Development of a predictive model of rock fragmentation for Nui Phao open-pit mine in Vietnam using multiple-output Neural Networks and Monte Carlo Dropout technique

Trong Vu (✉ trongvu@qui.edu.vn)

Quang Ninh University of Industry: Truong Dai hoc Cong nghiep Quang Ninh <https://orcid.org/0000-0001-8765-387X>

Tran Bao

Hanoi University of Mining and Geology

Original Research

Keywords: blast fragmentation, cumulative distribution function, multiple-output neural network, single-output neural network, Monte Carlo dropout, uncertainty

Posted Date: January 29th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-171960/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Abstract

Precise and reliable prediction of blast fragmentation is essential for the success of mining operations. Over the years, various machine learning models using artificial neural network have developed and proven to be efficient in predicting the blast fragmentation. In this research, we design multiple-output neural networks to forecast the cumulative distribution function (CDF) of blast fragmentation to improve this prediction. The model architecture contains multiple response variables in the output layer that correspond to the CDF curve's percentiles. We apply Monte Carlo dropout procedure to estimate the uncertainty of the predictions. Data collected from a Nui Phao open-pit mine in Vietnam are used to train and validate the performance of the model. Results suggest that multiple-output neural network models provide better accuracy than single-output neural network models that predict each percentile on a CDF independently. Whereas, Monte Carlo dropout technique can give valuable and relative reliable information during decision making.

Article highlights:

- Precise and reliable prediction of blast fragmentation is essential for the success of mining operations.
- A predictive model based on the multi-output neural network and Monte Carlo dropout technique was designed to predict the fragmentation CDF curve in the blasting operation of an open-pit mine.
- The predictive model was proven reliable and provided better accuracy than models based on a single-output neural network.

1. Introduction

Blasting is an essential process in open-pit mining operations. Their results significantly impact the efficiency of downstream processes such as crushing and grinding, loading, haulage, and processing in terms of cost and production. Proper fragmentation is the primary goal of blasting. Therefore, a precise and reliable model for predicting blast fragmentation distribution is incredibly beneficial for mining operations. Blast parameters are defined as the optimisation of blast design to obtain a good fragmentation.

Open-pit mine blasts result in a widespread muck pile containing thousands of individual rock fragments with different geometric properties. The cumulative distribution function (CDF) curve of fragmentation is usually deployed to evaluate the distribution of rock fragments within a blast muck pile and understanding the physics of blasting parameters. Given the percentile values of fragmentation size obtained by sieving analysis, the CDF curve computes rock fragments' distribution as mass weights of the remaining fragments on each screen with a known mesh size.

Different empirical models have been introduced for constructing the CDF curve. For instance, blast fragmentation distribution can be referred to as the Rosin-Rammler or Weibull distributions (Bennett

1936; PAUL Rosin 1933; Paul Rosin, Rammler, and Sperling 1933). The works by Koshelev et al. (Koshelev et al. 1971) and Kuznetsov (Kuznetsov 1973) explored the relations between fragment sizes and the explosive charging parameters or the hardness of blasted rock. Cunningham (Cunningham 1983) proposed the Kuz-Ram model using Kuznetsov's equation and Rosin-Rammler distribution for estimating the fragment sizes and the uniform index of blast muck piles. Nevertheless, some restrictions on using these experimental models due to the intensive labour work needed to determine the factors in the equations and the experiments' bias in some specific situations.

In recent years, blasting researchers have expressed their concerns about the usefulness of machine learning models when predicting blast fragmentation. Actual data collected from fragmentation measurement using computer vision programs have motivated researchers to design more effective prediction models. Several multi-variate regression algorithms have been applied to capture most blast parameters (Chakraborty et al. 2004; Hudaverdi, Kulatilake, and Kuzu 2011). However, the nonlinear relationship of blast parameters is still a challenging task for these models. Shi *et al.* (X. Z. Shi et al. 2012) introduced the Support Vector Machines (SVM) to forecast the mean fragmentation size for bench blasting, accounting for blast design parameters and rock mass properties. The result indicates that the SVM model is scientific and feasible with high accuracy.

