Shahrood University of Technology

Journal of Mining and Environment (JME)

Journal homepage: www.jme.shahroodut.ac.ir

Iranian Society of Mining Engineering (IRSME)

Prediction and Minimization of Blast-induced Back Break, Using Deep Neural Network, Extreme Gradient Boosting, Lasso Regression, and Gravitational Search Algorithm

Eslam Ghojoghi[1], Hamid Mansouri[1*], Mohamad Ali Ebrahimi Farsangi[1], and Esmat Rashedi[2]

1. Department of Mining Engineering, Shahid Bahonar University of Kerman, Iran
2. Department of Electrical and Computer Engineering, Graduate University of Advanced Technology, Kerman, Iran

Article Info

Received 16 September 2025
Received in Revised form 11 October 2025
Accepted 7 February 2026
Published online 7 February 2026

DOI: 10.22044/jme.2026.16834.3303

Keywords

Blasting; Back break
Deep Neural Network
Extreme Gradient Boosting
Lasso regression
Gravitational Search Algorithm

Abstract

Back break is an undesirable consequence of rock blasting, which causes energy loss from the explosive and reduces operational efficiency. Consequently, it is essential to forecast it to enable control and prevent operational cost losses. The objective of this scientific research is to utilize Deep Neural Networks, Extreme Gradient Boosting, and Lasso Regression in conjunction with the Gravitational Search Algorithm to predict and minimize the occurrence of blast-induced back break at Gol-e-Gohar 4 iron ore mine, Sirjan, Kerman, Iran. The constructed models comprise a set of nine input parameters, including blasting design parameters and rock geomechanical properties, and produce back break as a single output. The datasets used for training and evaluation consist of 266 blasting records extracted from Gol-e-Gohar 4 iron ore mine. The results obtained show that the Deep Neural Network model, with an R^2 of 0.81 and an MSE of 0.70, outperforms the Extreme Gradient Boosting and Lasso Regression models in predicting back break. Furthermore, the application of the optimization algorithm resulted in optimized parameter values that minimize back break.

1. Introduction

Back break (BB) is an undesirable consequence of suboptimal blast design, characterized by the fracture of rocks outside the intended blast area, extending beyond the last row of the blasting pattern. This phenomenon can lead to mine bench instability, inadequate fragmentation, rock falls, and subsequent increases in both mining costs and operational time. Therefore, given the potential ramifications of blasting-induced back break, it is crucial to identify and optimize the factors influencing its extent. This endeavor is vital for controlling back break quantities, mitigating adverse effects, and minimizing potential damages [1, 2].

In recent years, numerous scientific investigations have been undertaken with the objective of predicting back break and discerning the associated influential factors. While these approaches have provided valuable insights, many are limited by linear assumptions, insufficient handling of nonlinear interactions between blasting and rock mass parameters, or reliance on small datasets, which reduces predictive reliability [3-5]. Monjezi et al. (2012) utilized Artificial Neural Networks (ANN) to predict back break and conducted a comparative analysis with conventional regression analysis. The outcomes demonstrated the good performance of ANN modelling [6]. Also, Faramarzi et al. (2012) introduced a novel model that incorporated the fundamental principles of Rock Engineering Systems (RES) approach to predict back break and assess the associated blast risk. By comparing the model's results with actual measurements of back

Corresponding author: hmansouri@uk.ac.ir (H. Mansouri)

break, a consistent correlation between the predicted risk level and the measured back break was observed, indicating the model's strong performance [7]. Moreover, Monjezi et al. (2012) conducted a study with the objective of developing an innovative Neuro-Genetic (NG) model for predicting flyrock and back break. This study, utilized Genetic Algorithm (AG) to optimize the parameters of neural network. The performance and accuracy of the developed model were evaluated, using rigorous statistical methods, which unequivocally confirmed the remarkable effectiveness of the NG modelling approach in comparison to alternative methods [8]. Furthermore, Esmaeili et al. (2012) explored the utilization of ANN and Adaptive Neuro-Fuzzy Inference System (ANFIS) for predicting back break. Also, the performances of these models were compared with multiple regression model. Results revealed that the ANFIS model demonstrated good performance compared to both the ANN and multiple regression models [9]. In addition, Sari et al. (2014) investigated the use of stochastic modelling for back break prediction and examined the effects of controllable blasting parameters. In this research, an empirical regression equation was developed to relate back break to various blasthole attributes through multiple regression analysis. Also, a Monte Carlo (MC) simulation approach was employed to probabilistically model the occurrence of back break accounting for variability in blasting conditions. The results demonstrated that stochastic modelling provided an effective means for assessing the impact of parameter uncertainty on back break predictions [10]. Ebrahimi et al. (2015) employed the ANN in combination with an Artificial Bee Colony (ABC) algorithm to predict the fragmentation of rocks and back break. The findings of their research demonstrated the robustness and accuracy of the ANN model, indicating its high reliability in this context [11].

Also, Hasanipanah et al. (2017) coupled the Particle Swarm Optimization (PSO) algorithm with the ANFIS for predicting blast-induced back break. To evaluate its performance, the accuracy of the proposed model was assessed in comparison with the Multiple Linear Regression (MLR) technique. The obtained results demonstrated the potential of the PSO-ANFIS model for accurately predicting back break [12]. Moreover, Katasabanis (2020) reviewed fragmentation prediction methods and highlighted key limitations in engineering models, emphasizing the need for improved treatment of blastability and energy partitioning. The study discussed rock damage in terms of microcracks, based on both experiments and numerical simulations, and noted that grindability is influenced by energy consumption, powder factor, and powder distribution. Findings showed that blast geometry, borehole pressure, delay timing, and stress-wave interactions significantly affect damage, while linking blasting to milling performance requires integrated analysis of lithology, fracture networks, detailed blasting parameters, and mill outcomes [13].

Additionally, Kumar et al. (2021) applied the Random Forest (RF) and linear regression methods to predict back break. Their findings revealed that the RF method outperformed linear regression, demonstrating better predictive performance [1]. On the other hand, YU et al. (2021) employed a Support Vector Machine (SVM) model along with the Moth-Flame Optimization (MFO) and Whale Optimization Algorithm (WOA) to predict back break. The performance of these two models was assessed in comparison to other hybridized SVM models. The findings of the study revealed that both the SVM-WOA and SVM-MFO models exhibited enhanced performance compared to the standard SVM. Moreover, the SVM-MFO model demonstrated superior performance relative to the SVM-WOA model and other hybridized SVM models [14]. Also, Khajouei Sirjani et al. (2022) developed a multilayer perceptron ANN as well as six multiple linear and nonlinear statistical models to predict back break. The study findings indicated that the ANN model's back break prediction accuracy was acceptable and showed less errors in comparison to the statistical models tested [15]. Moreover, Kumar et al. (2023) formulated equations, using genetic programming, incorporating selected parameters related to blast design, to predict ground vibration and back break. The findings of this study demonstrated that this approach yielded acceptable accuracy in predicting both back break and ground vibration [16]. Furthermore, Sorabi et al. (2024), employed the WOA, Multi-Verse Optimizer (MVO), Ant Lion Optimizer (ALO), and Sine Cosine Algorithm (SCA) for predicting back break. Through a comparative analysis of the results obtained from these four prediction models, it was observed that the MVO-based model showed the highest level of accuracy and the lowest amount of error, indicating its good performance in predicting back break [17].

Also, many scholars have employed machine learning techniques to for predicting back break [18-21]. For instance, Khandelwal and Monjezi explored the utilization of the SVM method and

compared it with the Multi-Variate Regression Analysis (MVRA). The results indicated that the SVM method exhibited superior performance in predicting back break compared to MVRA [22]. Moreover, Sharma et al. (2021) developed Multivariate Regression Analysis (MVRA) and Random Forest (RF) models for the prediction of back break. The findings indicated that the RF model outperforms MVRA in relation to accuracy to back break prediction [23]. Additionally, Dai et al. (2022) developed a novel hybrid intelligence approach based on RF and PSO for predicting back break. The results illustrated the good performance of the PSO–RF model in predicting back break [24].

In addition, Soltanalinejad and Moomivand (2024) investigated the influence of blasthole design on overbreak (LOZ), slope angle deviation ($\alpha° - \beta°$), and damaged zone width (LDZ). They found that shock wave reflections from bench faces amplify breakage toward the crest, and that LOZ, ($\alpha° - \beta°$), and LDZ follow power-law increases with blasthole inclination, charge, burden, spacing, stemming, and stiffness ratio. In contrast, specific charge and blastability index showed weak correlations. To improve prediction, they developed multivariable nonlinear models with strong accuracy, indicating that reduced parameter values could minimize overbreak and slope deviation [25]. Furthermore, Kunkyin-Saadaari et al. (2024) conducted a comparative assessment of four advanced machine learning technique including Lagrangian Support Vector Machine, Radial Basis Function Neural Network, Gaussian Process Regression, and Extreme Gradient Boostin for backbreak prediction. Their results indicated that XGBoost outperformed the other models, emphasizing its potential as a robust and reliable approach in this application [26]. Moreover, Jethro et al. (2024) employed an artificial neural network (ANN) to predict backbreak, achieving high predictive accuracy and revealing strong correlations between blasting parameters and backbreak. The study further identified stiffness ratio as a more influential factor than rock strength in controlling backbreak [27]. Additionally, Kannavena et al. (2025) utilized back propagation neural networks (BPNN) to predict backbreak and benchmarked the results against decision tree regression and linear regression models. The findings demonstrated the superior predictive capability of the BPNN model, while also identifying burden, spacing, stemming length, bench height, number of holes, and powder factor as the most critical parameters influencing backbreak [28]. Also, Shahani and Zheng (2025) proposed a least squares support vector machine (LSSVM) model optimized with metaheuristic algorithms, including genetic algorithm (GA), PSO, and grey wolf optimization (GWO), for backbreak prediction. Among these, the GWO-LSSVM model achieved the highest accuracy, with sensitivity analysis indicating powder factor as the most influential parameter and burden as the least [29].

