



Original Paper

Measuring and Predicting Blast-Induced Flyrock Using Unmanned Aerial Vehicles and Lévy Flight Technique-Based Jaya Optimization Algorithm Integrated with Adaptive Neuro-Fuzzy Inference System

Hoang Nguyen^{1,2,5} Tran Dinh Bao,^{1,2} Xuan-Nam Bui,^{2,3} Van-Viet Pham,^{1,2}
Dinh-An Nguyen,^{1,2} Ngoc-Hoan Do,^{1,2} Le Thi Thu Hoa,^{1,2} Qui-Thao Le,^{1,2}
and Tuan-Ngoc Le⁴

Received 3 November 2024; accepted 31 December 2024
Published online: 9 February 2025

Predicting flyrock is key to safety and efficiency in open pit mining. In this study, we developed and tested four hybrid models utilizing an adaptive neuro-fuzzy inference system (ANFIS) integrated with metaheuristic optimization techniques: Lévy-enhanced Jaya (ANFIS-LJ), bat algorithm (ANFIS-BA), firefly algorithm (ANFIS-FA) and social spider optimization (ANFIS-SSO). Remarkably, the Lévy technique was applied to enhance the JA algorithm and improve the performance of the ANFIS model for predicting flyrock distance. The models were trained and tested using a dataset from Ta Phoi copper mine with 204 blast events and flyrock distance as the target variable. A drone was used to measure flyrock distance in this study with high resolution to capture the entire flyrock phenomenon of each blast. The k-fold cross-validation technique (with 5 folds) was applied to ensure that AI-based models are not only accurate but also generalize well to new data. It helps in evaluating model performance, tuning hyperparameters, reducing overfitting, and providing a more reliable estimate of how the model will perform in predicting blast-induced flyrock. The models were evaluated using MAE (mean absolute error), RMSE (root mean-squared error) and R^2 . The result showed that ANFIS-LJ outperformed the other models with MAE of 1.423, RMSE of 1.895 and R^2 of 0.981 on the testing dataset. It was also validated through 13 blasts in practice and achieved a high R^2 of 0.988, indicating excellent agreement between predicted and observed flyrock distances. Besides, the low MAE (1.322) and RMSE (1.825) values confirmed the model's precision and reliability in predicting flyrock distances. These results confirmed its potential as a valuable tool for optimizing blast designs, enhancing safety, and reducing environmental impacts in real-world engineering

¹Department of Surface Mining, Mining Faculty, Hanoi University of Mining and Geology, 18 Vien Str., Duc Thang Ward, Bac Tu Liem Distr., Hanoi 100000, Vietnam.

²Innovations for Sustainable and Responsible Mining (ISRM) Research Group, Hanoi University of Mining and Geology, 18 Vien Str., Duc Thang Ward, Bac Tu Liem Distr., Hanoi 100000, Vietnam.

³Vietnam Mining Science and Technology Association, 226 Le Duan Str., Trung Phung Ward, Dong Da Distr., Hanoi 100000, Vietnam.

⁴Vinacomin – Minerals Holding Corporation, 193 Nguyen Huy Tuong Str., Thanh Xuan Trung Ward, Thanh Xuan Dist., Hanoi 100000, Vietnam.

⁵To whom correspondence should be addressed; e-mail: nguyenhoang@humg.edu.vn

applications. This study showed that combining ANFIS with metaheuristic algorithms, especially Lévy-enhanced Jaya algorithm, can produce accurate flyrock prediction. The result can be used to improve the predictive model in open pit mining and decision making. Future study can focus on refining the models and applying them in different mining environments to improve the accuracy.

KEY WORDS: Flyrock prediction, ANFIS, Metaheuristic algorithms, Lévy-enhanced Jaya, Open-pit mining, Blast safety modeling.

INTRODUCTION

Flyrock is a major hazard in mining and construction blasting; people, equipment and structures nearby are at risk. Flyrock is the projection of rock beyond the blast area driven by the explosive energy used to break the rock (Fig. 1). Blasting operations near residential areas can raise concerns about safety and property damage. By studying and controlling flyrock, mining operations can alleviate community concerns and maintain positive relationships. Uncontrolled flyrock can result in costly property damage, leading to compensation claims and project delays. Accurate prediction reduces such risks. In addition, uncontrolled flyrock can damage critical equipment, leading to downtime and increased maintenance costs. Managing flyrock helps maintain operational efficiency. A better understanding of flyrock behavior allows engineers to design more efficient blast patterns, reducing excessive use of explosives while achieving desired fragmentation.

Many countries impose strict environmental regulations on blasting activities. Effective flyrock management ensures compliance, reducing the risk of legal liabilities and fines. Therefore, predicting and controlling flyrock is key to safety and minimizing accidents in blasting. Traditional empirical methods for predicting flyrock have shown to be limited due to the complexity and multivariable nature of the phenomenon, influenced by many controllable factors (i.e., blast design) and uncontrollable factors (i.e., rock properties, geology) (Ghasemi et al., 2012; Armaghani et al., 2016b; Faradonbeh et al., 2018; Bhatawdekar et al., 2021).

Blast-induced flyrock equations for open-pit mines have been developed using site-specific data and general observations of blasting (Raina et al., 2015; Armaghani et al., 2016c). These equations use variables such as blast hole diameter, charge weight, and distance from the blast to establish a direct

relationship between these parameters and flyrock distance. While empirical models are simple and easy to use, they fail to capture the complex and nonlinear interactions between blasting variables and environmental factors (Faradonbeh et al., 2016). This results in low accuracy and limited applicability across different sites. Moreover, empirical models cannot adapt to new data and cannot account for changes in geological conditions (Zhou et al., 2020).

To overcome the above-mentioned limitations, artificial intelligence (AI) models like artificial neural networks (ANN), random forest (RF), and recent advancements in AI like hybrid models combining metaheuristic algorithms with machine learning techniques have been proposed and they have opened up new avenues for improving flyrock prediction (Monjezi et al., 2013; Trivedi et al., 2016; Asl et al., 2017; Bakhtavar et al., 2017; Li et al., 2023). These AI models can handle large datasets and multiple input parameters, learn complex relationships and improve prediction accuracy, so they are more flexible and reliable for a broader range of blasting conditions.

Many studies have contributed to predicting flyrock, each one helping to develop more accurate AI models. Manoj & Monjezi (2013) used support vector machine (SVM) for flyrock prediction, and it outperformed traditional methods like multivariate regression analysis (MVRA) because it can model nonlinear relationships. This is the foundation of AI in flyrock prediction. Building on this, Armaghani et al. (2016c) compared ANN with adaptive neuro-fuzzy inference systems (ANFIS). They found ANFIS to be more effective because it combines fuzzy logic and neural networks and had a higher R^2 value (0.964 vs. 0.925 for ANN) and is more suitable for handling the uncertainties in flyrock prediction. Following this, Hasanipanah et al. (2017) used particle swarm optimization (PSO) for flyrock prediction and compared it with multiple linear regression (MLR). PSO gave better prediction accuracy than



Figure 1. Blast-induced flyrock in mining and associated risks (Law, 2024; Plus, 2024): (a) flyrocks induced by mine blasting; (b)–(e) damages caused by flyrock in Vietnam.

MLR by reducing RMSE (root mean squared error) and increasing R^2 . This study proved that meta-heuristic algorithms can optimize the predictive models.

In 2020, several key studies pushed the boundaries further. Lu et al. (2020) introduced extreme learning machine (ELM) and outlier robust ELM (ORELM), the latter being a first in flyrock prediction. Both models demonstrated superior performance over ANN and multiple regression

models, with lower RMSE and higher R^2 , highlighting the advantages of integrating robust algorithms to handle outliers in complex datasets. Meanwhile, Rad et al. (2020) proposed a hybrid genetic algorithm (GA) combined with recurrent fuzzy neural network (RFNN), which outperformed standalone ANN and nonlinear regression models. This hybrid approach allowed for more precise flyrock predictions, particularly by better accounting for critical parameters like maximum charge per

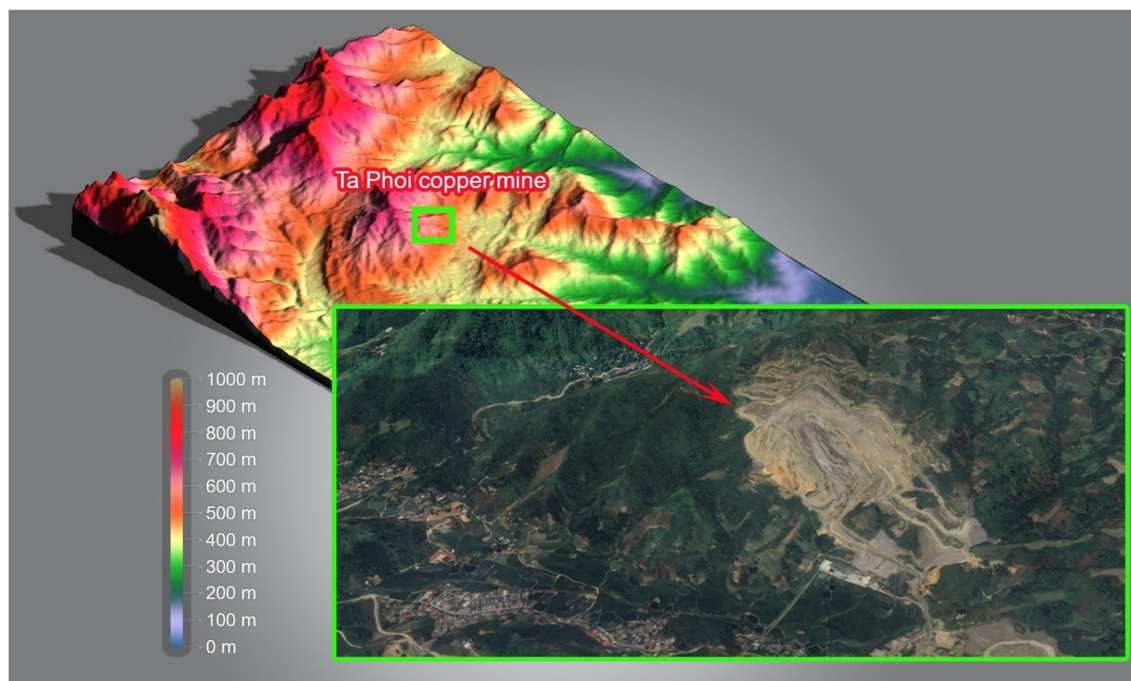


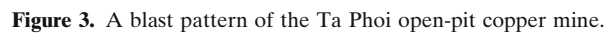
Figure 2. Location of the Ta Phoi copper mine (Vietnam).

delay, as revealed in the sensitivity analysis. Additionally, Han et al. (2020) applied Bayesian networks (BN) and RF for probabilistic prediction of flyrock. By combining these models, they successfully modeled the uncertainty inherent in flyrock prediction, with high accuracy in determining the probabilities of flyrock occurrences.

Barkhordari et al. (2022) further advanced the field by applying ensemble learning techniques, including Bayesian–XGBoost and stacking models, for flyrock prediction. Their separate stacking model with a bagging meta-learner achieved a high R^2 of 0.974 and low RMSE, outperforming traditional machine learning models and illustrating the strength of ensemble approaches for complex predictions like flyrock. Shakeri et al. (2023) explored several metaheuristic algorithms, such as gray wolf optimization (GWO) and multi-verse optimizer (MVO), for flyrock prediction. Their results indicated that MVO provided the best predictive performance, with the highest R^2 and lowest error rates, further supporting the role of metaheuristic techniques in enhancing model accuracy. Finally, Taiwo et al. (2024) used advanced deep learning

models, including long short-term memory (LSTM) networks, to predict flyrock. LSTM achieved the highest predictive accuracy, with an R^2 of 0.99, outperforming SVM models. This study highlighted the effectiveness of deep learning in modeling complex time-sequenced data for flyrock prediction.

Similar studies and models can be referred to the references (Ghasemi et al., 2014; Armaghani et al., 2016a; Jamei et al., 2021; Babaeian et al., 2023; Wang et al., 2023b; Ghoghghi et al., 2024). These studies and models demonstrated the evolution of flyrock prediction models from simple empirical methods to sophisticated AI and hybrid models. Each study has contributed to improving accuracy and reliability, with metaheuristic algorithms and deep learning proving to be the most promising for future advancements. Despite advancements in AI-based flyrock prediction models, many existing approaches still face challenges in processing complex datasets and adapting to real-time environmental variability. Previous models, while improving accuracy, often encounter difficulties with local optima and depend on limited data sources.



search process, and deliver a more flexible and robust solution for flyrock forecasting. The need for this study arose from the current models' limitations in handling large datasets and real-time conditions. By integrating UAVs for richer data collection and the Lévy flight-based Jaya algorithm for enhanced optimization, the proposed method advances the accuracy and reliability of flyrock prediction models. Moreover, the contribution of this work lies in its novel combination of technologies, which not only enhances safety and accuracy in predicting flyrock but also sets the foundation for future research into integrating advanced AI with real-time monitoring in mining operations.

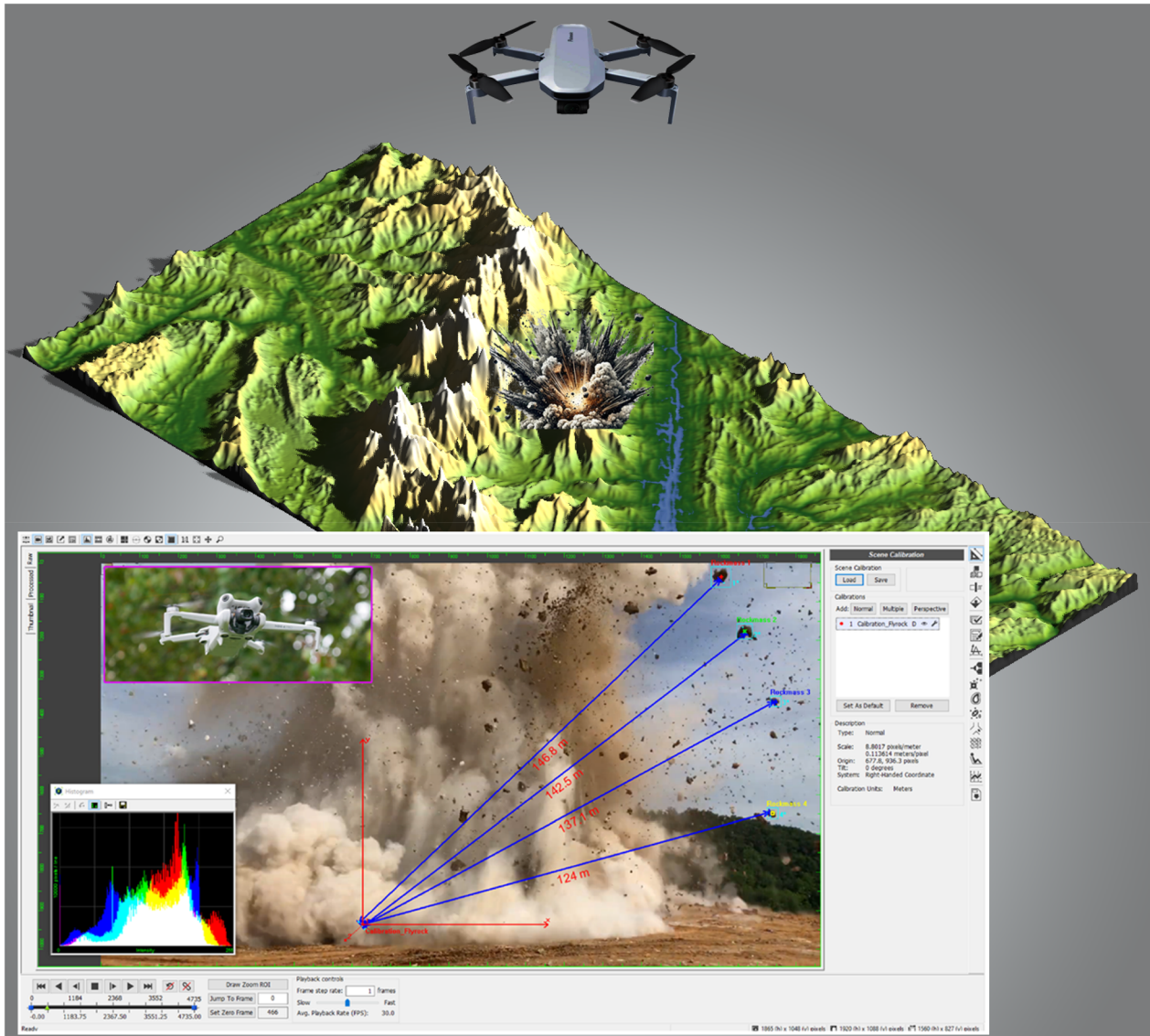


Figure 4. Utilization of UAV to measure flyrock distance.

