algorithm that uses a combination of several decision trees and uses the boosting technique. The boosting technique extracts n data samples from the training data through random sampling and updates the weight by reflecting the error of the previous weak learner. It then creates a strong prediction model by sequentially reflecting it on the next learning model (Fig. 5). Gradient boost is a representative algorithm implemented using the boosting technique. The library that implements this algorithm to support parallel learning is called XGBoost.

For XGBoost model training, a Python-based mean fragmentation size prediction model was implemented using XGBoost 1.7.2, an open-source library. Hyperparameter tuning is required to train the XGBoost model, as with SVM. In this study, hyperparameter tuning was performed by adjusting the values of learning_rate, n_estimators, subsample, and colsample_bytree among the various hyperparameters. Learning _rate refers to the amount or step of learning, and its value is updated when the boosting step is repeatedly performed. n_estimators determine the number of weak learners to be created, and its value must be increased to prevent overfitting when the learning rate is low. Subsample determines the proportion of data sampling used by weak learners for learning, and a value between 0.5 and 1 is typically used. colsample_bytree determines the percentage of features used for each tree. For such hyperparameter tuning, the RandomSearchCV class of the Scikit-learn library was also used, as with the SVM model training. Table 7 summarizes the results of tuning the hyperparameters for XGBoost training.

4. Results

Among the 70 datasets used in this study, 63 were used for model training and the 7 datasets in Table 4 were used for model evaluation. The coefficient of determination (r^2) and root mean squared error (RMSE) were used to evaluate the performance of the mean fragmentation size prediction models derived using DNN, SVM, and XGBoost techniques. r^2 is an evaluation index that represents the ratio of the variance of the predicted values to that of the measured values. An r^2 value close to 1 indicates a model with good performance, whereas an r^2 value close to 0 represents a model with poor performance. RMSE is the square root of the average of the squares of the difference between predicted and measured values. A lower RMSE value indicates a model with a superior predictive performance. r^2 and RMSE are given by Eqs. (5) and (6), respectively.

$$r^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - y_i')^2}{\sum_{i=1}^{n}(y_i - y_{iave})^2} \tag{5}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2} \tag{6}$$

where r^2 is the coefficient of determination, n is the number of datasets, y_i is the measured X50, y_i' is the predicted X50, y_{iave} is the average value of the measured X50, and $RMSE$ is the root mean squared error.

Fig. 6 and Table 8 show the training and evaluation results for each model. In the training results, the DNN, SVM, and XGBoost models exhibited RMSE values of 3.090, 2.710, and 1.890, and r^2 values of 0.885, 0.911, and 0.959, respectively. The XGBoost model showed the lowest RMSE value, with an the r^2 value close to 1. When

