RESEARCH ARTICLE

Assessment of Selected Machine Learning Models for Intelligent Classification of Flyrock Hazard in an Open Pit Mine

IAN KROP[1,2], **YOSHIAKI TAKAHASHI**[3], **TAKASHI SASAOKA**[1], **HIDEKI SHIMADA**[1], **AKIHIRO HAMANAKA**[1], **AND JOAN ONYANGO**[1,2]

[1]Rock Engineering and Mining Machinery Laboratory, Department of Earth Resources Engineering, Kyushu University, Fukuoka 819-0395, Japan
[2]Department of Mining, Materials and Petroleum Engineering, Jomo Kenyatta University of Agriculture and Technology, Nairobi 62000-00200, Kenya
[3]Research Institute of Science for Safety and Sustainability, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba 305-8569, Japan

Corresponding author: Akihiro Hamanaka (hamanaka@mine.kyushu-u.ac.jp)

ABSTRACT This paper presents an alternative methodology for the study of flyrock hazards in mining, utilizing Artificial Intelligence (AI) through machine learning by classification. By using distance as a delineator to denote the consequences of a blast, the models generated two classes of blasts: safe and unsafe. In this study, statistical learning models best suited for classification, that is, K Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree (DT), and Artificial Neural Networks (ANNs), were used, and their classification abilities were assessed. Machine performance was evaluated using a Confusion Matrix (sensitivity and specificity) and Receiver Operating Characteristic (ROC) curve. A higher weight was assigned to the minority class (unsafe blasts). Overfitting assessment was also performed. The Wide Neural Network (WNN) demonstrated the highest classification superiority. During training and validation, 75% sensitivity, 100% specificity, and an ROC of 0.9853 were achieved. In the test phase, perfect stratification (100 %) was maintained, with an ROC of 1. The Cubic SVM exhibited 50% sensitivity, 100% specificity, and an ROC of 0.9412 during training and validation. In the test set, it achieved 100% sensitivity, 100% specificity, and a ROC of 1. Fine KNN showed 50% sensitivity, 94.1% specificity, and an ROC of 0.7206 in the validation set. The test set displayed 100% sensitivity, 100% specificity, and an ROC of 1. Conversely, Coarse DT had a higher misclassification rate, resulting in a 25% sensitivity, 76.5% specificity, and an ROC of 0.5221 during the validation phase. In the test set, it showed 50% sensitivity, 100% specificity, and an ROC of 0.75. A feedforward neural network (FNN) was designed, trained, and demonstrated to be a highly flexible classification tool. The FNN achieved an excellent classification score of 100%. These findings demonstrate the potential for the broad applicability of machine learning through classification in addressing flyrock challenges in open-pit mines.

INDEX TERMS Algorithm, artificial intelligence, classification, flyrocks, machine learning, neural networks, prediction, regression, rock blasting.

I. INTRODUCTION

Recently, Machine Learning (ML) has received significant attention in science and engineering. This is because of their ability to improve the quality of human life [1] by solving complex problems associated with different phenomena [2].

Machine learning (ML) is a subfield of artificial intelligence (AI) dedicated to training machines to automatically learn from data and past experiences [3] without the need for explicit programming. This process involves identifying patterns to make predictions, with minimal human intervention. ML employs various statistical learning methods to improve the performance of machines over time.

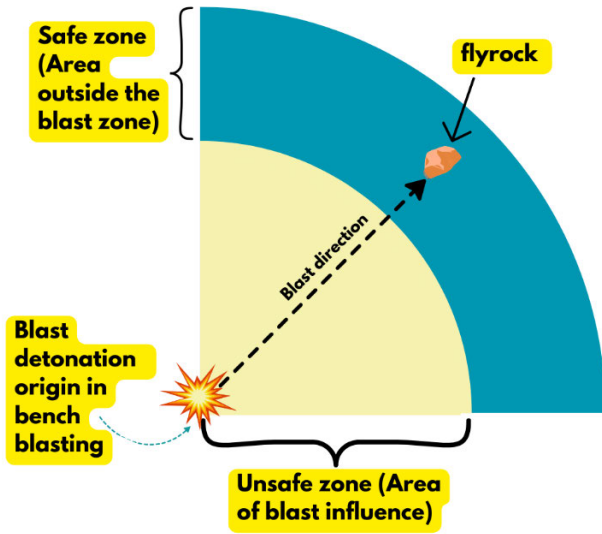ML has wide application in various fields [4]. In the mining industry, the deployment of intelligent machines has

The associate editor coordinating the review of this manuscript and approving it for publication was Alberto Cano.

FIGURE 1. Blast zone delineation.

promoted efficiency and productivity, which are crucial to profitability. One of the key safety concerns in mineral exploitation is the generation of flyrock from poor blast designs. As depicted in Fig. 1, flyrocks are rock fragments from a blast propelled beyond the limits of the blast area [5]. This is an undesirable occurrence in blasting operations owing to its adverse environmental impact. These rock fragments can cause significant damage to infrastructure such as machinery, buildings, and other equipment. It can also cause minor to severe injuries. Additionally, fatalities have been reported [6].

Flyrock is a complex phenomenon [7]. Bajpayee et al. postulated that this arises because of a mismatch between the mechanical strength of the rock mass, confinement of the charge in a blast hole, and distribution of the explosive energy [8]. Other investigations [9], [10] point to controllable parameters such as the use of inappropriate blast patterns, incorrect burdens, improper stemming lengths, excessive powder factors, and ill-suited delay times. In addition, uncontrollable parameters, such as density, porosity, Uniaxial Compressive Strength (UCS), primary wave and secondary wave velocities, and geological conditions, such as rock mass discontinuities, have been noted as the most probable causative agents.

Traditional methods such as the use of empirical techniques have proven cumbersome (technically difficult to perform), unreliable, time-consuming, and prone to large errors [11]. These pitfalls have led to a rapid interest in ML technology, which, when trained properly, can be fast, reliable, and can accommodate many parameters. The possibility of applying ML interventions to flyrock problems is of increasing interest. Therefore, several studies on flyrock distance prediction have focused on the use of AI through ML to solve flyrock-associated problems. For example,

Monjezi et al. showed that an Artificial Neural Network (ANN) was effective in predicting flyrock at the Sangan iron mine in Iran [12]. Hasanipanah et al. found that the Genetic Algorithm (GA) model had high accuracy compared to the Imperialist Competitive Algorithm (ICA)-and Particle Swarm Optimization (PSO)-based models [13]. Marto et al. demonstrated that ICA – ANN was a more accurate model than the Back Propagation-ANN [14]. Trivedi et al. proved that an ANN using Back Propagation was a superior predictive tool than the multivariate regression analysis (MVRA) tool [15]. Jamei et al. compared Kernel Extreme Learning Machine (KELM) with Response Surface Methodology (RSM), Boosted Regression Tree (BRT) and Local Weighted Linear Regression (LWLR) and discovered KELM to have good predictive capability since it was computationally efficient [16]. These studies were aimed at flyrock distance predictions and how ML can inform proper decision-making that minimizes or eliminates flyrock hazards.

ML can be used to solve problems via regression or classification. Regression models output continuous variables. However, for classification models, there are a discrete number of possible outcomes [17]. For an imbalanced dataset (data with unequal distribution of classes), which is usually the case for flyrocks, regression solvers, especially linear solvers, have been found to be less than ideal as they are highly sensitive. Machine learning through classification offers an alternative approach for understanding flyrock problems more realistically than the regression approach. For example, Hudaverdi and Akyildiz used a novel classification approach, Multiple Discriminant Analysis (MDA), to group the blasts. The study estimated the severity of the flyrock rather than estimating a numerical value (flyrock distance). They noticed that even though their regression model exhibited high correlation coefficients, some of those regression estimations had huge deviances of more than 10 -20 metres; therefore, they could not be relied upon because accuracy was as crucial as it could mean, in extreme cases, the difference between life and death. Their study demonstrated that flyrock throw distance can be presented using categorical variables rather than numerical values [18].

As outlined in Table 1, many recent studies have focused on how ML can solve regression problems for safe blast outcomes. However, few studies have focused on how ML through classification can be utilized to address the negative impacts of flyrocks. Therefore, as an alternative approach, more work is required to demonstrate the potential of using classification models in the study of flyrock generation and mitigation.

The aim of this paper to illustrate the capability of ML classification in flyrock control through a case study. In this study, the classification was based on a predetermined distance criterion that delineated the blast-clearance zone. The goal was to classify, on the premise of distance, a set of blasts in an open pit mine into 'safe' and 'unsafe' blast denoting their severity. A safe blast in the context of this study was one

whose blast-induced fragment throw was less than 90 m. Any throw distance from 90 m onwards would result in a flyrock and therefore the blast would be considered 'unsafe' since it posed the highest risks. Thus, the primary objective was to use different ML methods to assess classification performance. This comparative study determined the best classifier based on its superior performance. Based on this ML approach, this study describes the selected machine learning methods: K Nearest neighbors (KNN), Support Vector Machines (SVM), Artificial Neural Networks (ANNs) and Decision Tree (DT). This paper shows that classification can be relied upon to solve flyrock-related problems. To the best of our knowledge, this is the first study that has focused purely on classification techniques to characterize blast safety with reference to flyrock generation. Moreover, it has considered rock mass conditions, an aspect recommended by a number of studies [19], [20] owing to insufficient attention in past studies.

II. MATERIALS AND METHODS

This section discusses the data acquisition process and the learning methods used in this study, that is, the DT, KNN, SVM, and ANNs.

A. MATERIALS

The current investigation involved sampling and analysis of blast data from an open-pit gold mine in the southern part of Kyushu Island, Japan. A total of 29 experimental blasts conducted in the mine formed the dataset used in this study. The blast design parameters are presented in Table 2. To examine the relationship between the input variables and the output variable (flight distance) for each blast, seven blast design parameters were chosen for this study. They included the burden, stemming length, Brazilian Tensile Strength (BTS), Uniaxial Compressive Strength (UCS), rock density, powder factor, and crack density of the rock mass. Crack density is the number of joints found within a 1 metre-traverse of the bench face, as shown in Fig. 3.

The dataset consisted of blast geometry, explosive parameters, and rock mass properties. The majority of other studies have rarely incorporated adequate rock parameters in their study. However, in this study, a significant number of key rock parameters, such as rock density, BTS, UCS, and crack density, were investigated.

After analyzing the collected blast data, a 90 m boundary was chosen. This decision was aimed at creating an imbalanced dataset, ensuring sufficient data for training in machine learning, while leaving an ample amount for testing. An 'unsafe' blast (represented by 0) denotes a flyrock distance of 90 m and above, presenting the highest risk to infrastructure and potential harm to personnel in the mine. On the other hand, a 'safe' blast (represented by 1) has a flyrock throw length below 90 m. This two-class grouping was achieved through binning using Python in the Pandas library, consequently transforming it into a binary classification problem.

TABLE 1. Examples of some of the most recent research on flyrock problem using ML.

Reference	Algorithm	Type of model
Murlidhar et al. (2023)	EO-ELM, PSO-ANN, PSO-ELM	Regression
Murlidhar et al. (2021)	RF, SVM, WOA-MLP, HHO-MLP	Regression
Jamei et al (2021)	KELM, LWLR, BRT	Regression
Lu et al (2019)	ELM, ORELM, MLR, ANN	Regression
Rad et al (2019)	RFNN-GA, GA-ANN, ANN	Regression
Hudaverdi & Akyildiz (2019)	MRA	Regression but classified the predicted flyrock values
Hasanipanah (2019)	GA, PSO, ICA	Regression
Hasanipanah (2016)	PSO, MLR	Regression
Trivedi et al (2016)	MVRA, BPNN	Regression
Faradonbeh et al (2016)	GP, GEP	Regression
Raina et al (2015)	ANN	Regression
Marto et al (2014)	ICA-ANN, BP-ANN	Regression
Trivedi et al. (2014)	ANN, MVRA	Regression
Monjezi et al. (2010)	MVRA, GA-ANN	Regression
Rezaei et al (2010)	Mamdani Fuzzy	Regression
Monjezi et al (2009)	ANN	Regression

Rock blasting is a rapid occurrence; hence, a proper qualitative analysis of any blast is not possible with the aid of the naked eye. Recently, high-speed cameras have gained popularity because of their fast capture speed, high resolution, and light sensitivity. They can also be used to obtain quantitative measurements, such as the flyrock direction and velocity. In this study, a high-speed camera (Vision Research, Phantom Ver.7.3) was set up perpendicular to the face direction, and the blast footage was recorded at a frame rate of
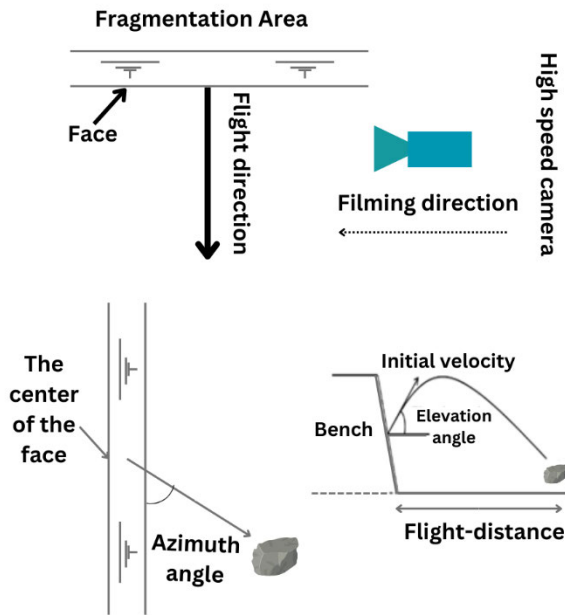
FIGURE 2. The experimental setup.