Remarkably, Artificial Neural Networks (ANN) models have been attracted to many researchers in areas of blast fragmentation prediction due to their ability of self-learning and organisation, nonlinear pattern recognition, and generalisation. Various authors applied the ANN model using blast design parameters and rock mass properties to predict the fragmentation. Oraee and Asi (Oraee and Asi 2006) represented an ANN model trained by 13 input parameters and three output parameters in Gol-e-Gohar iron ore mine to predict three percentile values of the fragmentation CDF curve. Other authors developed different ANN models with multiple inputs and a single output to predict a single percentile value of the fragmentation CDF curve or the average fragmentation size. For instance, Kulatilake *et al.* (Kulatilake et al. 2010) evaluated four learning algorithms to train the ANN model and show the ability to predict more accurate fragmentation sizes of the ANN model and the multi-variate regression model. Sayadi *et al.* (Sayadi et al. 2013) performed a comparative study on applying two different ANN model structures to predict blast fragmentation and back break. Shi *et al.* (X. Shi et al. 2013) and Enayatollahi *et al.* (Enayatollahi, Bazzazi, and Asadi 2014) proved better predictability of the ANN model than the empirical formulas obtained by regression analysis. JA Rosales-Huamani *et al.* (Rosales-Huamani et al. 2020) designed an ANN model that outputs three results, defining 80th, 50th, and 20th percentiles of fragmentation sizes.

Even though the developed ANN models above tend to produce better fragmentation estimates than other empirical formula and linear regression models, direct estimation of individual percentiles often has nonmonotonic CDF curves. Hence, postprocessing is required to replace violating percentiles or resorting predicted percentiles. These models often ignore the relation between percentiles and the connection between each percentile and the associated blast parameters. JA Rosales-Huamani *et al.* (Rosales-Huamani et al. 2020) also reported that their model produced a moderate accuracy since it tends to focus

on the target of the 80th percentile while decreasing the effect on the 50th and 20th percentile. There is a pressing need for multi-output learning paradigms as a solution.

Another issue is that standard ANN models for regression only return value predictions given some input variables. By definition, the value prediction is an estimate and an approximation and contains some uncertainty. The uncertainty may result from the model errors or noise in training data or unknown parameters like blasting engineers' skills or blasted rock's properties. Therefore, the value predictions are insufficient for describing the real uncertainty of the forecast. Gal and Ghahramani (Gal and Ghahramani 2016) introduced a simple Monte Carlo Dropout (MCD) method to capture model uncertainty. They have discovered that training any ANNs with dropout is used to prevent overfitting and could be interpreted as an approximate of weight's posterior. The trained ANNs with dropout can make multiple predictions to estimate uncertainty. The MCD mechanism has mainly been applied for image processing tasks with convolutional neural networks (CNNs), where the uncertainty can be evaluated visually (Kim, Kim, and Choe 2020; Myojin et al. 2019; Stoean et al. 2020). Uber rides also used MCD combined with Bayesian NN for the large-scale time series anomaly detection (Zhu and Laptev 2017). A probabilistic wind power forecasting model based on Bayesian theory and MCD is presented in (Wen et al. 2019). The employment of MCD within the deep learning structure in healthcare was shown in (Wickstrøm, Kampffmeyer, and Jenssen 2020).

This study aims to improve the prediction of the CDF curve of blast fragmentation size in bench blasting with a multi-output neural network (MNN) and MCD. With this aim, we undertake a hold-out sample forecasting competition and compare the performance of MNN and single-output neural network (SNN). We demonstrate the ability of MNN in simultaneously predicting multiple and monotonically increasing percentiles of the fragmentation CDF, resulting in better predictions than SNN. MCD is applied to obtain an empirical distribution of the CDF curve, rather than a single CDF curve, allowing us to access the uncertainty in predictions.

The paper proceeds as follows. The first section introduces and reviews the literature on forecasting the blast fragmentation distribution with ANNs. The second section then describes the data used in the study and the design of the MNN model. In the following section, experimental the results are presented and compared. Finally, the discussion and conclusion complete the paper.

2. Methods

2.1 Data

The data consists of 903 samples of blasting records collected at Nui Phao open-pit mines, Vietnam, from 2013 until 2019. The mine applies the bench blasting technique for breaking waste rock and ore. The blasting parameters can be classified into controllable and uncontrollable groups. The controllable parameters, including explosive type, burden, blast holes spacing, bench height, drilling depth, explosive charge length, sub-drilling length, and stemming length, are shown in Fig. 1 and Table 1. The hardness of rock and ore, which is classified into soft, medium, and hard, is uncontrollable. Fortis7030, Extra100,

Extra7030, Anfo, Package, Extra6535, ANE, or mixing Package&Anfo, Fortis7030&Anfo, Fortis7030&Flexigel are primary explosives used in the mine. In Table 1, percentile values were mapped using the image analysis method. High-quality images representing the blasted muck pile are chosen to be analysed using Split-desktop software (LLC 2016).