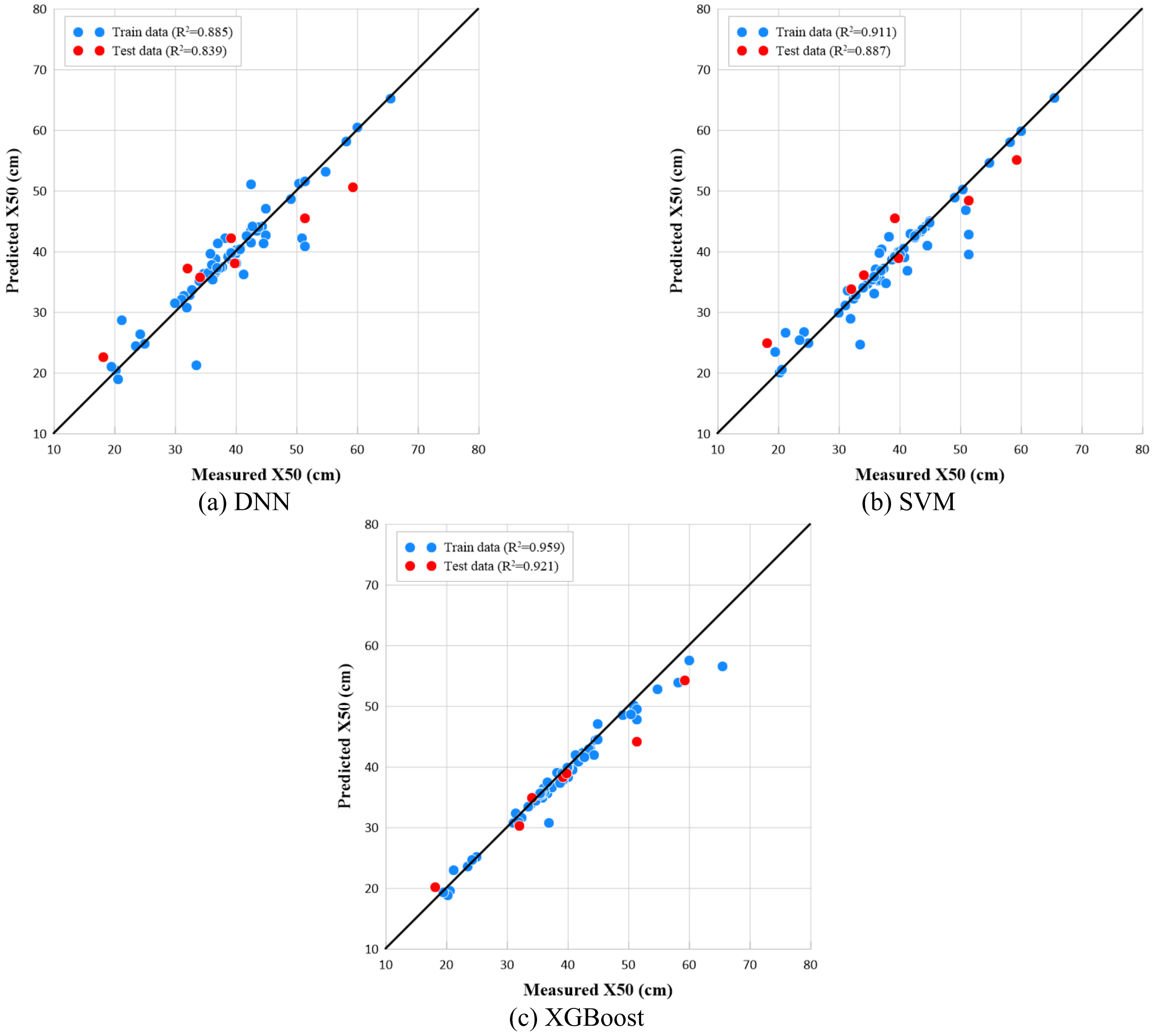


Fig. 6 Correlation among the measured and predicted X50 by the DNN, SVM, and XGBoost models

Table 7 Hyperparameters used for XGBoost model training

Parameter	Range of input values	Optimal values
N_estimators	50 – 1,000	900
Learning_rate	0.1 – 2.0	0.3
Subsample	0.5 – 1.0	0.8
Colsample_bytree	0.5 – 1.0	0.8

Table 8 Evaluation results for the DNN, SVM, and XGBoost models

Model	Training data		Test data	
	RMSE	r^2	RMSE	r^2
DNN	3.090	0.885	4.980	0.839
SVM	2.710	0.911	4.160	0.887
XGBoost	1.890	0.959	3.490	0.921

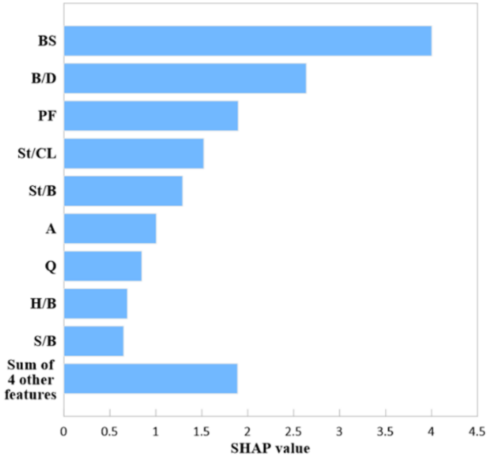
applying the test data in Table 4, the XGBoost model exhibited the lowest RMSE value (3.490), and its r^2 value (0.921) was the closest to 1. These results confirm that the XGBoost technique had the highest predictive performance among the machine learning techniques used in this study.