1000fps. A 2-metre white line, serving as a scale, was marked perpendicular to the bench face to aid in image analysis. For each blast, 12 trackable blast-induced rock fragments were analyzed using Phantom Camera Control (PCC) software. The average velocity of each fragment was computed and determined as the 'initial velocity' of the fragment. The 'maximum initial velocity' was defined as the largest 'initial velocity' among the analyzed fragments. The experimental setup is shown in Fig. 2.

To investigate the influence of rock mass conditions, a digital camera was used to capture the joints on the bench face prior to each blast. A 1-metre traverse across the entire face was established, as shown in Fig. 3.

An overview of the key stages of this study is presented in Fig. 3. The data generated by the research are presented in Table 4, and the statistical summary of the dataset is presented in Table 5.

TABLE 2. Standard blast design.

Parameter	Value	Parameter	Value
Spacing (m)	1.5	Hole length (m)	12
Borehole diameter (mm)	76	Powder factor (g/t)	170
Bench height (m)	10	Charge quantity (kg/hole)	23.7
Bench angle (°)	80	Drilling angle (°)	80

The powder factor, stemming length, and burden were used to generate a prediction equation for the flight distance of fragmented rocks. This was due to their lower p-values compared to the other parameters, which had higher p values upwards of 0.5 (statistically insignificant). The high p-values for rock density (0.50), BTS (0.51), UCS (0.74), and crack density (0.99) rendered them insignificant parameters for predictive value; thus, for the sake of the equation, these parameters were omitted. A significance F of 0.008886, which is below the recommended p-value of 0.05, inferred that the selected features were statistically significant.

TABLE 3. Regression statistics.

Multiple correlation coefficient		0.60
sample number		29
Significance F		**0.008886**
Powder factor (kg/t)	Coefficient	297.13
	P-Value	0.0042
Stemming length (m)	Coefficient	5.05
	P-Value	0.0318
Burden (m)	Coefficient	4.92
	P-Value	0.2620
Intercept	Coefficient	-53.24
	P-Value	0.0425

Multiple regression analysis, summarized in Table 3, was conducted to determine the influence of the selected parameters on the initial velocity (V_o). Thus, the linear equation of the initial velocity (V_o) is presented.

$$V_o = 297.13 \, \mathrm{PF} + 5.05 \, \mathrm{SL} + 4.92 \, \mathrm{B} - 53.24 + \varepsilon \quad (1)$$

where V_o is the initial velocity in m/s, PF is Powder Factor in kg/t, B is Burden in metres, SL is the stemming length in metres and ε is the error term.

The initial average velocity (μ) was determined to be the average of the measured maximum initial velocities which were estimated by comparing the distance traveled by the fragmented rocks after 1.0 ms with the scale. The travel distance of the fragmented rocks was calculated every 50 ms from the initiation time up to 600 ms.

Using the three-sigma (3σ) rule on the assumption that the measured maximum initial velocity could be expressed with a normal distribution curve, the potential maximum initial velocity(V_{max}) was calculated using Equation 2 below.

$$V_{max} = \frac{\mu + 3\sigma}{\mu} V_0 \quad (2)$$

where μ is the average initial velocity and σ is the associated standard deviation. In order to incorporate the riskiest of
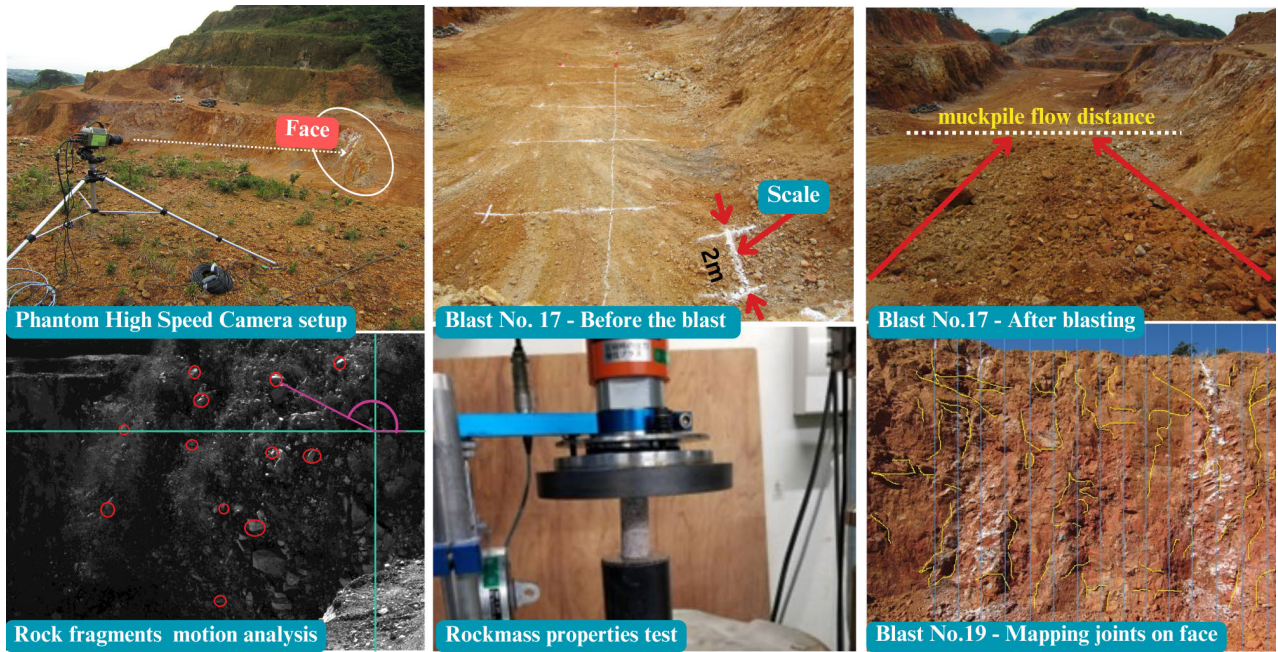
FIGURE 3. The critical stages of the research.

FIGURE 4. Correlation matrix of the blast dataset.

scenarios, as in the case of typical hazard analysis, a 3 σ was adopted.

Thus, the maximum flight distance of the fragmented rocks became a cross product of the horizontal component of the

TABLE 4. Data set is presented.

Blast No.	Stemming Length (m)	Burden (m)	Powder Factor (kg/t)	Crack density (cracks/m)	BTS (MPa)	Rock density (kg/m³)	Uniaxial Compressive Strength (MPa)	Flight distance (m)	Safety Level
1	3.96	1.5	0.141	1.95	16.8	2376	135.5	97.53	0
2	2.85	1.5	0.149	0.75	16.1	2327	137.6	102.14	0
3	2.57	2.0	0.120	0.44	21.6	2321	108.1	11.14	1
4	2.16	2.0	0.130	0.94	9.7	2283	147.7	14.13	1
5	3.96	2.0	0.149	0.99	19.7	2311	187.3	228.80	0
6	2.16	2.0	0.150	0.62	7.8	2190	63.9	64.81	1
7	2.37	2.0	0.150	0.57	17.5	2333	125.0	198.45	0
8	2.16	2.0	0.160	1.71	6.0	2377	121.8	87.62	1
9	2.16	2.0	0.161	2.54	3.1	1970	47.0	77.13	1
10	1.67	2.0	0.170	1.08	14.1	2363	175.3	94.63	0
11	1.65	2.0	0.170	0.78	11.5	2247	130.2	107.34	0
12	2.75	2.5	0.120	0.31	17.8	2567	237.5	35.99	1
13	2.05	2.5	0.131	1.69	14.1	2359	256.6	26.94	1
14	2.05	2.5	0.131	1.04	13.6	2545	304.0	24.09	1
15	2.00	2.5	0.131	0.44	8.0	1505	81.8	26.15	1
16	2.09	2.5	0.133	1.77	9.5	2492	223.1	23.72	1
17	1.56	2.5	0.150	0.93	14.4	2315	163.9	35.05	1
18	1.19	2.5	0.150	0.78	8.3	2266	145.3	24.80	1
19	1.62	3.0	0.130	0.37	14.0	2269	119.0	19.92	1
20	2.00	2.5	0.139	0.38	10.7	2357	123.2	35.68	1
21	2.00	2.5	0.139	0.40	13.5	2260	134.6	61.28	1
22	2.00	2.5	0.139	0.18	11.2	2297	73.1	41.12	1
23	2.00	2.5	0.139	0.51	26.0	2024	158.6	56.26	1
24	2.00	2.5	0.139	0.55	19.5	1613	79.1	40.06	1
25	2.00	2.5	0.139	0.29	9.2	1703	34.4	29.16	1
26	2.00	2.5	0.139	0.33	3.6	1644	49.6	62.38	1
27	2.00	2.5	0.139	0.34	23.2	2414	181.0	40.97	1
28	2.00	2.5	0.139	0.36	11.0	1758	131.4	28.35	1
29	2.00	2.5	0.139	0.59	6.6	1804	174.1	30.55	1

potential maximum initial velocity (V_x) of the fragmented rock projectiles and the total flight time.

$$V_x = V_{max} Cos\theta \qquad (3)$$

And total flight time (t) is given by:

$$t = \frac{V_{max} Sin\theta}{g} + \frac{\sqrt{(V_{max} Sin\theta)^2 + 2Hg}}{g} \qquad (4)$$

TABLE 5. Descriptive statistics of the dataset.

Parameter	Count	Mean	std	min	25%	50%	75%	max
Stemming Length (m)	29	2.17	0.60	1.19	2.00	2.00	2.16	3.96
Burden (m)	29	2.29	0.34	1.50	2.00	2.50	2.50	3.00
Powder Factor (kg/t)	29	0.14	0.01	0.12	0.13	0.14	0.15	0.17
Crack density (cracks/m)	29	0.82	0.59	0.18	0.38	0.59	0.99	2.54
BTS (MPa)	29	13.04	5.70	3.12	9.23	13.45	16.83	26.03
Rock density (kg/m^3)	29	2182.48	295.12	1505.49	2024.34	2297.06	2359.12	2566.52
Uniaxial Compressive Strength (MPa)	29	139.65	63.27	34.42	108.13	134.58	174.14	303.98
Flight distance (m)	29	59.52	51.16	11.14	26.94	40.06	77.13	228.80

where H is the ench height, g is the acceleration due to gravity (9.81m/s^2), and θ is the elevation angle in degrees.

Therefore, the maximum flight distance (FD) is given by the following equation:

$$FD = V_x t \tag{5}$$

where FD is in metres, V_x is the horizontal component of the potential maximum initial velocity in m/s, and t is the total flight time of the blast-induced fragmented rock in seconds. FD was calculated on the assumption that flyrock occurred from the top of the face (face bursting) at an elevation angle of 45 degrees. The nature of face bursts can be indicative of overcharging or can easily express the impact of geological weaknesses, such as joints or cracks present on the bench face.

Fig. 4 shows the correlation matrix developed using Matplotlib in the Scikitlearn library. It clearly indicates how the selected parameters are associated with each other.

B. METHODS

This section introduces and provides a description of the machine learning models (Decision Tree, KNN, ANNs and SVM) that were proposed and applied in this study. Machine learning tools, that is, classification, regression, and pattern recognition tool (nprtool) in MATLAB R2023b, were used in a Windows 10 environment to evaluate the performance of the models on the blast dataset.

1) K – NEAREST NEIGHBORS (KNN)

KNN is one of the simplest and most widely used ML algorithms, making it a baseline classifier for many pattern recognition problems [21]. It is a supervised learning method primarily used for classification tasks. KNN is non-parametric, meaning that there are no fixed parameters for any given data size [22]. The number of parameters depended on the size of the training dataset. In addition, no assumptions must be made regarding the underlying data distribution.

The algorithm works on the assumption that '*similar things exist in close proximity*,' that is, similar things are most likely to be found closer to each other. This means that an unforeseen point or tested example can be classified based on the values of the closest existing points (training examples).

The "K" value refers to the number of nearest neighbor data points to include in the majority voting process. The assignment of the class label on a test example is a majority vote by the K-nearest neighbors closest to it.

$$y(d_i) = argmax_k \sum_{xj \in kNN} y(x_j, C_k) \tag{6}$$

where d_i is a test example, x_j is one of its k nearest neighbors in the training set, and y (x_j, c_k) indicates whether x_j belongs to class c_k. Equation (6) indicates that the prediction will be the class with the most members in the k-nearest neighbors.

The distance metric is a method for determining the distance between a new data point and an existing training dataset [23]. KNN has different distance metrics (over 50 in number). The use of the best distance metric yields good results for the test data, that is, the highest precision, recall, and accuracy [24]. For this study, Euclidean, Manhattan, Cosine and Minkowski distance metrics were selected

2) SUPPORT VECTOR MACHINE (SVM)

SVM has proven to be dependable owing to its balanced accuracy and reproducibility when learning data classification patterns. The objective of every SVM is to optimize

for correct labelling and ensure that the classifier can generalize well to the new data. Although it can be used to perform regression, the SVM has been widely developed as a tool for classification. This is because it is highly versatile, making it a convenient tool across a range of data science scenarios [25].