The scale and distribution of the input data are different from each other. Hence, the numeric data are scaled to lie between 0 and 1. Simultaneously, the categorical data such as rock hardness and explosive type are converted into dummy encoding technique.

Figure 2 shows the strong and positive correlations between all output parameters. These correlations are inherent due to the monotonic relationship between output parameters as percentiles of the CDF curve. Besides, we observed no correlations for all input parameters.

Table 1
Description of the input and output parameters

Parameters	Description	Min	Max
Input	B: Burden (m)	2.2	4.0
	S: Spacing (m)	2.2	4.5
	L _s : Stemming length (m)	0.0	3.5
	Ls: Sub-drilling length (m)	0.0	1.0
	H: Bench height (m)	1.1	10.5
	L: Drilling depth (m)	1.3	17.2
	D: Drill hole diameter (mm)	5.6	115.0
	N: Number of drill holes	34	794
	P: Powder factor (kg/m ³)	0.36	1.21
	V: Blasted Volume (m ³)	1031	48027
	F10: 10th percentile (cm)	0.11	27.45
Output	F20: 20th percentile (cm)	0.3	44.91
	F30: 30th percentile (cm)	0.39	60.17
	F40: 40th percentile (cm)	0.49	73.07
	F50: 50th percentile (cm)	0.6	83.74
	F60: 60th percentile (cm)	0.73	93.63
	F70: 70th percentile (cm)	0.86	104.57
	F80: 80th percentile (cm)	1.0	119.08
	F90: 90th percentile (cm)	1.2	137.15

2.2 ANN models

2.2.1 Basic structure of an ANN model

The basic ANN configuration comprises artificial neurons or nodes that simulate the human brain's biological neurons. Figure 3 illustrates the simple operations of an artificial neuron. A neuron receives and sums all signals with the associated weights from its connected neurons on each connection. Each neuron then executes an activation function to give the final output. Logistic, Tanh, and Rectified Linear Unit (ReLU) are commonly used functions. For each layer, a neuron also has a bias to adjust the output following the weighted sum of inputs.

Neurons are grouped into one input layer, one or more hidden layers, and one output layer (Fig. 4). Each neuron uses a transfer function on hidden layers to fully connect to all neurons on the next layer. The back-propagation algorithm is widely used for training ANN models. It propagates and minimises the loss (error) between the output and the target back into the network. The associated weights on each connection are modified along with the initial loss. The input data then fed forward again, generating a new output and loss. The process is repeated until the loss obtains an acceptable value.

2.2.2 MNNs against SNNs

A typical regression task involves predicting a single outcome of interest. Consequently, the output layer has one single neuron (SNN model). However, some tasks require predicting more than a single result. These tasks are referred to as multiple-output regression (Borchani et al. 2015). The output layer has two or more neurons (MNN model), simultaneously predicting a set of outcomes. While SNN models ignore the correlations between outputs, MNN models leverage them to improve predictive accuracy (Breiman and Friedman 1997). Our goal is to leverage the monotonic relationship between fragmentation percentiles, as shown in Fig. 2. In the training process, the loss is computed for each target output before averaging them to adjust the connection weights.

2.2.3 ANN model uncertainty using Monte Carlo dropout

Dropout is commonly used in ANN models to avoid over-fitting (Srivastava et al. 2014) or prevent hidden neurons from learning unnecessary training data features (Hinton et al. 2012). In training time, standard dropout randomly eliminates neurons from their layer by a given probability p (Fig. 4). However, during the test time, dropout is not applied, and the model always gives the same single value. For MCD (Gal and Ghahramani 2016; Srivastava et al. 2014), the dropout is used at both training and test time. At the test time, the model could predict different values each time, depending on which neurons are randomly chosen to keep. Ghahramani (Gal and Ghahramani 2016) demonstrated that MCD could be used as a Bayesian approximation of a Gaussian process to capture model uncertainty. For the same input, one can run the ANN model multiple times to obtain a set of predictions resulting in a distribution or determine statistical values such as mean or standard deviation.

