Enhancing blasting vibration prediction accuracy using hybrid ML models

Paulo Lopes[1]
Hernani Lima[2]

Abstract

Blasting is the main method of rock excavation in mining and civil engineering. However, ground vibration induced by blasting often causes cracks and even collapses of surrounding structures. Traditional empirical methods struggle to provide a reliable explosion prediction model to design blasting parameters due to the complexity of explosion physics and mining area topography. The PSO-GBDT, PSO-XGboost, and Random forest(RF) hybrid ensemble machine learning (ML) models are given and compared in this research. To prevent overfitting, K-cross validation is used in hyper-parameters tuning. A hybrid stacking ML model based on PSO-RF-XGboost and a data-augmented method is developed with the goal of increasing the accuracy of blasting prediction. To shed light on the viability and interpretability of the model, a variety of interpretability analyses—including sample required numbers analysis and dimension contribution analysis based on SHAP—are discussed.

Keywords

blasting, vibration, accuracy, machine learning

[1] Co-founder - Beyond Mining - paulo@beyondmining.tech
[2] Researcher and Professor - UFOP - hernani.lima@ufop.edu.br

Introduction

In the process of open-pit mining, blasting is the most important means of rock crushing (Navarro Torres et al. - 2018). The exploitation of blasting has a negative effect on ground vibration-induced structure damage and human suffering in adjacent areas (Raina et al. - 2004). Researchers typically utilize the particle peak vibration velocity (PPV) to assess the degree of damage caused by blasting operations on buildings because the damage caused by ground vibration is frequently challenging to quantify. Many academics forecast PPV and evaluate the prediction using R2, the fraction of the variation for PPV, primarily taking into account the maximum charge quantity (Qmax) of each delay and the distance (R) between the blasting surface and the monitoring point.

Due to the numerous influencing elements brought on by intricate blasting operations and diverse topography, the process of predicting ground vibration is challenging. In this situation, empirical methods are frequently applied. Two classic empirical equations are shown below, proposed by Holmberg and Persson (1979) and Ambraseys and Hendron (1968), respectively (Khandelwal and Singh - 2009).

$$PPV = aQ^{b}DC \tag{1}$$

$$PPV = \lambda\left(\frac{D}{\sqrt[2]{Q}}\right)^{-\alpha} \tag{2}$$

a, b, c, α, λ are the constants related to local rock property and explosive design.

Despite being explicit and practical, these empirical prediction methods occasionally fail to produce the expected results within the permitted margin of error with R2 fluctuating below 0.5. First, a variety of influencing factors related to blasting operations are difficult to account for using empirical methods based on on-site blasting tests and mathematical modeling analysis. Second, the geography of distinct open-pit mines varies in terms of the types of rock, soil, and distribution of rocks, etc. The fact that these parameters are interwoven and ultimately form a complex nonlinear relation- ship with the blasting result makes it much more complicated to predict. As a result, achieving a steady prediction accuracy for blasting operation analysis using empirical methods is sometimes challenging.

The traditional empirical modeling's limited capacity for generalization can be compensated for by supervised learning techniques. For the purpose of being able to fit the nonlinear relationship, models based on supervised learning theory can take into account all aspects impacting PPV simultaneously (Kulatilake et al. - 2010). Therefore, supervised learning theory has been applied to the prediction of blasting vibrations with examples including principal component analysis, support vector machines, and adaptive neural networks. The BP neural network method was

proposed by Kulatilake and Hudaverdi to forecast the blasting effect (Esmaeili et al. - 2015). In order to estimate the impact of blasting, in 2014 Mohammad Eamaeil employed principal component analysis, support vector machines, and adaptive neural networks into the prediction of blasting (Armaghani et al. - 2015).

At the same time, hybrid ML models are also chosen by researchers in this field. Khandelwal used artificial neural network (ANN) technology to predict the ground vibration with rock properties and blasting parameters (Saghatforoush - 2016). Saghatforoush combines artificial neural network (ANN) and ant colony algorithm to establish an optimized model to predict the hazards in throwing blasting (Marto et al. - 2014).The risks of vibration impact are reduced to assure the secondary risks in throwing blasting by adjusting the blasting mode parameters (Hudaverdi - 2012). Turker Hudaverdi used a multivariate analysis procedure to predict blasting (Zhang et al. - 2021). He grouped the blasting data of rock mining by hierarchical clustering and classified them by linear discriminant equation and analyzed the survey data set by using the multiple regression analysis method in statistics.