The SVM decision function is an 'optimal' hyperplane. The hyperplane distinguishes observations belonging to one class from another (see Fig. 5). This is a result of patterns of information about these observations, which we call 'features' or support vectors. Thus, the most probable label for unseen data can be determined from the hyperplane. In addition to the raw data, interpolation can be conducted to generate derivative data that are used to make up the features that are used to infer the hyperplane. Therefore, the SVM can classify both linear and nonlinear data.

A good SVM algorithm maximizes the marginal distance between the two classes while minimizing the classification error.
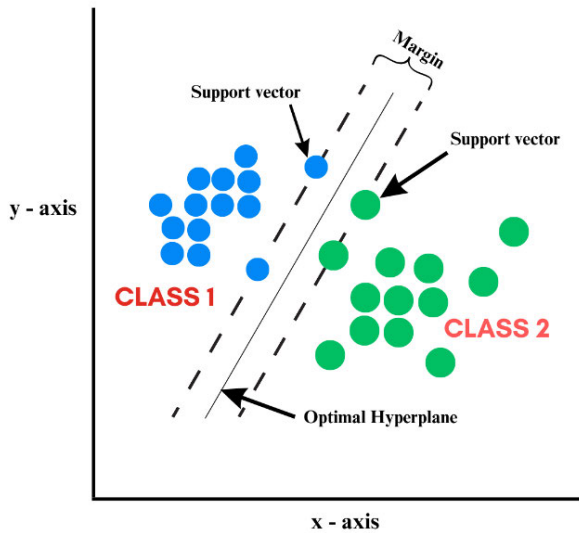
FIGURE 5. Linear classification.

Transformation is made possible using a kernel function. The kernel maps non-linearly separable data into a feature space or higher-dimensional space, where it is linearly separable [26]. SVM models have different kernels. In this study, the following kernels were investigated: Linear, Quadratic, Cubic, SVM – Fine Gaussian, SVM – Medium Gaussian and SVM –coarse Gaussian.

3) ARTIFICIAL NEURAL NETWORKS (ANN)

The human biological nervous system inspired the development of Artificial Neural Networks (ANNs) [27]. The goal of neural networks is to build machines using components that mimic the biological neurons. Consequently, machines are capable of executing complex tasks similar to those of the human brain.

A neural network, as shown in Fig. 6, is a mathematical model that consists of an input layer, hidden layer, and output layer. The mathematical relationship between the input and output values is obtained once the historical data is trained [28]. Weights are associated with the input depending on the degree of relevance for each input that is associated with the output. An activation (transfer) function is then applied to produce the output value. The unit step (threshold), piecewise, linear, Gaussian, and sigmoid functions are among the most commonly used activation functions.
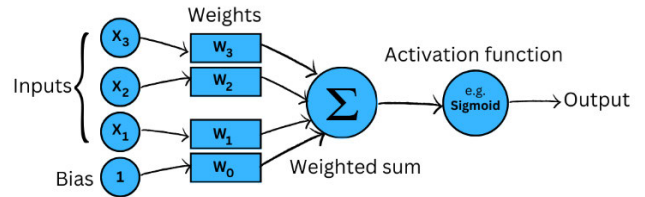

FIGURE 6. Shows a perceptron with its four basic parts: input, bias, weights, activation or step function.

ANNs can have different architectures, causing them to function or behave differently [29].

Feed-Forward Neural Network (FNN): This is also known as multilayer perceptron (MLP), or simply Neural Networks. In FNN, the connections between nodes do not form a cycle; that is, no loops are formed (see Fig. 7). This approach is simple because information processing is unidirectional. Once data is passed through hidden layers (nodes), it can never move backwards.
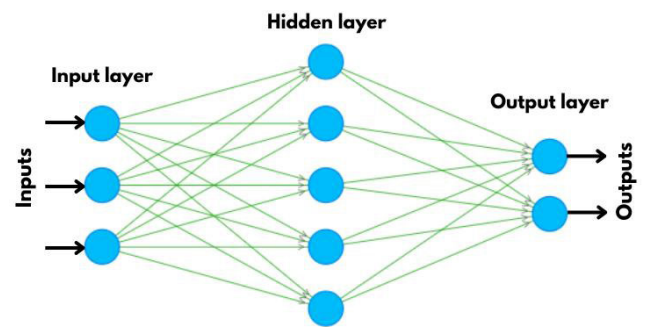

FIGURE 7. Feed forward neural network (MLP).

4) DECISION TREE (DT)

This is a non-parametric type of supervised machine learning, in which the decision-making process results in a visual flowchart that resembles an upside-down tree-like structure (see Fig. 8). The decision starts at the top of the tree, referred to as the root node. It adopts a data-separating sequence that results in Boolean values, that is, Yes or No/True or False. Therefore, information processing using this sequence flows from the root node to the internal nodes (branches), and terminates at the leaf nodes. Each node represents a feature,
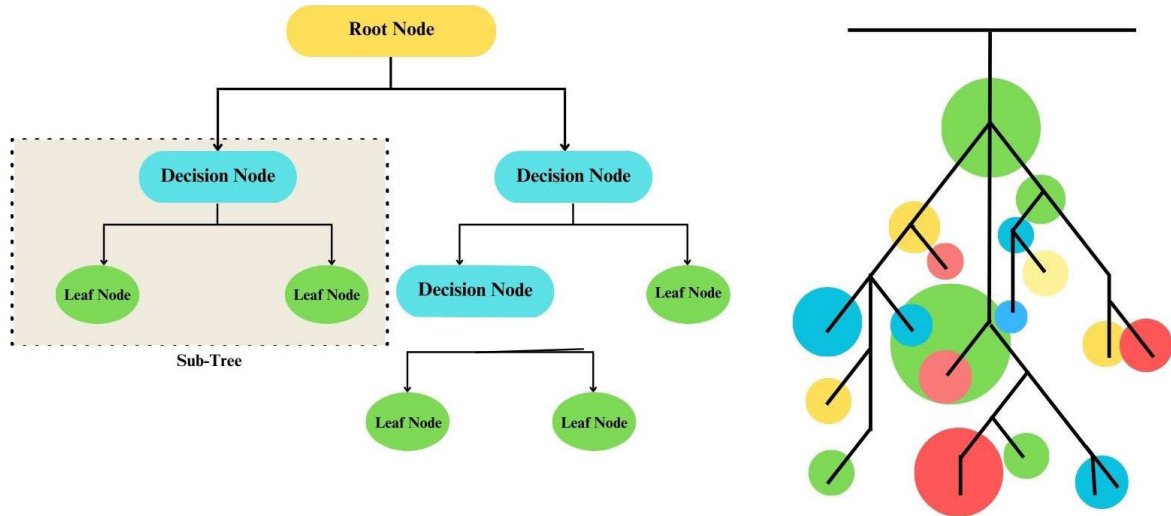
FIGURE 8. Decision tree.

each branch represents a decision rule, and each leaf indicates a categorical value (for classification) or a continuous value (regression case). In DT, decisions are represented by root and intermediate nodes, whereas the outcomes are represented by leaf nodes.

This algorithm is best utilized in solving classification problems but can also be used for regression applications [30].

Hyperparameter tuning is performed to optimize the performance of the algorithm. This can involve proper splitting i.e. division of the main nodes into sub-nodes. This is attained using a split measure function that selects the 'best' splitting attribute based on the impurity measure. As indicated in equations (7), (8), and (9), information gain/entropy (E), Gini index (G), and misclassification error (M) are the most popular impurity measures [31].

$$g^E(S) = -\sum_{k=1}^{K} \frac{n^k(S)}{n(S)} log_2 \left(\frac{n^k(S)}{n(S)} \right) \qquad (7)$$

$$g^G(S) = 1 - \sum_{k=1}^{K} \left(\frac{n^k(S)}{n(S)} \right)^2 \qquad (8)$$

$$g^M(S) = 1 - \frac{max_k \left\{ n^k(S) \right\}}{n(S)} \qquad (9)$$

C. PERFORMANCE MEASURES

For this study, the following performance indicators were used to measure machine performance:

1) CONFUSION MATRIX TECHNIQUE

In this study, the minority class, that is, the important class with fewer samples, had a higher risk loss compared to the majority class (with more samples). The main goal was to correctly classify minority data points. To avoid poor generalization owing to hyperplanes that are more biased towards the majority class, standard machine learning aimed

at achieving an optimized overall accuracy should be avoided. For such an imbalanced set (because flyrock does not occur in every blast), the overall accuracy of a classifier's goodness is usually biased. To overcome this, a confusion matrix (shown in Fig. 9) with information such as False Positives/Negatives and True Positives/Negatives, sensitivity, and specificity offers a more reliable measure of the classifier performance [32].

Because the data classification is divided into two classes, the resulting confusion matrix is made up of four cells, with each cell corresponding to True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN). All parameters are evaluated to determine their accuracy, sensitivity, and specificity.

		Predicted Class	
		Positive Class	Negative Class
True Class	Positive Class	TP	FN
	Negative Class	FP	TN

FIGURE 9. Confusion matrix.

A description of the essential terms is given below:

i. **Accuracy** – The number of data points that were classified correctly by the classification algorithm.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \qquad (10)$$

ii. **Positive class** – Represents the minority class and is the class that holds utmost significance to the designer.

iii. **True Positive (TP)**– Details the number of data points from the positive class that have been correctly classified.

iv. **False Positive (FP)/Type 1 error** – The number of data points from the negative class that have been incorrectly classified as belonging to the positive class.

v. **True Negative (TN)** – the number of data points from the negative class that have been correctly classified.

vi. **False Negative (FN)/Type 2 error** – The number of data points from the positive class that have been incorrectly classified as belonging to the negative class.

vii. **Specificity/True Negative Rate** – Specifies how well the data points in the negative class have been classified.

$$Specificity = \frac{TN}{TN + FP} \qquad (11)$$

viii. **Sensitivity/True Positive Rate/Recall Rate** –Specifies how well the data points in the positive class have been classified.

$$Sensitivity = \frac{TP}{TP + FN} \qquad (12)$$

2) RECEIVER OPERATING CHARACTERISTIC (ROC) CURVES

This is another indicator that graphically illustrates the useful information that details the performance of a classifier on an imbalanced dataset. ROC Curves are generated by plotting true positives (sensitivity) against false-positive rates, that is, (1-specificity). Sensitivity and Specificity are inversely correlated; that is, a decrease in sensitivity indicates an increase in specificity. As shown in Fig. 10, the line of equality (50% specific and 50 % sensitive) where no discriminative value exists is represented by an area under curve (AUC) of 0.5. This is a diagonal line that runs from the lower-left corner to the upper-right corner. The incorrectly predicted results are represented by a curve below the line of equality. A curve close to the line of equality signifies a low accuracy and is not different from that obtained by chance. A curve above the line of equality indicate correct predictions or possesses a high discriminatory power [33], [34].

3) CROSS ENTROPY ERROR

The cross-entropy error or loss is a metric that evaluates the performance of a classification model during machine learning. The model weights are adjusted during training to minimize loss. A better model is one that has comparatively lower loss values in the range of 0 to 1. Cross-entropy loss of zero points to a perfect model.

III. RESULTS AND DISCUSSION

In this study, several machine classification algorithms were developed to classify flyrock hazard. The models would be able to predict based on severity whether a blast would be
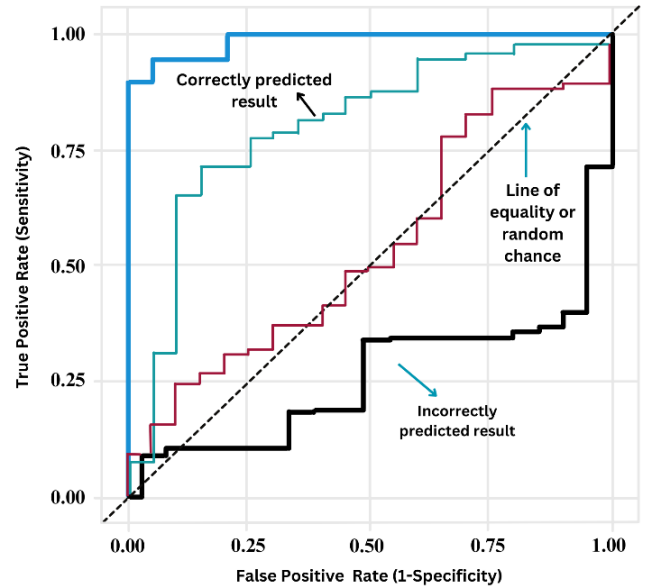

FIGURE 10. An example of a ROC curve.

'safe' or 'unsafe'. Decision Tree (DT), K Nearest neighbors (KNN), SVM and ANNs were investigated.

As proven by Hudaverdi and Akyildiz [18], a classification approach can be utilized to indicate the potential risk of a blast.

Regression models were also evaluated to justify the use of the classification. Among the four machine learning models, namely, Support Vector Regression (SVR), KNN, ANNs and Decision Tree, the SVR with a linear kernel demonstrated the highest performance. It achieved the lowest Root Mean Square Error (RMSE) of 10.90, indicating the smallest prediction errors. Additionally, the SVR model showed a commendable R-squared value of 0.83, signifying a strong ability to explain variance in the data. The Variance Accounted For (VAF) was also impressively high, at 84.20%, further validating the model's capability.