In this research, the primary goal is to devise a predictive model for estimating back break. For this purpose, Deep Neural Network (DNN) regression, Least Absolute Shrinkage and Selection Operator (Lasso), and Extreme Gradient Boosting (XGBoost) approaches are applied. Also, hyperparameter optimization is used to identify the optimal network conFigureurations. Additionally, the Gravitational Search Algorithm (GSA) is utilized to optimize the blasting parameters with the goal of minimizing BB. The study further contributes by integrating predictive modeling with metaheuristic optimization to provide a practical tool for minimizing back break. It evaluates the influence of key blasting and rock parameters through sensitivity analysis, offering actionable insights for optimizing blast design and improving operational efficiency.

2. DNN, Lasso, and XGBoost

2.1. DNN

The sequential deep neural network model operates by leveraging a series of interconnected layers to process input data and make predictions. This model follows a feedforward approach, where information flows from the input layer through the hidden layers to the output layer. Each layer comprises multiple neurons that perform computations and transmit signals to the subsequent layer. The model's ability to capture intricate relationships within the data is enhanced through the use of nonlinear activation functions (Figure 1) [30-32].

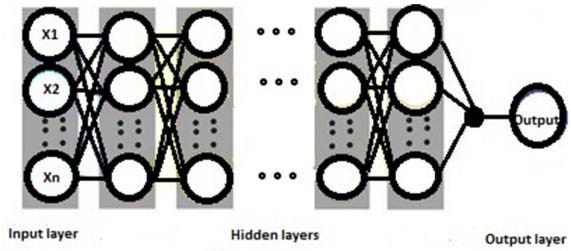


Figure 1. Deep neural network structure

Throughout the training phase, the model iteratively fine-tunes its parameters to minimize the disparity between the predicted output and the actual target values. This process, referred to as backpropagation, involves the computation of the gradient of the loss function concerning the weights and biases of the model. By utilizing optimization algorithms such as stochastic gradient descent, the model updates its parameters in the direction that minimizes the loss. This iterative optimization continues until the model achieves a satisfactory level of accuracy and generalization [33].

Once trained, the sequential deep neural network regression model can be employed for prediction by feeding new input data through the trained network. The input data propagates through the layers, and the final output is generated at the output layer. The model's performance can be assessed by evaluating various metrics, such as mean squared error or coefficient of determination, which measure the discrepancy between predicted and actual values. The ability of the sequential deep neural network regression model to capture intricate patterns and nonlinear relationships makes it a powerful tool for predictive modeling in various domains [34].

2.2. Lasso

The Lasso regression approach has gained significant attention in statistical modeling and machine learning for its capacity to manage high-dimensional datasets and perform variable selection. It introduces a regularization term, controlled by the hyperparameter alpha, into the objective function of a variant of linear regression. This regularization term penalizes the sum of the absolute values of the coefficients, encouraging sparsity in the coefficient vector and shrinking less important coefficients toward zero. This sparsity-inducing property not only aids in feature selection but also enhances interpretability by identifying influential predictors. The Lasso approach is a powerful tool for predictive modeling and data analysis across various domains. The working mechanism of Lasso regression involves an optimization process that minimizes a combined objective function. This function combines the Residual Sum of Squares (RSS) with the product of the regularization term (alpha) and the sum of the absolute values of the coefficients. Different optimization algorithms, such as coordinate descent or least angle regression, can be employed to estimate the coefficients. These algorithms iteratively update the coefficient values by selecting influential predictors and shrinking coefficients of less important predictors. The process continues until convergence or a predetermined stopping criterion is satisfied. By tuning the hyperparameter alpha, Lasso regression provides a flexible trade-off between model complexity and prediction accuracy [35, 36].

2.3. XGBoost

XGBoost regression is a robust and extensively utilized machine learning algorithm that has gained significant attention in various fields. It is an extension of the gradient boosting framework that combines the strengths of boosting and gradient descent techniques. XGBoost is particularly effective for regression tasks as it employs an ensemble of weak prediction models, typically decision trees, to generate a strong predictive model. The algorithm operates by progressively incorporating decision trees into the ensemble, with each subsequent tree correcting the errors made by the previous trees. The learning process involves optimizing a loss function by finding the best splits in the data that minimize the loss. By leveraging gradient descent optimization and employing advanced regularization techniques, XGBoost achieves high predictive performance, robustness to overfitting, and improved interpretability [37].

The working mechanism of XGBoost regression can be summarized in a few key steps. First, an initial prediction model is created by fitting a base decision tree to the training data. Then, the algorithm computes the residuals (the disparities between the actual target values and the predictions generated by the preceding model). In subsequent iterations, additional decision trees, called weak learners, are added to the ensemble. These trees are built to predict the residuals, with the objective of minimizing the errors made by the preceding models. The final prediction of the XGBoost model is obtained by summing the predictions from all the decision trees. The learning process involves optimizing a loss function, which can be customized based on the specific regression task. XGBoost employs gradient descent optimization to find the best splits in the data that minimize the loss function. Additionally, it incorporates regularization techniques to control the model's complexity and prevent overfitting. The combination of boosting, gradient descent optimization, and regularization makes XGBoost

regression a highly effective and versatile algorithm suitable for diverse regression tasks [38].

2.4. Bayesian Hyperparameter Optimization

Bayesian Hyperparameter Optimization (BHO) is a sophisticated and efficient technique used to find the optimal hyperparameters for machine learning models. It addresses the challenge of selecting appropriate hyperparameter values, which significantly impact a model's performance, by treating it as a sequential decision-making problem. Unlike traditional methods that rely on grid search or random search, BHO employs Bayesian inference to guide the search process. It constructs a probabilistic model that captures the relationship between hyperparameters and model performance based on observed data. By iteratively updating the model with new observations, it intelligently explores the hyperparameter space to identify promising regions and utilizes this information to search for the optimal hyperparameter conFigureuration. This approach enables faster convergence and more efficient exploration of the hyperparameter space, leading to improved model performance and reduced computational costs [39].

The working mechanism of BHO can be summarized in a few key steps. First, a prior probability distribution is defined over the hyperparameter space, representing initial beliefs about the hyperparameters. Then, an acquisition function is used to determine the next set of hyperparameters to evaluate. The acquisition function balances exploration and exploitation by considering both the uncertainty of the model and the potential performance improvement. By evaluating the model with the selected hyperparameters, the corresponding performance metrics are obtained, which are used to update the probabilistic model through Bayesian inference. This update incorporates the newly observed data to refine the posterior distribution over the hyperparameters. The process iterates, with the acquisition function guiding the selection of subsequent hyperparameters for evaluation, until a specified termination criterion is satisfied. BHO efficiently leverages the probabilistic model to concentrate the search on promising areas of the hyperparameter space, leading to improved efficiency and effectiveness in finding the optimal hyperparameter conFigureuration for machine learning models [40, 41].

2.5. Talos Hyperparameter Optimization

Talos Hyperparameter Optimization is a technique used to automatically search and optimize the hyperparameters of a machine learning model. It functions by methodically investigating various combinations of hyperparameters to find the conFigureuration that yields the best performance. Talos employs a combination of grid search and random search algorithms to efficiently explore the hyperparameter space [42].

To begin the optimization process, Talos defines the range or set of values for each hyperparameter to be tuned. It then generates different combinations of these hyperparameters and trains the model using each conFigureuration. The model's performance is assessed, using a specified evaluation metric, such as accuracy or mean squared error. Talos tracks the performance for each conFigureuration and selects the hyperparameter combination that produces the best results.

Additionally, Talos employs an intelligent search strategy that adapts based on the performance of previous conFigureurations. It dynamically adjusts the search space by focusing on promising areas and gradually refining the hyperparameter values. This iterative process continues until a satisfactory set of hyperparameters is found. By automating the hyperparameter tuning process, Talos aids in identifying the optimal conFigureuration for a machine learning model, improving its performance and generalization capabilities [43, 44].

2.6. Gravitational Search Algorithm

The GSA, a metaheuristic optimization method developed by Rashedi et al., is based on the principles of gravity and mass interaction [45]. It has found applications across various research fields for optimization purposes [46, 47]. This algorithm, based on Newtonian gravity principles and mass interaction, demonstrates considerable promise as an innovative optimization method [48]. In this algorithm, agents are considered entities, and their performance is evaluated according to their respective masses. Each object in this algorithm represents either a comprehensive solution or a constituent part of a solution to the specified problem. All objects interact with one another through gravitational forces, leading to a collective motion of objects toward those with higher mass. Objects with greater mass, reflecting

higher fitness values, signify the optimal solutions to the problem. Therefore, these objects display slower motion compared to lighter ones, which are associated with suboptimal solutions [32, 49-51].

3. Methodology

The methodology employed in predicting blasting back break, using a DNN, Lasso, and XGBoost, optimized with hyperparameter tuning, involves a series of steps to ensure the robustness and applicability of the modeling (Figure 2). This methodology encompasses careful consideration of the data, model architecture, and hyperparameters, along with extensive testing and validation.