But when it comes to open-pit mine blasting, the current supervised learning models still have some serious drawbacks. Due to the high cost of sensors, on-site blasting tests are costly to be recorded. It is hard to provide the abundant data size that supervised learning models need. The property of imbalanced sample distribution also makes the ML model sensitive to tuning changes in the hyper- parameters. In order to use data wisely and fully, effective approaches based on interpretability analysis should be put into practice (Khandelwal and Singh - 2009).

Objective and Solutions

The characteristics of this project, the ML model's structure, and the novelty and excellence of the solution are all summarized in this section. The mine used to collect experimental blasting data, Akdaglar Quarry, is situated in northern Istanbul's Cendere Basin. The goal of this study is to forecast ground vibration in a particular mine given nine blasting design parameters and to further optimize blasting design parameters using ML models. In the Akdaglar quarry, 88 blasting monitoring sites were used to gather data. Nine blasting parameters' distribution characteristics are shown in Figure 1.

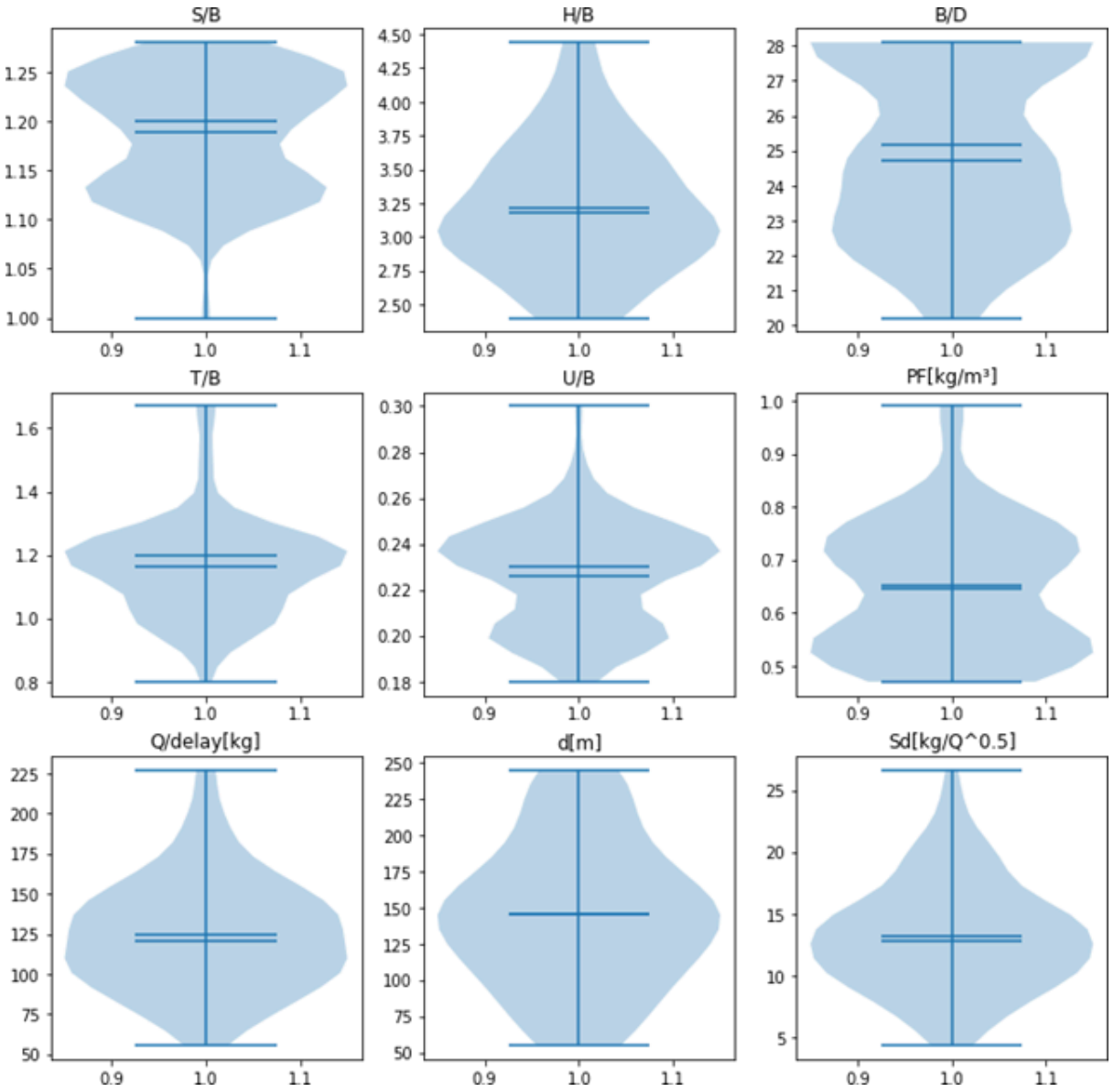

Figure 1: violin plot of data distribution.