In regression analysis, indirect inference techniques based on the trained model and data are employed to analyze the impact of input parameters on the output parameter. These inference techniques include permutation feature importance (PFI) and shapley additive explanations (SHAP), among others. SHAP is widely used in various research fields to evaluate the importance of parameters (Lundberg and Lee 2017, Li *et al*. 2023, Gebreyesus *et al*. 2023, McFall *et al*. 2023). In this study, SHAP analysis was used to estimate the impact of the input parameters on the output parameter.

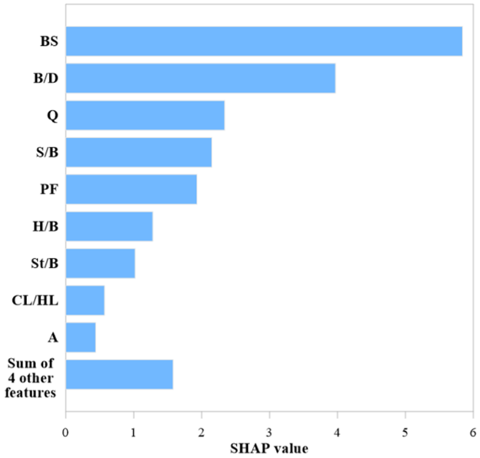
When analyzing the importance of the top three parameters based on the SHAP analysis results, in the DNN model, BS showed the highest importance, followed by B/D and PF (Fig. 7(a)). In the SVM model, the importance was highest for BS, followed by B/D and Q (Fig. 7(b)). In the XGBoost model, the importance was highest for BS, followed by UCS and CL/HL (Fig. 7(c)). Consequently, BS was found to be the most influential parameter for the mean fragmentation size in all models.

5. Discussion

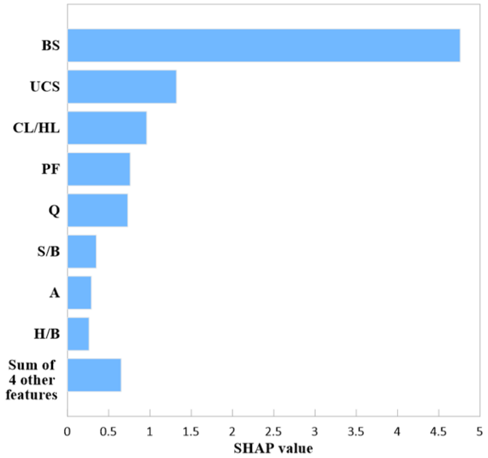
The machine learning models used in this study were



(a) DNN



(b) SVM



(c) XGBoost

Fig. 7 Feature importance of the DNN, SVM, and XGBoost models using SHAP

compared with the Kuz–Ram model (Cunningham 1983), which is a versatile statistical prediction model. For this purpose, the seven datasets in Table 4 were used, and the performance of each model was compared and analyzed by comparing the measured and predicted values. Table 9 lists the absolute errors between the predicted values of each model and the measured values, whereas Fig. 8 shows both the measured and predicted values.

Among the machine learning models, the SVM model showed the largest error rate of 37.76% in test case 2. The XGBoost model exhibited the smallest error rate (1.98%) in test case 5. When the error rates of the machine learning models were analyzed for each test case, the DNN model showed the largest error in test cases 1, 4, and 6, whereas the SVM model exhibited the largest error in test cases 2 and 5. By contrast, the XGBoost model showed the largest error only in test case 3. The average error rate for all test cases was largest in the DNN model (12.15%) and smallest in the XGBoost model (6.53%). Consequently, as confirmed by comparing and analyzing the errors between the measured and predicted values, the performance of the XGBoost model was found to be higher than that of the other models.

The results of the machine learning analysis and the applicability of the Kuz-Ram model were examined. This study attempted to apply the recently proposed Kuz–Ram (2005) model, but there are insufficient data for the calculation of parameters related to the rock factor anddelay time in the study by Choi (2005) because it was conducted with a focus on the Kuz–Ram model proposed in 1983. Therefore, in this study, the predicted values were calculated using Eq. (2), which is the Kuz–Ram model proposed in 1983.