The supervised modeling is divided into three main steps: pre-processing, processing, and post-processing. Initially, the main objective and problem statement are defined. The pre-processing stage involves data collection through field sampling, followed by descriptive statistics and correlation analysis to gain insights into the data. Subsequently, the data are normalized to ensure uniformity. To evaluate the model's performance, the data are then divided into training and testing sets. The models are developed through feature extraction, hyperparameter tuning, calibration, and validation. These steps collectively contribute to building an optimized model capable of effectively predicting blasting back break. In the final step, the GSA was utilized with the prediction models to determine the optimal blast parameters focused on minimizing back break.

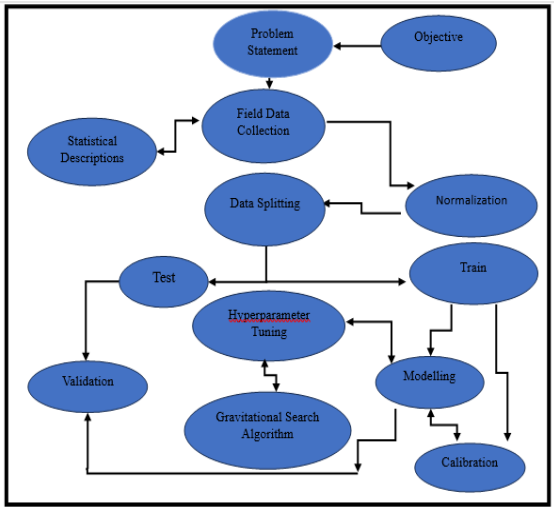


Figure 2. Flowchart illustrating the methodology of the study

3.1. Data Collection

266 field datasets were collected from eight zones, each characterized by different geomechanical properties of the rocks, at Gol-e-Gohar 4 iron ore mine, which is situated approximately 55 km southwest of Sirjan, Kerman, Iran (Figure 3).

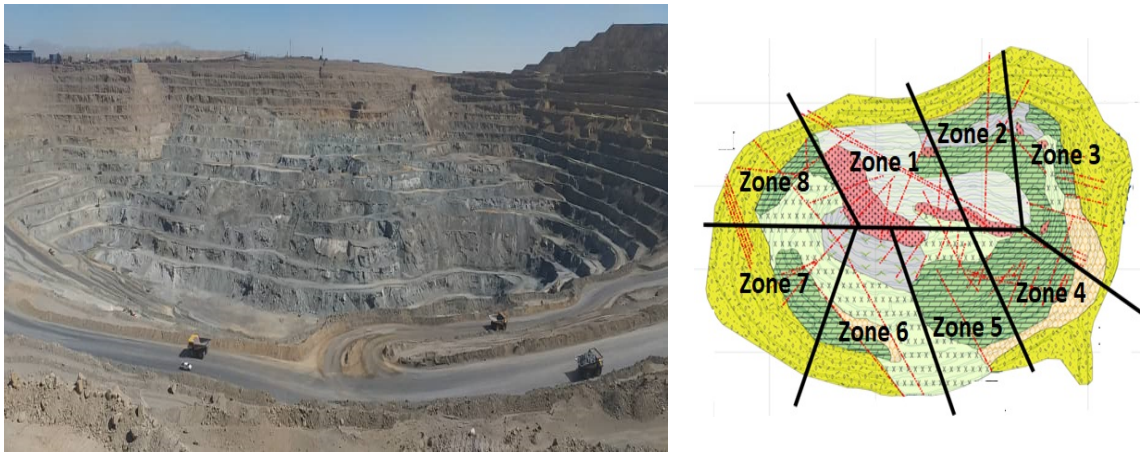


Figure 3. A view of Gol-e-Gohar 4 iron ore mine

Each dataset includes burden (B), spacing (S), powder factor (PF), hole diameter (D), number of rows (NR), and last row charge (Charge). To incorporate the geomechanical properties of rocks into the study, various parameters were measured for eight rock types: Conglomerate (CM), Alluvium (AL), Talcum-Schist (TS), Chlorite-Schist (CS), Quartz-Schist (QS), Gneiss (GS), Magnetite (MG), and Hematite (HM). These measurements encompassed uniaxial compressive strength (UCS), density (Den), and geological strength index (GSI) (Table 1). UCS and density were determined through laboratory testing of representative rock samples. GSI values were

estimated in the field based on the Hoek–Brown method using compass-based mapping of joint orientation and evaluation of joint conditions, and in some cases cross-checked with Bieniawski's classification charts. Back break was quantified by surveying the blast sites before and after detonation; the difference between the designed excavation line and the actual post-blast profile was taken as the back break value.

These nine blasting and geomechanical parameters were used as input parameters to develop the models, with back break as the output. The descriptive statistics for the input and output variables are presented in Table 2. Additionally, the frequency distribution of the data is illustrated in Figure 4, providing a visual representation of the data distribution.

Table1. Different rocks geomechanical parameters range in Gol-e-Gohar 4 iron ore mine

Rock Type	CM	AL	TS	CS	QS	GS	MG	HM
UCS (MPa)	2.46-11.41	2.46-5.37	0.5	21.81-73.89	18=51.8	46.75-56.5	40.5-47.8	24.47-41.05
Den (gr/cm^3)	2.35-2.65	2.36-2.65	2.5	2.71	2.67	2.72	3.36	3.69
GSI	22-35	22-29	15	36-43	45-56	51-57	62-65	64-65

Table 2. Statistical analysis of input and output data for modelling blast-induced back break

Parameter	B (m)	S (m)	PF (gr/ton)	D (inch)	NR	Charge (Kg)	UCS (MPa)	Den (gr/cm3)	GSI	BB (m)
Mean	5.72	7.65	204.14	7.86	3.44	10784.20	22.91	2.65	37.11	9.99
Std	0.69	0.93	101.91	0.54	1.15	7124.62	18.29	0.25	15.08	1.69
Min	2.50	3.50	90.35	6.00	1.00	413.26	0.50	2.35	15.00	3.25
Median	6.00	8.00	171.69	8.00	3.00	10023.74	24.75	2.67	40.00	10.39
Max	6.00	8.00	799.91	8.50	7.00	30367.50	73.89	3.69	65.00	12.85

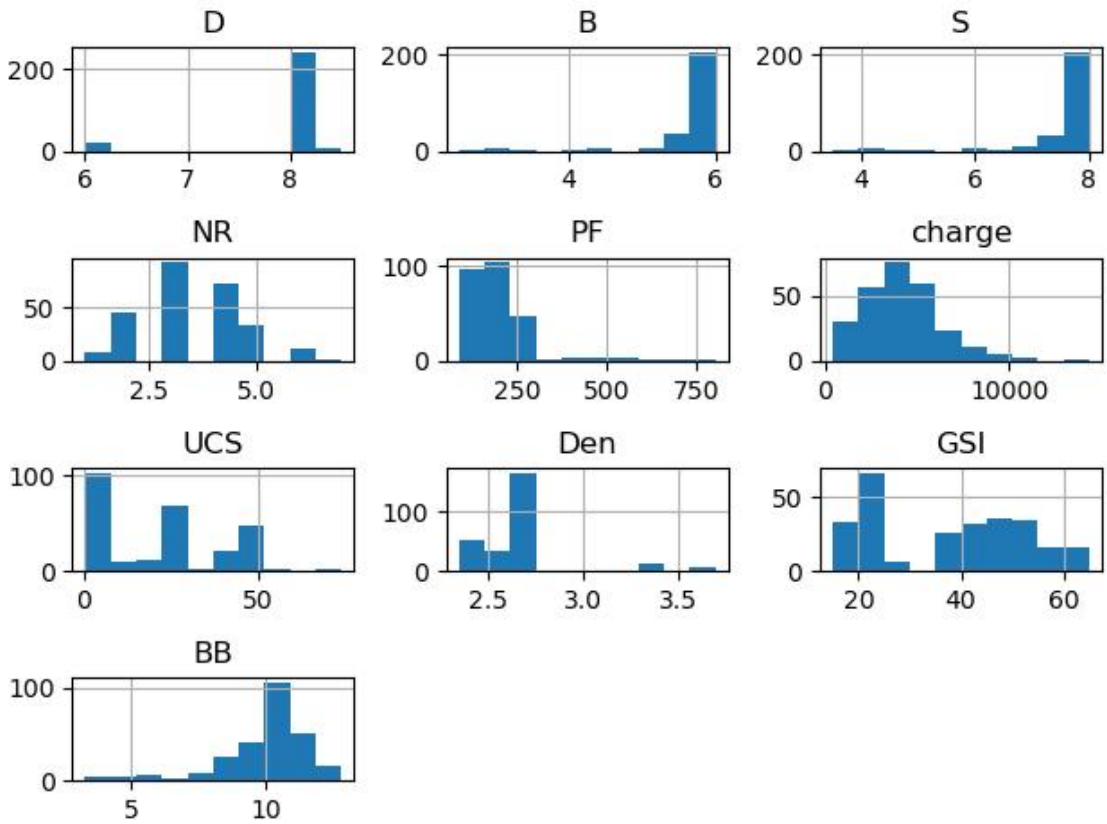


Figure 4. Frequency distribution of the data

The initial pre-processing stage involves loading the data, partitioning it into training and testing sets, and performing data scaling as needed. In this case, 75% of the datasets are allocated as training samples for model calibration, while the remaining 25% serve as test samples for model validation. The distribution of the target variable (back break) for the training and testing datasets is

compared in Figure 5. As shown, there is an almost uniform distribution of data for both training and testing datasets.