In this project, In addition to the property of small data size and huge kinds of data dimensions discussed in the last section, parameter distribution also needs careful consideration in establishing ML models. There are two peaks in the distributions of four parameters, such as ratio of spacing to burden (S/B) and ratio of subdrilling to burden (U/B), because there are different parameter design standards for rocks blasting with different hardness in the mine. Even for the burden to hole diameter ratio (B/D), many data points have gathered at the highest level. Most parameter distributions do not follow the normal distribution. Due to this, the ML models based on the characteristics of the normal distribution, such as the multivariate linear regression model, may perform poorly. These models aim to learn the normal distribution's mean (μ) in the context of linear relationships between independent variables. Data-hungry ML models like Artificial Neural Networks (ANN) also perform poorly due to the property of limited data size here.

According to the discussion above, compared with other ML models, ensemble models such as XGboost, GBDT, and Random Forest perform better on small sample problems. XGboost and GBDT models have excessive hyper-parameters needed for tuning, so the grid research method for hyper- parameters tuning is ineffective to implement (Zhang and Jung - 2021). PSO is used to tune hyper-parameters. There is no publication examining the viability and efficiency of PSO-XGboost or PSO-GBDT for controlling ground vibration.

XGboost and GBDT are boosting models that perform well on gradient boosting capabilities (Hudaverdi and Akyildiz - 2017). Theoretically, The former's prediction accuracy is higher than the latter's, but the XGboost model takes longer to train since it requires more hyper-parameter tuning and has a more intricate algorithm structure. Random Forest is a common bagging ensemble model that performs exceptionally well in terms of anti-overfitting and nonlinear characteristics. However, attributes with excessive divisions have a higher effect on random forests for the data with an imbalanced distribution.