SITE STUDY, DATA COLLECTION AND DATA ANALYSIS

Study Site

To conduct this study and examine the performance of the proposed models, the Ta Phoi copper mine (Lao Cai, Vietnam) was selected. This mine is located on the right bank of the Red River (side of Vietnam—Fig. 2), with elevations ranging from 210 to 720 m. The northern part of the mine stretches in a NE–SW direction, while the southern terrain

gradually shifts towards a N–S orientation. The central area of the mine rises above 700 m, with elevations gradually tapering off to the N and S. The terrain is heavily dissected by river systems and numerous small ravines, predominantly oriented in an E–W or sub-latitudinal direction. Geologically, the mine lies in the northeastern wing of the PoSen anticline. It is composed of metamorphosed sedimentary formations from the Sin Quyen, Da Dinh, and Cam Duong formations, intersected by a large intrusive magma body from the PoSen complex (Vinacomin, 2014a).

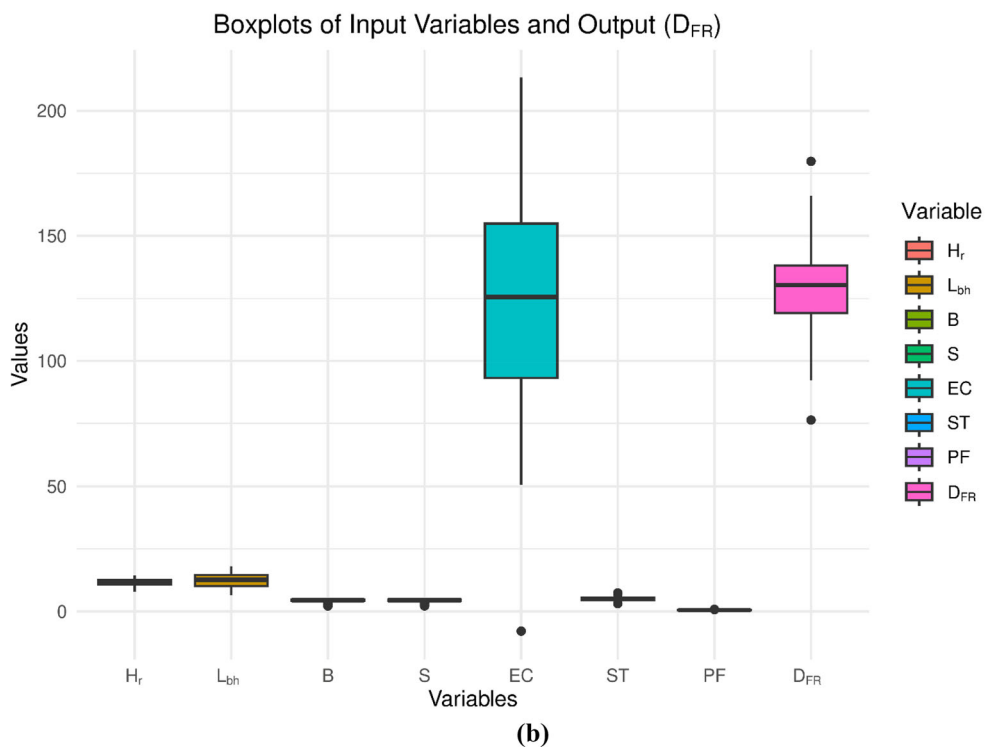
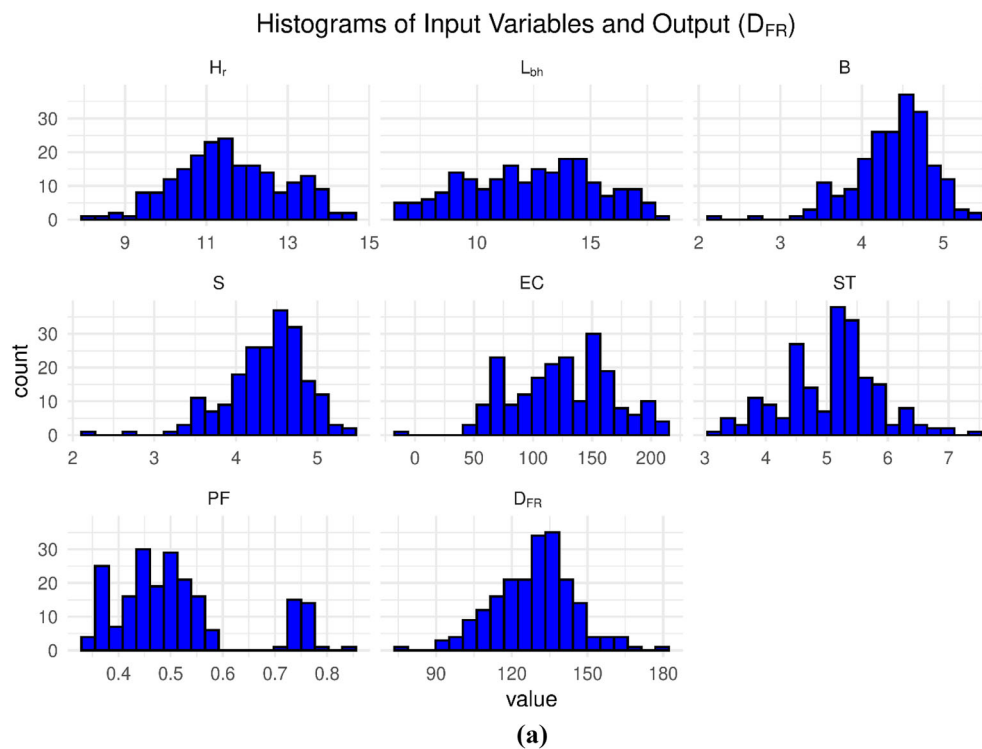


Figure 5. (a) Histograms to visualize the distribution of individual parameters. (b) Box plots to detect outliers and understand the spread of data.

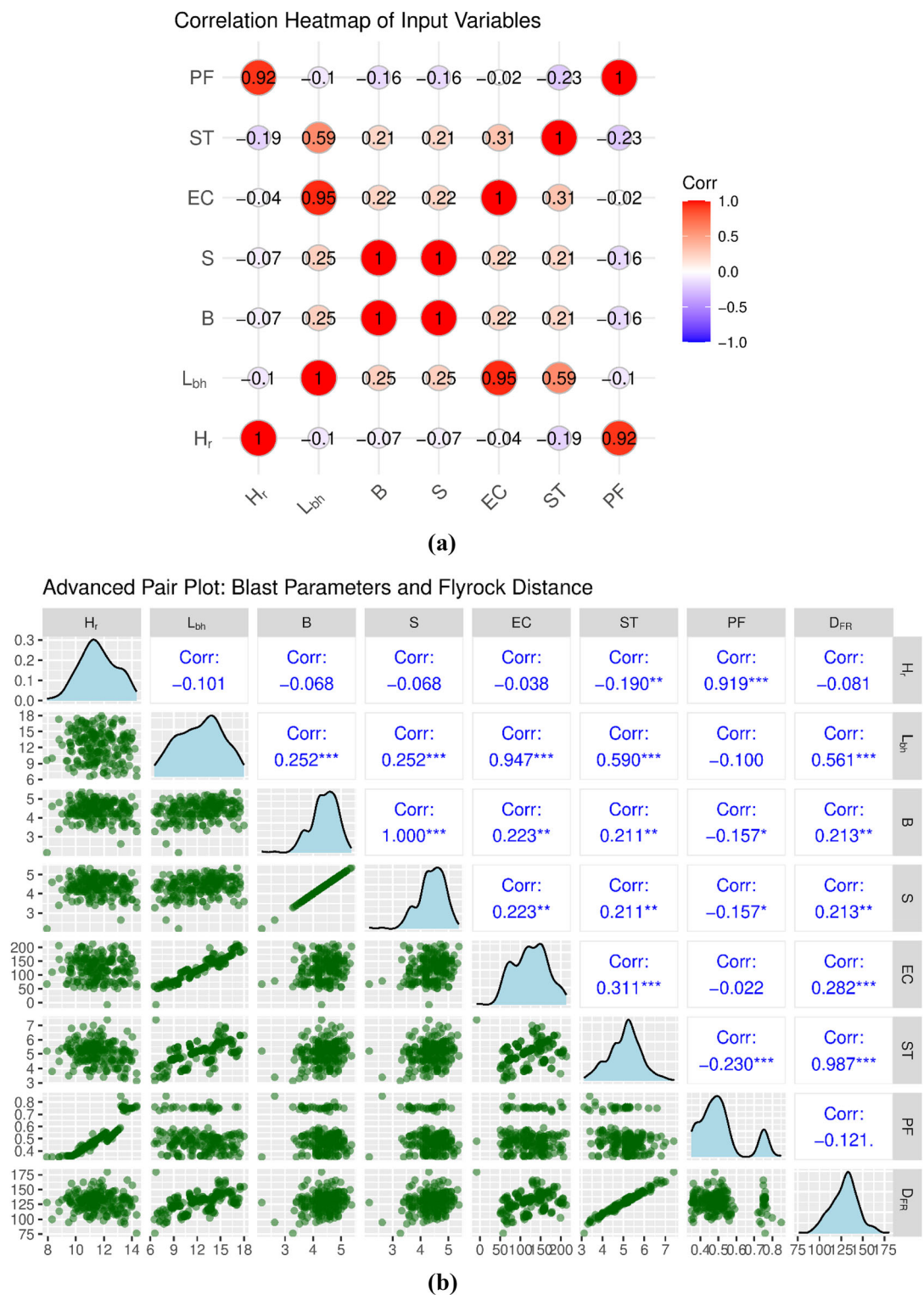


Figure 6. (a) Correlation matrix to examine correlations among all numerical variables. (b) Pair plots to visualize relationships among pairs of variables.

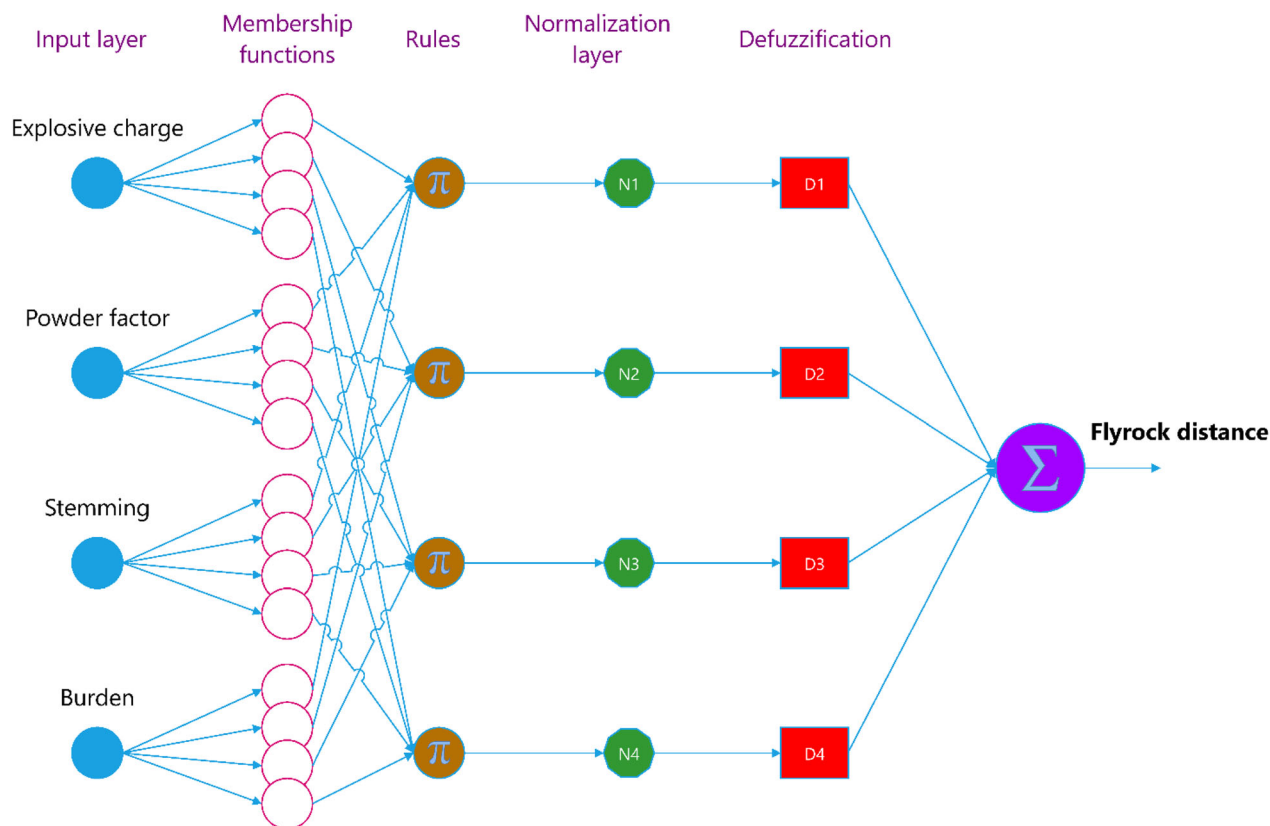


Figure 7. Structure of an ANFIS model.

Spanning 396.3 hectares, the Ta Phoi copper mine is located in Ta Phoi commune, Lao Cai city, Lao Cai province, on the eastern slope of the Cam Thang mountain range. In addition to its primary copper ore deposits, the mine contains significant reserves of pyrite and limestone, which are valuable for construction (Vinacomin, 2014a).

The mineral characteristics of the Ta Phoi copper mine include Lang Coc pyrite and construction limestone. Pyrite in the Lang Coc area occurs as rolling boulders distributed intermittently in a zone running NW–SW for about 2 km, with a width of 20–50 m. The boulders vary in size from a few centimeters to 1–2 dm, with some as large as 1.5 m. The outer surface of the pyrite is often weathered, exhibiting a brown to dark brown color due to limonite alteration, while the interior remains brass yellow. Pyrite grains are euhedral to subhedral, with sizes ranging 0.5–2 cm. The mineral occurs in dense clusters, disseminations, and vein fillings within quartz fractures, with sulfur content ranging 12.31–36.75% (Vinacomin, 2014b).

Meanwhile, the mine also features limestone deposits in the Da Dinh formation, exposed as low mountain ridges with steep cliffs. These deposits consist of layers of limestone and marble, making them an important resource for construction in the region (Vinacomin, 2014b). The ore and rock in this mine have high hardness, ranging 9–15 on the Protodiakonov scale (Protodiakonov et al., 1964), necessitating the use of blasting for fragmentation. Due to this level of hardness, the demand for explosive energy is substantial, leading to significant flyrock, especially under complex geological conditions or when blasting techniques are not well executed.