To ensure uniform scaling of the input variables, all data were normalized to a range of [0, 1], using Equation 1 [52].

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

where the variable x represents the values before normalization, and xnorm represents the values after normalization. The variables xmin and xmax denote the minimum and maximum values, respectively, within the dataset before normalization.

Also, a Principal Component Analysis (PCA) decomposition was carried out on training and testing data. PCA is a technique for reducing dimensionality that is employed to convert a dataset with high dimensions into a lower-dimensional space while preserving the most critical information. It achieves by identifying the orthogonal axes, called principal components, that capture the maximum variance in the data, allowing for a simplified representation and improved data visualization [53, 54]. A plot on the concatenated input dataset, which is formed by combining the training and testing data, is shown in Figure 6. As it can be observed from this Figure, there is relatively a good distribution of data for both training and testing datasets.

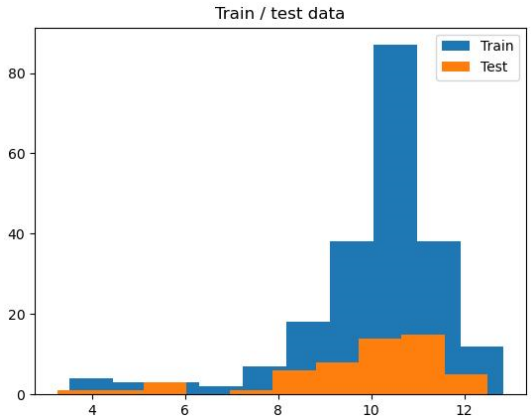


Figure 5. The distribution of target variable (back break) for the training and testing datasets

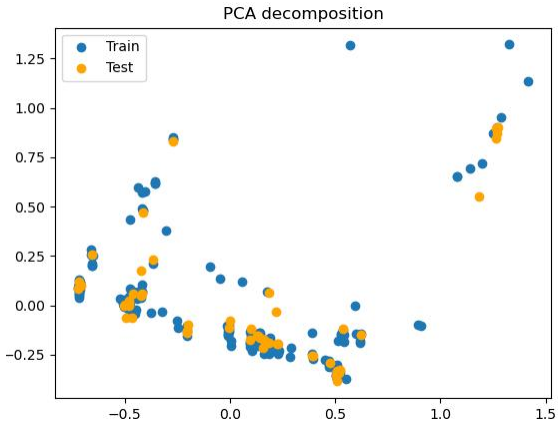


Figure 6. Principal Component Analysis for training and testing datasets

3.2. Model Development

3.2.1. DNN Model

In this study, the DNN approach was employed to model and predict blasting back break. The model was optimized using Talos hyperparameter optimization tuning. The DNN architecture is illustrated in Figure 7. It includes an input layer, four hidden layers, and an output layer, each with different activation functions. The number of neurons in each hidden layer and the choice of activation functions were considered as hyperparameters and specified in the params dictionary for hyperparameter search. The output layer consists of a single neuron with a linear activation function. The model was compiled with the mean squared error loss function and the Adam optimizer. The training process utilized the training data, and the model's performance was evaluated on validation data. The number of epochs and batch size were also considered hyperparameters and included in the params dictionary for hyperparameter tuning. The search spaces for hyperparameters are shown in Table 3.

The Talos scan function was employed to conduct comprehensive hyperparameter search, which facilitates the training and evaluation of multiple models, each employing distinct combinations of hyperparameters. By systematically exploring various hyperparameter conFigureurations, the best model based on the minimum validation mean squared error function was determined. This approach allowed for fine-tuning the model's hyperparameters and identifying the optimal settings to enhance its predictive performance.

Figure 8 shows the parameter correlation heatmap plot from hyperparameter tuning. The values in this Figure indicate the correlation between each pair of hyperparameters. A correlation of 1 means they are perfectly positively correlated, -1 means perfectly negatively correlated, and 0 means no correlation. In summary, this correlation heatmap shows that

based on the hyperparameter tuning with Talos, the different hyperparameters tuned for this DNN model have very low correlations with each other. This means they can likely be optimized independently without worrying about interaction effects between hyperparameters.

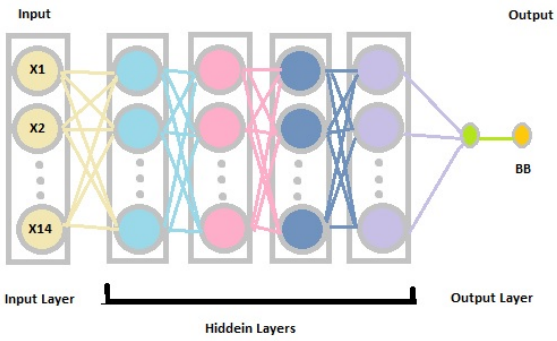


Figure 7. The architecture of DNN model for predicting back break

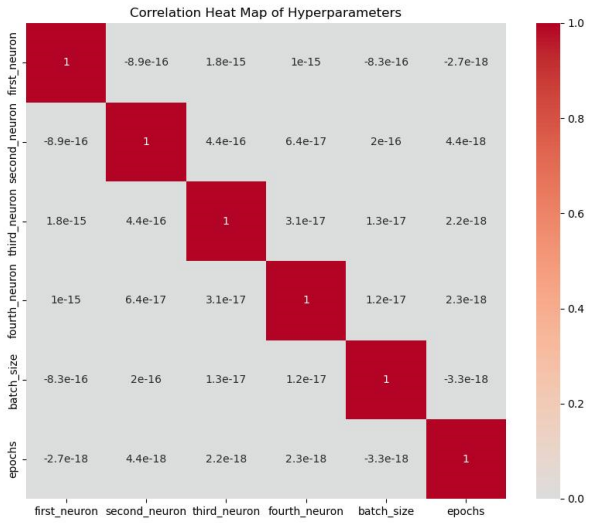


Figure 8. The parameter correlation heatmap plot from hyperparameter tuning

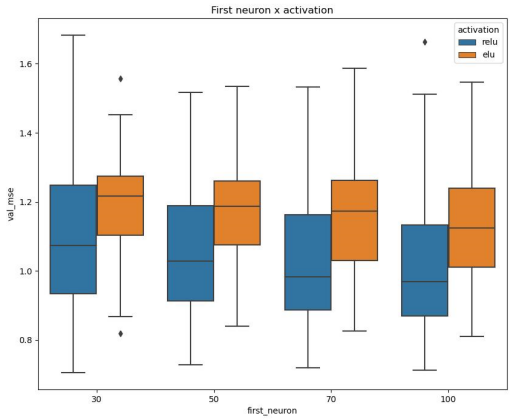
Also, the relationship between hyperparameters and the validation MSE visualized as box plots, shown in Figure 9, which indicate that ReLU activation function and 50 neurons for the first layer, ReLU activation function and 50 neurons for the second layer, ReLU activation function and 50 neurons for the third layer, and ReLU activation function and 70 neurons for the fourth layer, ReLU activation function and 50 batch size , and ReLU activation function and 200 epochs, are the optimal values for hyperparameter tuning. Also, the optimal values of hyperparameter tuning are shown in Table 3.

Table 3. Search space and optimal values of hyperparameter tuning

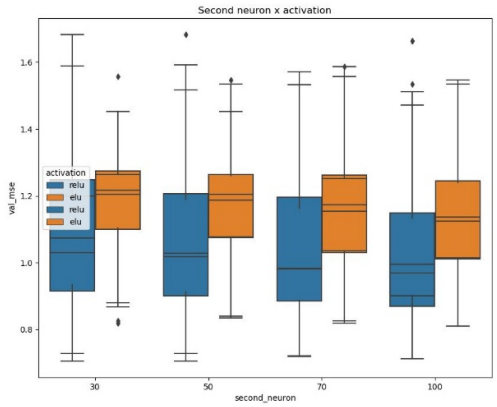
Parameter	Search Space	Optimal Value
First layer neuron	30,50,70,100	30
Second layer neuron	30,50,70,100	50
Third layer neuron	30,50,70,100	70
Fourth layer neuron	30,50,70,100	70
Activation function	relu , elu	relu
Batch size	20,50	20
Epochs	100,200	200

3.2.2. Lasso Modelling

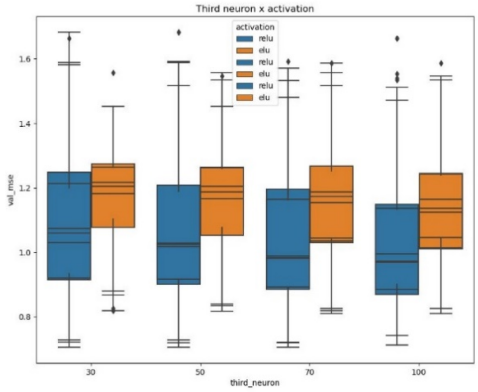
A Lasso regressor model was also constructed to predict back break. The data were split and pre-processed in the same manner as for the DNN model. Subsequently, Bayesian Hyperparameter Optimization (BHO) was performed to fine-tune the model for optimal performance. The hyperparameters included in the search space were "alpha" (the regularization parameter), "fit_intercept," "copy_X," "positive," "precompute," and "max_iter" (maximum number of iterations). Hyperparameter optimization was conducted using the BayesSearchCV class with a 5-fold cross-validation (cv=5). The BayesSearchCV class employs Bayesian optimization to efficiently explore the hyperparameter space and identify the optimal conFigureuration. The model was fitted to the training data using this method. Once the hyperparameter search was complete, the model with the best conFigureuration was used for back break prediction. The search space and optimal hyperparameters for the Lasso model are shown in Table 4.