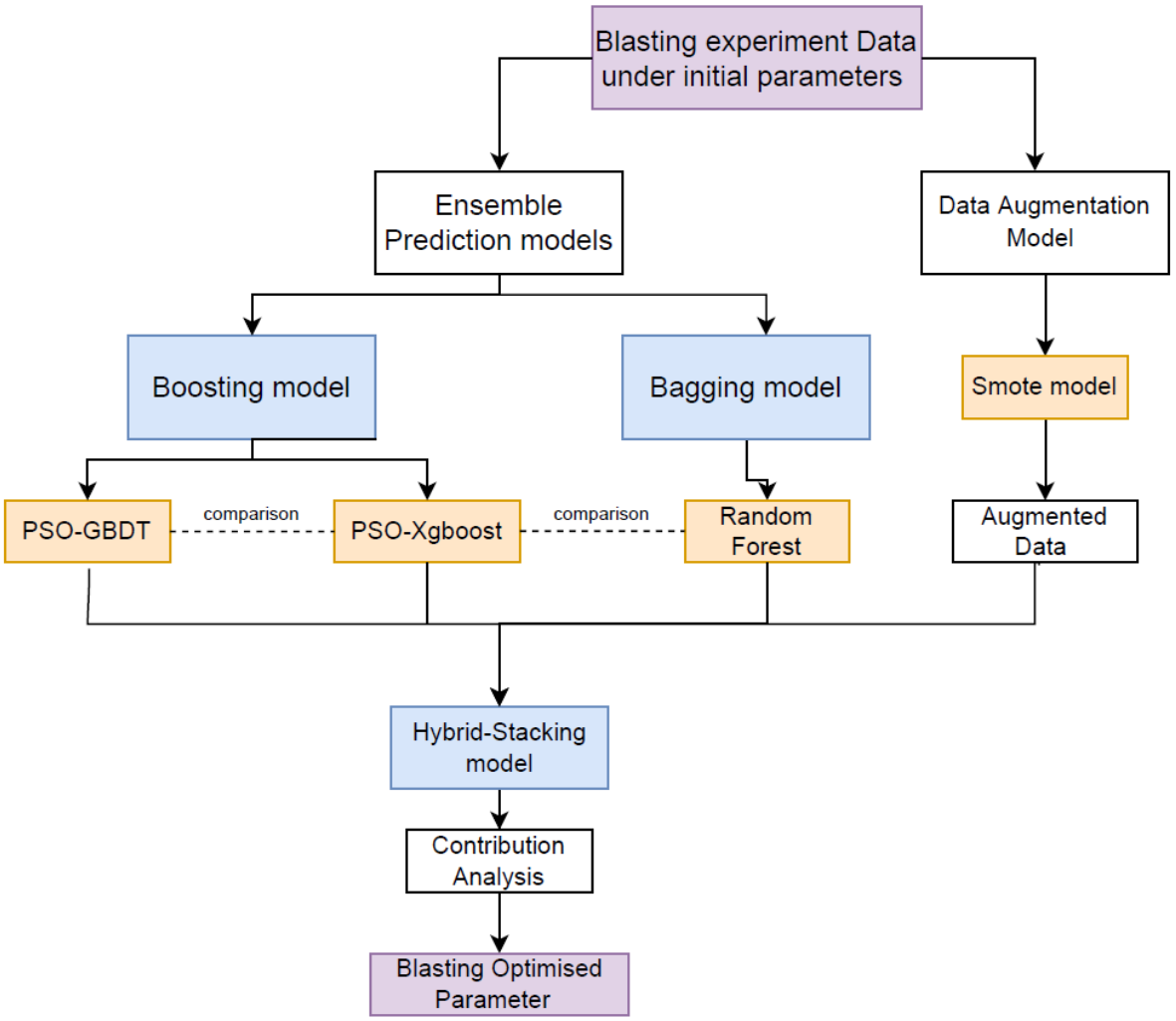

Figure 2: Flowchart of tasks in the project.

To qualify the discussion above, the three hybrid models are established. Accuracy is compared, and a series of interpretability analyses are proposed, including sample size effect, the significance of dimensions, and dimension contribution analysis based on SHAP. In order to generalize models and prevent over-fitting, this project adds k-fold validation to the PSO algorithm. In order to fulfill the goal of improving vibration prediction accuracy, the models are further optimized. It uses a SMOTE data-augmented approach and stacking models after discovering that the amount of data severely limits prediction accuracy as revealed by sample numbers analysis. Performance analyses are proposed to confirm their viability and superiority. The Flow chart of the implementations are shown in figure 2.

Methodology

Pre-processing

This section consists of the illustration of dimensions of data, training data split method, and evaluation method in this project. The indicator of ground vibration is PPV(mm/s). The distance between blasting location and the monitoring point d(m), and maximum charge per delay Q(kg) are recorded and designed by rock blasting standard firstly. The scaled distance (Sd) is calculated by equation 3.

$$Sd = \frac{m}{\sqrt[2]{kg}} \tag{3}$$

Other six parameters are designed including power factor PF(kg/m3), bench height H(m) , drilled burden B(m) , spacing S(m), blasting hole diameter D(m), sub-drilling U(m) and stemming T(m), which are represented for conciseness as ratios: PF,H/B,S/B ,B/D ,U/B and T/B . The inputs and output of the model are represented as: X1(S/B), X2(H/B), X3(B/D), X4(T/B), X5(U/B), X6(PF), X7(Q), X8(d), X9(Sd) and Y1(PPV) . The distribution of the nine blasting parameters is represented in Figure 1.

After normalization, the dataset is splitted into a train set (64%), validation set (16%). and test set (20%). There are two reasons for specifically dividing the test set (20%) instead of using cross-validation. The first is that the test set is necessary to verify that the model does not lead to over-fitting because the validation set and 5-fold cross-validation method have been used by the PSO algorithm implemented to tune the hyper-parameters. The second is to confirm the test set is not fed into the process of model training to ensure the sufficient interpretability of the models.

R2 is used to evaluate the accuracy of the model (Calabrese et al. - 2014), as shown in equation 4.

$$R^2 = \frac{\sum_{i=1}^{n} \left(y_{truei} - y_{predi}\right)^2}{\sum_{i=1}^{n} \left(x_{truei} - x_{predi}\right)^2} \quad (4)$$

Algorithm

Prediction Algorithm

The concepts of gradient boosting are implemented in GBDT and XGBoost. The primary idea of the algorithms is to use gradient boosting to determine the best variables to minimize a specified objective function (Yin et al. - 2018 and Nobre and Neves - 2019). They use gradient boosting decision trees and gradient boosting machines as examples of parallel tree boosting.