Input Parameters Used

Prior to blasting, a blast pattern must be carefully designed to ensure safety (e.g., minimize ground vibration and flyrock) and maximize the effectiveness of the blasting process. Proper blast

Pseudo-code of the Lévy-JA algorithm (LJ)

```

1  Initialize the population of candidate solutions ( $x$ ) with size  $N$ .

    Each candidate solution  $x_i$  ( $i=1,2,...N$ ) is randomly generated within the
    problem's search space.

2  Set the maximum number of iterations ( $Max_{iter}$ ) and the Lévy flight parameter ( $\lambda$ ).

3  Evaluate the objective function for each candidate solution in the population.

4  Identify the best ( $x_{best}$ ) and worst ( $x_{worst}$ ) solutions in the current population.

5  Repeat until the termination criterion is met (e.g.,  $Max_{iter}$  is reached):

6      For each candidate solution  $x_i$  in the population:

7          Generate two random numbers  $r_1$  and  $r_2$  uniformly distributed in  $[0,1]$ .

8          Perform Lévy flight step:

9              Generate a step size from Lévy distribution:  $L = \text{Lévy}(\lambda)$ 

10             Update the candidate solution using the Jaya formula enhanced with
            Lévy flight using Eq. (2).

11             If the updated solution  $x_i^{new}$  is better than the current solution  $x_i$ :

12                 Replace  $x_i$  with  $x_i^{new}$ .

13             Update the best and worst solutions in the population ( $x_{best}$  and  $x_{worst}$ ).

14             Store the best solution found so far.

15  Output the best solution found ( $x_{best}$ ) and its corresponding objective function value.

16  End.

```

Figure 8. Pseudo-code of the Lévy-JA algorithm.

design aids in achieving controlled rock fragmentation, reducing the risk of flyrock, and optimizing the use of explosives. Figure 3 illustrates a blast pattern of the Ta Phoi open-pit copper mine. The input parameters selected in this study were chosen based on their established influence on blast outcomes, particularly flyrock, and their ease of extraction from standard blast design records. These parameters include:

- Length of borehole (L_{bh}): this directly influences the explosive energy distribution, which impacts rock displacement and the likelihood of flyrock. This parameter was measured on-site using a calibrated measuring tape or borehole logging tools before blasting. The measurements were

recorded with an accuracy of ± 0.1 meters.

- Borehole diameter (D_{bh}): this affects the explosive charge capacity and energy intensity, which are critical factors in flyrock generation. This parameter was obtained directly from the drilling equipment specifications and verified using calipers at selected boreholes. This ensured consistency across all blast patterns.
- Burden (B) and spacing (S): these define the confinement of explosives. Improper burden or spacing can lead to inadequate confinement, increasing the potential for excessive flyrock. These parameters were determined based on the blast design plan provided by the mining engineering team. The burden represents the distance between the borehole and the nearest free face,

Pseudo-code of the BA algorithm	
1	Initialize the population of bats (x_i) for $i = 1, 2, \dots, N$
2	Initialize the position x_i and velocity v_i of each bat randomly within the search space.
3	Initialize the frequency (f_i), loudness (A_i), and pulse emission rate (r_i) for each bat.
4	Define the objective function to be optimized.
5	Identify the current best solution x_{best} in the initial population.
6	Set the maximum number of iterations (Max_{iter}).
7	Repeat until the stopping criterion is met (e.g., Max_{iter} is reached):
8	For each bat i in the population:
9	Update the frequency (f_i) for bat i using Eq. (3).
10	Update the velocity (v_i) of bat i using Eq. (4).
11	Update the position (x_i) of bat i using Eq. (5).
12	Generate a local solution around the best solution if a random number is greater than r_i :
13	If $rand(\) > r_i$ then applying Eq. (6).
14	Evaluate x_{new} using the objective function.
15	If the fitness of x_{new} is better than the fitness of x_i^{t+1} :
16	Accept x_{new} as the new position x_i^{t+1} .
17	Update the loudness (A_i) and pulse emission rate (r_i) for bat i using Eqs. (7) and (8).
18	Evaluate the fitness of each bat's position in the population.
19	Update the best solution x_{best} if a better solution is found.
20	Output the best solution found x_{best} and its corresponding fitness value.
21	End.

Figure 9. Pseudo-code of the BA algorithm.

while spacing is the distance between adjacent boreholes. Both were validated in the field using measuring tapes and laser distance meters.

- Powder factor (PF): representing the ratio of explosive weight to the amount of rock blasted, this is a key indicator of energy application. An

inappropriate powder factor can lead to excessive energy release, increasing flyrock risks. This parameter, calculated as the ratio of the total weight of explosives used to the volume of rock blasted, was determined from the blast design records. Explosive weights were measured using

Pseudo-code of the FA algorithm	
1	Initialize the population of fireflies (x_i) for $i = 1, 2, \dots, N$
2	Initialize the position x_i of each firefly randomly within the search space.
3	Define the light intensity i_i at each firefly's position x_i based on the objective function.
4	Set the algorithm parameters: attractiveness β_0 , light absorption coefficient γ , and randomization parameter α .
5	Define the objective function to be optimized.
6	Repeat until the stopping criterion is met (e.g., maximum number of iterations):
7	For each firefly i in the population:
8	For each firefly j in the population:
9	If $i_j > i_i$ (i.e., firefly j is brighter than firefly i):
10	Calculate the distance r_{ij} between fireflies i and j :
	$r = \ x_i - x_j\ $
11	Calculate the attractiveness β_{ij} of firefly j as seen by firefly i using Eq. (9).
12	Move firefly i towards firefly j using the equation using Eq. (10).
13	Evaluate the new position of firefly i and update its light intensity i_i .
14	Rank the fireflies based on their light intensity and identify the current best solution.
15	Output the best solution found (position of the brightest firefly) and its corresponding light intensity (objective function value).
16	End.

Figure 10. Pseudo-code of the FA algorithm.

electronic weighing scales, while the blasted rock volume was estimated from pre-blast surveys conducted with UAV-based photogrammetry.

- Explosive charge (EC) and length of explosive (L_{exp}): these parameters dictate the total energy released during blasting and its distribution along the borehole, both of which are critical in controlling the extent of flyrock. The EC was computed based on the borehole dimensions and the explosive type used. The charge weights were verified during explosive loading by trained per-

sonnel, and the L_{exp} was determined based on the length of the explosive column in each borehole as specified in the blast design. Field checks were conducted during loading to confirm adherence to the design.

- Stemming (ST): stemming material and length provide borehole confinement, preventing explosive gases from escaping prematurely, which is crucial for minimizing flyrock. This parameter placed at the top of each borehole was recorded during the blasting preparation phase.

Pseudo-code of the SSO algorithm	
1	Initialize the population of spiders (x_i) for $i = 1, 2, \dots, N$
2	Initialize the position x_i of each spider randomly within the search space.
3	Assign gender to each spider (male or female) based on a predefined ratio.
4	Evaluate the fitness of each spider using the objective function.
5	Identify the best and worst spiders in the initial population based on fitness.
6	Set the algorithm parameters: attraction coefficient, repulsion coefficient, random perturbation factor, and maximum number of iterations (Max_{iter}).
7	Repeat until the stopping criterion is met (e.g., maximum number of iterations):
8	For each spider i in the population:
9	Compute the social attraction force:
10	Calculate the attraction towards other spiders with better fitness values.
	The attraction is stronger towards spiders with significantly better fitness.
11	Calculate the attractiveness β_{ij} of firefly j as seen by spider i using Eq. (9).
12	Compute the social repulsion force:
13	Calculate the repulsion from spiders in close proximity or with worse fitness values.
14	The repulsion helps to avoid overcrowding and maintains diversity.
15	Update the position of spider i based on the combined effects using Eq. (11).
16	If the spider is male:
17	Perform additional behavior such as territorial fights or mating.
18	Update the position of the male spider based on these behaviors.
19	Evaluate the fitness of the new position $x_i(t+1)$.
20	Generate offspring (new spiders) by simulating mating between male and female spiders:
21	Select pairs based on fitness and generate new solutions.
22	Add the offspring to the population.
23	Rank all spiders based on their fitness values and update the best and worst solutions found so far.
24	If necessary, remove the worst-performing spiders to maintain population size.
25	Output the best solution found (position of the spider with the highest fitness) and its corresponding fitness value.
26	End.

Figure 11. Pseudo-code of the SSO algorithm.

Measurements were taken using measuring sticks to ensure consistency across boreholes.

- Rock hardness (H_r): the hardness of the rock influences its resistance to fracturing. Harder rocks may require more energy to fragment, potentially increasing the risk of flyrock if not properly controlled. This parameter was assessed using a Schmidt hammer on representative rock samples collected from the blast site. Multiple readings were taken at different locations, and the average rebound value was recorded to account for variability.

The above-described parameters were selected because they are the most commonly documented in blasting design practices and are widely reported in the literature as significant predictors of blast-induced flyrock. Moreover, they represent a balance between design-related (controllable) factors and site-specific geological conditions, enabling a comprehensive analysis of flyrock generation. This selection ensures that the model captures the critical aspects influencing flyrock while maintaining practicality for real-world applications.

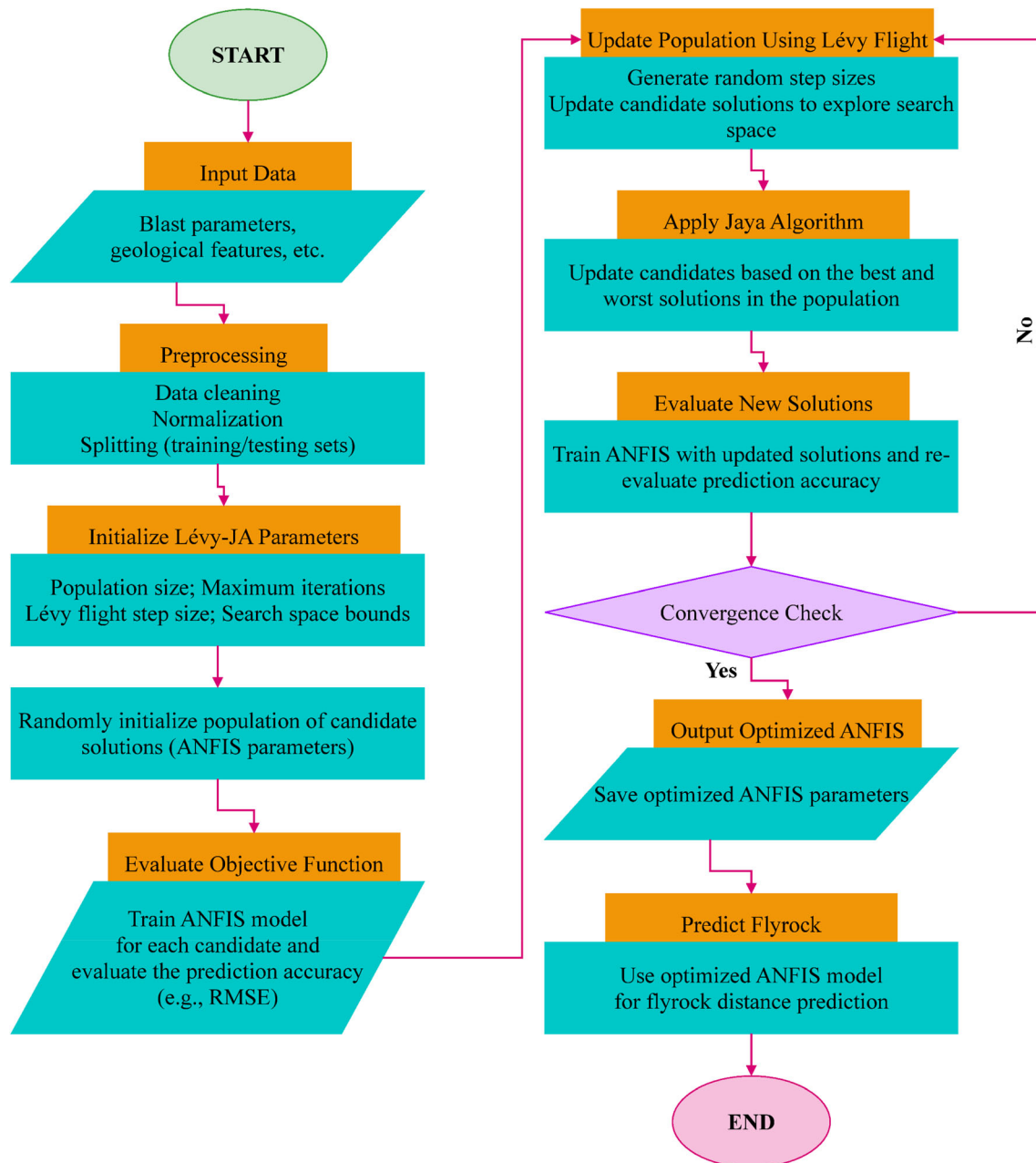


Figure 12. Flowchart of the ANFIS-LJ model for predicting blast-induced flyrock.

Measuring Flyrock Distance

For measuring flyrock distance (D_{FR}), this study utilized a drone (UAV) with high resolution to capture the entire flyrock phenomenon of each blast. The captured video was then imported into ProAnalyst® software to analyze the flyrock distance

(Fig. 4). Finally, data from 204 blasting events were collected and compiled as a dataset for this study.

In this study, a single UAV was used for measuring flyrock distances. However, using multiple UAVs offers several advantages. It ensures improved coverage by monitoring the blast area from multiple angles, reducing the chances of missing flyrock fragments. This approach enhances accuracy

Table 1. Selection of metaheuristic algorithms' parameters

Parameters	Optimization algorithms			
	LJ	BA	FA	SSO
Population size	50	50	50	50
Epoch	1000	1000	1000	1000
Loudness	None	0.8	None	None
Pulse rate	None	0.95	None	None
Pulse frequency min-max	None	0–10	None	None
Total number of sparks generated by the population size fireworks	None	None	50	None
Number of sparks generated in each explosion generation	None	None	5	None
Maximum and minimum explosion amplitudes	None	None	40–0	None
Amplitude proportionality parameter	None	None	0.04	None
Spark count proportionality parameter	None	None	0.8	None
Female percent (min-max)	None	None	None	0.65–0.9

through overlapping data, allowing cross-verification and eliminating errors caused by occlusions or shadowed areas. Multiple UAVs also enable faster data collection by surveying the site concurrently, which is critical for operations with high blasting frequencies. Additionally, they improve safety by minimizing the need for personnel to enter hazardous zones and provide scalability for large sites where a single UAV may be insufficient. Redundancy is another benefit, ensuring uninterrupted data collection in case of technical failures. However, deploying multiple UAVs involves challenges such as coordination, requiring advanced flight planning to prevent collisions. It also incurs higher costs for acquisition, maintenance, and training. Data integration from multiple sources adds complexity, and regulatory compliance may be stricter for multi-UAV systems.

Data Analysis

Before developing the D_{FR} predictive models, data analysis is crucial for understanding parameter correlation, variable sensitivity, outlier detection, exploratory insights, feature engineering, distribution of D_{FR} , multicollinearity check, and ensuring UAV data quality. By analyzing dataset beforehand, we can ensure that the model developed is well-informed by the characteristics and patterns of the data, leading to better predictive accuracy and more reliable outcomes in flyrock distance predictions.

In Figure 5a, we can see that some variables, like B and S, seem to have a fairly uniform distribution, while others, such as EC and D_{FR} , show

more skewed distributions. This suggests potential nonlinear relationships and possibly the need for transformation of some variables for modeling purposes. From Figure 5b, it is conspicuous that the data contain some potential outliers, especially in variables like D_{FR} and spacing S. This indicates variability across the dataset, which could impact model performance. Some of the variability in D_{FR} may be due to extreme flyrock events, which are critical to account for when building predictive models.

From Figure 6a, we can see that there is very strong positive correlation (0.95) between L_{bh} and PF. This implies that, in practice, as the borehole length increases, the powder factor also increases, which makes sense in blasting operations. This strong relationship suggests that both variables are essentially providing similar information, and one of them might be redundant for the model.

The S and B have a perfect correlation (1.0), indicating that they are either calculated or set in tandem during the blast design. Because they provide identical information, removing one of them variables to avoid issues of multicollinearity during model development might be considered. Herein, S was removed from the dataset to leave the B variable.