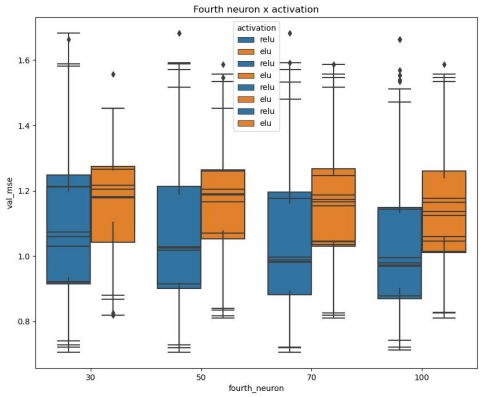
a) First neuron x activation



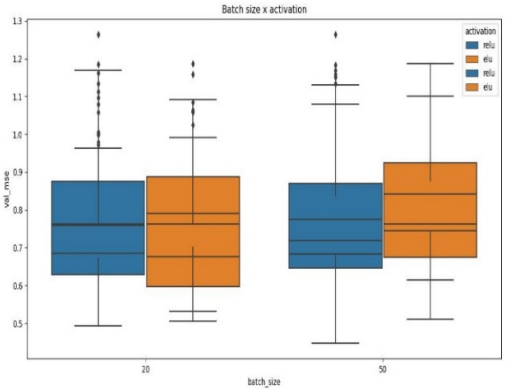
b) Second neuron x activation



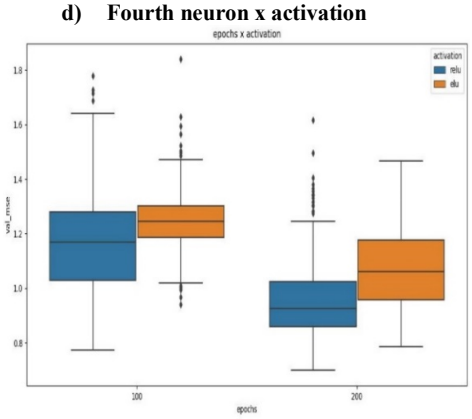
c) Third neuron x activation



d) Fourth neuron x activation



e) Batch size x activation



f) Epoch x activation

Figure 9. The relationship between hyperparameters and the validation mean squared error

Table 4. Search space and optimal values of Lasso model

Parameter	Search Space	Optimal Value
alpha	0.0001,0.001,0.01,0.1,0.5,0.6,0.7,0.8,0.9,1	0.01
fit_intercept	True, False	True
copy_X	True, False	True
positive	True, False	False
precompute	True, False	False
max_iter	100,200,500,1000,2000	1000

3.2.3. XGBoost Modelling

Also, to predict the back break, an XGBoost regressor approach, capable of handling nonlinearity and interactions between predictors through ensemble learning, was developed. The splitted and pre-processed data were utilized for this purpose. Then, BHO was performed to tune the

model for optimal performance. The hyperparameters specified in the search space are "n_estimators" (number of boosting stages), "max_depth" (maximum depth of each tree), "min_samples_split" (minimum number of samples required to split an internal node), and "learning_rate" (shrinkage factor for the contribution of each tree). The ranges for these hyperparameters are specified in Table 5. The hyperparameter optimization was performed, using the BayesSearchCV class with 5 cross-validation folds, which efficiently explore the hyperparameter search space and find the best combination of hyperparameters. Finally, the best model found by the hyperparameter search was evaluated. The search space and optimal values of the XGBoost modelling are illustrated in Table 5.

Table 5. Search space and optimal values of the XGBoost modelling

Parameter	Search Space	Optimal Value
learning_rate	0.1,0.2,0.5	0.1
n_estimators	100, 500, 1000	100
max_depth	1, 10,20	10
min_samples_split	Range 2 to 10	7

3.3. GSA algorithms implementation

In this study, the GSA optimization algorithm was used to optimize the parameters of the DNN, Lasso, and XGBoost models with the goal of minimizing back break. The algorithm was run iteratively, producing optimization statistics as the final output. After constructing an initial predictive model, the algorithms were employed as a metaheuristic approach to minimize back break. The BB prediction function was established as the objective function. It is generally recommended to perform multiple independent runs of each evaluated algorithm to ensure reliable statistical outcomes during the hyperparameter optimization process. This approach accounts for the intrinsic stochastic nature of metaheuristic techniques like GSA and helps reduce the influence of random initialization or other sources of variability. In this study, the number of repetitions (NumOfRuns) for each experiment, which focused on optimizing the hyperparameters of the models using the algorithms, was set to 3. The general parameters for all optimizers, including population size and number of iterations, were conFigured in this study. Specifically, a population size of 50 was designated, with the iteration count fixed at 100. After setting the hyperparameters of the algorithms, defining the objective function, specifying the model, and conFigureuring other experimental parameters, the optimization process was initiated by calling the "run" function.

4. Results and Discussion

The optimized models were assessed based on their performances, using the determination coefficient (R^2) and Mean Squared Error (MSE) metrics for both the training and testing datasets. R^2 is a measure of how well the model can explain the variance in the target variable. MSE quantifies the average squared deviation between the predicted and actual values. Lower MSE and higher R^2 values indicate better performance of the model (Equations 2 and 3) [52].

$$R^2 = \frac{(n\sum A_i P_i - \sum A_i \sum P_i)^2}{(n\sum A_i^2 - (\sum A_i)^2)(n\sum P_i^2 - (\sum AP_i)^2)} \quad (2)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(A_i - P_i)^2 \quad (3)$$

where the measured values are denoted as Ai, and the corresponding predicted values are denoted as Pi.

The correlation between the predicted values and real target data for train and test data for the XGBoost, Lasso, and DNN modelling are illustrated Figure 9.

The results of three different modelling: XGBoost, Lasso, and DNN are presented in Table 6 and Figure 10. For the XGBoost modelling, a good performance on the training dataset, with an R^2 value of 0.86 was achieved, indicating a high degree correlation. Also, a low MSE value of 0.36 for training data, suggesting that the model's predictions match the actual values. However, there is a notable drop in its performance when the XGBoost model was tested on test data, as R^2 decreases to 0.77 and MSE increases to 0.87. This indicates that the model may be overfitted on the training data and struggling to generalize well to new data.

In comparison, the Lasso model performs less impressively than XGBoost on both the training and test datasets. The R^2 values for Lasso are 0.51 and 0.62 for the training and test datasets,

respectively, indicating a moderate level of correlation. The MSE values for Lasso are also higher than those of XGBoost, with a training MSE of 1.27 and a test MSE of 1.41. This suggests that the Lasso model has a higher prediction error and may not capture the underlying patterns as effectively as XGBoost.

The DNN model demonstrates good performance on both the training and test datasets. It achieved R^2 values of 0.86 and 0.81 for the training and test datasets, respectively, indicating a relatively high degree of correlation. The MSE values for DNN are relatively low, at 0.38 and 0.70 for the training and test datasets, respectively. These results suggest the superiority of the DNN model over XGBoost and Lasso models for predicting blasting back break.

Table 6. The performance of the XGBoost, Lasso, and DNN models for predicting back break

Model	Evaluation Metrics	Train	Test
XGBoost	R^2	0.86	0.77
	MSE	0.36	0.87
Lasoo	R^2	0.51	0.62
	MSE	1.27	1.41
DNN	R^2	0.86	0.81
	MSE	0.38	0.70