The way they are implemented is as follows: There are n data samples and m features in D = (Xm, y). To minimize the objective function, the sample space is partitioned along the feature dimensions. The objective function consists of two parts, the first part is used to determine the difference between the predicted value and the real value, and the other part is the regularization term. The regularization term also contains two parts, T represents the number of leaf nodes, and w represents the weight of leaf nodes. γ can control the number of leaf nodes, and λ can control the weight of leaf nodes not to be too large to prevent overfitting, as shown in equation 5 and 6 (Ma et al. - 2021, Raipal et al. - 2020 and Sagi and Rokach - 2021).

$$Obj = \sum_{i=1}^{n} l\left(y_i, y_{i'true} + f_k\left(x_i\right)\right) + \sum_{k=1}^{k} \omega\left(f_k\right) \quad (5)$$

$$\omega(f) = \gamma * T + 0.5 * \lambda||\omega||^2 \quad (6)$$

The next step is to find variables that minimize the objective function. The idea of GBTD is to use the first derivative, shown in equation 7, but XGBoost is to determine it using its Taylor second-order expansion at ft=0, shown in equation 8, Where gi is the first derivative and hi is the second derivative (Qiu et al. - 2021):

$$Obj_{GBDT} = \sum_{i=1}^{n} l\left(y_i, y_{i'true} + g_i f_i\left(x_i\right)\right) + \sum_{k=1}^{k} \omega\left(f_k\right) \quad (7)$$

$$Obj_{XGBoost} = \sum_{i=1}^{n} l\left(y_i, y_{i'true} + g_i f_i\left(x_i\right) + 0.5 h_i f_i^2\left(x_i\right)\right) + \sum_{k=1}^{k} \omega\left(f_k\right) \quad (8)$$

The models combine multiple tree models to create powerful ensemble models. To grow a tree, the algorithms employ feature splitting and continuous tree addition

(Parsa - 2021). Each time a tree is added, it is actually learning a new function to fit the residual of the last prediction. The final prediction can be shown in equation 9 (Hajihassani et al. - 2015).

$$y = \sum_{k=1}^{k} f_k\left(x_i\right), f_k \in F \quad (9)$$

The Random Forest model contains multiple decision trees, and its output, based on random sampling, is determined by the accumulation of individual trees.

PSO is used to optimize the hyper-parameters in the XGboost and GBDT model. There are nine hyper-parameters involved in the PSO-XGboost model, including boost iteration , maximum depth, subsample percentage, shrinkage rate, column subsample ratio, subsample ratio of the training in- stances, regularization term on weights (α and λ), minimum child weight, and minimum loss reduction γ . Five hyper-parameters are involved in the PSO-GBDT model, including number of estimators, max depth of the tree, learning rate, the fraction of samples and the alpha-quantile of the huber-loss function.

The maximum tree depth and the number of trees are the two dominating hyper-parameters in the Random Forest model. The other two parameters, minimum samples split partition and minimum sample leaf, are used to prevent overfitting and are not significant.

Because XGboost uses the second order derivative and regulation term, it is theoretically more accurate than GBDT. However, because the XGboost algorithm is more complex, there are a lot more hyper-parameters to tune, and doing so requires more computer power and memory when training a model. Such computational costs vary by orders of magnitude over accumulation of numerous training processes by PSO. So in order to quantify the difference in accuracy between XG boost and GBDT in different situations, a grid research method is used in PSO.

To prevent overfitting, 5-fold cross-validation is applied on the fitness functions. The flowchart of the PSO-XGBoost and PSO-GBDT is shown in figure 3.

Hyper-Parameter Searching Algorithm

A machine learning technique called particle swarm optimization (PSO) is based on the characteristics of microscopic particles and the behavior of social animals (Armaghani et al. - 2014). The algorithm follows these steps: Initialize the first particle swarm and its associated velocity before measuring particle fitness and selecting an appropriate position to make it the local and global optimal (Wang et al. - 2020). The second step is to update the particle velocity. In the search space, each particle travels at a specified speed, and in each iteration, the particle's speed is

determined by both the global and local optimum. The global optimum is the best position obtained by the particle in all the iterative processes, and the local optimum is the best position of the particle in the current iteration. The third step is to update the position of the particle. After calculating the new velocity of the particle, the particle moves in the search space with the new velocity, Equation 10 and 11 is used to update the velocity of each particle.

$$v_j^{i+1} = wv_j^i + \left(c_1 * r_1 * \left(localbest_j - x_j^i\right)\right) + \left(c_2 * r_2 * \left(globalbest_j - x_j^i\right)\right) \quad (10)$$

$$v_{min} < v_j^i < v_{man} \quad (11)$$

where xi and vi represent the position and velocity of the j-th particle at the i-th iteration, respectively; w represents the inertia weight coefficient; i is the number of iterations; r1 and r2 represents the number in the interval [0, 1].