The EC variable shows moderate positive correlation (0.59) with both L_{bh} and PF, suggesting that higher explosive charges are used with longer boreholes and higher powder factors. This relationship is expected because these parameters often scale together in blasting designs to ensure effective fragmentation. The ST has weaker correlations with other parameters. For example, its highest correla-

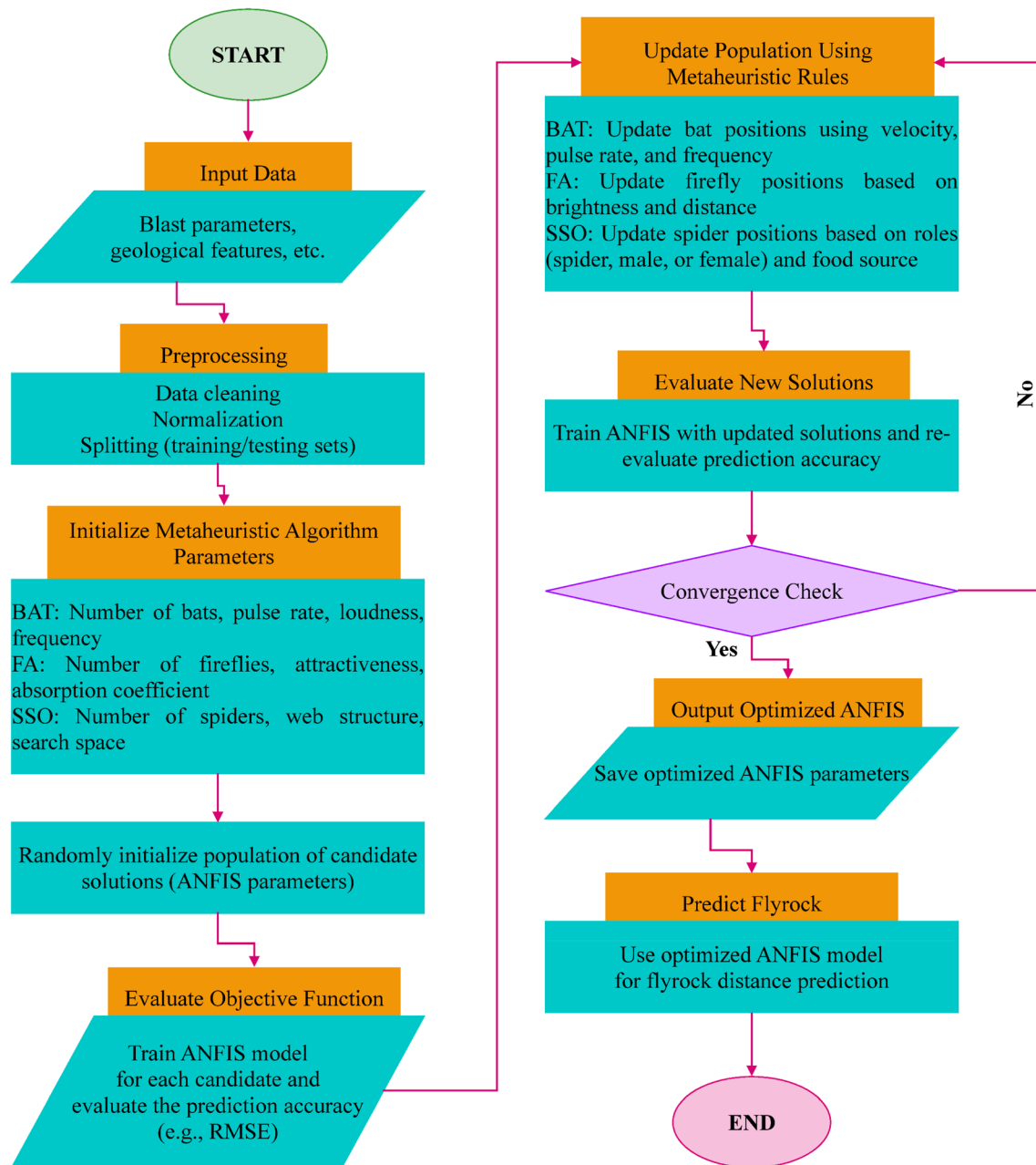


Figure 13. General flowchart of the ANFIS-BA, ANFIS-FA, and ANFIS-SSO models for predicting blast-induced flyrock.

tion is with L_{bh} (0.31) but it does not show a strong relationship with any other input variables. This might indicate that stemming is controlled independently in the blast design process or has less direct interaction with other variables like explosive charge or borehole length.

The H_r shows moderate negative correlation (-0.19) with EC, suggesting that, as the hardness of rock increases, the explosive charge may decrease, potentially because harder rock requires more precision in charge placement rather than higher amounts of explosive. Normally, as the hardness of rock increases, the PF also increases, and their cor-

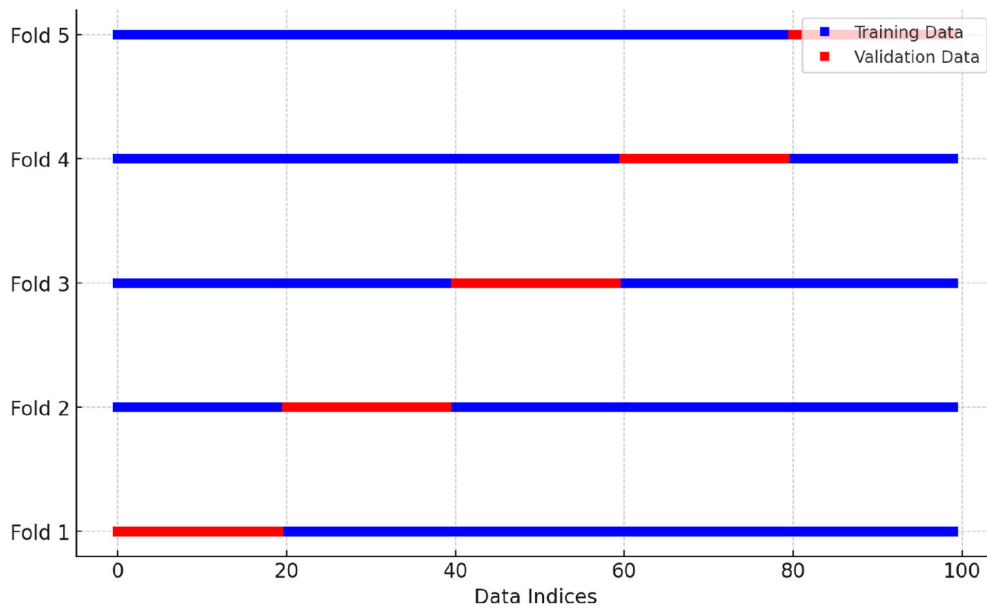


Figure 14. Illustration of fivefold cross-validation used in this study.

relation is 0.92 in this study. In addition, the H_r also has weak correlations with the other variables, indicating that it might play a more isolated role in the blast design.

Figure 6b indicates high correlation between EC and D_{FR} , as well as PF and L_{bh} , as seen previously. Other correlations, such as between H_r and D_{FR} , are weaker but still relevant to the analysis. The visual relationships, combined with the correlation values, suggest that EC, PF, ST, and B are the key predictors of D_{FR} , making them critical for modeling.

MACHINE LEARNING AND METAHEURISTIC ALGORITHMS APPROACHES

Principle of ANFIS

The ANFIS is a hybrid model that merges the advantages of fuzzy logic and neural networks to represent complex systems. It combines the human-like reasoning approach of fuzzy systems with the learning and networked structure of neural networks (Andre, 2024), making it highly effective for approximating nonlinear functions. The ANFIS is

based on a fuzzy inference system (FIS), typically a Sugeno-type fuzzy system, which uses fuzzy if-then rules to model the input-output relationship. FIS maps input to output using fuzzy logic, where the input variables are transformed into linguistic terms with corresponding membership functions.

To address relevant problems, the ANFIS employs a hybrid learning approach that merges gradient descent with least-squares estimation to adjust both the membership functions and the parameters of the fuzzy rules (Momeni et al., 2023). The learning process involves two stages: a forward pass and a backward pass. During the forward pass, the input data flow through the network, and the consequent parameters of the rules are determined using least-squares estimation. In the backward pass, the error between the predicted and actual output is propagated backward, updating the premise parameters (related to the membership functions) using gradient descent. The ANFIS typically consists of five layers, each performing a specific function in the learning and inference process (Fig. 7).

- Layer 1 (input layer): the input variables are passed to this layer, where membership functions are applied to transform crisp inputs into fuzzy values.
- Layer 2 (rule layer): this layer calculates the fir-

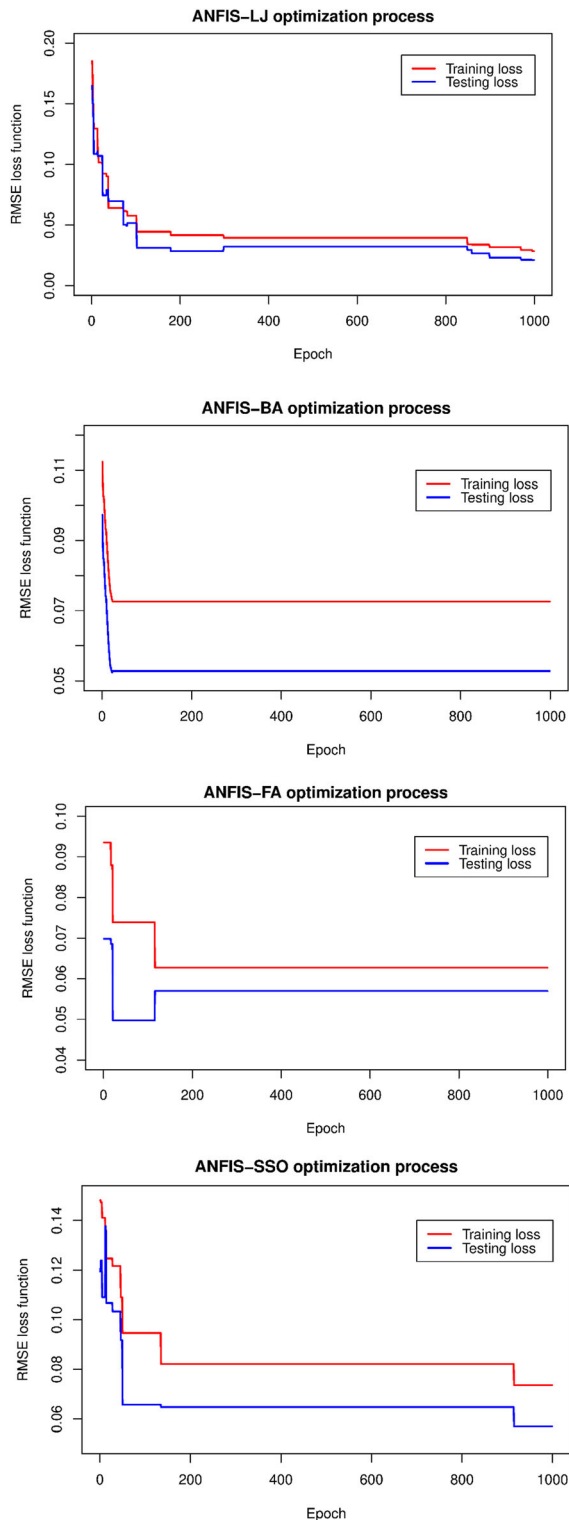


Figure 15. Iterative optimization process of the ANFIS model based on the LJ, BA, FA, and SSO algorithms for predicting blast-induced flyrock at the Ta Phoi copper mine.

ing strength of each rule by taking the product of the membership values for each rule.

- Layer 3 (normalization layer): the firing strengths are normalized, i.e., the ratio of each rule's firing strength to the sum of all firing strengths is computed.
- Layer 4 (defuzzification layer): the consequent parameters are applied to the normalized firing strengths to produce the output for each rule.
- Layer 5 (output layer): the outputs from all rules are summed to produce the final output of the system.

The learning in ANFIS is adaptive, meaning the system can adjust its parameters based on the input–output data it is trained on. This adaptability makes it effective in handling systems where the relationship between input and output is complex and not easily modeled using traditional methods. The hybrid learning process allows ANFIS to optimize both the structure (by adjusting the fuzzy rules) and the parameters (by tuning the membership functions and rule outputs).

Principle of Jaya Optimization Algorithm Under the Enhancement of Lévy Flight Technique (LJ)

The Jaya optimization algorithm is a straightforward but highly effective population-based optimization method, introduced for addressing continuous optimization challenges (Rao, 2019). The term “Jaya” comes from a Sanskrit word meaning “victory”, symbolizing the algorithm's objective to achieve the optimal solution by consistently progressing towards the best outcome and steering away from the worst (Wang et al., 2023a). When enhanced with the Lévy flight technique, the algorithm gains additional exploration capabilities, making it more effective in escaping local optima and improving convergence speed.

Jaya Optimization Algorithm (JA)

The algorithm begins by randomly generating a population of potential solutions within the specified search space, with each candidate representing a possible answer to the optimization problem. Afterward, each candidate is assessed using an objective function, which measures how well the

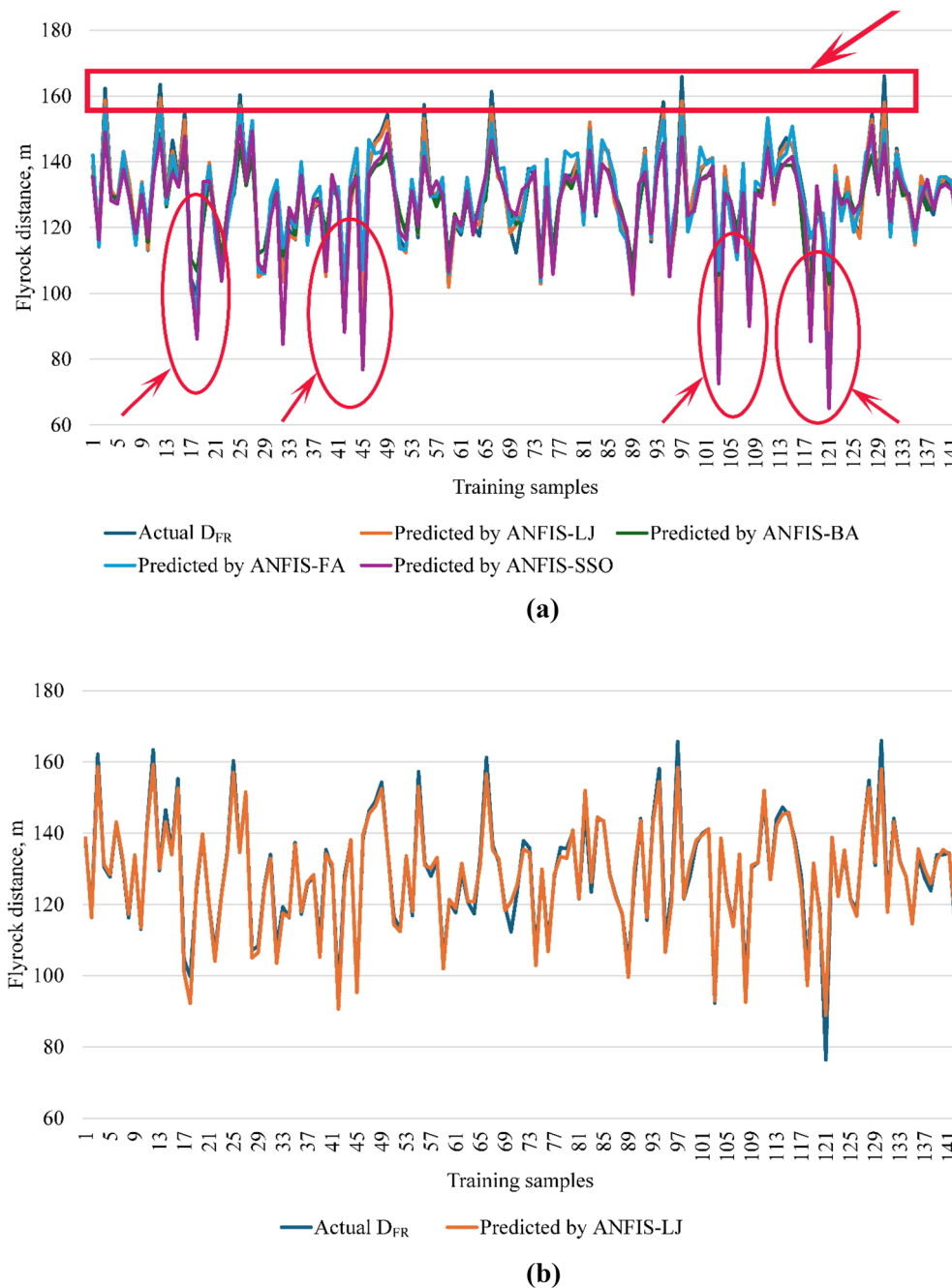


Figure 16. Outcome predictions of the developed hybrid models on the training dataset: (a) comparison between all models; (b) comparison between the actual model and the proposed ANFIS-LJ model.