However, the Decision Tree model performed poorly, displaying negative values for both R-squared (-0.13) and VAF (-0.61), suggesting a weak fit to the data. The Neural Network model showed a high RMSE (48.95) and negative R-squared value (-2.50), indicating inadequate predictive power. In this regression context, the K-Nearest Neighbors model displayed moderate performance, with an RMSE of 22.95 and an R-squared value of 0.23. SVR emerged as the most suitable regression model for predicting flyrocks in the experimental mine blasts, offering the highest accuracy and reliability compared to other models. As observed in Fig. 12, despite the SVR performing well, it still exhibited a high magnitude of differences between the true and predicted values, more so, in the unsafe blasts whose consequences are significant. Overall, this analysis, as illustrated in Fig. 11, proved that machine learning via regression was not optimal in our research context. Thus, we could explore the potential of classification learning to derive meaningful learning from
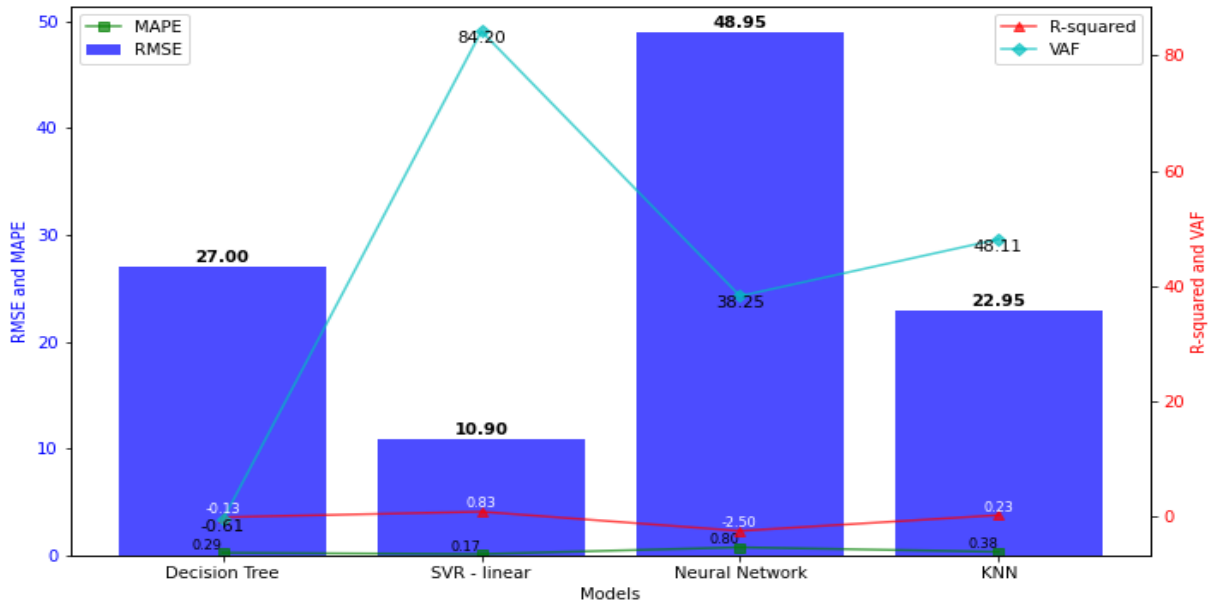
FIGURE 11. Performance metrics comparison in regression models.
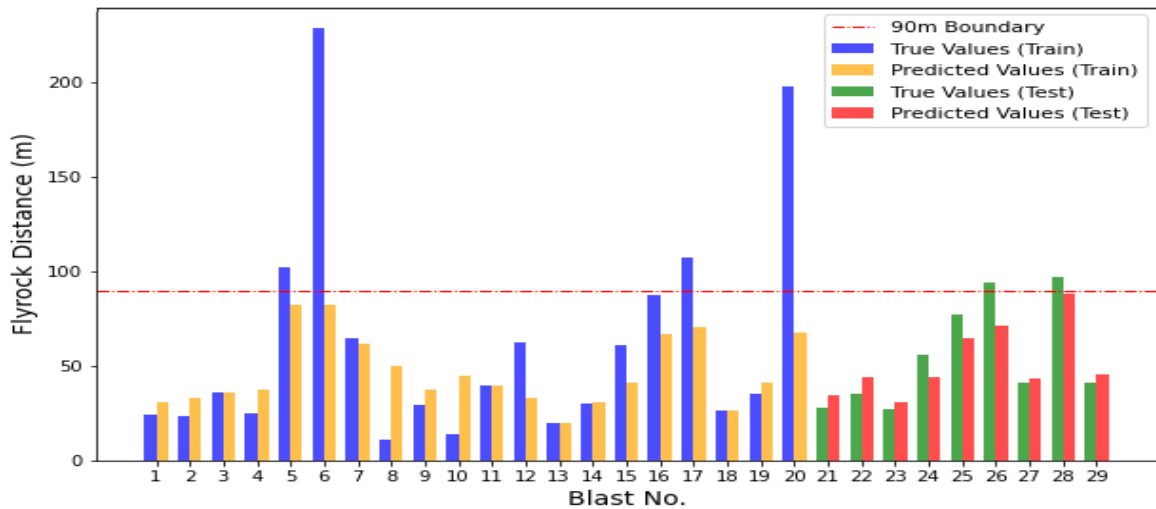

FIGURE 12. True vs predicted flyrock distances for SVR - linear model.

machine performance, which is crucial in decision-making with regard to blast safety in a mine.

For classification, this study utilized different criteria for the hyperparameters, particularly within each modeling technique. For example, in SVM, a number of SVM methods, such as Linear SVM, Quadratic SVM, Medium Gaussian SVM, have been examined to determine the best technique. This section presents the results of the learning models based on their classification abilities.

Initially, the dataset was stripped into x- and y-features. The x features represent all the explanatory variables. Consequently, the y features (response variable) displayed categorical values for two types of blasts, that is, safe and unsafe blasts.

Training was conducted using the default settings of the MATLAB Classification Learner. Feature selection was not performed. Misclassification costs were set to default. No optimizer properties were used, and Principal Component Analysis (PCA) was deactivated. This was performed to preserve the integrity of the original dataset. Thus, the machine performance was judged based purely on the original data. Twenty models were trained using the Levenberg–Marquardt algorithm (LMA).

At the end of the training, all models were assessed according to their performance. Accuracy, Total cost and Training Time were the three key performance indicators that informed the selection of the most ideal machine models. Simultaneously, the hyperparameter attributes of each model were

TABLE 6. Performance of the models (70% training & validation, 30% testing).

Model No.	Model Type	Accuracy % (validation)	Total Cost (Validation)	Accuracy % (Test)	Total Cost (Test)	Training Time (sec)
2.1	Tree - Fine	66.67	7	87.5	1	3.34
2.2	Tree - Medium	66.67	7	87.5	1	3.05
2.3	Tree - Coarse	66.67	7	87.5	1	2.19
2.11	SVM - Linear	90.48	2	87.5	1	1.13
2.12	SVM - Quadratic	90.48	2	100	0	1.88
2.13	SVM - Cubic	90.48	2	100	0	1.95
2.14	SVM - Fine Gaussian	80.95	4	75	2	1.20
2.15	SVM - Medium Gaussian	80.95	4	87.5	1	1.99
2.16	SVM - Coarse Gaussian	80.95	4	75	2	1.06
2.17	KNN - Fine	85.71	3	100	0	2.32
2.18	KNN-Medium	80.95	4	75	2	0.96
2.19	KNN - Coarse	80.95	4	75	2	1.37
2.20	KNN - Cosine	80.95	4	75	2	2.43
2.21	KNN - Cubic	80.95	4	75	2	1.08
2.22	KNN - Weighted	80.95	4	87.5	1	2.08
2.28	Neural Network - Narrow	90.48	2	100	0	3.12
2.29	Neural Network -Medium	95.24	1	100	0	5.04
2.30	Neural Network - Wide	95.24	1	100	0	2.47
2.31	Neural Network - Bilayered	95.24	1	100	0	2.13
2.32	Neural Network - Trilayered	85.71	3	100	0	1.34

TABLE 7. The top four selected models.

Models	Accuracy % (validation)	Total Cost (validation)	Accuracy % (Test)	Total Cost (Test)	Training Time (sec)
Coarse DT	66.67	7	87.5	1	2.19
Neural Network Wide	95.24	1	100	0	2.47
Fine KNN	85.71	3	100	0	2.32
SVM-Cubic	90.48	2	100	0	1.95

investigated. This would show which specific tuning parameters were responsible for the training, validation, and test results.

A summary of the classification training and testing is presented in Table 6. Once their performances were ranked, the best four models were selected.

The Cubic SVM performed better than the other SVM kernels. It produced a hyperplane that was able to completely separate the vectors into two non-overlapping classes. This means that the nonlinear region was capable of separating the data more efficiently. Usually, the higher the degree of the polynomial, the more curved the resulting hyperplane line. Cubic SVM, which is a third-order polynomial when compared to quadratic SVM, was therefore more robust.

From the ANNs tested, the Wide Neural Network (WNN) performed comparatively better, perhaps due to the high

TABLE 8. Hyperparameters for the top four selected models.

Models and Hyperparameters				
Coarse DT	Maximum No. of splits	Split Criterion	Surrogate Decision Splits	Other information
	4	Gini's Diversity Index	Off	None
Wide NN	No. of fully connected layers	First layer size	Activation	Other Information
	1	100	ReLu	Iteration limit: 1000, Regularization strength (Lambda):0, Standardize data: Yes
Fine KNN	No of Neighbors	Distance Metric	Distance Weight	Other Information
	1	Euclidean	Equal	Standardize data: Yes
Cubic SVM	Kernel Function	Kernel Scale	Box Constraint Level	Other Information
	Cubic	Automatic	1	Multiclass method: One-vs-One; Standardize data: Yes

TABLE 9. Ranking of the top four machine models.

Models	Train			Test				Rank
	Sensitivity (%)	Specificity (%)	AUC	Sensitivity (%)	Specificity (%)	AUC	Training Time (sec)	
WNN	75	100	0.9853	100	100	1	2.47	**1**
Cubic SVM	50	100	0.9412	100	100	1	1.95	**2**
Fine KNN	50	94.1	0.7206	100	100	1	2.32	**3**
Coarse DT	25	76.5	0.5221	50	100	0.75	2.19	**4**

number of neurons (100 in its first layer compared to 25 in the Medium Neural Network (MNN) and 10 neurons in the Narrow Neural Network (NNN). It also proved superior to Bilayered and Trilayered Neural Networks, which have two and three fully connected layers, respectively.

Using the Gini index, all DT algorithms yielded similar results. However, Coarse DT was unique because it managed the shortest training time. A possible inference is that the number of splits has a direct bearing on the machine performance, that is, the lower the number of splits, the faster the training time.

The Fine KNN has 1 neighbor compared to 10 in Medium KNN or to 100 in Coarse KNN. The classifier accuracy increased when the number of neighbors decreased. Fine KNN is therefore able to make finely-detailed distinctions between classes. Euclidean distance associated with Fine KNN was observed to be the best determiner of nearest neighbors compared to other KNN models. It performed better than

the distance weighting used in the Weighted KNN, which was the second best KNN model.

The corresponding hyperparameters are listed in the Table 8.

The main purpose of this study was to assess the classification ability of SVM, KNN, ANN, and DT for the flyrock problem. After determining the best models, the next step involved carrying out a further comparative analysis among the selected models based on the main performance indicators. In this research, the Confusion Matrix and the Receiver Operating Characteristic (ROC) curve, denoted by the Area Under Curve (AUC), were the main statistical indicators used to measure the performance of the selected models.