When the error rates between the values predicted by the Kuz–Ram model and the measured values were compared and analyzed, test case 5 exhibited the highest error rate, and test case 2 exhibited the lowest error rate. The error rates for the two test cases were 58.98 and 64.97%, respectively. The average error rate for all test cases was 47.75%, which was approximately 35.6% higher than the average error rate of the DNN model, which showed the highest error rate among the machine learning models. Consequently, the Kuz–Ram model exhibited a large error because it could not reflect the characteristics of the data used in this study. This is because the Kuz–Ram model cannot represent the blasting results of all sites; however, it collects and statistically analyzes data from various sites. In other words, because the Kuz–Ram model is a statistical model derived based on the collected data, it can predict the blasting results relatively well if data with characteristics similar to those of the data used in the analysis are applied.

Table 9 Prediction error of machine learning and Kuz–Ram model

Test case	Measured X50 (cm)	Predicted X50 (cm)				Absolute value of prediction error (Error rate, %)			
		Machine learning			Kuz–Ram	Machine learning			Kuz–Ram
		DNN	SVM	XGBoost		DNN	SVM	XGBoost	
1	59.30	50.65	55.15	54.27	25.62	**8.65 (14.59)**	4.15 (6.99)	5.03 (8.49)	33.68 (56.79)
2	31.95	37.21	33.78	30.29	18.93	4.55 (25.13)	**6.83 (37.76)**	2.07 (11.41)	6.33 (34.97)
3	18.10	22.65	24.93	20.17	24.43	5.85 (11.40)	2.86 (5.57)	**7.13 (13.89)**	23.73 (46.25)
4	39.12	42.25	45.49	38.35	16.05	**5.26 (16.46)**	1.83 (5.73)	1.66 (5.21)	13.02 (40.76)
5	51.30	45.45	48.44	44.17	27.57	3.13 (7.99)	**6.37 (16.27)**	0.77 (1.98)	23.07 (58.98)
6	39.81	38.09	38.95	38.95	16.80	**1.72 (4.33)**	0.86 (2.16)	0.86 (2.17)	23.01 (57.79)
7	34.03	35.79	36.11	34.90	20.85	1.76 (5.18)	**2.08 (6.12)**	0.87 (2.56)	13.12 (38.72)
Average						4.42 (12.15)	3.57 (11.52)	2.63 (6.53)	19.43 (47.75)

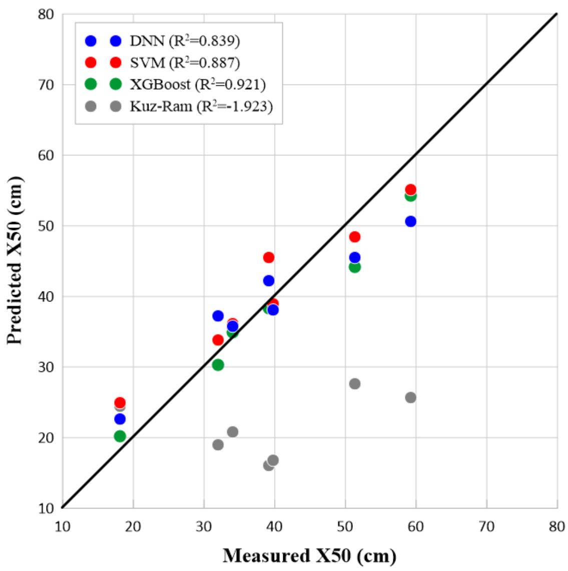


Fig. 8 Correlation of the measured X50 and predicted X50 by each model

Otherwise, results different from measured values are derived. Therefore, Cunningham introduced the correction factor (C(A)) to the Kuz–Ram model in 2005 to correct the predicted values considering site characteristics.

The Kuz–Ram model is relatively easy to use because it contains the main parameters that affect the blasting results and is expressed as a formula. Thus, it has been widely used for various purposes but cannot represent the blasting results of all sites, as mentioned above. Owing to this limitation, researchers modified the Kuz–Ram model considering the data characteristics of the target site and proposed modified Kuz–Ram models after evaluating their applicability (Gheibie *et al*. 2009, Hekmat *et al*. 2019).