solution addresses the problem. During each iteration, the best solution (with the highest objective function value in a maximization problem or the lowest in a minimization problem) and the worst solution (with the lowest objective function value in

a maximization problem or the highest in a minimization problem) are determined (Platt et al., 2018). The positions of the candidate solutions are updated based on the following equation:

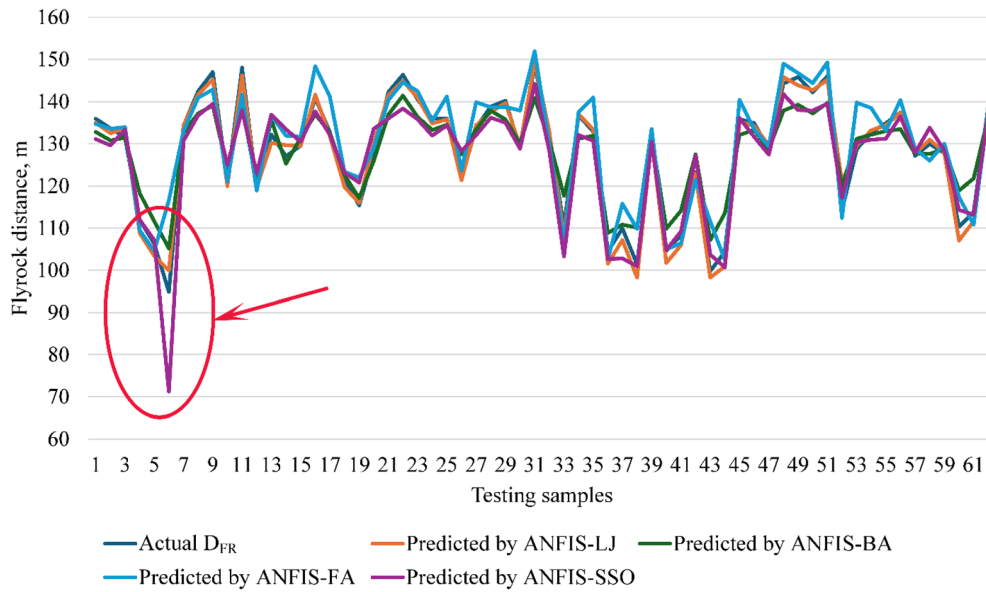


Figure 17. Outcome predictions of the developed hybrid models on the testing dataset.

$$x_i^{\text{new}} = x_i + r_1 \times (x_{\text{best}} - |x_i|) - r_2 \times (x_{\text{worst}} - |x_i|) \quad (1)$$

where x_i^{new} is the new position of the i th solution, x_i is the current position of the i th solution, x_{best} is the best solution in the current population, x_{worst} is the worst solution in the current population, and r_1 and r_2 are random numbers uniformly distributed in the range $[0, 1]$.

Unlike many other optimization algorithms, Jaya does not require algorithm-specific control parameters (e.g., crossover rate, mutation rate, inertia weight), which simplifies its implementation and tuning (Yu et al., 2017; Rao, 2019). The algorithm iteratively updates the candidate solutions until a termination criterion is met, typically a maximum number of iterations or convergence to a satisfactory solution.

Lévy Flight Technique

Lévy flight is a random walk characterized by step lengths that have a probability distribution that is heavy-tailed (often a power-law distribution) (Barthelemy et al., 2008). This means that Lévy flights consist of many small steps combined with occasional large steps. This technique helps in

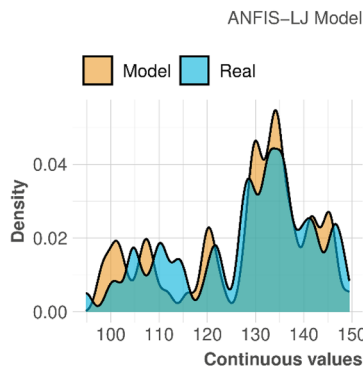
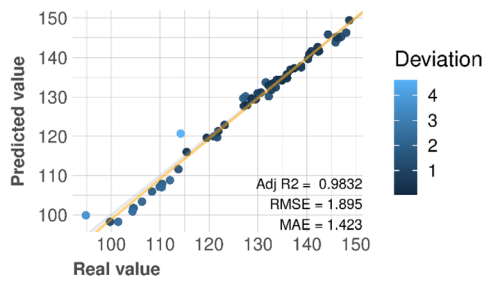
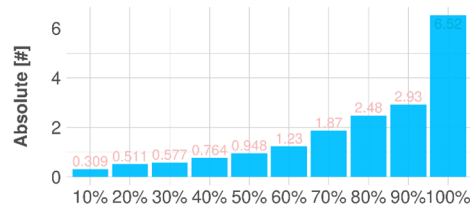
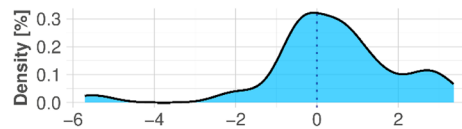
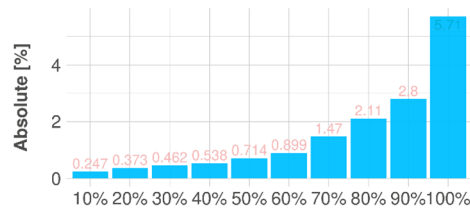
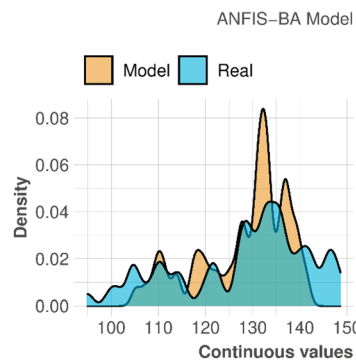
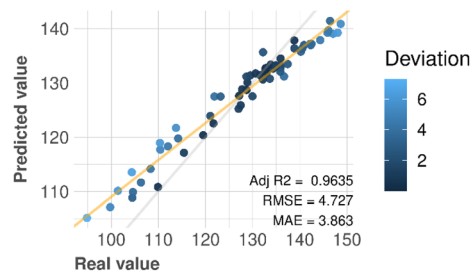
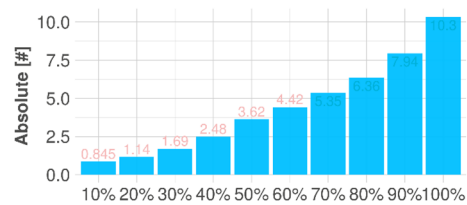
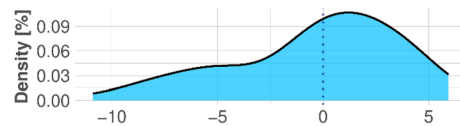
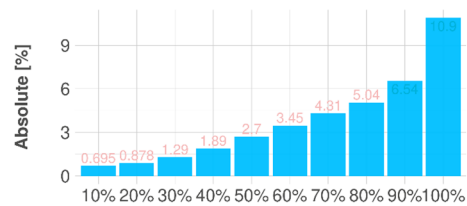
enhancing the exploration ability of the optimization algorithm by allowing the algorithm to make long jumps, thereby reducing the chances of getting trapped in local optima (Jamil & Zepernick, 2013; Haklı & Uğuz, 2014; Emary et al., 2019; Wu et al., 2023). The Lévy flight can be integrated into the JA by modifying the solution update equation as:

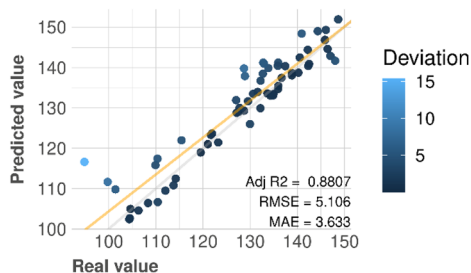
$$x_i^{\text{new}} = x_i + \text{Levy}(\lambda) \times (r_1 \times (x_{\text{best}} - |x_i|) - r_2 \times (x_{\text{worst}} - |x_i|)) \quad (2)$$

where $\text{Levy}(\lambda)$ represents the step size drawn from a Lévy distribution, which is controlled by the parameter λ (Yang & He, 2016). Typically, λ is set between 1 and 3.

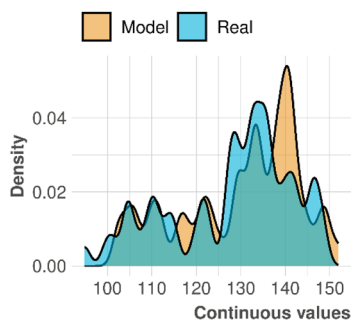
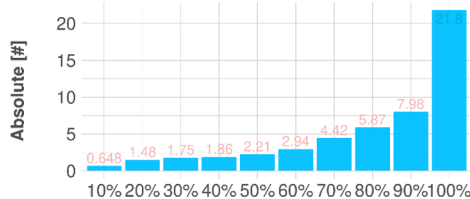
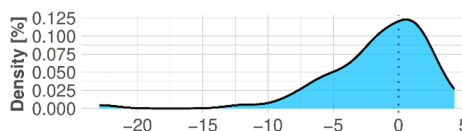
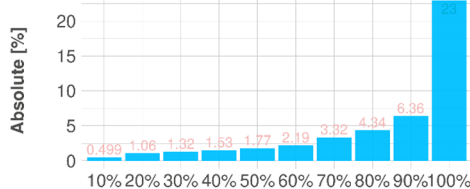
The Lévy flight helps the algorithm to explore the search space more effectively, especially in the early stages of the optimization process, by allowing the solutions to make larger jumps in the search space (Haklı & Uğuz, 2014). This increases the likelihood of escaping local optimal and finding a global optimum. The combination of Jaya's inherent exploitation (moving towards the best and away from the worst solutions) with Lévy flight's exploration (large random jumps) creates a balance that enhances the overall performance of the algorithm.

In summary, the enhanced JA with Lévy flight works by iteratively improving the candidate solu-

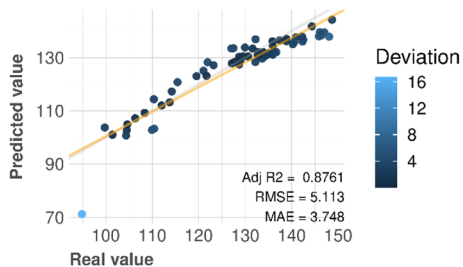
ANFIS-LJ Model Results*Flyrock Prediction Model**Cuts and distribution by absolute error**Cuts and distribution by absolute percentage error***(a)** ANFIS-LJ's performance**ANFIS-BA Model Results***Flyrock Prediction Model**Cuts and distribution by absolute error**Cuts and distribution by absolute percentage error***(b)** ANFIS-BA's performance**Figure 18.** Models' performance for predicting blast-induced flyrock at the Ta Phoi copper mine.

ANFIS-FA Model Results*Flyrock Prediction Model*

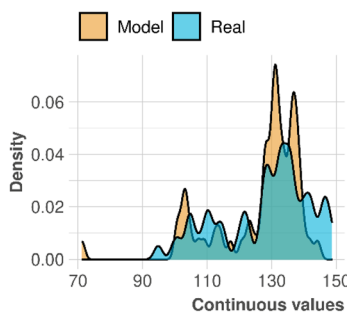
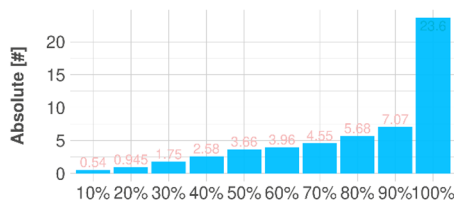
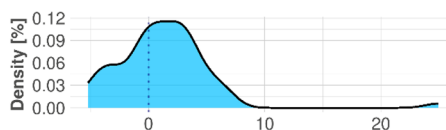
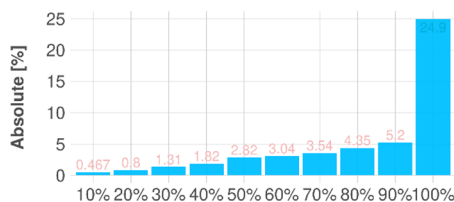
ANFIS-FA Model

*Cuts and distribution by absolute error**Cuts and distribution by absolute percentage error*

(c) ANFIS-FA's performance

ANFIS-SSO Model Results*Flyrock Prediction Model*

ANFIS-SSO Model

*Cuts and distribution by absolute error**Cuts and distribution by absolute percentage error*

(d) ANFIS-SSO's performance

Figure 18. continued.

tions in a population. It balances exploration and exploitation by using the JA's simple yet effective mechanism for solution improvement and the Lévy flight's ability to make large, random jumps. This balance helps the algorithm to converge more quickly to a global optimum while avoiding local optima. Figure 8 shows the pseudo-code of the Lévy-JA algorithm.

Table 2. Performance metrics of ANFIS models optimized by metaheuristic algorithms for predicting blast-induced flyrock at the Ta Phoi copper mine

Model	Training dataset			Testing dataset		
	MAE	RMSE	R^2	MAE	RMSE	R^2
ANFIS-LJ	1.719	2.544	0.974	1.423	1.895	0.981
ANFIS-BA	4.724	6.509	0.832	3.863	4.727	0.881
ANFIS-FA	4.016	5.623	0.874	3.633	5.106	0.861
ANFIS-SSO	4.782	6.599	0.827	3.748	5.113	0.860

Principle of Bat Algorithm (BA)

The BA is a nature-inspired optimization technique introduced by Yang (2010). It draws inspiration from the echolocation behavior of microbats, which use sound waves to locate prey, navigate, and avoid obstacles. This behavior is simulated in the algorithm to address optimization challenges, particularly in continuous optimization tasks. In the BA, echolocation mechanism is used. Bats emit ultrasonic pulses and listen to the echoes that bounce back from the surrounding objects to detect the prey and navigate (Singh et al., 2022). The distance to the prey or obstacle is calculated by the time delay between the emission of the pulse and the reception of the echo. The loudness and frequency of the pulses change according to the distance to the target and bats adjust these parameters dynamically.