If the new particle velocities are more matched, the positions of the global optimum and the local optimum will also change. Equation 12 is used to update the position of the local optimum for each particle.

$$x_j^{i+1} = x_j^i + v_j^{i+1} \quad (12)$$

Cross-validation is applied in the cost function to generalize the model. Grid search method is implemented to determine hyper-parameters of PSO. The number of particles was set to 3, 7, and 20 to observe the accuracy of the model and the number of iterations was set to 20. Implementation of PSO is shown in Fig 3

Post-Processing

In this project, in addition to developing an efficient machine learning model, significant consideration must be given to the analysis of the data dimensions and sample size (Nobre and Neves - 2019, Ke et al. - 2021 and Douzas and Bacao - 2019). By studying which features play a decisive role and the influence of sample size on the prediction results, the constructive instructions for further optimisation of the model can be proposed.

Interpretability Analysis

SHAP analyzes model interpretability by the marginal benefits of individual dimensions. The importance of the individual dimension is determined by the gain of a certain feature in the combination. It can be obtained by removing the benefit regarding that the feature is not included in the dimension combination. The total contribution of this feature is then determined by adding up all combinations and averaging the results.

$$I_j = \frac{1}{n}\sum_{i=1}^{n}\left|\theta_j^i\right| \quad (13)$$

where Ij represents the contribution of the dimension. n is the number of dimensions. θ is j th dimension weight by removing the benefit when the i feature is not included in the dimension combination in this project, the cost of increasing data size is quite high. The impact of data size on prediction accuracy needs to be quantified in order to achieve a balance between the cost and accuracy.

Data Augmented Algorithm

The predicted accuracy is constrained by the sample size, as shown by the sample size analysis in the Result section. A data-augmented method is used to increase the training data set in order to further increase the accuracy.

The general data-augmented methods such as Gaussian multivariate simulation based on normal distribution does not perform well in this project as a result of the sample's significant imbalanced distribution.

Smote is an oversampling technique based on feature space for handling imbalanced data (Douzas and Bacao - 2019). Smote creates new samples with the same class target label by synthesizing new characteristics from samples from minority classes and samples from those classes' closest neighbors. Only classification-related problems can be resolved by using the conventional Smote approach. But by switching from directly copying labels to utilizing interpolation to calculate target values for new samples from the existing labels, Smote can be used to fix the regression problem.