The limitation of the Kuz–Ram model is that it is also applied to statistical analysis based on data and machine learning techniques based on regression analysis. These techniques are inevitably affected by the characteristics of the collected data, and the main influencing parameters and their importance vary depending on these characteristics. In addition, because the analysis results can be distorted when the amount of data biased toward a specific result increase, both the quantity and quality of data are important.

In this study, 70 datasets were used to predict the mean fragmentation size using machine learning. As mentioned above, the data used in this study cannot represent all results at the site where they were collected. Owing to the nature of the field blasting operation being restricted by the production schedule, there are limitations in collecting all data, including uncontrollable parameters, for each blasting operation; however, it is necessary to secure and reanalyze a sufficient amount of data to derive a more accurate prediction model and improve the performance of the model.

6. Conclusions

In this study, the applicability of DNN, SVM, and XGBoost, which are machine learning techniques, and the Kuz–Ram model, a statistical prediction technique, was analyzed to predict the mean fragmentation size in open-pit mines. For machine learning training, 70 field blasting datasets were collected, which included parameters such as the uniaxial compressive strength, tensile strength, rock factor, and mean in-situ block size, which are characteristics of in-situ rock.

A DNN structure with three hidden layers was used to train the DNN model, and hyperparameter tuning was performed using the RandomSearchCV class provided by the Scikit-learn library for SVM and XGBoost model training.

The DNN, SVM, and XGBoost models were trained using 63 datasets, and the performance of each model was evaluated using seven datasets. The performance was evaluated using RMSE and r^2. The RMSE of the XGBoost model (3.490) was the smallest, and its r^2 value (0.921) was closest to 1. The XGBoost model also exhibited the lowest error rate, with an average error rate of 6.53% when the error between the measured and predicted values was analyzed using the seven datasets. Consequently, the performance of the XGBoost model was found to be the best in this study, confirming that it is the most suitable model for predicting mean fragmentation size.

When the Kuz–Ram model was applied to seven datasets, the average error rate between the measured and predicted values was 47.75%, which was approximately 35.6% higher than the average error rate of the DNN model, which showed the highest error rate among the machine learning models. The difference in the error rate was large because the characteristics of the data used for the construction of the Kuz–Ram model were different from those of the data used in this study. Owing to the differences in the characteristics of the field data, a field correction factor was introduced to the recently proposed Kuz–Ram model to expand its applicability.

Because ore deposits and geological characteristics are different for each target area and there is uncertainty that the interior of the rock cannot be clearly identified, it is practically impossible to develop a fragmentation size prediction model suitable for all sites. Therefore, to predict the fragmentation size suitable for the target site, it is desirable to modify the universally used Kuz–Ram model or to develop a prediction model optimized for the target site using various machine learning techniques, as in this study.

Acknowledgments

This research was partly supported by Energy & Mineral Resources Development Association of Korea (EMRD) grant funded by the Korea government (MOTIE) (Educational-Industrial Cooperation Consortium of Energy and Mineral Resources Development-Training Program for Specialists in Smart Mining) and National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT, ME, MOTIE) (NRF-2017M3D8A2085342, the National Strategic Project, Carbon Upcycling).

References

Adams, T., Demuth, R., Margolin, L. and Nicholas, B. (1983), "Simulation of rock blasting with the shale code", *Proceedings of the 1st International Symposium on Rock Fragmentation by Blasting*, Lulea, Sweden, August.

Amoako, R., Jha, A. and Zhong, S. (2022), "Rock fragmentation prediction using an artificial neural network and support vector regression hybrid approach", *Mining*, **2**(2), 233-247. https://doi.org/10.3390/mining2020013.

Ash, R.L. (1963), "The mechanics of the rock breakage (Part 1)", *Pit and Quarry*, **56**(2), 98-100.

Bamford, T., Esmaeili, K. and Schoellig, A.P. (2021), "A deep learning approach for rock fragmentation analysis", *Int. J. Rock Mech. Min. Sci.*, **145**, 104839. https://doi.org/10.1016/j.ijrmms.2021.104839.

Bengio, Y., Courville, A. and Vincent, P. (2013), "Representation learning: a review and new perspectives", *IEEE T. Pattern Anal.*, **35**(8), 1798-1828. https://doi.org/10.1109/tpami.2013.50.