2.2.4 Model training and testing

The objective of MNN and SNN configuration is to optimise the prediction accuracy. The networks' input layer has 27 neurons, one for each blasting parameters after being normalised and encoded. The MNN contains nine neurons in the output layer representing nine percentiles of fragmentation distribution. The SNN has one neuron in the output layer generating one percentile to be estimated. Consequently, nine individual SNN models need to build to calculate each required percentile. We provided later details on other features of the models in Sect. 2.2.5. Both models were fit using Keras 2.3.1 (Chollet 2015) with a TensorFlow backend.

A tenfold cross-validation approach was used to tune the model hyperparameters and evaluate the model performance. The procedure involves the following steps:

1. The dataset is randomly partitioned into ten parts, 90% for training, and the remaining 10% as a hold-out set for testing.

2. The 90% training set is further randomly separated into two groups: 90% for training and 10% for validation.

a. First, the network is initialised using random weights. The weights and inputs traverse through all neurons and activation functions from the first to the last layer. This forward pass outputs nine percentile values for each observation for both training and validation sets.

b. In the next step, the network measures how good the outputs

$$(y_j)$$

is by a cost function called mean squared error (MSE):

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

The MSE for the training set is backpropagated and updates the weights and biases to optimise the output values. The stochastic gradient descent is employed as the optimiser for this step. The network continues to adjust by running through new observations from the training set.

c. The early stopping strategy was applied to prevent the network from overfitting the training dataset. The training process is terminated after 500 epochs when the MSE is not improved significantly. The final weights are then evaluated on the hold-out set, which was wholly partitioned in step 1.

3. Step 1 and 2 are repeated ten times, giving each observation a chance to train the models nine times and in the hold out set one time. The ten-fold cross-validation process results are the mean of the MSE, which measures the model performance on an unseen dataset.

2.2.5 Hyperparameter optimisation

Hyperparameter optimisation aims to find the set of ANN model configuration arguments that yield the best performance measured on the dataset. The criteria for selecting the final hyperparameter include choosing the model with the lowest cross-validated MSE. Besides, the chosen hyperparameters should generate zero monotonic violations in forecasting the nine percentiles of the CDF curve. The number of monotonic violations was monitored after each training epoch by:

$$e_m = \sum_{i=1}^n \sum_{j=1}^m (\max(y_j) \neq y_j) \quad (2)$$

where n is the number of samples in the training dataset, m is the number of percentiles ($m=9$), max refers to the maximum value of percentiles in the sequence. Each predicted percentile value is checked against the max. If the percentile value is not equal to the max, it will break the monotonicity. For instance, if the nine percentiles were 10, 20, 30, 15, 25, 28, 40, 50, and 60 cm, there would be a total of three monotonic violations because 15, 25, and 28 are not the max (30 cm) in their sequence.

There are many ways to implement hyperparameter optimisation, such as manual, grid, or random search. In this study, we chose Bayesian Optimization to optimise the hyperparameters of the two models. Bayesian Optimization searches the optimum hyperparameters using a surrogate probability model of hyperparameter values. The objective function is much easier to optimise than the actual one in training the machine learning model. Bayesian approaches work by finding the next set of hyperparameters that perform the best on the surrogate model's objective function before evaluating them on the actual objective function. It is noted that the time spent on selecting the next promising hyperparameters from the past results is not significant in comparison with the time spent on the objective function. Therefore, Bayesian Optimisation fast and effective in comparison with other methods.

The range of hyperparameters for the MNN and SNN models are shown in Table 2. The Bayesian Optimisation was run for 50 iterations using ten-fold cross-validations on the full dataset (903 samples). The Adam optimisation algorithm (Kingma and Ba 2014) was fixed to optimise the models. The hyperparameters with the best model performance (lowest MSE) on hold-out sets were chosen to be trained in the next section. We deployed the Python library Hyperopt (Bergstra, Yamins, and Cox 2013) to run the Bayesian Optimisation.

Table 2
Hyperparameters considered for the SNN and MNN models

Hyperparameters	Values
Number of hidden layers	{1,2,...,10}
Number of neurons in the input layer	{30,40,...,450}
Number of neurons in hidden layers	{30,40,...,450}
Activation function	{Relu, Tanh, Sigmoid}
Dropout rate after input and hidden layers	[0-0.3]
Learning rate	{0.01,0.001,0.0001 }
Batch size	{10,20,...,300}

The final model is structured by one input layer, three hidden layers and one output layer. The input layer contains 27 neurons using the dropout rate of 0.045. The first hidden layer is formed with 260 neurons with a dropout rate of 0.143 while both second and the third layers consist of 440 neurons with a dropout rate of 0.101. For the output, the final layer owns nine neurons corresponding to nine percentiles of the CDF curve. All neurons execute the ReLU as an activation function on the input value. In the training process, the learning rate of 0.001 and the batch size of 210 were applied.