Fig. 13 illustrates the performance of the Coarse DT algorithm. An inspection of its performance revealed that it correctly predicted 13 of 17 safe blasts (76.5% specificity). It gave 1 correct prediction out of 4 (25% True Positive Rate) for the unsafe blasts during training and validation. However,
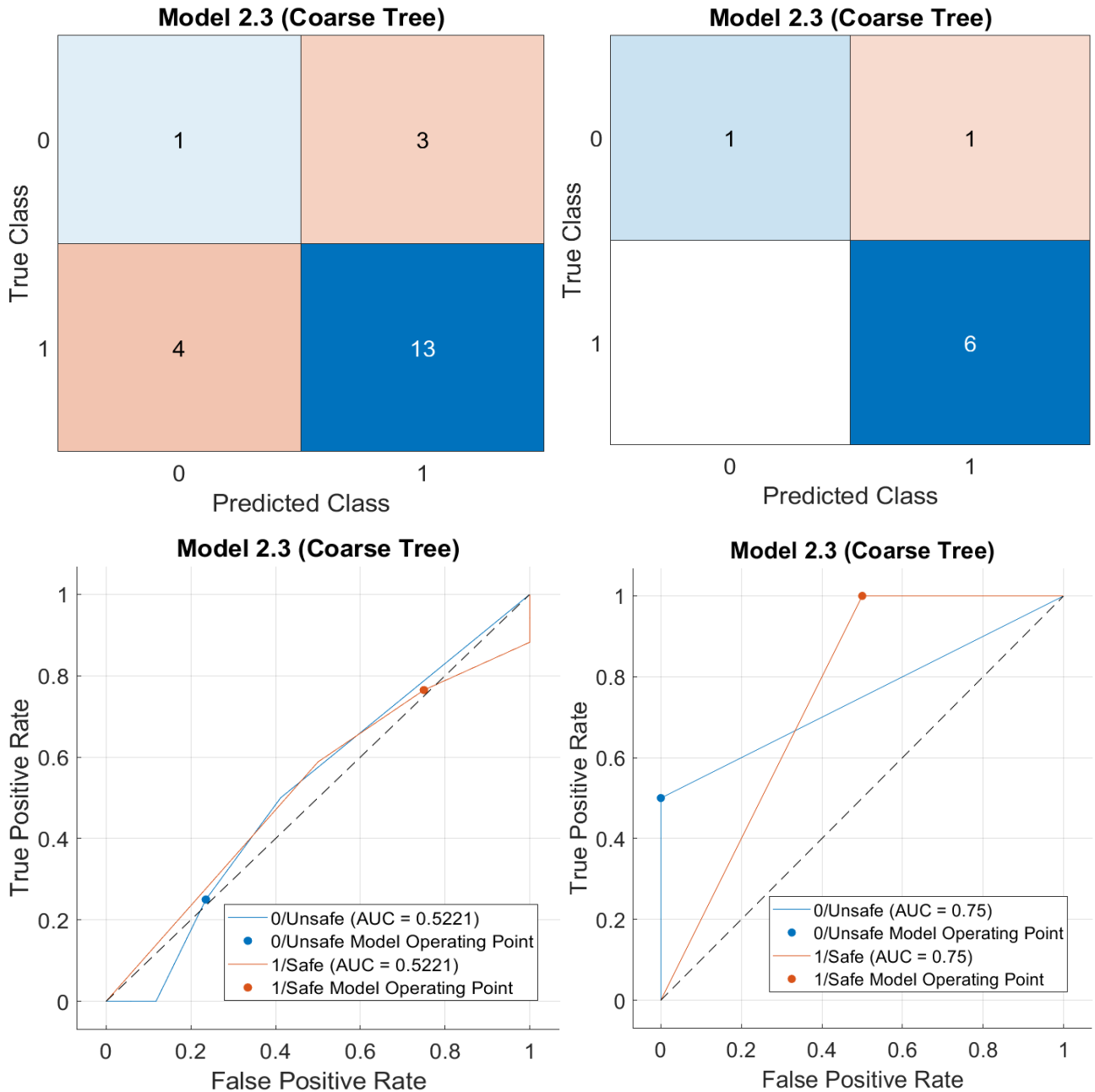
FIGURE 13. Confusion matrix and ROC for coarse DT during validation and testing phase.

it attained a 50% sensitivity for unsafe blasts during the test phase. In contrast, it correctly classified all safe blasts and thus achieved a specificity of 100% for the same test sample.

Moreover, it achieved an ROC of 0.5221 in the validation phase, and an ROC of 0.75 during testing. Therefore, it can be deduced that the Coarse DT algorithm showed a slight improvement in performance on the test set.

Analysis of the Cubic SVM model during validation indicated outstanding classification performance with 100% specificity for safe blasts. 2 of the 4 unsafe blasts were correctly classified (50% sensitivity). In the test set, the model

accurately pinpointed the two categories of blasts with 100% scores for both sensitivity and specificity.

As presented in Fig. 14, using the Cubic SVM model, an ROC of 0.9412 was achieved during training and valida-tion, whereas the test dataset recorded a perfect ROC score of 1.

In Fig. 15, it can be observed that the classification ability for unsafe blasts was similar to that of Cubic SVM, that is, 50% sensitivity (2 correct predictions of 4 unsafe blasts). However, it had a high specificity of 94.1% for the vali-dation set. In the test phase, the Fine KNN model showed
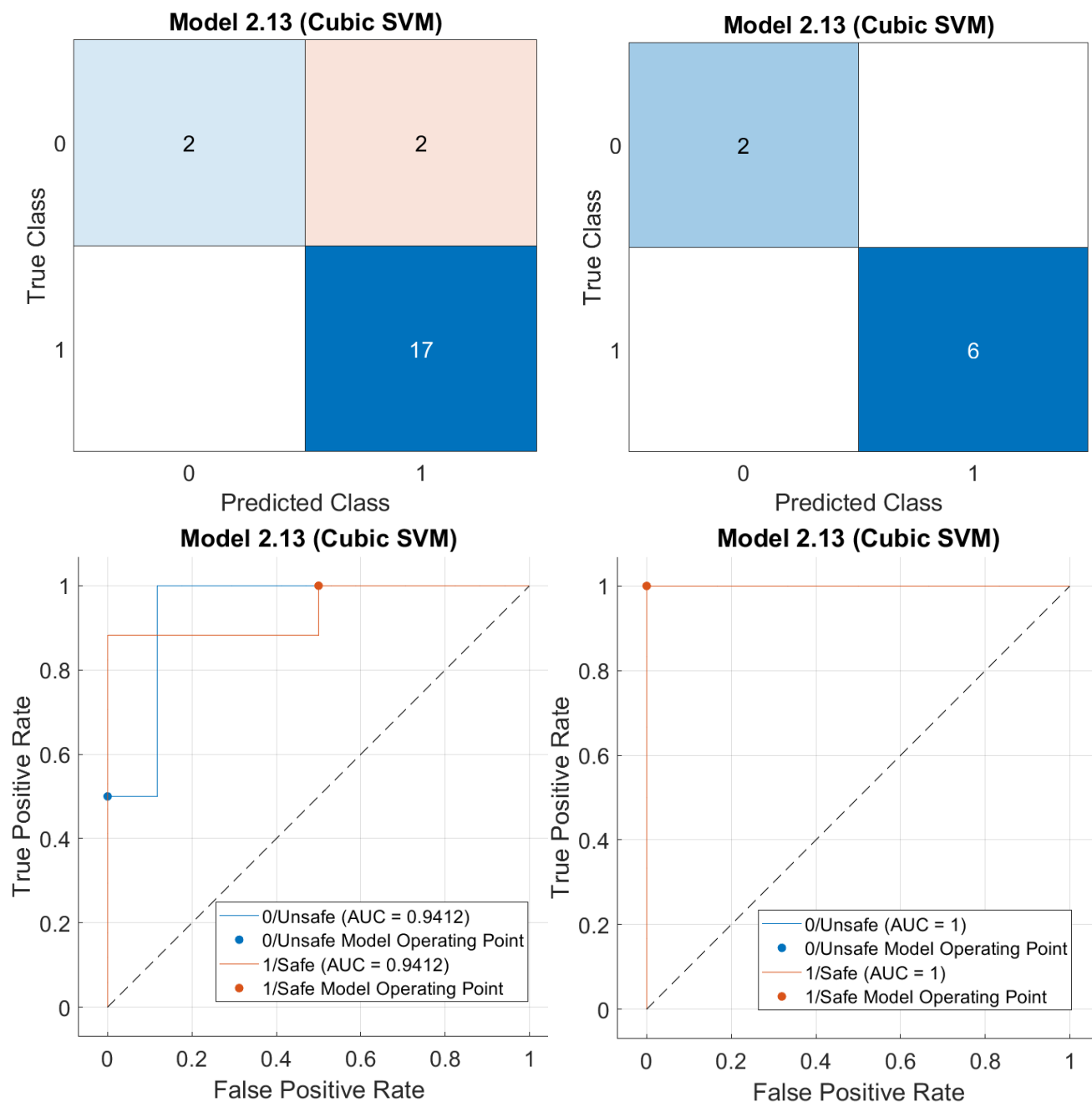
FIGURE 14. Confusion matrix and ROC for cubic SVM during training and testing phase.

good distinction power because it appropriately classified all blasts.

The ROC measurements indicated an improvement from 0.7206 in the validation set to 1 in the test data.

As shown in Fig. 16, the WNN generated the highest classification accuracy. The confusion matrix illustrated that all 17 safe blasts and 3 of 4 unsafe blasts were correctly identified. Moreover, it registered a 100% recognition rate for both types of blasts in the test set.

The ROC also demonstrated exceptional performance powers associated with the WNN. It also had the highest ROC scores of 0.9853 for the validation set and 1 for the test set.

Finally, all models were compared, and a ranking was established to ascertain the superiority of the classifiers (see Table 9).

Overall, in the assessment of the results based on the sensitivity and specificity measures from the confusion matrix and
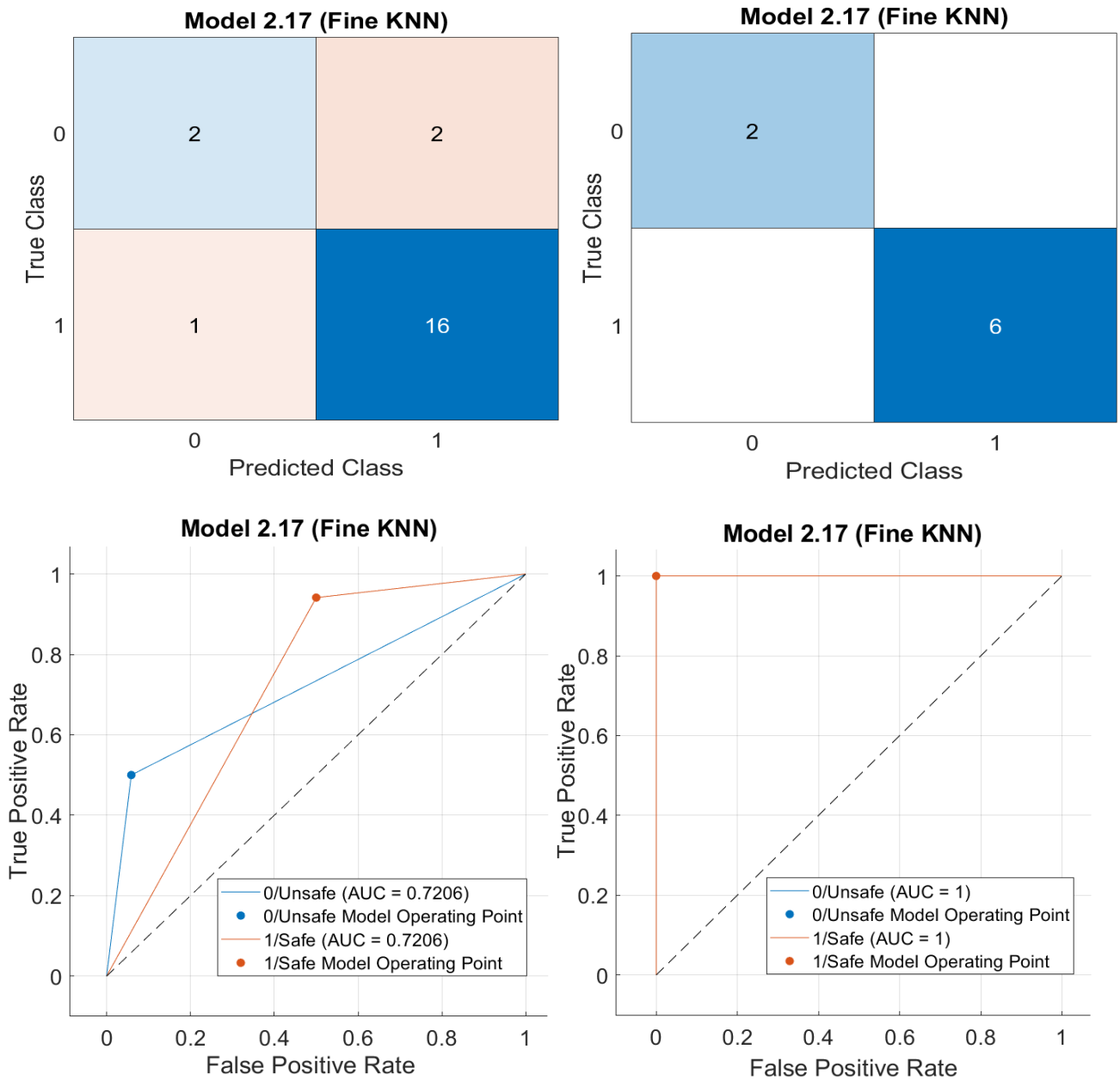
FIGURE 15. Confusion matrix and ROC for Fine KNN during training and testing phase.

ROC measures, WNN and Cubic SVM were found to be the most capable algorithms. The rest also displayed significant classification abilities. WNN and Cubic SVM displayed better ROC performances and were able to classify blasts better than Fine KNN and Coarse DT. Coarse DT was the least desirable model. It had a misclassification (1 FN) in the test sample compared with the other models.

To validate the results of the machine-learning exercise, an assessment of overfitting was conducted. Given the small dataset, a 5-fold cross-validation was performed. Subsequently, the obtained performance was compared with the test accuracy of the four models. The "5% points or less" rule of

thumb i.e. a heuristic guideline commonly used in machine learning, was chosen as a useful estimate for assessing the potential presence of overfitting.