Among the initial population of virtual bats, each bat is a candidate solution in the search space. Each bat has position, velocity, frequency, loudness and pulse emission rate. Each bat flies with a varying frequency f_i , which controls the velocity of the bat and helps it explore the search space, thus:

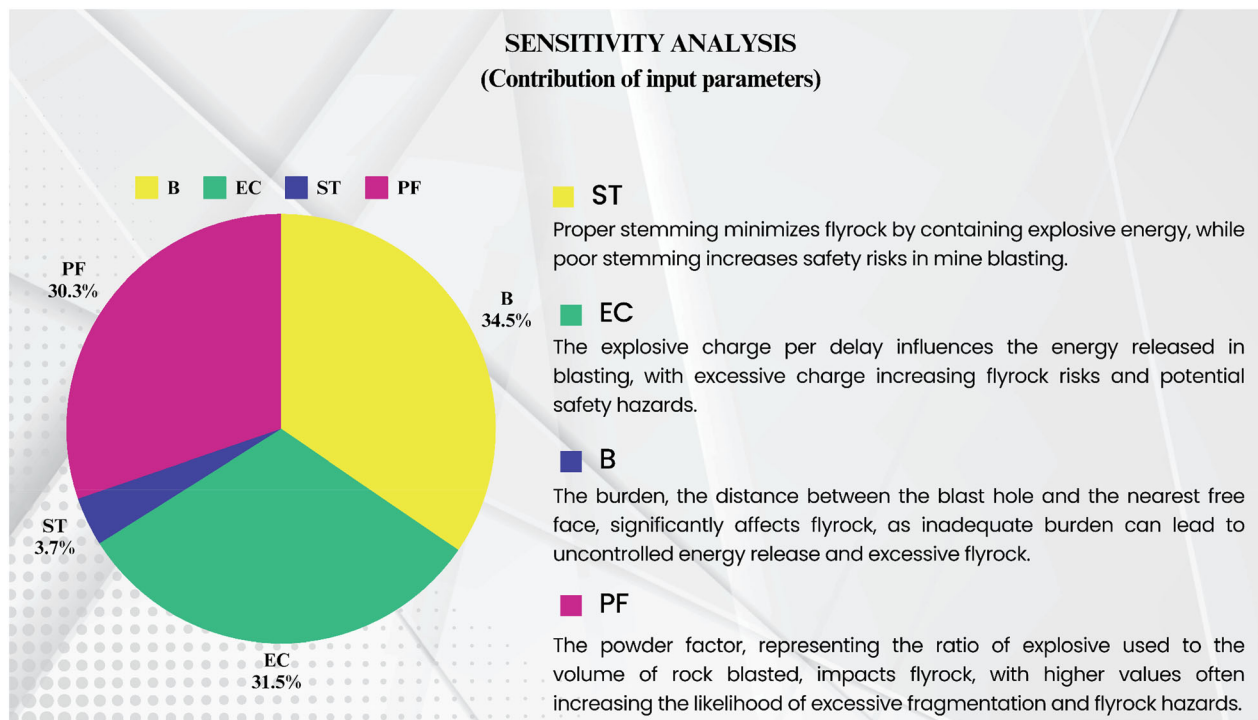


Figure 19. LHS sensitivity analysis results based on the best model (i.e., ANFIS-LJ).

$$f_i = f_{\min} + (f_{\max} - f_{\min}) \times \beta \quad (3)$$

where $f_{\min}$ and $f_{\max}$ are the minimum and maximum frequencies, respectively, and β denotes a random number uniformly distributed in $[0, 1]$.

The velocity of each bat v_i is updated based on the frequency and the distance to the current best solution x_{best} , thus:

$$v_i^{t+1} = v_i^t + (x_i^t - x_{\text{best}}) \times f_i \quad (4)$$

The position of each bat x_i is updated using the new velocity, thus:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (5)$$

In the BA, bats conduct a local search near the current optimal solution by taking small random steps. If a randomly generated number (uniformly distributed within $[0, 1]$) exceeds the bat's pulse emission rate, a new solution is produced locally, thus:

$$x_{\text{new}} = x_{\text{best}} + \varepsilon \times A_i^t \quad (6)$$

where ε is a random number uniformly distributed in $[-1, 1]$, A_i^t stands for the current loudness of the bat. As bats approach their prey, their loudness A_i decreases and their pulse emission rate r_i increases. These parameters are adjusted dynamically:

$$A_i^{t+1} = \alpha \times A_i^t \quad (7)$$

$$r_i^{t+1} = r_i^0 \times [1 - \exp(-\gamma \times t)] \quad (8)$$

where α and γ are constants, typically $0 < \alpha < 1$ and $\gamma > 0$, and r_i^0 is the initial pulse rate. After updating the positions of all bats, the algorithm evaluates the fitness of each bat (candidate solution) using the objective function. The best solution found so far is updated if a better solution is found.

The algorithm repeats the above steps until a termination condition is satisfied, such as reaching a predefined maximum number of iterations or obtaining an acceptable fitness value. The optimal solution discovered throughout the optimization process is provided as the final result. Figure 9 presents the pseudo-code for the BA.

Principle of Firefly Algorithm (FA)

The FA is a nature-inspired optimization algorithm developed by Yang (2009). It is based on the

Table 3. Real-world blasting data for validation in practical engineering

ST	EC	L_b	PF	D_{FR}
5.1	158.2	5.8	0.45	142.67
4.7	128.6	5.4	0.44	134.32
4.7	108.1	5.0	0.37	125.43
4.0	108.0	3.8	0.46	102.91
5.0	64.6	4.3	0.42	114.19
4.1	128.6	5.4	0.49	135.55
4.5	88.6	5.7	0.51	142.90
4.5	114.6	5.1	0.53	130.63
4.9	158.2	5.8	0.42	141.89
4.2	157.1	4.4	0.50	114.87
5.0	140.5	6.8	0.36	160.84
4.4	166.9	6.0	0.58	150.62
3.9	57.8	4.0	0.57	109.40

flashing behavior of fireflies. The primary purpose of the algorithm is to solve optimization problems by mimicking the way fireflies attract each other using their bioluminescent light. In nature, fireflies use bioluminescence to communicate and attract mates. The intensity of the light emitted by a firefly is perceived by others and is a key factor in their interaction (Jain et al., 2021). The brighter a firefly appears, the more attractive it is to other fireflies. This attraction is proportional to the intensity of the light, which diminishes as the distance between fireflies increases (Fister et al., 2013).

In the context of the algorithm, fireflies are considered to be attracted to each other based on their brightness (which corresponds to the quality of the solution they represent). The brightness of a firefly corresponds to the value of the objective function at its current location. In a maximization problem, a firefly with greater brightness signifies a superior solution (higher objective function value). The goal is for each firefly to move towards brighter (better) fireflies in the search space. The appeal of a firefly increases with its brightness and decreases with the square of the distance between two fireflies (Ullah & Hussain, 2019). This can be expressed as:

$$\beta(r) = \beta_0 \times \exp(-\lambda \times r^2) \quad (9)$$

where $\beta(r)$ denotes the attractiveness of the firefly at a distance r , β_0 is the initial attractiveness at $r = 0$, and γ stands for the light absorption coefficient, which determines the rate at which the light intensity decreases with distance.

A firefly i is attracted to another firefly j that is brighter, and it moves towards j according to the following equation:

$$x_i = x_i + \beta(r) \times (x_j - x_i) + \alpha \times (\text{rand} - 0.5) \quad (10)$$

where x_i and x_j are the positions of fireflies i and j in the search space, $\beta(r)$ is the attractiveness function, α presents a randomization parameter that controls the movement randomness, and rand is a random number uniformly distributed in $[0, 1]$. The random term $\alpha \times (\text{rand} - 0.5)$ adds randomness to the movement, allowing fireflies to explore the search space and avoid getting stuck in local optima.

The intensity of light decreases with distance, and it is also affected by the medium through which it travels. In the algorithm, this is modeled by the light absorption coefficient γ . A higher value of γ means that the light intensity decreases rapidly, leading to a more localized search. A lower value allows for a more global search.

The algorithm balances exploration (searching globally) and exploitation (refining locally) through the control of the parameters α (randomization) and γ (light absorption). The process of fireflies moving towards brighter ones represents exploitation, while the randomness in movement allows for exploration.

The algorithm continues to run until a stopping condition is satisfied, which could be reaching a predefined number of iterations or achieving convergence to the optimal solution. Finally, the algorithm outputs the position of the brightest firefly, which corresponds to the best solution found during the optimization process. Figure 10 shows the pseudo-code of the FA algorithm.

Principle of Social Spider Optimization (SSO)

The SSO algorithm is a population-based optimization technique inspired by the social behavior of spiders, particularly those species that live in colonies and exhibit social interactions. The SSO algorithm was introduced to solve optimization problems by mimicking the way spiders in a colony interact with each other, share information, and search for food (James & Li, 2015). In nature, social spiders live in colonies where they interact with each other to survive. They communicate through vibrations on the web, which help them detect prey, predators, or other spiders (Cuevas et al., 2015). These vibrations are key to their ability to locate

food and coordinate their activities. In the SSO algorithm, spiders represent candidate solutions, and the social interactions between them are modeled to explore the search space and find the optimal solution.

The algorithm starts by initializing a population of spiders (candidate solutions) randomly within the search space. Each spider has a position in the search space, representing a potential solution to the optimization problem. The population of spiders is divided into male and female spiders, with each gender having different roles and behaviors in the optimization process. The proportion of males and females is typically predefined. Female spiders are generally more active and responsible for searching for food (exploring the search space), while male spiders focus more on mating and territorial behavior. The fitness of each spider is evaluated based on the objective function of the optimization problem. This fitness value represents the quality of the solution. Spiders communicate through vibrations on the web, which are modeled as information exchange between spiders in the SSO algorithm. This communication helps them adjust their positions (solutions) based on the information received from other spiders. The social interaction mechanism in SSO is based on two main components:

- Attraction: spiders are attracted to other spiders with better fitness values. This attraction guides their movement towards better solutions.
- Repulsion: to maintain diversity in the population and avoid premature convergence, spiders are repelled by others in close proximity or those with much worse fitness values.

The position of each spider is updated based on the combined effects of attraction and repulsion. This can be modeled as:

$$x_i(t+1) = x_i(t) + \text{Social_attraction} - \text{Social_repulsion} + \text{Random_perturbation} \quad (11)$$

where $x_i(t)$ is the position of spider i at time t , Social_attraction represents movement towards spiders with better fitness values, Social_repulsion represents movement away from spiders in close proximity or those with worse fitness values, and $\text{Random_perturbation}$ represents a random term to introduce diversity and exploration in the search space.

Male spiders may also perform additional behaviors like mating and territorial fights, which influence their position updates. In the SSO algorithm, mating occurs between male and female spiders, producing offspring (new candidate solutions). The offspring inherit characteristics (positions) from their parents and are added to the population. This process helps in exploring new areas of the search space and maintaining diversity in the population.

After each iteration, the fitness of all spiders is evaluated, and the best solution found so far is recorded. Similar to the FA algorithm and previous algorithms, the SSO algorithm continues to run until a stopping condition is satisfied, which could be reaching a predefined number of iterations or achieving convergence to the optimal solution. The best solution found during the optimization process is returned as the final result. Figure 11 shows the pseudo-code of the SSO algorithm.

New Hybrid Models based on ANFIS and Metaheuristic Algorithms

In this study, we developed four new hybrid models for predicting blast-induced flyrock, each of which is a combination of the ANFIS model and one of the four metaheuristic algorithms presented above.

The ANFIS–Lévy–Jaya (ANFIS–LJ) hybrid model integrates ANFIS with the Jaya optimization algorithm enhanced by the Lévy flight technique (Lévy–JA) to predict flyrock distance. The Lévy flight technique improves the exploration capabilities of the Jaya algorithm, making it more effective in optimizing ANFIS parameters. The working principle of the ANFIS–LJ model includes the following four steps:

1. Initialization: start with a basic ANFIS model, including an initial set of membership functions and fuzzy rules.
2. Optimization: use the Lévy–JA to optimize the ANFIS parameters, such as the membership function parameters and rule base. The Lévy flight mechanism enables the algorithm to make large, random jumps in the search space, avoiding local optima and ensuring better global exploration.
3. Evaluation: continuously evaluate the performance of the ANFIS model in predicting flyrock

distance, using a suitable error metric such as mean squared error (MSE).

4. Output: the final optimized ANFIS model is expected to have superior predictive accuracy for flyrock distance due to the enhanced exploration and exploitation capabilities of the Lévy–JA algorithm.

The flowchart of the ANFIS–LJ model is proposed in Figure 12.

The ANFIS–Bat algorithm (ANFIS–BA) hybrid model combines ANFIS with the BA to predict flyrock distance. The BA’s ability to balance exploration and exploitation through echolocation principles makes it an effective optimizer for ANFIS parameters. The working principle of the ANFIS–BA model includes the following four steps:

1. Initialization: initialize the ANFIS model with an initial set of parameters, including membership functions and fuzzy rules.
2. Optimization: apply the BA to optimize these parameters. Bats (candidate solutions) update their positions based on the current best solution, with a focus on moving towards better solutions (exploitation) while occasionally exploring new areas (exploration).
3. Evaluation: assess the ANFIS model’s predictive performance after each iteration and refine the parameters until the model achieves optimal prediction accuracy for flyrock distance.
4. Output: the optimized ANFIS model, refined through the BA, is expected to accurately predict flyrock distance, benefiting from the BA’s dynamic balance between exploration and exploitation.

The ANFIS–firefly algorithm (ANFIS–FA) hybrid model integrates ANFIS with the FA to predict flyrock distance. The FA uses the concept of light intensity and attractiveness to optimize ANFIS parameters, guiding the model towards better solutions. The working principle of the ANFIS–FA model includes the following four steps:

1. Initialization: begin with a basic ANFIS model, defining the initial membership functions and fuzzy rules.
2. Optimization: use the FA to optimize the parameters of the ANFIS model. Fireflies (candidate solutions) are attracted to brighter (bet-

- ter) solutions, and their positions are updated accordingly, which helps in refining the ANFIS model.
3. Evaluation: continuously evaluate the model's performance in predicting flyrock distance and adjust the parameters to minimize prediction errors.
 4. Output: the resulting ANFIS model, optimized by the FA, should offer enhanced accuracy in predicting flyrock distance, thanks to the FA's effective search and optimization process.

The ANFIS–social spider optimization (ANFIS–SSO) hybrid model combines ANFIS with SSO to predict flyrock distance. SSO models the social behavior of spiders to optimize ANFIS parameters, making the algorithm particularly effective in maintaining diversity and avoiding premature convergence. The working principle of the ANFIS–SSO model includes the following four steps:

1. Initialization: start with an initial ANFIS model configuration, including membership functions and fuzzy rules.
2. Optimization: apply the SSO algorithm to fine-tune the ANFIS parameters. Spiders (candidate solutions) interact through attraction and repulsion, ensuring a balance between exploration and exploitation in the optimization process.
3. Evaluation: the ANFIS model's performance in predicting flyrock distance is evaluated after each iteration, and parameters are adjusted accordingly.
4. Output: the optimized ANFIS model, tuned by the SSO algorithm, is expected to provide accurate predictions for flyrock distance, leveraging the SSO algorithm's ability to explore the search space effectively while maintaining diversity.

Each of the hybrid models combines the strengths of ANFIS with those of the respective metaheuristic algorithms to optimize the prediction of flyrock distance induced by mine blasting. The choice of a specific hybrid model can be based on the nature of the dataset (i.e., flyrock dataset in this study), the complexity of the problem, and the desired balance between exploration and exploitation in the optimization process. The flowchart of the last three hybrid models described above for predicting flyrock distance is proposed in Figure 13.

TRAINING AND OPTIMIZATION OF THE ANFIS MODEL FOR PREDICTING BLAST-INDUCED FLYROCK

Before training and optimization of the ANFIS model using the LJ, BA, FA, and SSO algorithms,

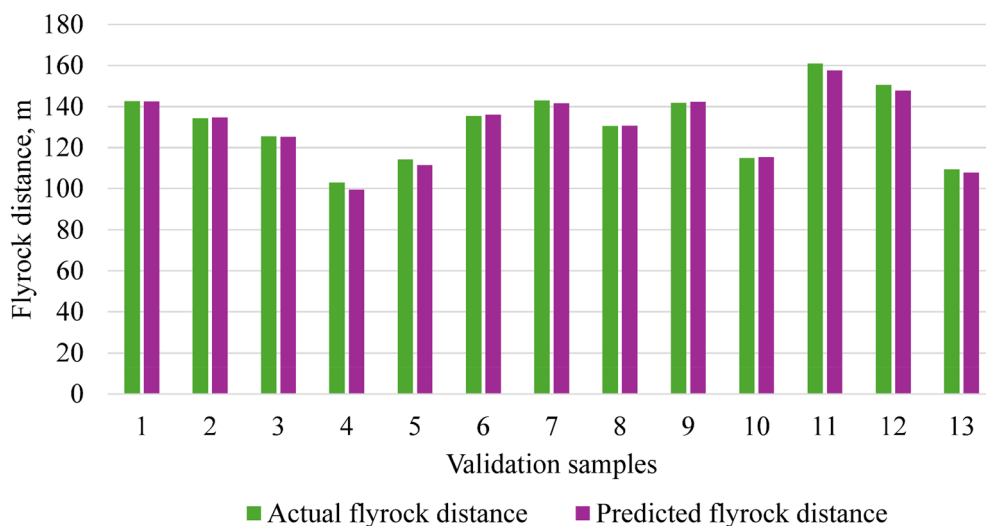


Figure 20. Comparison between actual and predicted flyrock distances by the proposed ANFIS–LJ model in practical engineering.

the dataset was divided into two parts, one with 70% of the whole dataset was used for training and optimization, and the remaining 30% was used for testing separately the trained and optimized models.