3. Results

3.1 Training and testing

Figure 5 shows a similar pattern for each fold (represented by the individual green lines) in the training and validation process. The error MSE of the MNN models on the validation data drops dramatically for the first 50 epochs and then slowly decreases for another 100–150 epochs before getting stability after the 200th epoch. The number of monotonic violations also goes down for the first 50 epochs but with a slower rate. It arrives at zero after the 400th epoch. The SNN models generate 145 or approximately 22% of predicted percentile monotonic violations on the validation data.

Not only producing nonmonotonic CDF curves on validation data, but the SNN models are also generally less accurate than MNN models on testing data. Some examples of predictions by SNN and MNN models on testing data are illustrated in Fig. 6a. The absolute error in prediction accuracy tends to increase for larger percentiles for both SNN and MNN models, as shown in Fig. 6b. Nevertheless, the MNN models generate a lower absolute error than the SNN models for all percentiles, except for the 90th percentile, which is the same for both. The discrepancy in prediction accuracy is distinctive in lower percentiles. Given a better performance of the MNN models, they are used for further analysis and all results hereafter refer to the MNN models.

3.2 Dropout uncertainty

Figure 7 shows three examples of 1000 predicted CDF curves using the Monte Carlo dropout procedure. The realisations of CDF curves mostly capture the true CDF curves. The coverage probabilities were computed to determine how well the uncertainty from MCD is calibrated to the observations. For well-calibrated predictions, the green shade in Fig. 8 should be as close as possible the diagonal black dashed line. This would correspond to the situation in which a given prediction interval should contain the fraction of observations. For example, a 50% prediction interval should include 50% of the observations. In Fig. 8, the coverage probabilities for prediction intervals ranging from 5–100% were constructed for each percentile. The prediction intervals for all percentiles are relatively well-calibrated. Nevertheless, the MNN model tends to be overconfident, mostly for large prediction intervals. The overconfident prediction can be observed clearly in 20th percentile where its coverage probabilities are below the “ideal” line.

4. Discussion

The loss for each output is averaged before back-propagation in training an MNN. A single training epoch for an MNN is computationally equivalent to that for an SNN. Therefore, training an MNN does not increase the computation burden in learning the model weights than training an SNN. Furthermore, SNN requires optimising hyperparameters and training independently for each output of interest. Whereas, MNN is a more straightforward approach since it only needs a single-time method for optimising hyperparameters and training.

Unlike the SNN where a single value is predicted for each percentile of the CDF curve of blast fragmentation, the MNN leverages the relation between percentiles to forecast a CDF curve that does not violate the monotonicity simultaneously. For instance, if the MNN predicts a CDF curve of small-sized fragmentations for a particular blast, the value for the 90th percentile will be smaller than it might have been if predicted independently because the MNN shares the information between percentiles. Compared to the SNN, the MNN can provide better accuracy in predictions. The MNN, by its structured inference, is suitable for the multi-variate nature and complex interactions of the multiple outputs used to represent the CDF curve of blast fragmentation.

One of the most prominent and essential benefits of MCD is the uncertainty information it provides. Since blasting operation is affected by many complicated factors, one cannot guarantee a perfect prediction model. MCD can be considered an ensemble of networks to generate a set of predictions (Fig. 7). This prediction set's mean and variance can be used to estimate how confident the model is with its predictions and provide valuable information during decision making. For instance, in the case of high uncertainty, more tests and data collection should be implemented or eventually passing the case to a human to avoid potentially wrong results.

The reliability plots in Fig. 8 suggest that prediction uncertainty under the MNN model is relatively reliable for all percentiles of the CDF curve. However, more work is needed to improve the calibration of the SNN model. The model seems to be overconfident when predicting the unseen test data. This miscalibration is generally observed for model uncertainty provided by MCD (Bishop 2006; Kendall and Gal 2017) since it is only applied to obtain uncertainty in the model weights due to training with limited data and knowledge (Gal, Hron, and Kendall 2017; Guo et al. 2017).