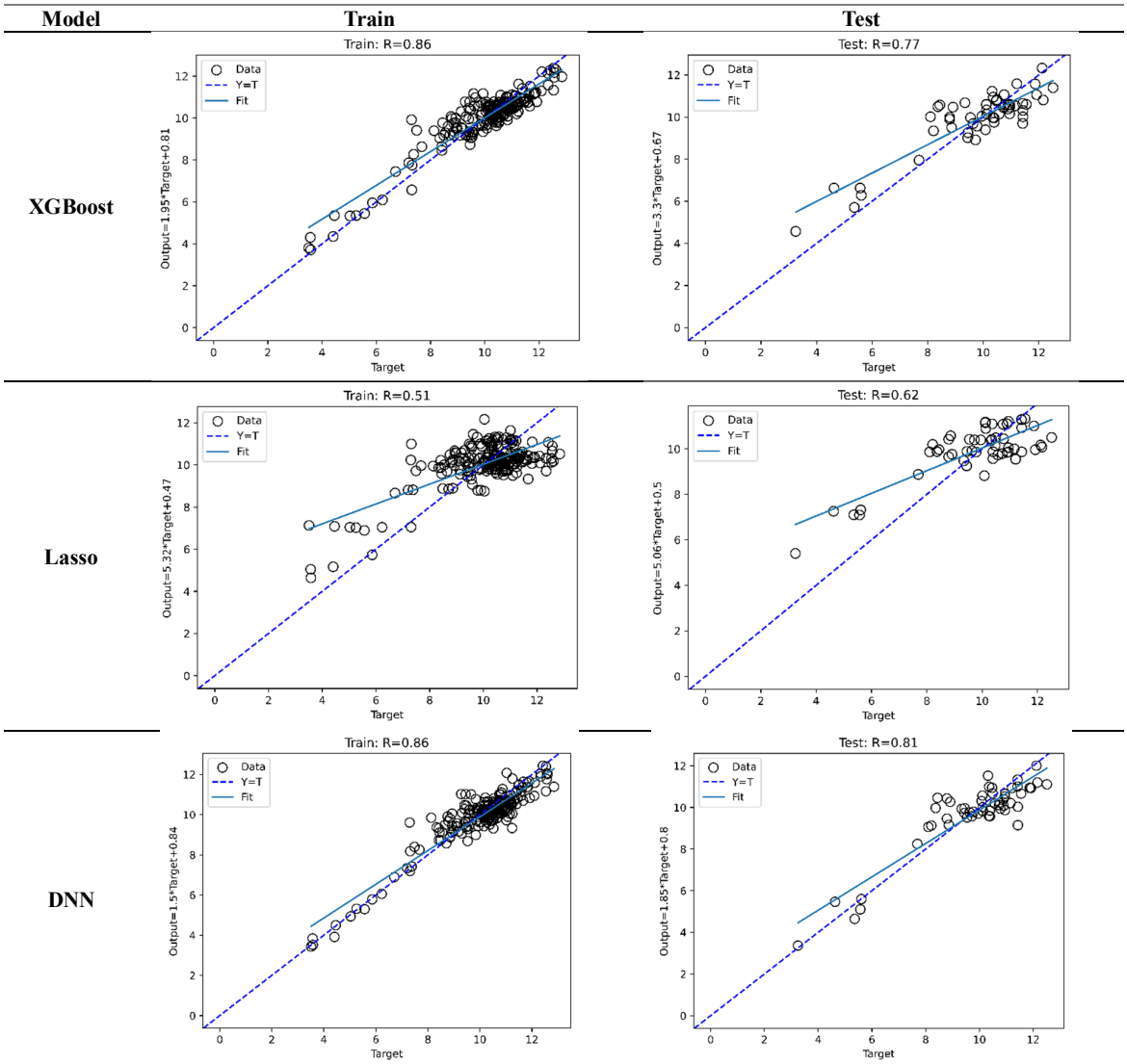


Figure 10. The performance of XGBoost, Lasso, and DNN models in predicting back break (solid blue line: 45-degree line; dashed line: the fitted linear regression model)

Additionally, a comparison between the predictive performances of the three models is shown in Figure 11. As evident from the results, the DNN model achieved the highest prediction accuracy for back break compared to the Lasso and XGBoost models.

To minimize back break, the GSA algorithm was employed in conjunction with the XGBoost, Lasso, and DNN models. The comparative performances of these algorithms and models are shown in Figure 12. Based on this Figure, it can be concluded that the DNN model demonstrated gradual convergence, reaching its minimum at iteration 34, while LASSO exhibited rapid convergence, achieving its minimum by iteration 2. XGBoost showed quick initial improvement, reaching its minimum at iteration 9. The DNN required significantly more execution time compared to LASSO and XGBoost. Although LASSO demonstrated the fastest convergence, it yielded the highest final back break value. XGBoost balanced quick convergence with competitive performance. DNN achieved the lowest back break prediction, suggesting a superior ability to capture the complex, non-linear relationships in the blasting data. The rapid convergence of LASSO and its higher back break prediction point indicate potential limitations in modeling complex relationships, likely due to its linear nature. All models maintained stable minimum values over numerous iterations, indicating GSA's efficacy in identifying stable local or global minima.

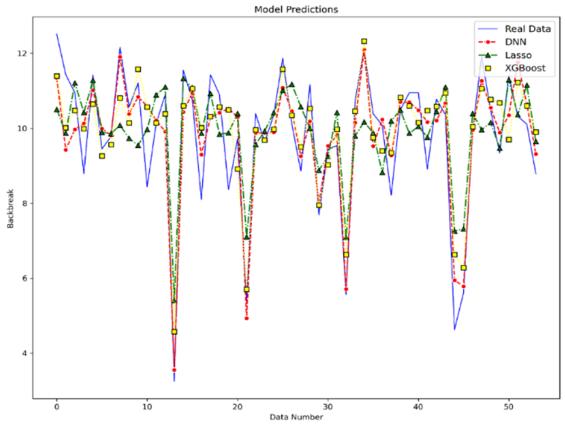


Figure 11. A comparison between the results obtained through the prediction models and the measured values of back break

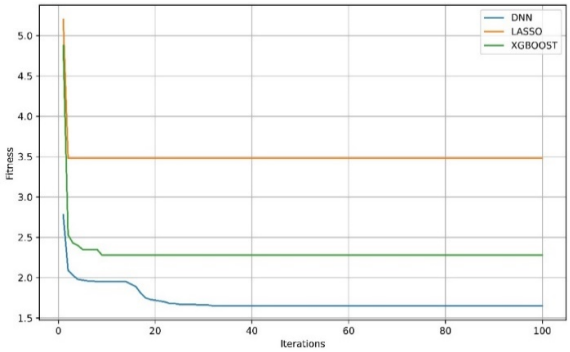


Figure 12. The performance of GSA, XGBoost, Lasso, and DNN models in minimization of the fitness function

The optimized parameter values resulting from the application of the optimization algorithms in these three models are detailed in Table 7. From these results, it can be observed that the DNN model achieved the lowest back break value of 1.64 m, demonstrating its effectiveness in minimizing back break in this context. In contrast, the LASSO model produced a higher back break value of 3.48 m, indicating that its performance in mitigating back break was less effective than that of the DNN model. The XGBoost model yielded a back break value of 2.28 m, positioning it between the outcomes of the DNN and LASSO models.

These results underscore the potential of machine learning models, including DNN, LASSO, and XGBoost, in optimizing blasting back break. Notably, the DNN model exhibited superior performance in minimizing back break relative to the other models evaluated in this study.

The consistency of certain parameters, such as PF and UCS, across the models reinforces their significance in influencing back break. However, the notable differences observed in other parameters suggest that each model interprets the relationship between inputs and back break in distinct ways. This divergence in optimized parameters across models highlights the complexity of the problem of blasting-induced back break.

The DNN model's capacity to capture non-linear relationships suggests that, within the context of the dataset analyzed in this study, an optimized strategy for minimizing back break involved employing more rows with higher charges in cases associated with lower GSI values. This outcome reflects model-driven optimization results specific to the studied data, as back break is known to depend on multiple interacting factors beyond

GSI alone. In comparison, the optimization outcomes from the LASSO and XGBoost models pointed toward a different strategy, favoring a single row with relatively lower charges under conditions associated with higher GSI values. These contrasting results highlight the influence of model architecture and parameter sensitivity on the optimization process, rather than implying a deterministic link between rock mass strength and back break behavior. It is important to note that the minimum recorded back break value from the collected dataset was 3.25 m. By utilizing the DNN model in conjunction with GSA optimization, a reduction of approximately 49.5% in the back break value was achieved compared to this minimum observed value. This finding underscores the potential of the proposed technique for enhancing back break control within the specific context of the study.

Table 7. Optimization results for the enhanced algorithm and XGBoost, Lasso, and DNN models performance

Model	Parameter									
	D (inch)	B (m)	S (m)	NR	PF (gr/ton)	Charge (Kg)	UCS (MPa)	Den (gr/cm3)	GSI	BB (m)
DNN	6.80	3.71	3.54	4	90.35	6005.47	0.5	3.33	18.72	1.64
LASSO	6	6	8	1	90.35	413.26	0.5	3.69	65	3.48
XGBoost	7.62	2.5	3.5	1	90.35	413.26	0.5	3.69	65	2.28

5. Sensitivity Analysis

To assess the relative influence of input factors on back break prediction, a sensitivity analysis was conducted using the cosine amplitude (CA) method proposed by Yang and Zhang (Equation 4) [32, 55, 56].

$$r_{i,j} = \frac{\sum_{k=1}^{l} g_{ik} g_{ik}}{\sqrt{\left(\sum_{k=1}^{l} g_{ik}^2\right) \cdot \left(\sum_{k=1}^{l} g_{jk}^2\right)}} \quad (4)$$

Where gik represents the inputs, gjk the corresponding output(s), and k denotes the number of output(s). A higher rij value indicates greater significance or importance of the input. Figure 11 visually illustrates the impact of each parameter on back break. According to the findings, the NR variable has the greatest sensitivity, with a value of 4.03, followed by Den with a sensitivity of 3.35 and GSI with a sensitivity of 2.94. The variables B, S, D, PF, and Charge exhibited moderate sensitivity, with values of 2.85, 2.80, 2.56, 2.30, and 1.45, respectively. Conversely, the UCS variable demonstrated relatively lower sensitivity, with a value of 1.45. These findings suggest that NR, Den, GSI, B, S, D, PF, Charge, and UCS contribute to varying degrees in influencing back break. Notably, these sensitivity analysis results align with previous studies conducted within this mine, reinforcing the consistency of the findings within the broader context of research in this mine [18].