The repeated k-fold cross-validation (CV) technique was applied to ensure that AI-based models are not only accurate but also generalize well to new data. It helps in evaluating model performance, tuning hyperparameters, reducing overfitting, and providing a more reliable estimate of how the model will perform in predicting blast-induced flyrock. In this study, fivefold CV was applied during training and optimization of the ANFIS-LJ, ANFIS-BA, ANFIS-FA, and ANFIS-SSO models for predicting blast-induced flyrock (Fig. 14). The fivefold CV was applied with three repeats to (a) help reduce the variance in performance estimates, (b) provide a more reliable and robust evaluation of the model, (c) mitigate the effect of any particular data split, (d) enhance hyperparameter tuning, and (e) allow for a more accurate estimation of the model's performance and confidence intervals. This strategy leads to a more trustworthy assessment of how a model is likely to perform on new, unseen data.

Normalizing the dataset before training AI models is essential for ensuring that all features contribute equally, improving the efficiency and stability of the training process, enhancing the performance of distance-based models and neural networks (e.g., the ANFIS model), avoiding numerical instability, enabling fair comparisons among features, and ultimately improving the model's ability to generalize to new data. In this study, the Min-Max normalization technique was applied in the range of $[-1, 1]$.

The objective function, also known as the loss function or cost function, plays a crucial role in training AI models. It is a mathematical function that a model aims to minimize the error during the training process, and in this study, RMSE was used, thus:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (12)$$

where y_i is the i th actual value of flyrock distance measured (m), $\hat{y}_i$ is the i th predicted value of flyrock distance (m), n stands for the number of blasts measured in this study. The RSME guides the optimization of model parameters, evaluates model performance, aids in hyperparameter tuning, pro-

vides feedback for model adjustments, and ensures consistency with the problem type. By minimizing the objective function, the model learns to make more accurate predictions, ultimately leading to better performance on unseen data.

In this study, fuzzy membership functions were optimized using the LJ, BA, FA, and SSO algorithms with 16 selected rules, each rule base contains four fuzzy memberships. During optimization of fuzzy membership functions, a special objective fitness function based on RMSE and MAPE (mean absolute percentage error), each with a coefficient of 0.5, was applied to evaluate the parameters sets of the metaheuristic algorithms and select the best one, i.e., LJ, BA, FA, and SSO, thus:

$$\text{Fitness}_{\text{algorithms}} = 0.5 \times \text{RMSE} + 0.5 \times \text{MAPE} \quad (13)$$

Once the best set of parameters of the algorithms were selected, they are used to re-train the ANFIS model for predicting blast-induced flyrock. In this step, a new fitness function was applied to select the best model during retraining the models, thus:

$$\begin{aligned} \text{Fitness}_{\text{retrain}} = & 0.1 \times \text{RMSE}_{\text{train}} + 0.1 \times \text{MAPE}_{\text{train}} \\ & + 0.6 \times \text{RMSE}_{\text{test}} + 0.1 \times \text{MAPE}_{\text{test}} \end{aligned} \quad (14)$$

The goal of using Eqs. 13 and 14 was to evaluate the performance of the models for more objective with higher reliability based on multiple metrics. This multifaceted approach helps balance competing goals (e.g., accuracy and interpretability, or precision and recall), mitigating metric bias, highlighting trade-offs between competing priorities (e.g., speed vs. accuracy or complexity vs. performance), and enabling better-informed decisions about model deployment.

Moreover, the use of multiple metrics to evaluate the AI-based models allows suitability for diverse applications, for models applied to multiple scenarios or datasets like predicting flyrock, evaluating with various metrics ensures the model performs adequately across different conditions. Moreover, the use of multiple metrics allows the identification of weak spots in the model, which will be robustness assessment. Last but not least, different stakeholders may value different outcomes. Using multiple metrics helps address diverse requirements, improving the likelihood of model adoption.

As introduced in Figures 12 and 13, after initializing the ANFIS parameters, metaheuristic algorithms were also initialized, including the number of populations, epochs, and other parameters (Table 1). It should be noted that these parameters were optimized and selected using the fitness function in Eq. 13, as mentioned above. Herein, 50 populations were applied with iteration limit of 1000 times. A population size of 50 strikes a balance between exploration (searching a wide solution space) and exploitation (refining the best solutions). Larger population sizes increase exploration but at the cost of higher computational demands, while smaller populations may converge prematurely to suboptimal solutions. In addition, given the computational intensity of evaluating fitness functions for each individual in the population, a population size of 50 was deemed sufficient to ensure effective convergence without imposing excessive computational costs. Subsequently, the candidate solutions (parameter sets for ANFIS model) were randomly generated to configure the ANFIS model for predicting flyrock distance. The metaheuristic algorithms—LJ, BA, FA, and SSO—were applied to reconfigure the ANFIS model with the best parameters. The iterative optimization process of the ANFIS model based on the LJ, BA, FA, and SSO algorithms is shown in Figure 15.

RESULTS AND DISCUSSION

Comparison between Predicted and Real D_{FR}

Based on the trained and optimized results of the proposed models (Fig. 15), it can be seen that the optimization process of the ANFIS-LJ model was superior to those of the other hybrid models under the optimization of BA, FA and SSO algorithms. The convergence plot for the LJ algorithm likely shows rapid initial convergence, which then plateaus as the optimization reaches closer to the optimal solution. The Lévy flight enhancement allowed this algorithm to escape local minima, leading to a more global exploration of the solution space early on. This results in efficient optimization, especially when the problem involves highly nonlinear behavior, as seen in the flyrock prediction. The LJ is expected to show good balance between exploration and exploitation. It can avoid local minima, making it robust for this specific prediction task.

The BA plot may show moderate initial convergence followed by slower, more gradual optimization as it approaches the best solution (Fig. 15). The BA typically uses a balance between frequency tuning and echolocation to explore the solution space, which can lead to more gradual optimization. The BA performed well in medium to complex scenarios but might be slower to converge compared to the LJ.

The FA is usually highly explorative, as suggested by its characteristic random walks based on light intensity attraction. The convergence plot might show a more erratic initial optimization phase, which then stabilizes as the brighter solutions (better-performing solutions) attract other solutions. The FA's optimization performance could show good exploration but may be less efficient in exploitation, potentially requiring more iterations to converge to a fine solution compared to the LJ.

The SSO plot likely reflects a methodical, steady convergence due to its social behavior-based mechanisms. The communication between “spiders” in the population leads to a systematic approach to exploring and exploiting the solution space. The SSO might show strong convergence properties, particularly in exploiting the solution space efficiently after initial exploration. This algorithm is expected to perform well in finding an accurate solution, but it may require more computational time compared to the LJ or FA. Figure 16 shows the predictive results of the models during training processes.

The results in Figure 16a demonstrate that the proposed models learned the existing dataset very well, especially the ANFIS-LJ model, and the prediction trends aligned closely with the actual data, showing strong overall fit. The models, particularly the ANFIS-LJ model, performed exceptionally well, capturing even the peak values with high accuracy (Fig. 16a). The ANFIS-LJ model exhibited the highest accuracy on the training samples (Fig. 14b). While some models slightly overestimated the D_{FR} at the lower values, none of the models predicted D_{FR} values exceeding the actual maximum of 160 m (Fig. 16a). However, Figure 16b clearly shows that the ANFIS-LJ model consistently predicted D_{FR} with excellent accuracy during the training process.

To assess the practical applicability of the developed models in engineering, an independent evaluation was conducted using a testing dataset. In other words, these samples were not used during

model training and represent unseen data. The predicted results of the models on this testing dataset are presented in Figure 17. The results in Figure 17 indicated that the proposed models predicted D_{FR} accurately on the testing dataset, with their prediction trends closely following the actual D_{FR} pattern. Notably, the ANFIS-LJ model achieved the highest accuracy, while the ANFIS-SSO model appeared to introduce some bias in D_{FR} predictions, including a significant outlier (Fig. 17).

Analysis of Errors

To further evaluate the models, Figure 18 illustrates the relationship between predicted and actual D_{FR} values, as well as their distribution and associated errors.

In Figure 18, the absolute and percentage errors were calculated and sorted from the smallest errors to the worst. From this, we can determine the portion of our test set with errors below a certain value or percentage using the first two plots (on the right) (Lares, 2021). The third plot (also on the right) is a density plot showing the actual percentage error, with a reference line at 0, where we can observe the ranges where errors occur most frequently (Lares, 2021). Meanwhile, in the first plot on the left side, we can see how close the predicted and real D_{FR} values are to the reference line of 45° (intercept = 0, slope = 1). Finally, the second plot on the left side presents a straightforward comparison between the actual values and the predicted values. The closer these two curves align, the more accurate the prediction (Lares, 2021).

Looking at Figure 18, we can see that the ANFIS-LJ model showed the best accuracy across the entire dataset of flyrock at the Ta Phoi copper mine. The model aligned closely with the actual D_{FR} values, indicating excellent training and generalization performance on the test data. The ANFIS-BA model also performed well, although it showed slightly higher errors in some instances compared to the ANFIS-LJ model. Its balance between speed and accuracy remained competitive but with minor deviations in lower prediction ranges. The ANFIS-FA model had more dispersed errors, as evident in the density plot. It tended to exhibit greater variability in its D_{FR} predictions, suggesting that it may struggle in certain regions of the dataset, particularly

at extreme values of D_{FR} . The ANFIS-SSO model exhibited some bias in its D_{FR} predictions, with noticeable outliers, as seen in the overall distribution of errors. This model appeared to perform less reliably compared to the others, particularly at certain lower ranges of flyrock distances.

Besides, the comparison between predicted and real D_{FR} values highlights how close the predictions are to the reference line (45° slope). The closer the model's predictions to this line, the better its performance. The ANFIS-LJ displayed the closest fit to this line, reinforcing its strong predictive ability. The ANFIS-SSO, on the other hand, had the largest deviations from the reference line, with more scattered points that indicate higher prediction errors in some cases.

Based on the results in Figure 18, it is conspicuous that the ANFIS-LJ model stood out as the most effective for predicting blast-induced flyrock, with the lowest error rates and highest alignment with actual values, while the ANFIS-SSO model presented the most challenges in terms of bias and error distribution.

Calculation of Performance Metrics and Evaluation

For further assessing the effectiveness of predictive models by quantifying how well a model performs in relation to actual outcomes, performance metrics, including MAE, RMSE, and R^2 were calculated using Eqs. 12, 15, and 16, as they provide objective basis for comparing different models and understanding their strengths and weaknesses (Table 2).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (15)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (16)$$

where y_i is the i th actual value of flyrock distance measured (m), $\hat{y}_i$ is the i th predicted value of flyrock distance (m), $\bar{y}$ denotes the mean of actual values of flyrock distance measured (m), and n stands for the number of blasts measured in this study.

The results in Table 2 show that the ANFIS-LJ model clearly outperformed the other models in all performance metrics for both the training and test-

ing datasets. The high R^2 and MAE and RMSE indicated that the ANFIS-LJ model is highly accurate and generalizes well on unseen data in predicting D_{FR} . This model is best at capturing the relationship between the input parameters and the flyrock distance, making it a reliable choice.

Good generalization occurs when the performance on the testing dataset is similar to that on the training dataset. The ANFIS-LJ model had minimal degradation between training and testing, showing that it generalized well. Its testing MAE and RMSE were slightly lower than the training values, and the R^2 score remained very high. In contrast, the ANFIS-SSO, ANFIS-BA, and ANFIS-FA models exhibited larger differences between training and testing metrics, indicating they may be less robust in predicting new, unseen data.

The ANFIS-LJ did not show significant overfitting or underfitting, as the errors and R^2 values on the training and testing datasets were close. In contrast, the ANFIS-BA, ANFIS-FA, and ANFIS-SSO models displayed lower R^2 values and higher errors, suggesting that they might have either be underfitted the training data or have trouble generalizing to new data. For example, the ANFIS-SSO model had relatively high training errors (MAE = 4.782, RMSE = 6.599) and only slightly improved results on the test set (MAE = 3.748, RMSE = 5.113), which indicates suboptimal model performance.

The results also revealed that the ANFIS-LJ model had an R^2 of 0.974 on the training dataset and 0.981 on the testing dataset, which indicate that over 97–98% of the variance in the data was explained by the model. In comparison, the ANFIS-BA model (R^2 of 0.832 on training and 0.881 on testing), ANFIS-FA model (R^2 of 0.874 on training and 0.861 on testing), and ANFIS-SSO model (R^2 of 0.827 on training and 0.860 on testing) had lower R^2 values, meaning that they explained less of the variance in the dataset and, thus, were less effective at capturing the underlying patterns.

Given that the ANFIS-LJ model achieved the highest accuracy, explained the most variance (highest R^2), and had the lowest errors, it would be the most suitable model for practical applications where reliable prediction of flyrock distance is crucial. Meanwhile, the ANFIS-SSO, ANFIS-BA, and ANFIS-FA models may require further tuning or optimization, as their performance metrics indicate lower suitability for high-precision predictions.

Sensitivity Analysis

The Latin hypercube sampling (LHS) is a popular method for performing sensitivity analysis, particularly in scenarios where the model is expensive to evaluate, or the parameter space is large (Iman, 2008). It ensures that the entire range of possible inputs is sampled effectively. The results from LHS are generally robust, especially when the sample size is sufficiently large, as it provides a comprehensive exploration of the input space. In this study, the LHS was applied to analyze the sensitivity of the input variables used in the best model selected, i.e., ANFIS-LJ model.

In LHS, each input variable is divided into equal probability intervals based on the desired number of samples (N). If N samples are used, the range of each variable is split into N intervals. For each input variable, a single sample is randomly selected from each interval. The sampling ensures that each interval is only sampled once, so the sample set covers the full range of each input variable. The values sampled from the intervals of different input variables are combined randomly to generate a multidimensional set of input samples. This creates an N -sample matrix where each row represents a unique combination of sampled inputs. The model is evaluated for each combination of input samples. This provides N model outputs corresponding to the N sets of input variables. The mathematical principle of this algorithm is described as follows:

- Partition the range of each variable: for each input variable X_i , the range $[X_{\min}, X_{\max}]$ is divided into N intervals of equal probability. For example, if X_i is uniformly distributed, the intervals will be $[X_{\min} + (k - 1)\Delta X, X_{\min} + k\Delta X]$, where $\Delta X = \frac{(X_{\max} - X_{\min})}{N}$ and $k = 1, 2, \dots, N$.
- Random sampling within intervals: from each interval, a single random value $X_{i,k}$ is drawn, ensuring that every interval has one sample point, but samples are not repeated.
- Permutate the sampled points: the sampled points for different variables are combined into unique sets by permuting their indices. This ensures the independence of the input variables and creates diverse combinations of input values.
- Generate output samples: each combination of input values (row in the sample matrix) is used to evaluate the model, resulting in a corresponding output for each sample set.

In this study, four variables were used to develop the blast-induced flyrock distance models, namely S, MEC, ST, and PF, as introduced earlier. The sensitivity analysis results of these variables based on the best selected model (i.e., ANFIS-LJ) are shown in Figure 19.

The results indicate that the B variable had the highest sensitivity values (0.3057–34.5%), suggesting it was the most influential input variable in the model according to the LHS method. The EC variable was also highly influential, with sensitivity values of 0.2789–31.5%, indicating that it significantly impacted the blast-induced flyrock distance of the ANFIS-LJ model. The PF variable had sensitivity values of 0.2678–30.3%, making it another important variable but slightly less influential than S and MEC. Finally, the ST variable had very low sensitivity values of 0.0327–3.7%, implying that it had minimal impact on the model's output.