Table 10 illustrates slight variations between the test accuracy obtained from the 70-30 train-test split dataset and the results obtained from 5-fold cross-validation for both the WNN and Cubic SVM. This implies a reduced probability of overfitting in the models. Conversely, the notable gaps in performance between Fine KNN and Coarse DT indicate a higher potential for overfitting. These results highlight the robust generalization capability of the WNN and Cubic SVM models.
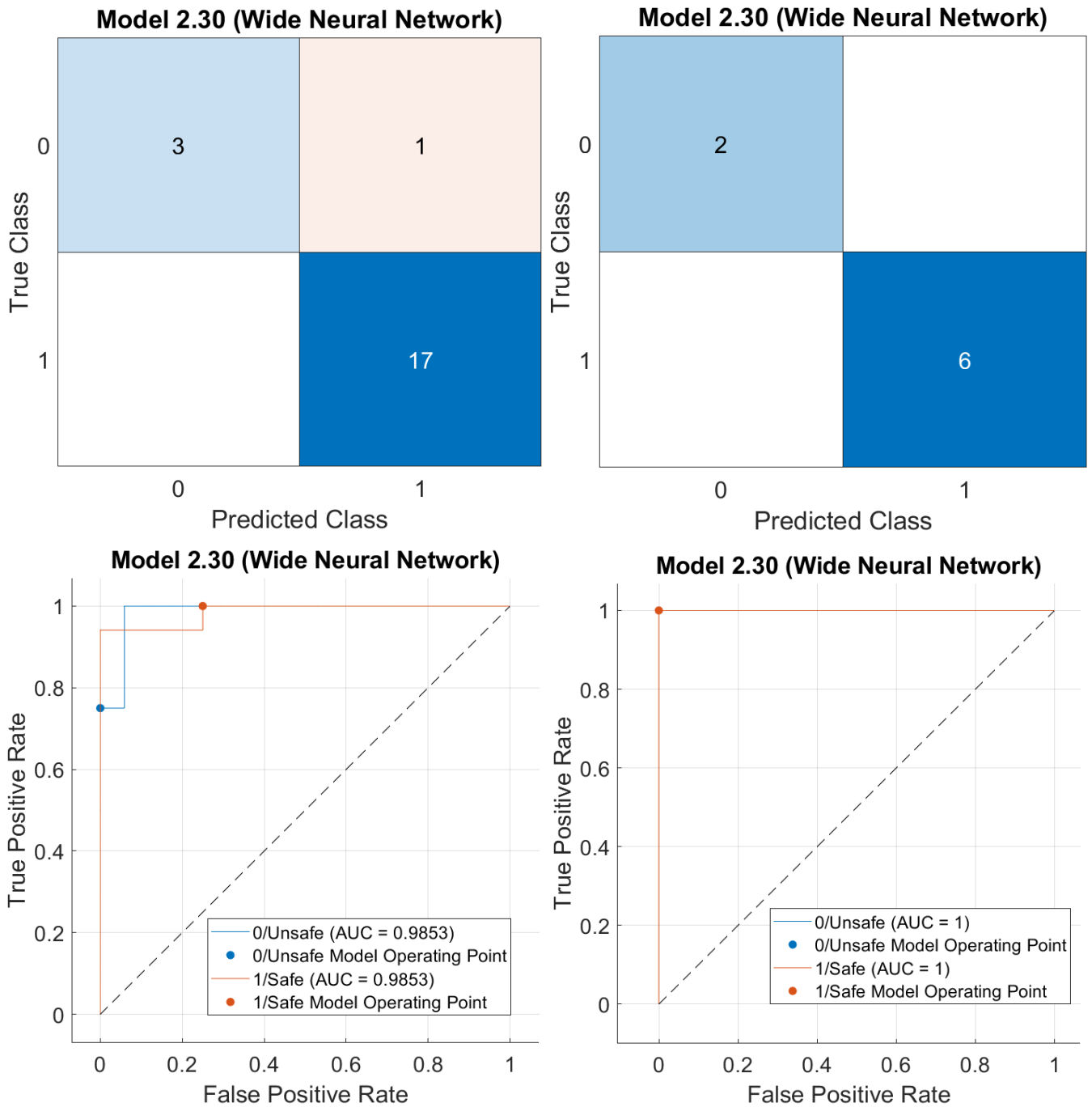
FIGURE 16. Confusion matrix and ROC for WNN during validation and testing phase.

After determining that the WNN had the best overall performance, the second phase of the study investigated how ANNs can be trained for classification tasks using a different approach. The goal of the second phase was to develop a new neural network model that could achieve a 100% accuracy rate in both the training-validation and testing phases, surpassing the performance of the WNN model. This would be achieved by building a new model and assigning new

parameters and functions that would tune the machine to achieve a 100% classification accuracy for all types of blasts and in all datasets.

The nprtool in MATLAB was used in this study. This tool can be used to create models suited to pattern-recognition problems. Therefore, it is the best tool for training learning models for classification. The network architecture shown in Fig. 17 is a two-layer feedforward neural network (FNN).

TABLE 10. Overfitting analysis.

Models	70-30 Test Accuracy (%)	5-fold Cross Validation (%)	Gap (%)
WNN	100	96.55	3.45
Cubic SVM	100	96.55	3.45
Fine KNN	100	89.66	10.34
Coarse DT	87.5	75.86	11.64
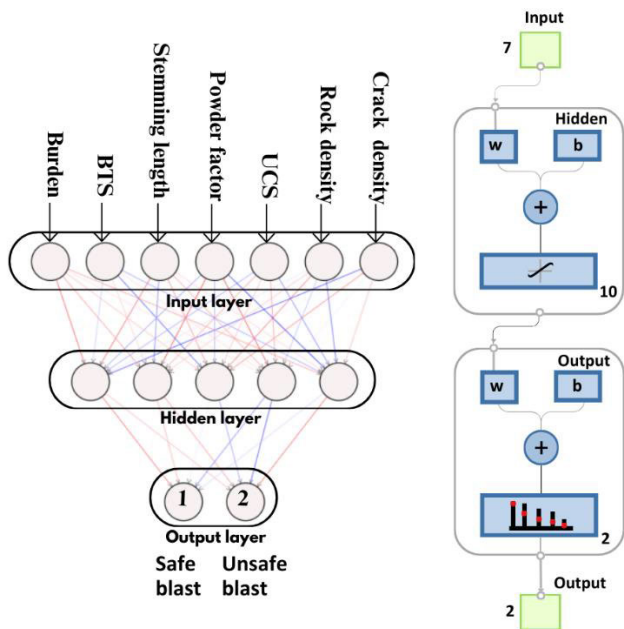
FIGURE 17. Two layered feed-forward neural network architecture.

It contains sigmoid hidden neurons and softmax output neurons, both of which are ideal for classification. The training dataset was set to 70%, validation to 15%, and testing to 15%.

These optimal solutions can also be verified using ROC curve results. In all stages, from training to validation and testing, the two-layered FNN achieved a perfect score of 1 for the true-positive rate (see Fig. 19).

The training outcomes obtained using this network architecture are shown in Fig. 18.

As can be seen in the confusion matrix below, the FNN model also exhibited exceptionally high performance after it was trained and tested. The sensitivity and specificity based on the predicted outputs during the training and validation phases were 100% for the safe and unsafe blasts. Out of a total of 21 blasts, 18 blasts were successfully categorized as 'safe' while the remaining 3 were 'unsafe.' Consequently, the validation phase predicted 2 'safe' blasts and 2 'unsafe'

blasts. To further confirm the classification power in this supervised environment, the test set successfully classified 3 'safe' blasts and 1 'unsafe' blast. No blast was misclassified. Finally, a review of its performance based on the 'All Confusion Matrix' revealed that it is also a far more accurate, flexible, and reliable model than classifiers such as Fine KNN and Coarse DT, which were trained using the classification learner app in MATLAB.

The best performance, as shown in Fig. 20, for the two-layered FNN occurred at epoch 30. The best model was selected based on the Cross-entropy and Error results. The cross-entropy loss function was selected as the objective function. Typically, high-accuracy classification models depict small values of cross-entropy. The goal was therefore to select the model with the smallest cross-entropy value as well as the smallest error values.

Table 11 lists the tabulated results from the training, validation, and testing of the two-layered FNN.

TABLE 11. Classification results using the two-layered FNN.

MODELS	OBSERVATIONS	CROSS-ENTROPY	ERROR
Training	21	0.0000	0
Validation	4	0.0000	0
Test	4	0.0138	0

The results therefore seem to indicate that machine learning models, such as KNN, SVM, DT, and ANN, can be applied to solve practical problems, such as flyrocks in mines. The experiment also implies that ANN have a significantly higher classification ability than KNN and DT. Among the ANN studied in this research, the two-layered FNN demonstrated how training on trained data can lead to the development of a superior learning model.

As earlier stated in this paper, most of the previous research studies that focused on flyrocks problems in mines were focused on building regression-based models for flyrock prediction. Upon closer examination of the data utilized in this study, it is evident that a nonlinear relationship exists between the input and output parameters. This nonlinearity is a challenging obstacle when attempting to solve regression problems. It is often difficult for regression models to predict with 100% accuracy compared to classification models. Although classification models have been used in other fields and in mining, very few, if any, have been used for flyrock distance determination. Therefore, this study tested classification algorithms as an alternative approach for solving flyrock-related problems. Moreover, it focused on the superiority of a selected number of classification models.

The aim of this study was to determine the safety aspects of a blast based on a predetermined distance. This made it possible to classify a blast into two classes: safe ($<$ 90 m) and unsafe ($>$ 90 m). Note that if the distance changes to let's say 60 m, the same process is repeated, and ML is applied
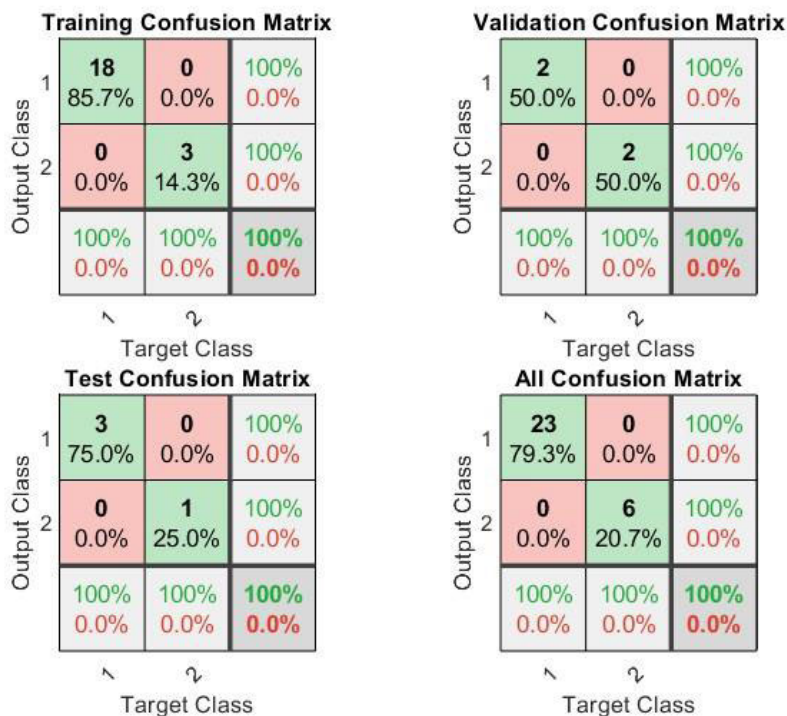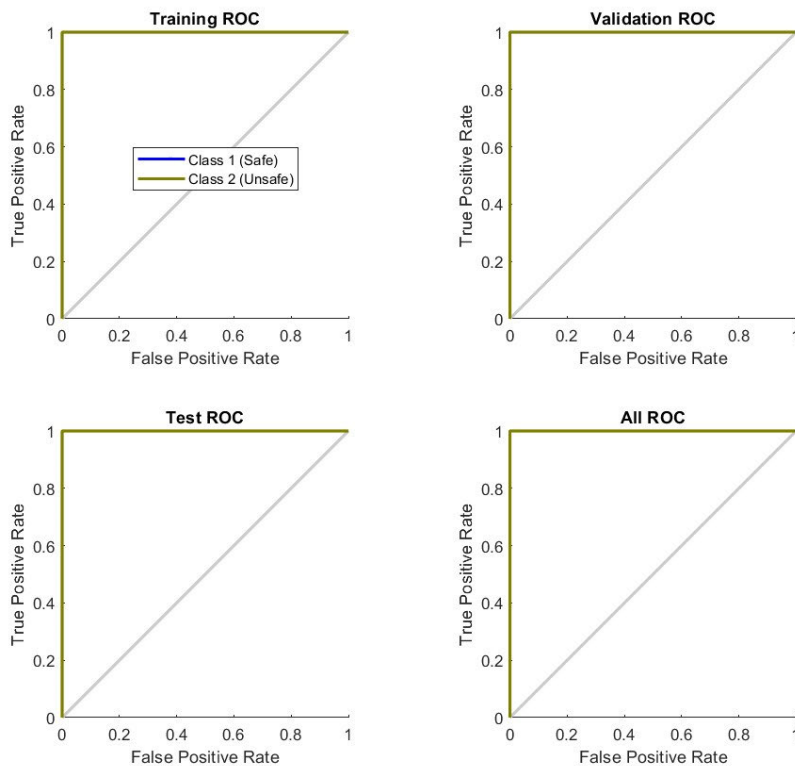
FIGURE 18. Confusion matrix for the FFNN.