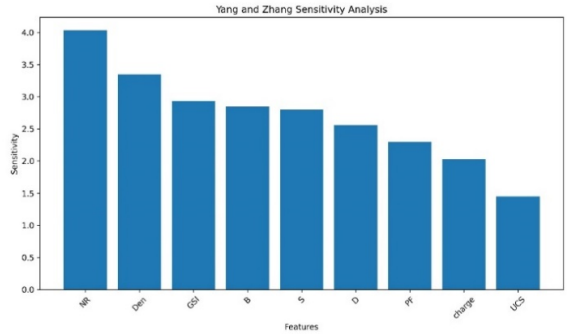


Figure 13. Assessing the impact of variables on back break intensity

6. Conclusions

In this study, three machine learning models—namely DNN, Lasso, and XGBoost—employing GSA optimization algorithms, were used to predict back break resulting from blasting. The findings revealed that the DNN model exhibited lower levels of overfitting, as indicated by an R^2 value of 0.81 and an MSE of 0.70 for the test data. These results demonstrate the superior performance of the DNN model compared to Lasso and XGBoost, suggesting its suitability for back break prediction. The results indicated that the use of the GSA algorithm alongside XGBoost, Lasso, and DNN models was effective in minimizing back break. The DNN model showed gradual convergence, reaching its minimum at iteration 34, while Lasso and XGBoost exhibited rapid convergence with varying final back break values. Despite requiring more computation time, the DNN model demonstrated superior ability to capture complex, non-linear relationships within the blasting data. The optimized parameter values confirmed the

DNN model's effectiveness in minimizing back break compared to Lasso and XGBoost, highlighting the potential of machine learning models for optimizing blasting outcomes. The sensitivity analysis emphasized the varying influence of parameters on back break, with NR, Den, and GSI showing the highest sensitivity, underscoring the complexity of blasting-induced back break and the need for tailored control strategies.

Beyond predictive accuracy, this study innovatively integrates machine learning models with the Gravitational Search Algorithm, enabling not only accurate predictions of back break but also the optimization of blasting parameters to reduce its occurrence. The DNN–GSA framework achieved a 49.5% reduction in back break compared to field-recorded minimums, demonstrating its practical effectiveness. This approach can be directly implemented onsite by blasting engineers to adjust burden, spacing, powder factor, and charge distribution, thereby enhancing mine safety, slope stability, and operational efficiency. Overall, the research demonstrates that combining advanced machine learning techniques with metaheuristic optimization provides a powerful decision-support tool for sustainable and cost-effective blast design, with potential for future integration of additional environmental and economic factors.

References

[1] Kumar, S., Mishra, A., Choudhary, B. (2021). Prediction of back break in blasting using random decision trees. *Engineering with Computers*. 1-7.

[2] Sayadi. A., Monjezi, M., Talebi, N., Khandelwal, M. (2013). A comparative study on the application of various artificial neural networks to simultaneous prediction of rock fragmentation and backbreak. *Journal of Rock Mechanics and Geotechnical Engineering*. 5(4), 318-24.

[3] Saghatforoush. A., Monjezi, M., Shirani Faradonbeh, R., Jahed Armaghani, D. (2016). Combination of neural network and ant colony optimization algorithms for prediction and optimization of flyrock and back-break induced by blasting. *Engineering with Computers*. 32, 255-66.

[4] Nabavi, Z., Mirzehi, M., Dehghani, H., Ashtari, P. (2023). A hybrid model for back-break prediction using XGBoost machine learning and metaheuristic algorithms in Chadormalu iron mine. *Journal of Mining and Environment*. 14(2), 689-712.

[5] Monjezi, M., Rezaei, M., Yazdian, A. (2010). Prediction of backbreak in open-pit blasting using fuzzy set theory. *Expert Systems with Applications*. 37(3), 2637-43.

[6] Monjezi, M., Ahmadi, Z., Varjani, A.Y., Khandelwal, M. (2013). Backbreak prediction in the Chadormalu iron mine using artificial neural network. *Neural Computing and Applications*. 23, 1101-7.

[7] Faramarzi, F., Ebrahimi Farsangi, M., Mansouri, H. (2013). An RES-based model for risk assessment and prediction of backbreak in bench blasting. *Rock Mechanics and Rock Engineering*. 46, 877-87.

[8] Monjezi, M., Amini Khoshalan, H., Yazdian Varjani, A. (2012). Prediction of flyrock and backbreak in open pit blasting operation: a neuro-genetic approach. *Arabian Journal of Geosciences*. 5(3).

[9] Esmaeili, M., Osanloo, M., Rashidinejad, F., Aghajani Bazzazi, A., Taji, M. (2014). Multiple regression, ANN and ANFIS models for prediction of backbreak in the open pit blasting. *Engineering with computers*. 30, 549-58.

[10] Sari, M., Ghasemi, E., Ataei, M. (2014). Stochastic modeling approach for the evaluation of backbreak due to blasting operations in open pit mines. *Rock Mechanics and Rock Engineering*. 47, 771-83.

[11] Ebrahimi, E., Monjezi, M., Khalesi, M.R., Armaghani, D.J. (2016). Prediction and optimization of back-break and rock fragmentation using an artificial neural network and a bee colony algorithm. Bulletin of Engineering Geology and the Environment. 7, 27-36.

[12] Hasanipanah, M., Shahnazar, A., Arab, H., Golzar, S.B., Amiri, M. (2017). Developing a new hybrid-AI model to predict blast-induced backbreak. *Engineering with Computers*. 33, 349-59.

[13] Katsabanis, P.D. (2020). Analysis of the effects of blasting on comminution using experimental results and numerical modelling. *Rock Mechanics and Rock Engineering*. 53, 3093-109.

[14] Yu, Q., Monjezi, M., Mohammed, A.S., Dehghani, H., Armaghani, D.J., Ulrikh, D.V. (2021). Optimized support vector machines combined with evolutionary random forest for prediction of back-break caused by blasting operation. *Sustainability*. 13(22), 12797.

[15] Sirjani, A.K., Sereshki, F., Ataei, M., Hosseini, M.A. (2022). Prediction of backbreak in the blasting operations using artificial neural network (ANN) model and statistical models (case study: Gol-e-Gohar iron ore mine No. 1). *Archives of Mining Sciences*. 107, 21.

[16] Kumar, S., Mishra, A.K., Choudhary, B.S. (2023). Estimation Equations for Back Break and Ground Vibration Using Genetic Programming. *Geotechnical and Geological Engineering*. 41(5), 3139-49.

[17] Sorabi, P., Ataei, M., Jazi, M.R.A., Dehghani, H., Shakeri, J., Habibi, M.H. (2024). Utilizing heuristic strategies for predicting the backbreak occurrences in

open-pit mines, Gol Gohar Mine, Iran. *Application of Soft Computing*. 10.1007/s00500-023-09613-8.

[18] Monjezi, M., Dehghani, H. (2008). Evaluation of effect of blasting pattern parameters on back break using neural networks. *International Journal of Rock Mechanics and Mining Sciences*. 45(8), 1446-53.

[19] Kamran, M. (2021). A state of the art catboost-based T-distributed stochastic neighbor embedding technique to predict back-break at dewan cement limestone quarry. *Journal of Mining and Environment*. 12(3), 679-91.

[20] Enayatollahi, I., Aghajani-Bazzazi, A. (2009). Evaluation of salt-ANFO mixture in back break reduction by data envelopment analysis. Proceedings of the 9th international symposium on rock fragmentation by blasting, Granada, Spain. 127-33.

[21] Shirani Faradonbeh, R., Monjezi, M., Jahed Armaghani, D. (2016). Genetic programing and non-linear multiple regression techniques to predict backbreak in blasting operation. *Engineering with Computers*. 32, 123-33.

[22] Khandelwal, M., Monjezi, M. (2013). Prediction of backbreak in open-pit blasting operations using the machine learning method. *Rock Mechanics and Rock Engineering*. 46, 389-96.

[23] Sharma, M., Choudhary, B.S., Agrawal, H. (2021). Prediction and assessment of back break by multivariate regression analysis, and random forest algorithm in hot strata/fiery seam of open-pit coal mine.

[24] Dai, Y., Khandelwal, M., Qiu, Y., Zhou, J., Monjezi, M., Yang, P. (2022). A hybrid metaheuristic approach using random forest and particle swarm optimization to study and evaluate backbreak in open-pit blasting. *Neural Computing and Applications*.1-16.

[25] Soltanalinejad, S., Moomivand, H. (2024). Development of a novel empirical approach to control overbreak, surface quality, and slope angle of benches following blasting. *Canadian Geotechnical Journal*. 62, 1-22.

[26] Kunkyin-Saadaari, F., Agadzie, V.K., Gyebuni, R. (2024). A comparative study on the application of intelligent models in the estimation of backbreak in mine blasting operations. *American Journal of Science, Engineering and Technology*. 9, 1e13.

[27] Jethro, M., Shehu, S., Agbalajobi, S., Thomas, A. (2024). Prediction of backbreak using artificial neural network at cgc abobo quarry during blasting operation. *Confluence University Journal of Science and Technology.* 1, 34.

[28] Kannavena, S.K., Pradeep, T., Chandrahas, N.S., Prasad, D. (2025). Prediction of back break using sensitivity analysis and artificial neural networks. *Journal of the Institution of Engineers (India)*.106, 383-98.

[29] Shahani, N.M., Zheng, X. (2025). Predicting backbreak due to blasting using LSSVM optimized by metaheuristic algorithms. *Environmental Earth Sciences*. 84, 156.

[30] Guo, H., Zhou, J., Koopialipoor, M., Jahed Armaghani, D., Tahir, M. (2021). Deep neural network and whale optimization algorithm to assess flyrock induced by blasting. *Engineering with Computers*. 37, 173-86.