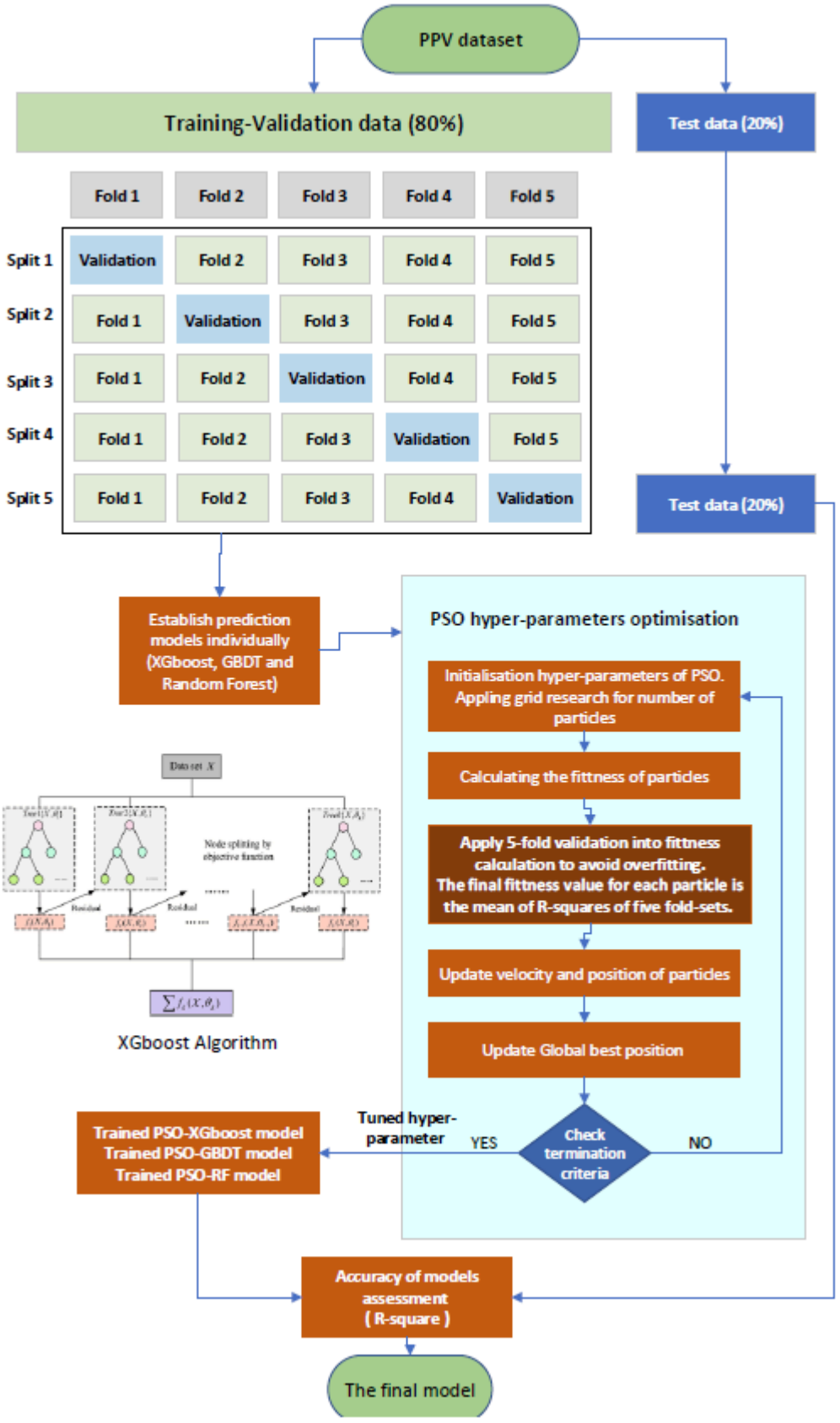

Figure 3: Flowchart of PSO and XGboost algorithms.

Stacking Method

The Stacking Method is used for the fact that the random forest model does not perform as well as the other two models in terms of prediction accuracy for the small data size here. This is probably due to the random forest model's sensitivity to the data's balanced distribution, which is why a hybrid stacking model is proposed here. Stacking regression is an ensemble learning technique that combines multiple regression models through a meta-regression.

Code metadata

This project is developed in python 3.6 using Google Colab and Pycharm(2022.2.1). The sample data can be directly obtained from the data file in the repository. The libraries are used in this project including Numpy, Pandas, Seaborn and Matplotlib for data processing and visualization of the result, Scikit-learn and Math for R2 importing, data splitting and comparison between models, SHAP(0.41.0) for data dimension analysis, SMOTE for augmentation of the data.

Results

The three models are compared here under the PSO algorithm. Dimension importance is analyzed using SHAP method and XGboost feature importance, calculated from the amount by which each attribute split point improves the performance metric. According to the comparative analysis results, the model is further optimized by data augmentation and stacking methods from the findings of the comparison analysis.

Models Performance

As demonstrated in the Fig 4, Random Forest Regression maintained accuracy with a R2 of roughly 0.72 during the course of the PSO iteration. At the beginning, the accuracy of it is much higher than that of XGboost and GBDT below 0.67. This is most likely due to the fact that there are only two main hyper-parameters in Random Forest: n-estimators and max-depth requiring tuning with 599 and 6, respectively.