5. Conclusion

Precise and reliable prediction of blast fragmentation is essential for the success of mining operations. In this paper, an attempt is made to design the MNN for predicting the fragmentation CDF curve in the blasting operation of an open-pit mine. Based on k-fold validation, we applied Bayesian Optimization to search the optimum hyperparameters and then trained the MNN with MCD. In the competition with the SNN, the MNN leverages the relation between percentiles of the CDF curve to provide better accuracy with no monotonic violations. Besides, MNN is a simpler approach in terms of structure and effort for tuning and training. To improve the reliability of MNN, we applied the MCD mechanism in both training and

predicting. The MNN model generates multiple CDF predictions, providing uncertainty estimates to the predictions with more trustworthy.

Similarly to other types of ANNs with MCD, the MNN tends to be miscalibrated. Future works will focus on improving the calibration of MNN model and more real data from different mines to verify the prediction accuracy and evaluate other associated costs.

Declarations

Acknowledgement

The authors would like to extend their sincere gratitude to Nui Phao mining company for providing data for this research and Hanoi University of Mining and Geology, Vietnam, and Quang Ninh University of Industry, Vietnam for computing resources and technical support.

Compliance with ethical standards

Conflict of interest The authors declare no conflict of interest.

References

Bennett, J G. (1936). Broken Coal. *J. Inst. Fuel* 10: 22–39.

Bergstra, James, Daniel Yamins, and David Cox. (2013). Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *International Conference on Machine Learning*, PMLR, 115–23.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Borchani, Hanen, Gherardo Varando, Concha Bielza, and Pedro Larrañaga. (2015). A Survey on Multi-output Regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5(5): 216–33.

Breiman, Leo, and Jerome H Friedman. (1997). Predicting Multivariate Responses in Multiple Linear Regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 59(1): 3–54.

Chakraborty, A K et al. (2004). Parametric Study to Develop Guidelines for Blast Fragmentation Improvement in Jointed and Massive Formations. *Engineering Geology* 73(1–2): 105–16.

Chollet, F. (2015). Keras. <https://github.com/keras-team/keras>.

Cunningham, Claude. (1983). The Kuz-Ram Model for Production of Fragmentation from Blasting. In *Proc. 1st Symp. on Rock Fragmentation by Blasting, Lulea*.

- Enayatollahi, Iman, Abbas Aghajani Bazzazi, and Ahamad Asadi. (2014). Comparison between Neural Networks and Multiple Regression Analysis to Predict Rock Fragmentation in Open-Pit Mines. *Rock mechanics and rock Engineering* 47(2): 799–807.
- Gal, Yarin, and Zoubin Ghahramani. (2016). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *International Conference on Machine Learning*, 1050–59.
- Gal, Yarin, Jiri Hron, and Alex Kendall. (2017). Concrete Dropout. In *Advances in Neural Information Processing Systems*, 3581–90.
- Guo, Chuan, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. (2017). On Calibration of Modern Neural Networks. *arXiv preprint arXiv:1706.04599*.
- Hinton, Geoffrey E et al. (2012). Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors. *arXiv preprint arXiv:1207.0580*.
- Hudaverdi, T, PHSW Kulatilake, and Cengiz Kuzu. (2011). Prediction of Blast Fragmentation Using Multivariate Analysis Procedures. *International Journal for Numerical and Analytical Methods in Geomechanics* 35(12): 1318–33.
- Kendall, Alex, and Yarin Gal. (2017). “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *Advances in Neural Information Processing Systems*, 5574–84.
- Kim, Yoon-Chul, Khu Rai Kim, and Yeon Hyeon Choe. (2020). Automatic Myocardial Segmentation in Dynamic Contrast Enhanced Perfusion MRI Using Monte Carlo Dropout in an Encoder-Decoder Convolutional Neural Network. *Computer methods and programs in biomedicine* 185: 105150.
- Kingma, Diederik P, and Jimmy Ba. (2014). Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.
- Koshelev, E A, V M Kuznetsov, S T Sofronov, and A G Chernikov. (1971). Statistics of the Fragments Forming with the Destruction of Solids by Explosion. *Journal of Applied Mechanics and Technical Physics* 12(2): 244–56.
- Kulatilake, PHSW, Wu Qiong, T Hudaverdi, and C Kuzu. (2010). Mean Particle Size Prediction in Rock Blast Fragmentation Using Neural Networks. *Engineering Geology* 114(3–4): 298–311.
- Kuznetsov, V M. (1973). The Mean Diameter of the Fragments Formed by Blasting Rock. *Soviet Mining* 9(2): 144–48.
- LLC, Split Engineering. (2016). Split-Desktop 3.0.
- Myojin, Tomoyuki et al. (2019). Improving Reliability of Object Detection for Lunar Craters Using Monte Carlo Dropout. In *International Conference on Artificial Neural Networks*, Springer, 68–80.