FIGURE 19. ROC curve for FFNN.

to determine the best classification algorithm. Although the distance is an important parameter, determining whether a

blast is safe is a more significant goal. This classification strategy attempts to interpret the consequence of the flyrock
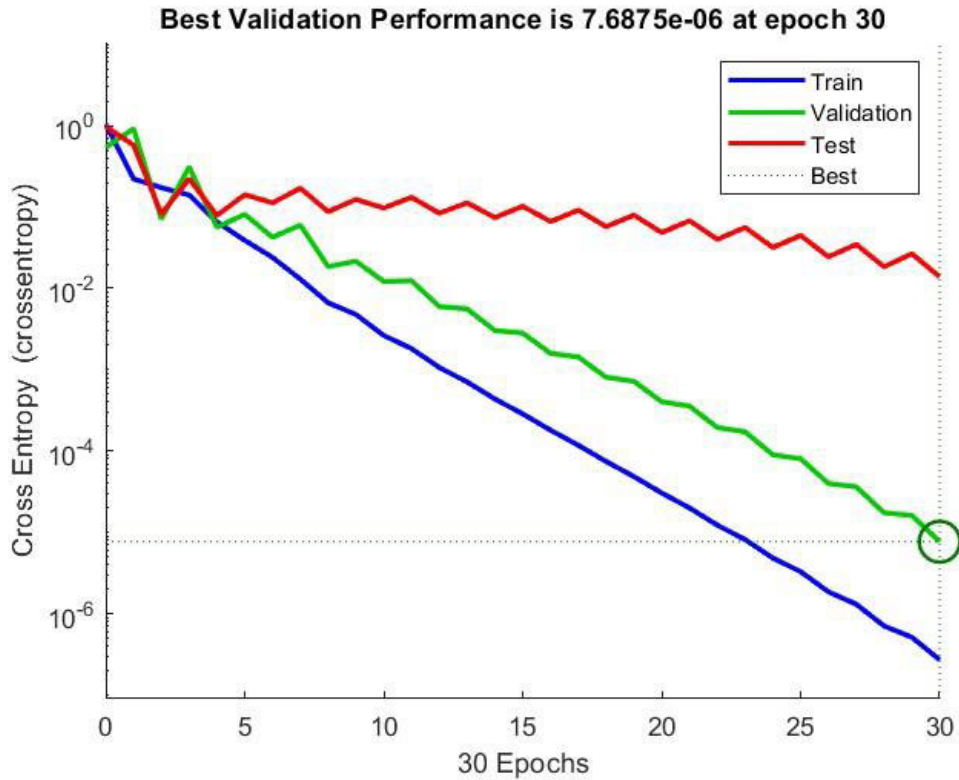
FIGURE 20. Best validation performance for two-layered FNN.

distance in terms of the safety level assigned to a given blast. The results of the investigation indicated that Coarse DT, Cubic SVM, Fine KNN, and WNN were the four best machine-learning models to use. Among these, WNN and Cubic SVM displayed higher-level performance outcomes with a lower likelihood of overfitting. Fine KNN and Coarse DT were the inferior models compared to the former two (WNN & Cubic SVM); with Coarse DT being the least performing model.

The underperformance of DT may have been due to its inability to properly learn the complex relationships between the features in the data. DTs are also highly susceptible to overfitting, that is, high accuracy on training data but low accuracy on test data, or exceptionally high results on the test data with relatively low results during the training and validation phases. For example, Coarse DT registered 66.7% validation accuracy and 87.5% test accuracy. In addition, DTs generally work well with large datasets compared with the one investigated in this research, which had 29 sample points and thus considered a small dataset.

KNN and DT are simplistic compared to SVM and ANNs. Only one hyperparameter, that is, the distance between a group of data points expressed by the value of K (total closest neighbor), is investigated for KNN, whereas ANNs can have many hyperparameters. ANNs operate via black-box

modeling, meaning that the mechanism that produces the output is not clearly understood.

A high degree of nonlinearity calls for complex operations in which models such as ANNs and SVM work optimally. SVM can use kernels that map nonlinear data points into a higher-dimensional space where linear separation is possible. In this case, Cubic SVM was the best kernel. A major advantage of SVM is its ability to classify datasets that have numerous attributes, even when a small sample size is available for training. This could be the reason for the exemplary performance of Cubic SVM presented in this paper. In ANNs, the hidden layers contain neurons that are largely responsible for their learning capabilities. In the WNN, the 100 neurons, which can be considered as an intermediate number of neurons in the hidden layer, encouraged proper learning. Small neuron numbers were associated with underfitting. The WNN also did not have a high number of neurons that resulted in overfitting, as evidenced by its ability to generalize well to the test data.

The research further investigated the classification ability of ANNs using the nprtool in MATLAB to train a Two-layered FNN with the sole aim of achieving 100% classification accuracy. This was attained and proved that the FNN was equally capable and that, overall, it was a highly flexible model since already trained data could be used for the next training phase, thus making the learning process quicker.
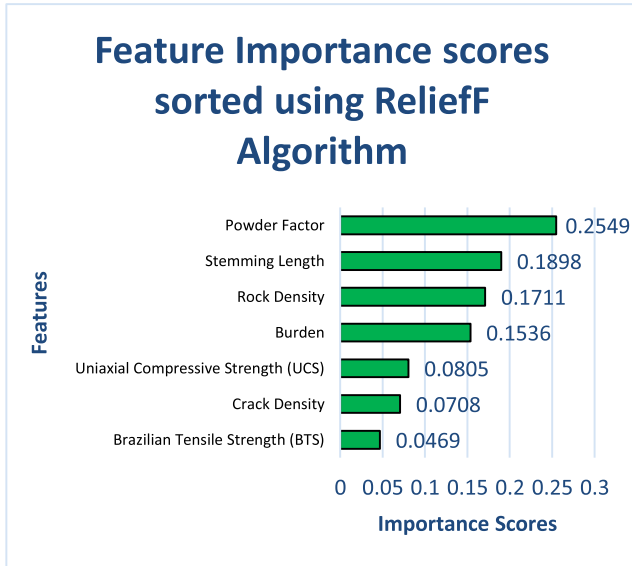
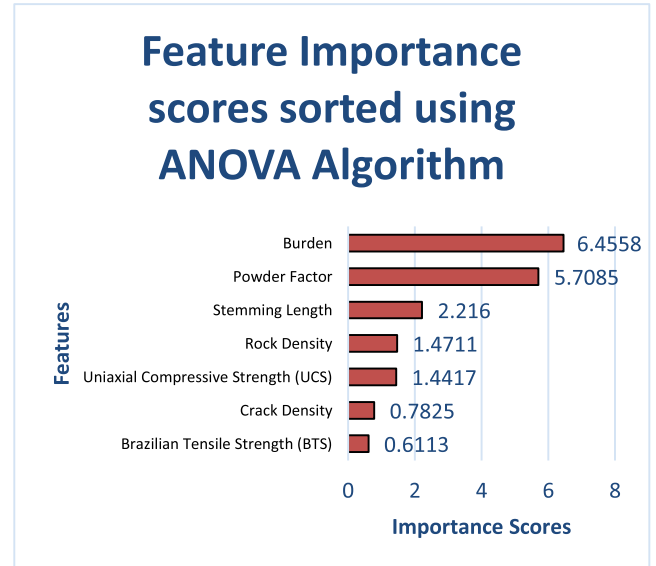**FIGURE 21.** Strength of relation between input parameters and flyrock distance using ReliefF algorithm.

FIGURE 22. Strength of relation between input parameters and flyrock distance using ANOVA.

A. SENSITIVITY ANALYSIS

Sensitivity analysis is a crucial process because it can demonstrate the relationship between input factors and their effect on the response variable. Therefore, it was possible to determine the importance of each feature on a relative basis. In this study, a sensitivity analysis was performed on all seven features to determine their degree of influence on the output. First, scatter plots for each feature against the dependent variable (i.e., the flyrock distance) were plotted. The statistical measure used to establish the influence of the features was R-squared, which indicates the proportion of variance in the dependent variable explained by the independent variable of interest. The R-squared values in Table 12 indicate the magnitude of the influence for each feature.

TABLE 12. Results from scatter plot analysis.

	FEATURE	R-SQUARE
1	BURDEN (M)	0.3157
2	STEMMING LENGTH (M)	0.3021
3	POWDER FACTOR (KG/T)	0.3001
4	BTS (MPA)	0.0393
5	CRACK DENSITY (CRACKS/M)	0.0324
6	ROCK DENSITY(KG/M³)	0.0264
7	UNIAXIAL COMPRESSIVE STRENGTH (MPA)	0.0016

To validate the results of the scatter plots, two additional statistical methods were employed: the ANOVA and ReliefF algorithms. These are two popular algorithms used for feature selection in machine learning. The results are presented in
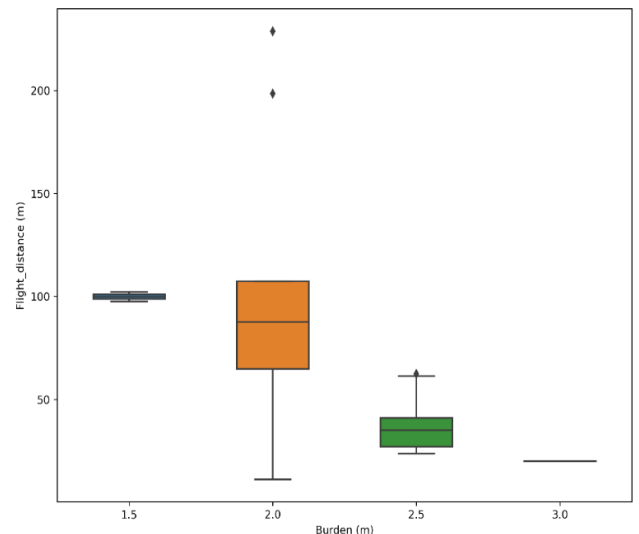

FIGURE 23. Burden vs Flyrock distance.

Fig. 21 and 22. To a large extent, all three methods were in agreement, suggesting that burden, stemming length, and powder factor were the most influential parameters on the throw distance. Additionally, they assigned varying degrees of importance to BTS, UCS, rock density, and crack density. The least influential parameters, as noted by all three methods, were the Uniaxial Compressive Strength, Crack density, and BTS.

Box plots were also used to examine the strength of relations among two selected features i.e. burden and powder factor. These features express sensitivity to the resulting flyrock distance. An inverse relationship was observed between burden and flight distance (see Fig. 23). This conforms
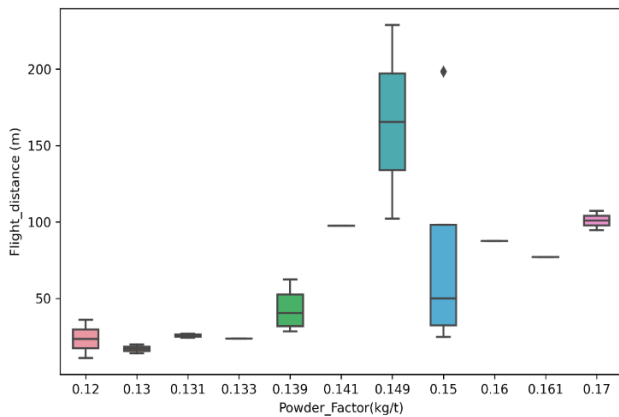
FIGURE 24. Powder factor vs Flyrock distance.

with most other studies that have established a similar relationship [35].

Generally, a positive correlation was observed between powder factor and the resulting flight distance (see Fig.24). This is supported by other research studies, which have shown that an increase in powder factor often results in a greater fragment throw distance.

IV. CONCLUSION

In this study, an alternative machine-learning method, that is, learning by classification for the prediction of flyrocks in a mine, was developed.

The following conclusions were drawn:

i. Wide Neural Networks (WNN) demonstrated the strongest classification power for two categories of blasts: safe and unsafe blasts. This is followed by a Cubic SVM. This means that such a model can be relied upon to predict the safety of future blasts in the mine.

ii. The performance of a model can be influenced by the dataset size. Coarse DT demonstrated poor machine-learning abilities. It had one false-negative result. It also exhibited the highest misclassification rate among the tested models. This may be attributed to the overfitting behavior common in most Decision Tree models. To a large extent WNN, Cubic SVM, and Fine KNN performed well, indicating that a sample size of 29 blast observations was sufficient. A larger sample size would have provided a clearer picture of the classification ability of the selected learning models, particularly with a clear improvement in the performance of the Decision Tree model. Future work should therefore focus on a larger dataset to ascertain, with very high confidence, the performance of machine learning models.

iii. In the case of the mine investigated, the Powder Factor (PF), Stemming Length (SL), and burden (B) were discovered to be the input parameters that had the most influence on the resulting fragment throw distance. Therefore, proper consideration of these variables during the blast design process should be performed to avoid flyrock accidents in mines.

iv. The input parameters in the dataset did not exhibit a strong linear relationship when gauged against the response variable. However, this study showed that machine learning by classification can be applied successfully, even for phenomena that may not be well represented using regression analysis.

v. Regression solutions are commonly validated using metrics, such as the Coefficient of Determination (R-square), Mean Absolute Error (MAE), and Mean Squared Error (MSE). Similarly, this study illustrates that for classification problems, combining the analysis from a confusion matrix, for example, the use of accuracy, recall, and Receiver Operating Characteristic (ROC) analysis, also offers a dependable evaluation of machine model performance. This alternative approach can aid domain experts in mining in making informed decisions.