Bergmann, O.R., Wu, F.C. and Edl, J.W. (1974), "Model rock blasting measures effect of delays and hole patterns on rock fragmentation", *Int. J. Rock Mech. Min. Sci. Geomech. Abstracts*, **175**(6), 124-127. https://doi.org/10.1016/0148-9062(74)90634-2.

Burges, C.J.C. (1998), "A tutorial on support vector machines for pattern recognition", *Data Mining and Knowledge Discovery*, **2**, 121-167. https://doi.org/10.1023/a:1009715923555.

Choi, Y., Lee, C., Lee, J. and Kim, J. (2004), "Analysis of in-situ rock conditions for fragmentation prediction in bench blasting", *Tunn. Undergr. Sp. Tech.*, **14**(5), 353-362.

Choi, Y.K. (2005), "Determinants analysis and prediction of rock fragmentation in bench blasting", Ph.D. Thesis, Seoul National University, Seoul, Republic of Korea

Cunningham, C. (1983), "The Kuz-Ram model for production of fragmentation from blasting", *Proceedings of the 1st International Symposium on Rock Fragmentation by Blasting*, Lulea, Sweden, August.

Cunningham, C. (1987), "Fragmentation estimations and the Kuz-Ram model–four years on", *Proceedings of the 2nd international symposium on rock fragmentation by blasting Keystone*, Colorado, USA, August.

Cunningham, C. (2005), "The Kuz-Ram fragmentation model–20 years on", *Proceedings of the Brighton Conference Proceedings*, Brighton, UK, October.

Dahl, G.E., Sainath, T.N. and Hinton, G.E. (2013), "Improving deep neural networks for LVCSR using rectified linear units and dropout", *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, May.

Dumakor-Dupey, N.K., Sampurna, A. and Jha, A. (2021), "Advances in blast-induced impact prediction-A review of machine learning applications", *Minerals*, **11**, 601. https://doi.org/10.3390/min11060601.

Ebrahimi, E., Monjezi, M., Khalesi, M.R. and Armaghani, D.J. (2015), "Prediction and optimization of back-break and rock fragmentation using an artificial neural network and a bee colony algorithm", *Bull. Eng. Geol. Environ.*, **75**, 27-36. https://doi.org/10.1007/s10064-015-0720-2.

Fukushima, K. (1975), "Cognitron: a self-organizing multilayered neural network", *Biol. Cybern.*, **20**(3), 121-136. https://doi.org/10.1007/BF00342633.

Gebreyesus, Y., Dalton, D., Nixon, S., Chiara, D.D. and Chinnici, M. (2023), "Machine learning for data center optimizations: feature selection using Shapley Additive exPlanation (SHAP)", *Future Internet*, **15**(88), 1-18. https://doi.org/10.3390/fi15030088.

Gheibie, S., Aghababaei, H., Hoseinie, S.H. and Pourrahimian, Y. (2009), "Modified Kuz-Ram fragmentation model and its use at the Sungun Copper Mine", *Int. J. Rock Mech. Min. Sci.*, **46**(6), 967-973. https://doi.org/10.1016/j.ijrmms.2009.05.003.

Hekmat, A., Munoz, S. and Gomez, R. (2019) "Prediction of rock fragmentation based on a modified Kuz-Ram model", *Proceedings of the 27th International Symposium on Mine Planning and Equipment Selection-MPES 2018*, New York, NY, USA.

Katsabanis, P.D., Tawadrous, A., Braun, C. and Kennedy, C. (2006), "Timing effects on the fragmentation of small-scale blocks of granodiorite", *Int. J. Blast. Fragmentation*, **10**, 83-93. https://doi.org/10.1080/13855140600858339.

Kim, Y., Hong, J., Shin, J. and Kim, B. (2022), "Shield TBM disc cutter replacement and wear rate prediction using machine learning techniques", *Geomech. Eng.*, **29**(3), 249-258. https://doi.org/10.12989/gae.2022.29.3.249.

Koulli, S. and Rustan, P. (1993), "Computerized design and result prediction of bench blasting", *Proceedings of the 4th International Symposium on Rock Fragmentation by Blasting*, Vienna, Austria, July.