- Oraee, Kazem, and Bahareh Asi. (2006). Prediction of Rock Fragmentation in Open Pit Mines, Using Neural Network Analysis. In *Fifteenth International Symposium on Mine Planning and Equipment Selection (MPES 2006)*,.
- Rosales-Huamani, Jimmy Aurelio, Roberth Saenz Perez-Alvarado, Uwe Rojas-Villanueva, and Jose Luis Castillo-Sequera. (2020). Design of a Predictive Model of Rock Breakage by Blasting Using Artificial Neural Networks. *Symmetry* 12(9): 1405.
- Rosin, Paul. (1933). Laws Governing the Fineness of Powdered Coal. *Journal of Institute of Fuel* 7: 29–36.
- Rosin, Paul, Erich Rammler, and K Sperling. (1933). *Korngrößenprobleme Des Kohlenstaubes Und Ihre Bedeutung Für Die Vermahlung*. VDI-Verlag.
- Sayadi, A, M Monjezi, N Talebi, and Manoj Khandelwal. (2013). A Comparative Study on the Application of Various Artificial Neural Networks to Simultaneous Prediction of Rock Fragmentation and Backbreak. *Journal of Rock Mechanics and Geotechnical Engineering* 5(4): 318–24.
- Shi, X Z et al. (2012). Support Vector Machines Approach to Mean Particle Size of Rock Fragmentation Due to Bench Blasting Prediction. *Trans Nonferrous Met Soc China* 22(2): 432–41.
- Shi, Xiuzhi, Dan Huang, Jian Zhou, and Shu Zhang. (2013). Combined ANN Prediction Model for Rock Fragmentation Distribution Due to Blasting. *Journal of information & computational science* 10(11): 3511–18.
- Srivastava, Nitish et al. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The journal of machine learning research* 15(1): 1929–58.
- Stoean, Catalin et al. (2020). Automated Detection of Presymptomatic Conditions in Spinocerebellar Ataxia Type 2 Using Monte Carlo Dropout and Deep Neural Network Techniques with Electrooculogram Signals. *Sensors* 20(11): 3032.
- Wen, Honglin, Jie Gu, Jinghuan Ma, and Zhijian Jin. (2019). Probabilistic Wind Power Forecasting via Bayesian Deep Learning Based Prediction Intervals. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, IEEE, 1091–96.
- Wickstrøm, Kristoffer, Michael Kampffmeyer, and Robert Jenssen. (2020). Uncertainty and Interpretability in Convolutional Neural Networks for Semantic Segmentation of Colorectal Polyps. *Medical Image Analysis* 60: 101619.
- Zhu, Lingxue, and Nikolay Laptev. (2017). Deep and Confident Prediction for Time Series at Uber. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, IEEE, 103–10.

Figures

Figure 1

Blasting parameters used in this study: S - spacing between two adjacent blast holes; B – burden; H – bench height; L – drilling depth; Le – explosive charge length; Ls – sub-drilling length; Lt – stemming length.

Figure 2

Correlation matrix of percentile values (output) of fragmentation size

Figure 3

Operation model of an artificial neuron

Figure 4

Example of ANN structure with i inputs, k hidden layers, and o outputs. The halted neurons are randomly dropped out with some probability p for each layer in the training process.

Figure 5

MSE (green lines) and the number of monotonic violations (red lines) against training epochs for the validation set. Each line records the predictions for the validation set from tenfold cross-validation.

a)

b)

Figure 6

(a) Three examples of predicted CDF curves generated by SNN model (red line) and MNN model (green line); (b) The median absolute error (dashed line) and the 90% interval of the error for each model across all nine predicted percentiles.

Figure 7

Examples of 1000 CDF realisations (the green lines) with Monte Carlo dropout. The black lines show the true CDFS.

Figure 8

A reliability plot of coverage probabilities for nine percentiles where the black dashed line indicates perfect reliability. The y-axis represents the percentage of observations within the prediction interval in the x-axis.