Lastly, to the best of our knowledge, this study is the first of its kind to purely use machine learning through classification intervention to show how flyrock problems can be analyzed. Therefore, it has opened up the possibility for deep consideration of machine learning by classification as a solution to tackle flyrock problems in other mines. By demonstrating the feasibility of using machine learning to solve flyrock problems, this study developed a framework for future studies to assess the performance of classification models. Future studies will aim to conduct additional test blasts to accumulate a robust dataset or sufficient sample of blast data for machine learning. Additionally, we intend to explore novel machine learning techniques, particularly those utilizing hybridization in machine learning (HML), to further evaluate their performance.

REFERENCES

[1] C. Rudin and K. L. Wagstaff, "Machine learning for science and society," *Mach. Learn.*, vol. 95, no. 1, pp. 1–9, Apr. 2014, doi: 10.1007/s10994-013-5425-9.

[2] M. Attaran and P. Deb, "Machine learning: The new 'big thing' for competitive advantage," *Int. J. Knowl. Eng. Data Mining*, vol. 5, no. 1, pp. 277–305, 2018, doi: 10.1504/ijkedm.2018.10015621.

[3] A. I. Lawal and S. Kwon, "Application of artificial intelligence to rock mechanics: An overview," *J. Rock Mech. Geotechnical Eng.*, vol. 13, no. 1, pp. 248–266, Feb. 01, 2021, doi: 10.1016/j.jrmge.2020.05.010.

[4] J. Sen, *Machine Learning: Algorithms, Models, and Applications*. London, U.K.: IntechOpen, Jan. 2022, doi: 10.5772/intechopen.94615.

[5] H. N. Rad, I. Bakhshayeshi, W. A. W. Jusoh, M. M. Tahir, and L. K. Foong, "Prediction of flyrock in mine blasting: A new computational intelligence approach," *Natural Resour. Res.*, vol. 29, no. 2, pp. 609–623, Apr. 2020, doi: 10.1007/s11053-019-09464-x.

[6] V. Kecojevic and M. Radomsky, "Flyrock phenomena and area security in blasting-related accidents," *Saf. Sci.*, vol. 43, no. 9, pp. 739–750, Nov. 2005, doi: 10.1016/j.ssci.2005.07.006.

[7] A. K. Raina, V. M. S. R. Murthy, and A. K. Soni, "Flyrock in surface mine blasting: Understanding the basics to develop a predictive regime," *Current Sci.*, vol. 108, no. 4, pp. 660–665, 2015.

[8] T. Bajpayee, S. Bhatt, T. Rehak, G. Mowrey, and D. Ingram, "Fatal accidents due to flyrock and lack of blast area security and working practices in mining," *J. Mines Metals Fuels*, vol. 51, no. 11, pp. 344–355, Jan. 2003.

[9] D. J. Armaghani, M. Hajihassani, E. T. Mohamad, A. Marto, and S. A. Noorani, "Blasting-induced flyrock and ground vibration prediction through an expert artificial neural network based on particle swarm optimization," *Arabian J. Geosci.*, vol. 7, no. 12, pp. 5383–5396, Dec. 2014, doi: 10.1007/s12517-013-1174-0.

[10] E. Ghasemi, M. Sari, and M. Ataei, "Development of an empirical model for predicting the effects of controllable blasting parameters on flyrock distance in surface mines," *Int. J. Rock Mech. Mining Sci.*, vol. 52, pp. 163–170, Jun. 2012, doi: 10.1016/j.ijrmms.2012.03.011.

[11] M. Rezaei, M. Monjezi, and A. Y. Varjani, "Development of a fuzzy model to predict flyrock in surface mining," *Saf. Sci.*, vol. 49, no. 2, pp. 298–305, Feb. 2011, doi: 10.1016/J.SSCI.2010.09.004.

[12] M. Monjezi, A. Bahrami, A. Y. Varjani, and A. R. Sayadi, "Prediction and controlling of flyrock in blasting operation using artificial neural network," *Arabian J. Geosci.*, vol. 4, nos. 3–4, pp. 421–425, Apr. 2011, doi: 10.1007/s12517-009-0091-8.

[13] M. Hasanipanah and H. B. Amnieh, "A fuzzy rule-based approach to address uncertainty in risk assessment and prediction of blast-induced flyrock in a quarry," *Natural Resour. Res.*, vol. 29, no. 2, pp. 669–689, Apr. 2020, doi: 10.1007/s11053-020-09616-4.

[14] A. Marto, M. Hajihassani, D. J. Armaghani, E. T. Mohamad, and A. M. Makhtar, "A novel approach for blast-induced flyrock prediction based on imperialist competitive algorithm and artificial neural network," *Sci. World J.*, vol. 2014, pp. 1–11, Jul. 2014, doi: 10.1155/2014/643715.

[15] R. Trivedi, T. N. Singh, and A. K. Raina, "Prediction of blast-induced flyrock in Indian limestone mines using neural networks," *J. Rock Mech. Geotechnical Eng.*, vol. 6, no. 5, pp. 447–454, Oct. 2014, doi: 10.1016/j.jrmge.2014.07.003.

[16] M. Jamei, M. Hasanipanah, M. Karbasi, I. Ahmadianfar, and S. Taherifar, "Prediction of flyrock induced by mine blasting using a novel kernel-based extreme learning machine," *J. Rock Mech. Geotechnical Eng.*, vol. 13, no. 6, pp. 1438–1451, Dec. 2021, doi: 10.1016/j.jrmge.2021.07.007.

[17] G. Bonaccorso, *Machine Learning Algorithms*. Birmingham, U.K.: Packt, 2017.

[18] T. Hudaverdi and O. Akyildiz, "A new classification approach for prediction of flyrock throw in surface mines," *Bull. Eng. Geol. Environ.*, vol. 78, no. 1, pp. 177–187, Feb. 2019, doi: 10.1007/s10064-017-1100-x.

[19] A. M. Kilic, E. Yasar, A. M. Kılıç, E. Yaar, Y. Erdoan, and P. G. Ranjith, "Influence of rock mass properties on blasting efficiency," *Sci. Res. Essay*, vol. 4, no. 11, pp. 1213–1224, 2009. [Online]. Available: http://www.academicjournals.org/SRE

[20] T. Sasaoka, Y. Takahashi, W. Sugeng, A. Hamanaka, H. Shimada, K. Matsui, and S. Kubota, "Effects of rock mass conditions and blasting standard on fragmentation size at limestone quarries," *Open J. Geol.*, vol. 5, no. 5, pp. 331–339, 2015, doi: 10.4236/ojg.2015.55030.

[21] G. E. A. P. A. Batista and D. F. Silva, *How K-Nearest Neighbor Parameters Affect Its Performance*. Accessed: Feb. 14, 2023. [Online]. Available: http://www.icmc.usp.br/

[22] A. A. Zolfaghari, R. Taghizadeh-Mehrjardi, A. R. Moshki, B. P. Malone, A. O. Weldeyohannes, F. Sarmadian, and M. R. Yazdani, "Using the nonparametric k-nearest neighbor approach for predicting cation exchange capacity," *Geoderma*, vol. 265, pp. 111–119, Mar. 2016, doi: 10.1016/j.geoderma.2015.11.012.

[23] K. Chomboon, P. Chujai, P. Teerarassammee, K. Kerdprasop, and N. Kerdprasop, "An empirical study of distance metrics for k-nearest neighbor algorithm," in *Proc. 3rd Int. Conf. Ind. Appl. Eng.*, 2015, pp. 280–285, doi: 10.12792/iciae2015.051.

[24] H. A. A. Alfeilat, A. B. A. Hassanat, O. Lasassmeh, A. S. Tarawneh, M. B. Alhasanat, H. S. E. Salman, and V. B. S. Prasath, "Effects of distance measure choice on K-nearest neighbor classifier performance: A review," *Big Data*, vol. 7, no. 4, pp. 221–248, Dec. 2019, doi: 10.1089/BIG.2018.0175.

[25] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine Learning: Methods and Applications to Brain Disorders*. Amsterdam, The Netherlands: Elsevier, 2019, pp. 101–121, doi: 10.1016/B978-0-12-815739-8.00006-7.

[26] T. Fletcher. (2008). *Support Vector Machines Explained*. [Online]. Available: https://www.cs.ucl.ac.uk/staff/T.Fletcher/

[27] A. Bahrami, M. Monjezi, K. Goshtasbi, and A. Ghazvinian, "Prediction of rock fragmentation due to blasting using artificial neural network," *Eng. Comput.*, vol. 27, no. 2, pp. 177–181, Apr. 2011, doi: 10.1007/s00366-010-0187-5.

[28] J. Zhao, Y. Yu, J. Lei, and J. Liu, "Multi-objective lower irrigation limit simulation and optimization model for *Lycium barbarum* based on NSGA-III and ANN," *Water*, vol. 15, no. 4, p. 783, Feb. 2023, doi: 10.3390/w15040783.

[29] K. L. Priddy and P. E. Keller, *Artificial Neural Networks: An Introduction (SPIE Tutorial Texts in Optical Engineering, Vol. TT68*. Bellingham, WA, USA: SPIE, 2005, Accessed: Feb. 17, 2023. [Online]. Available: https://books.google.com/books/about/Artificial_Neural_Networks.html?id=BrnHR7esWmkC

[30] M. Bansal, A. Goyal, and A. Choudhary, "A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning," *Decis. Anal. J.*, vol. 3, Jun. 2022, Art. no. 100071, doi: 10.1016/J.DAJOUR.2022.100071.

[31] M. Jaworski, P. Duda, and L. Rutkowski, "New splitting criteria for decision trees in stationary data streams," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2516–2529, Jun. 2018, doi: 10.1109/TNNLS.2017.2698204.

[32] M. Awad and R. Khanna, *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Berkeley, CA, USA: Apress, Jan. 2015, pp. 1–248, doi: 10.1007/978-1-4302-5990-9.

[33] J. Fan, S. Upadhye, and A. Worster, "Understanding receiver operating characteristic (ROC) curves," *CJEM*, vol. 8, no. 1, pp. 19–20, Jan. 2006, doi: 10.1017/S1481803500013336.

[34] J. V. Carter, J. Pan, S. N. Rai, and S. Galandiuk, "ROC-ing along: Evaluation and interpretation of receiver operating characteristic curves," *Surgery*, vol. 159, no. 6, pp. 1638–1645, Jun. 2016, doi: 10.1016/j.surg.2015.12.029.

[35] T. R. Rehak, T. S. Bajpayee, G. L. Mowrey, and D. K. Ingram, "Flyrock issues in blasting," in *Proc. 27th Conf. Explos. Blasting Tech.*, 2001, pp. 165–175.

IAN KROP received the B.S. degree in mining engineering from the Jomo Kenyatta University of Agriculture and Technology and the M.S. degree in mining engineering from the Wuhan University of Science and Technology. He is currently pursuing the Ph.D. degree in mining engineering with Kyushu University. His research interests include rock blasting engineering, artificial intelligence in mining, mine machinery, and materials handling.

YOSHIAKI TAKAHASHI received the Ph.D. degree in engineering from Kyushu University, Fukuoka, in 2019. He has been working with the Research Institute of Science for Safety and Sustainability, Advanced Industrial Science and Technology (AIST), since 2019. He currently focuses on blasting technology in mining and in the civil engineering field. He has engaged in projects on blasting and demolition of buildings in urban areas, as well as ground vibrations caused by explosions.

AKIHIRO HAMANAKA received the Ph.D. (Eng.) degree from Kyushu University, Fukuoka, in 2015. He is currently an Assistant Professor with Kyushu University. At that time, he studied environmental issues in open-cut coal mines, such as soil erosion, acid mine drainage, and revegetation. He has some projects related to environmental issues of open-cut mines in Southeast Asia.
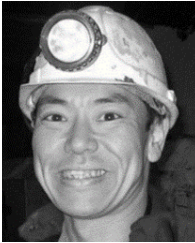
TAKASHI SASAOKA received the Dr. (Eng.) degree from Kyushu University, Fukuoka, Japan, in 2003. He is currently an Associate Professor with the Laboratory of Rock Engineering and Mining Machinery, Department of Earth Resources Engineering, Kyushu University. He has engaged in projects, such as controlled blasting and ground control in underground openings.

HIDEKI SHIMADA received the Dr. (Eng.) degree from Kyushu University, Fukuoka, Japan, in 1993. He is currently a Professor with the Laboratory of Rock Engineering and Mining Machinery, Department of Earth Resources Engineering, Kyushu University. He has participated in several projects, such as the utilization of cemented fly ash, development and application of pipe jacking in the civil engineering field, rehabilitation, and re-vegetation of open-cut coal mines in Asia.

JOAN ONYANGO received the B.S. degree in mechanical engineering from the Jomo Kenyatta University of Agriculture and Technology and the M.S. degree in mining engineering from the Wuhan University of Science and Technology. She is currently pursuing the Ph.D. degree in mining engineering with Kyushu University. Her research interests include rock mechanics, rock engineering, and pit slope stability.

● ● ●