The LHS method results showed that the B was the most significant variable, followed closely by EC and PF. ST was much less significant in this analysis. The relative importance of the variables according to LHS was thus $B > EC > PF > ST$.

PRACTICAL VALIDATION OF THE MODEL IN ENGINEERING APPLICATIONS

To enhance the model's reliability and applicability in practical engineering, it was validated using real-world blasting data with 13 blasting events (Table 3). The final model, ANFIS-LJ, was tested on a validation dataset representing new blasting events, with the outcome predictions shown in Figure 20 and performance metrics calculated to evaluate its predictive accuracy. The validation results were: MAE = 1.322; RMSE = 1.825; $R^2 = 0.988$.

The results in Figure 20 exhibit that the ANFIS-LJ model provided promising results with high accuracy in practical engineering at the Ta Phoi copper mine. The ANFIS-LJ model achieved high R^2 of 0.988, indicating excellent agreement between predicted and observed flyrock distances. Besides, the low MAE (1.322) and RMSE (1.825) values confirmed the model's precision and reliability in predicting flyrock distances. The outstanding performance metrics validated the model's ability to predict flyrock distances under varying blast conditions accurately. These results confirmed its potential as a valuable tool for optimizing blast designs,

enhancing safety, and reducing environmental impacts in real-world engineering applications.

CONCLUSIONS

We developed and assessed four hybrid ANFIS models—ANFIS-LJ, ANFIS-BA, ANFIS-FA, and ANFIS-SSO—for predicting blast-induced flyrock in an open-pit mining environment. The models were evaluated using several performance metrics, including MAE, RMSE, and R^2 , on both training and testing datasets. The results indicate that the ANFIS-LJ model outperformed all the other models, demonstrating the highest accuracy, the lowest prediction errors, and excellent generalization to unseen data.

Specifically, the ANFIS-LJ model achieved an MAE of 1.423 and RMSE of 1.895 on the testing dataset, with R^2 value of 0.981, indicating that it explained over 98% of the variance in the flyrock distance predictions. This model consistently produced the most accurate results and aligned closely with the actual values, making it the most reliable for practical engineering applications in the mining sector. In contrast, the ANFIS-SSO model exhibited the poorest performance, with higher MAE and RMSE values and the lowest R^2 scores, suggesting it is less suitable for precise predictions.

The performance analysis of ANFIS-BA and ANFIS-FA models revealed moderate accuracy, but their errors were notably higher than those of ANFIS-LJ, and they exhibited weaker generalization capabilities. While these models could still be useful in certain contexts, they would benefit from further optimization to improve their accuracy and robustness.

Overall, the findings highlight the importance of selecting the appropriate optimization algorithm for hybrid ANFIS models. Among the models tested, the ANFIS-LJ hybrid proved to be the most effective, making it an excellent choice for accurately predicting blast-induced flyrock distances in open-pit mining. This model could be applied in real-world engineering scenarios to enhance safety measures and operational efficiency, while minimizing the risk of flyrock-related accidents.

Future work could focus on further refining the optimization algorithms, testing the models in different mining environments, and incorporating additional parameters to improve the predictive capabilities of these models even further.

ACKNOWLEDGMENTS

This paper was supported by the Ministry of Education and Training (MOET) in Viet Nam under grant number B2023-MDA-04.

DECLARATIONS

Conflict of Interest We, the authors, claimed that we have no conflicts of interest.

REFERENCES

- Andre, D. (2024). What is Neuro Fuzzy? <https://www.allaboutai.com/ai-glossary/neuro-fuzzy/>. Accessed 22.10.2024.
- Armaghani, D. J., Mahdiyar, A., Hasanipanah, M., Faradonbeh, R. S., Khandelwal, M., & Amnieh, H. B. (2016a). Risk assessment and prediction of flyrock distance by combined multiple regression analysis and Monte Carlo simulation of quarry blasting. *Rock Mechanics and Rock Engineering*, 49(9), 3631–3641. <https://doi.org/10.1007/s00603-016-1015-z>.
- Armaghani, D. J., Tonnizam Mohamad, E., Hajiassani, M., Alavi Nezhad Khalil Abad, S., Marto, A., & Moghaddam, M. (2016b). Evaluation and prediction of flyrock resulting from blasting operations using empirical and computational methods. *Engineering with Computers*, 32, 109–121.
- Armaghani, D. J., Tonnizam Mohamad, E., Hajiassani, M., Alavi Nezhad Khalil Abad, S. V., Marto, A., & Moghaddam, M. R. (2016c). Evaluation and prediction of flyrock resulting from blasting operations using empirical and computational methods. *Engineering with Computers*, 32(1), 109–121. <https://doi.org/10.1007/s00366-015-0402-5>.
- Asl, P. F., Monjezi, M., Hamidi, J. K., & Armaghani, D. J. (2017). Optimization of flyrock and rock fragmentation in the Tajareh limestone mine using metaheuristics method of firefly algorithm. *Engineering with Computers*, 1–11.
- Babaeian, M., Sereshki, F., Ataei, M., Nehring, M., & Mohammadi, S. (2023). Application of soft computing, statistical and multi-criteria decision-making methods to develop a predictive equation for prediction of flyrock distance in open-pit mining. *Mining*, 3(2), 304–333. <https://doi.org/10.3390/mining3020019>.
- Bakhtavar, E., Nourizadeh, H., & Sahebi, A. (2017). Toward predicting blast-induced flyrock: A hybrid dimensional analysis fuzzy inference system. *International Journal of Environmental Science and Technology*, 14(4), 717–728.
- Barkhordari, M. S., Armaghani, D. J., & Fakharian, P. (2022). Ensemble machine learning models for prediction of flyrock due to quarry blasting. *International Journal of Environmental Science and Technology*, 19(9), 8661–8676. <https://doi.org/10.1007/s13762-022-04096-w>.
- Barthelemy, P., Bertolotti, J., & Wiersma, D. S. (2008). A Lévy flight for light. *Nature*, 453(7194), 495–498.
- Bhatawdekar, R. M., Armaghani, D. J., Azizi, A., Bhatawdekar, R. M., Armaghani, D. J., & Azizi, A. (2021). Applications of AI and ML techniques to predict backbreak and flyrock distance resulting from blasting. *Environmental Issues of Blasting: Applications of Artificial Intelligence Techniques*, 41–59.
- Cuevas, E., Cienfuegos, M., Rojas, R., & Padilla, A. (2015). A computational intelligence optimization algorithm based on the behavior of the social-spider. *Computational Intelligence Applications in Modeling and Control*, 123–146.
- Emary, E., Zawbaa, H. M., & Sharawi, M. (2019). Impact of Lévy flight on modern meta-heuristic optimizers. *Applied Soft Computing*, 75, 775–789.
- Faradonbeh, R. S., Armaghani, D. J., Amnieh, H. B., & Mohamad, E. T. (2018). Prediction and minimization of blast-induced flyrock using gene expression programming and firefly algorithm. *Neural Computing and Applications*, 29, 269–281.
- Faradonbeh, R. S., Jahed Armaghani, D., & Monjezi, M. (2016). Development of a new model for predicting flyrock distance in quarry blasting: A genetic programming technique. *Bulletin of Engineering Geology and the Environment*, 75, 993–1006.
- Fister, I., Fister, I., Jr., Yang, X.-S., & Brest, J. (2013). A comprehensive review of firefly algorithms. *Swarm and Evolutionary Computation*, 13, 34–46.
- Ghasemi, E., Amini, H., Ataei, M., & Khalokakaei, R. (2014). Application of artificial intelligence techniques for predicting the flyrock distance caused by blasting operation. *Arabian Journal of Geosciences*, 7(1), 193–202. <https://doi.org/10.1007/s12517-012-0703-6>.
- Ghasemi, E., Sari, M., & Ataei, M. (2012). Development of an empirical model for predicting the effects of controllable blasting parameters on flyrock distance in surface mines. *International Journal of Rock Mechanics and Mining Sciences*, 52, 163–170.
- Ghojoghi, E., Ebrahimi Farsangi, M. A., Mansouri, H., & Rashtedi, E. (2024). Prediction and minimization of blasting flyrock distance, using deep neural networks and gravitational search algorithm, JAYA, and multi-verse optimization algorithms. *Heliyon*. <https://doi.org/10.1016/j.heliyon.2024.e37876>.
- Hakli, H., & Uğuz, H. (2014). A novel particle swarm optimization algorithm with Levy flight. *Applied Soft Computing*, 23, 333–345.
- Han, H., Jahed Armaghani, D., Tarinejad, R., Zhou, J., & Tahir, M. M. (2020). Random forest and Bayesian network techniques for probabilistic prediction of flyrock induced by blasting in quarry sites. *Natural Resources Research*, 29(2), 655–667. <https://doi.org/10.1007/s11053-019-09611-4>.
- Hasanipanah, M., Jahed Armaghani, D., Bakhshandeh Amnieh, H., Majid, M. Z. A., & Tahir, M. M. (2017). Application of PSO to develop a powerful equation for prediction of flyrock due to blasting. *Neural Computing and Applications*, 28, 1043–1050.
- Iman, R. L. (2008). *Latin hypercube sampling*. John Wiley & Sons Ltd.
- Jain, A., Sharma, S., & Sharma, S. (2021). Firefly algorithm. *Nature-Inspired Algorithms Applications*, 157–180.
- Jamei, M., Hasanipanah, M., Karbasi, M., Ahmadianfar, I., & Taherifar, S. (2021). Prediction of flyrock induced by mine blasting using a novel kernel-based extreme learning machine. *Journal of Rock Mechanics and Geotechnical Engineering*, 13(6), 1438–1451. <https://doi.org/10.1016/j.jrmge.2021.07.007>.
- James, J., & Li, V. O. (2015). A social spider algorithm for global optimization. *Applied Soft Computing*, 30, 614–627.
- Jamil, M., & Zepernick, H.-J. (2013). Lévy flights and global optimization. In *Swarm intelligence and bio-inspired computation* (pp. 49–72): Elsevier.
- Lares, B. (2021). Machine Learning Results in R: one plot to rule them all! (Part 2 – Regression Models). <https://datascienceplus.com/machine-learning-results-in-r-one-plot-to-rule-them-all-part-2-regression-models/>. Accessed 22.10.2024.
- Law, P. O. T. (2024). Company's blasting operation sends rocks flying into homes: All mining activities temporarily suspended (In Vietnamese). <https://baovephapluat.vn/kinh-te/d-o-thi-xay-dung/vu-doanh-nghiep-no-min-lam-da-vang-vao-nh>

- a-dan-tam-dung-moi-hoat-dong-khai-thac-mo-155572.html. Accessed 04/09/2024.
- Li, C., Zhou, J., Du, K., Armaghani, D. J., & Huang, S. (2023). Prediction of flyrock distance in surface mining using a novel hybrid model of Harris Hawks optimization with multi-strategies-based support vector regression. *Natural Resources Research*, 32(6), 2995–3023. <https://doi.org/10.1007/s11053-023-10259-4>.
- Lu, X., Hasanipناه, M., Brindhadevi, K., Bakhshandeh Amnieh, H., & Khalafi, S. (2020). ORELM: A novel machine learning approach for prediction of flyrock in mine blasting. *Natural Resources Research*, 29, 641–654.
- Manoj, K., & Monjezi, M. (2013). Prediction of flyrock in open pit blasting operation using machine learning method. *International Journal of Mining Science and Technology*, 23(3), 313–316.
- Momeni, E., Armaghani, D. J., & Azizi, A. (2023). *Artificial Intelligence in Mechatronics and Civil Engineering: Bridging the Gap*. Springer Nature.
- Monjezi, M., Mehrdaneh, A., Malek, A., & Khandelwal, M. (2013). Evaluation of effect of blast design parameters on flyrock using artificial neural networks. *Neural Computing and Applications*, 23(2), 349–356.
- Platt, G. M., Yang, X.-S., & Neto, A. J. S. (2018). *Computational intelligence, optimization and inverse problems with applications in engineering*. Springer.
- Plus, E. (2024). Thanh Hoa: Rocks Fly into Residential Homes During Quarry Blasting (In Vietnamese). <https://moitruongplus.com.vn/11/17743/Thanh-Hoa-Da-bay-vao-nha-dan-khi-n-o-min-khai-thac-mo-da.htm>. Accessed 04/09/2024.
- Protodiakonov, M., Koifman, M., Chirkov, S., Kuntish, M., & Tedder, R. (1964). *Rock strength passports and methods for their determination*. Nauka.
- Rad, H. N., Bakhshayeshi, I., Jusoh, W. A. W., Tahir, M., & Foong, L. K. (2020). Prediction of flyrock in mine blasting: A new computational intelligence approach. *Natural Resources Research*, 29, 609–623.
- Raina, A. K., Murthy, V. M. S. R., & Soni, A. K. (2015). Flyrock in surface mine blasting: understanding the basics to develop a predictive regime. *Current Science*, 660–665.
- Rao, R. V. (2019). Jaya: an advanced optimization algorithm and its engineering applications.
- Shakeri, J., Bascompta, M., Alimoradjazi, M., & Dehghani, H. (2023). Application of artificial intelligence techniques for predicting the flyrock, Sungun mine, Iran. *Arabian Journal of Geosciences*, 16(8), 488. <https://doi.org/10.1007/s12517-023-11561-4>.
- Singh, A., Meyyazhagan, A., & Verma, S. (2022). Nature-inspired computing: bat echolocation to BAT algorithm. In *Nature-Inspired Intelligent Computing Techniques in Bioinformatics* (pp. 163–174), Springer.
- Taiwo, B. O., Fissaha, Y., Hosseini, S., Khishe, M., Kahraman, E., Adebayo, B., et al. (2024). Machine learning based prediction of flyrock distance in rock blasting: A safe and sustainable mining approach. *Green and Smart Mining Engineering*, 1(3), 346–361. <https://doi.org/10.1016/j.gsme.2024.08.006>.
- Trivedi, R., Singh, T., & Raina, A. (2016). Simultaneous prediction of blast-induced flyrock and fragmentation in opencast limestone mines using back propagation neural network. *International Journal of Mining and Mineral Engineering*, 7(3), 237–252.
- Ullah, I., & Hussain, S. (2019). Time-constrained nature-inspired optimization algorithms for an efficient energy management system in smart homes and buildings. *Applied Sciences*, 9(4), 792. <https://doi.org/10.3390/app9040792>.
- Vinacomin, T. P. C. J. S. C. (2014b). Project description and basic design.
- Vinacomin, T. P. C. J. S. C. (2014a). Geological report of the mine area.
- Wang, R., Zhou, M., Wang, J., & Gao, K. (2023a). An improved discrete Jaya algorithm for shortest path problems in transportation-related processes. *Processes*, 11(8), 2447.
- Wang, X., Hosseini, S., Armaghani, D. J., & Mohamad, E. T. (2023b). Data-driven optimized artificial neural network technique for prediction of flyrock induced by boulder blasting. *Mathematics*, 11(10), 2358. <https://doi.org/10.3390/math11102358>.
- Wu, L., Wu, J., & Wang, T. (2023). Enhancing grasshopper optimization algorithm (GOA) with levy flight for engineering applications. *Scientific reports*, 13(1), 124.
- Yang, X. -S. Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms*, 2009 (pp. 169–178), Springer.
- Yang, X. -S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65–74), Springer.
- Yang, X. -S., & He, X. (2016). *Nature-inspired computation in engineering* (Vol. 637), Springer.
- Yu, K., Liang, J., Qu, B., Chen, X., & Wang, H. (2017). Parameters identification of photovoltaic models using an improved JAYA optimization algorithm. *Energy Conversion and Management*, 150, 742–753.
- Zhou, J., Koopialipoor, M., Murlidhar, B. R., Fatemi, S. A., Tahir, M., Jahed Armaghani, D., et al. (2020). Use of intelligent methods to design effective pattern parameters of mine blasting to minimize flyrock distance. *Natural Resources Research*, 29, 625–639.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.