[31] Bai, R., Zhang, P., Zhang, Z., Sun, X., Fei, H., Bao, S., et al. (2023). Optimization of blasting parameters and prediction of vibration effects in open pit mines based on deep neural networks. *Alexandria Engineering Journal*. 70, 261-71.

[32] Ghojoghi, E., Farsangi, M.A.E., Mansouri, H., Rashedi, E. (2024). Prediction and Minimization of Blasting Flyrock Distance, Using Deep Neural Networks and Gravitational Search Algorithm, JAYA, and Multi-Verse Optimization algorithms. *Heliyon*. 10(19), e37876.

[33] Xu, S., Chen, T., Liu, J., Zhang, C., Chen, Z. (2021). Blasting vibration control using an improved artificial neural network in the Ashele copper mine. *Shock and Vibration*. 2021, 1-11.

[34] Ly, H.B., Nguyen, T.A., Tran, V.Q. (2021). Development of deep neural network model to predict the compressive strength of rubber concrete. *Construction and Building Materials*. 301, 124081.

[35] Nam, B.X, Hoang, N., Hieu, T.Q., An, N.D., Thao, L.Q. (2019). Utilizing the Lasso and Elastic-Net regularized generalized linear model for predicting blast-induced ground vibration in open-pit mines. *KHOA học trái đất và tài nguyên*.27

[36] Bui, X.N., Nguyen, H., Tran, Q.H., Bui, H.B., Nguyen, Q.L., Nguyen, D.A., et al. (2019). A lasso and elastic-net regularized generalized linear model for predicting blast-induced air over-pressure in open-pit mines. *Inżynieria Mineralna*. 21.

[37] Zhou, J., Qiu, Y., Khandelwal, M., Zhu, S., Zhang, X. (2021). Developing a hybrid model of Jaya algorithm-based extreme gradient boosting machine to estimate blast-induced ground vibrations. *International Journal of Rock Mechanics and Mining Sciences*. 145,104856.

[38] Qiu, Y., Zhou, J., Khandelwal, M., Yang, H., Yang, P., Li, C. (2021). Performance evaluation of hybrid WOA-XGBoost, GWO-XGBoost and BO-XGBoost models to predict blast-induced ground vibration. *Engineering with Computers*. 1-18.

[39] Wu, J., Chen, X.Y., Zhang, H., Xiong, L.D., Lei, H., Deng, S.H.(2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology.* 17(1), 26-40.

[40] Al-Taai, S.R., Azize, N.M., Thoeny, Z.A., Imran, H., Bernardo, L.F., Al-Khafaji, Z. (2023). XGBoost Prediction Model Optimized with Bayesian for the Compressive Strength of Eco-Friendly Concrete Containing Ground Granulated Blast Furnace Slag and Recycled Coarse Aggregate. *Applied Sciences*. 13(15), 8889.

[41] Valladares, H., Tovar, A. (2020). Design optimization of sandwich composite armors for blast mitigation using bayesian optimization with single and multi-fidelity data. *SAE Technical Paper.*

[42] Rajaraman, S., Siegelman, J., Alderson, P.O., Folio, L.S., Folio, L.R., Antani, S.K. (2020). Iteratively pruned deep learning ensembles for COVID-19 detection in chest X-rays. *Ieee Access*. 8, 115041-50.

[43] Yang, L., Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*. 415, 295-316.

[44] Kunang, Y.N., Nurmaini, S., Stiawan, D., Suprapto, B.Y. (2021). Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. *Journal of Information Security and Applications.* 58, 102804.

[45] Sabri, N.M,, Puteh, M., Mahmood, M.R. (2013).An overview of Gravitational Search Algorithm utilization in optimization problems. *IEEE 3rd International Conference on System Engineering and Technology*. 61-6.

[46] Banyhussan, Q.S., Hanoon, A.N., Al-Dahawi, A., Yıldırım, G., Abdulhameed, AA. (2020). Development of gravitational search algorithm model for predicting packing density of cementitious pastes. *Journal of Building Engineering*. 27, 100946.

[47] Ma, C., Jiang, Y., Li, T. (2019). Gravitational search algorithm for microseismic source location in tunneling: performance analysis and engineering case study. *Rock Mechanics and Rock Engineering.* 52, 3999-4016.

[48] Momeni, E., Yarivand, A., Dowlatshahi, M.B., Armaghani, D.J. (2021). An efficient optimal neural network based on gravitational search algorithm in predicting the deformation of geogrid-reinforced soil structures. *Transportation geotechnics*. 26, 100446.

[49] Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information Sciences*. 179(13), 2232-48.

[50] Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S. (2010). BGSA: binary gravitational search algorithm. *Natural Computing*. 9, 727-45.

[51] Rashedi, E., Rashedi, E., Nezamabadi-Pour, H. (2018). A comprehensive survey on gravitational search algorithm. *Swarm and Evolutionary Computation*. 41, 141-58.

[52] Fattahi, H., Hasanipanah, M. (2022). An integrated approach of ANFIS-grasshopper optimization algorithm to approximate flyrock distance in mine blasting. *Engineering with Computers*. 38(3), 2619-31.

[53] Hasan, B.M.S., Abdulazeez, A.M. (2021). A review of principal component analysis algorithm for dimensionality reduction. *Journal of Soft Computing and Data Mining*. 2(1), 20-30.

[54] Karamizadeh, S., Abdullah, S.M., Manaf, A.A., Zamani, M., Hooman, A. (2013). An overview of principal component analysis. *Journal of Signal and Information Processing*. 4(3),173.

[55] Yang, Y., Zhang, Q. (1997). A hierarchical analysis for rock engineering using artificial neural networks. *Rock Mechanics and Rock Engineering*. 30, 207-22.

[56] Wang, X., Hosseini, S., Jahed Armaghani, D., Tonnizam Mohamad, E. (2023). Data-driven optimized artificial neural network technique for prediction of flyrock induced by boulder blasting. *Mathematics*. 11(10), 2358.

انجمن مهندسی معدن ایران
نشریه مهندسی معدن و محیط زیست
نشانی نشریه: www.jme.shahroodut.ac.ir
دانشگاه صنعتی شاهرود

پیش‌بینی و کمینه‌سازی عقب زدگی ناشی از انفجار با استفاده از شبکه‌های عصبی عمیق، تقویت گرادیان شدید، رگرسیون لاسو و الگوریتم جستجوی گرانشی

اسلام قجقی[۱]، حمید منصوری[۲*]، محمد علی ابراهیمی فرسنگی[۲] و عصمت راشدی[۳]

۱. دکتری مهندسی معدن، دانشکده مهندسی معدن، دانشگاه شهید باهنر کرمان، ایران
۲. دانشیار مهندسی معدن، دانشکده مهندسی معدن، دانشگاه شهید باهنر کرمان، ایران
۳. دانشیار مهندسی مخابرات سیستم، دانشکده مهندسی برق و کامپیوتر، دانشگاه تحصیلات تکمیلی صنعتی و فناوری پیشرفته، کرمان، ایران

اطلاعات مقاله

تاریخ ارسال: ۲۰۲۵/۰۹/۱۶
تاریخ داوری: ۲۰۲۵/۱۰/۱۱
تاریخ پذیرش: ۲۰۲۶/۰۲/۰۷

DOI: 10.22044/jme.2026.16834.3303

کلمات کلیدی

انفجار
عقب‌زدگی
شبکه عصبی عمیق
تقویت گرادیان شدید
رگرسیون لاسو
الگوریتم جستجوی گرانشی
معدن سنگ‌آهن گل‌گهر ۴

چکیده

عقب‌زدگی یکی از پیامدهای نامطلوب انفجار سنگ است که موجب اتلاف انرژی ماده منفجره و کاهش بهره‌وری عملیاتی می‌شود. ازاین‌رو، پیش‌بینی این پدیده به‌منظور امکان کنترل آن و جلوگیری از افزایش هزینه‌های عملیاتی، امری ضروری است. هدف از این پژوهش علمی، به‌کارگیری شبکه‌های عصبی عمیق، تقویت گرادیان شدید و رگرسیون لاسو در کنار الگوریتم جستجوی گرانشی برای پیش‌بینی و کمینه‌سازی عقب‌زدگی ناشی از انفجار در معدن سنگ‌آهن گل‌گهر ۴، واقع در سیرجان، استان کرمان، ایران است. مدل‌های توسعه‌یافته شامل مجموعه‌ای از نه پارامتر ورودی هستند که پارامترهای طراحی انفجار و ویژگی‌های ژئومکانیکی سنگ را در بر می‌گیرند و عقب‌زدگی را به‌عنوان تنها خروجی تولید می‌کنند. داده‌های مورد استفاده برای آموزش و ارزیابی مدل‌ها شامل ۲۶۶ رکورد انفجار استخراج‌شده از معدن سنگ‌آهن گل‌گهر ۴ می‌باشد. نتایج نشان می‌دهد که مدل شبکه عصبی عمیق با R^2 برابر ۰.۸۱ و MSE برابر ۰.۷۰، عملکرد بهتری نسبت به مدل‌های تقویت گرادیان شدید و رگرسیون لاسو در پیش‌بینی عقب‌زدگی دارد. افزون بر این، به‌کارگیری الگوریتم بهینه‌سازی منجر به تعیین مقادیر بهینه‌ی پارامترها به‌منظور کمینه‌سازی عقب‌زدگی شده است.

نویسنده مسئول مکاتبات: hmansouri@uk.ac.ir (ح. منصوری)