Kuznetsov, V.M. (1973), "The mean diameter of the fragments formed by blasting rock", *Soviet Min. Sci.*, **9**, 144-148. https://doi.org/10.1007/bf02506177.

Kwak, N.S. and Ko, T.Y. (2022), "Machine learning-based regression analysis for estimating Cershar abrasivity index", *Geomech. Eng.*, **29**(3), 219-228. https://doi.org/10.12989/gae.2022.29.3.219.

Langefors, U. and Kihlström, B. (1978), *The modern technique of rock blasting*, 3rd Ed., Wiley, New York, NY, USA.

Lee, K.B., Shin, H.S., Kim, S.H., Ha, D.M. and Choi, I. (2019), "A study on automatic classification of characterized ground regions on slopes by a deep learning based image segmentation", *Tunn. Undergr. Sp. Tech.*, **29**(6), 508-522. https://doi.org/10.7474/TUS.2019.29.6.508.

Lee, S.J. (2016), "Prediction model for rock fragmentation based on field data and 3D measurement of muck pile with UAV", Ph.D. Thesis, Kangwon National University, Chuncheonsi, Ganwondo, Republic of Korea

Li, E., Yang, F., Ren, M., Zhang, X. and Zhou, J. (2021), "Prediction of blasting mean fragment size using support vector regression combined with five optimization algorithms", *J. Rock Mech Geotech. Eng.*, **13**(6), 1380-1397. https://doi.org/10.1016/j.jrmge.2021.07.013.

Li, J., Yang, Y., Hu, Y., Zhu, X., Ma, X. and Yuan, X. (2023), "Using multidimensional data to analyze freeway real-time traffic crash precursors based on XGBoost-SHAP algorithm", *J. Adv. Transport.*, **2023**, 1-18. https://doi.org/10.1155/2023/5789573.

Lilly, P. (1986), "An empirical method of assessing rock mass blastability", *Proceedings of the Large Open Pit Mining Conference*, Newman, Australia, October.

Lundberg, S.M. and Lee S.I. (2017), "A unified approach to interpreting model predictions", *Proceedings of the 31st Conference on Neural Information Processing System (NIPS 2017)*, Long Beach, CA, USA. https://doi.org/10.48550/arXiv.1705.07874.

Mahmoodzadeh, A., Nejati, H.R., Mohammadi, M., Ibrahim, H.H., Mohammed, A.H. and Rashid, S. (2022b), "Assessment of wall convergence for tunnels using machine learning techniques", *Geomech. Eng.*, **31**(3), 265-279. https://doi.org/10.12989/gae.2022.31.3.265.

Mahmoodzadeh, A., Nejati, H.R., Mohammadi, M., Ibrahim, H.H., Rashidi, S. and Mohammed, A.H., (2022a), "Meta-heuristic optimization algorithms for prediction of fly-rock in the blasting operation of open-pit mines", *Geomech. Eng.*, **30**(6), 489-502. https://doi.org/10.12989/gae.2022.30.6.489.

McFall, G.P., Bohn, L., Gee, M., Drouin, S.M., Fah, H., Han, W., Li, L., Camicioli, R. and Dixon, R.A. (2023), "Identifying key multi-modal predictors of incipient dementia in Parkinson's disease: a machine learning analysis and tree SHAP interpretation", *Front. Aging Neurosci.*, **15**, 1-16. https://doi.org/10.3389/fnagi.2023.1124232.

McHugh, S. (1983), "Computational simulations of dynamically induced fracture and fragmentation", *Proceedings of the 1st International Symposium on Rock Fragmentation by Blasting*, Lulea, Sweden, August.

Singh, P.K., Roy, M.P., Paswan, R.K., Sarim, M., Kumar, S. and Ranjan, R. (2016), "Rock fragmentation control in opencast blasting", *J. Rock Mech. Geotech. Eng.*, **8**(2), 225-237. https://doi.org/10.1016/j.jrmge.2015.10.005.

Vapnik, V. (1999), *The nature of statistical learning theory*, 2nd Ed., Springer science & business media, New York, NY, USA.

GC