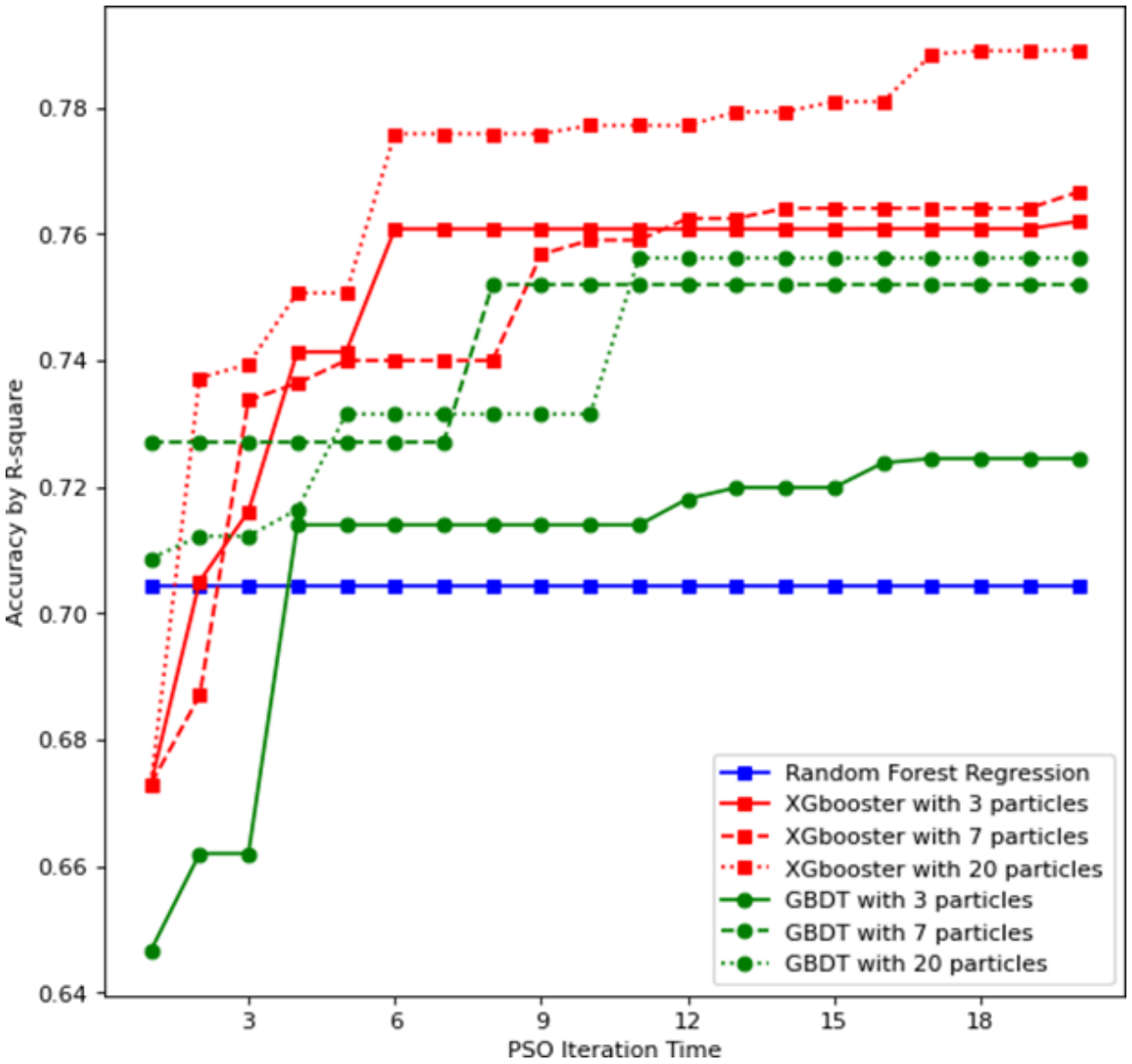

Figure 4: PSO-GBDT, PSO-XGboost and PSO-Random Forest Comparison.

After three PSO iterations, where the hyper-parameters of the two models begin to be tuned, XGboost and GBDT outperformed Random Forest in terms of accuracy. In general, XGboost outperforms GBDT in terms of prediction accuracy performance. For particles 7 and 20, the accuracy performance of GBDT is comparable to that of XGboost for particles 3 and 7. They settle after 10 rounds with R2 of roughly 0.76, reaching their accuracy peak at around 8 iterations. The highest precision of R2 of 0.7954 is attained by a 20-particle XGboost after 20 iterations. The best hyperparameters are listed below.

TABLE

Interpretability Analysis

In this section, interpretability analysis includes the importance of dimension analysis, the contribution of dimension SHAP-based analysis, and the effects of sample size and dimensionality on prediction accuracy.

In order to give a highly objective dimension analysis, the XGboost-based analysis and SHAP-based analysis are shown for comparison, as shown in Fig 5. The principle of the former is to calculate the number of features used when the XGboost subtree model is split, and the latter is analyzed based on the diminishing marginal effect of features.

The contribution of dimension analysis is shown in Fig 6. In the left figure, the positive and negative correlations between predictions of all sample points are quantified and presented. The colors represent the positive and negative correlations of dimensions, which are marked as red and blue, respectively. The contribution of each feature is shown in the image on the right by weighing and averaging the SHAP values across all samples and features. It should be emphasized even though the contribution analysis graphs are similar to importance analysis graphs, they significantly differ. The former reflects and quantifies the positive and negative correlation between dimensions and targets, while the latter just quantifies the influence of dimensions on the predicted value of the models.

The analyses of the relationship between the sample size, the number of dimensions and the prediction accuracy are shown in the figure 6. In the left image, the model's dimensions are ordered by decreasing relevance from right to left using the XGboost approach. When n=4, the elbow point happens. The elbow point appears at n=45 in the right picture. As the sample size lowers, the model prediction accuracy gradually declines.

The details of analysis calculations and the information gained from the analysis are presented in the Methodology section and the Discussion section, respectively.

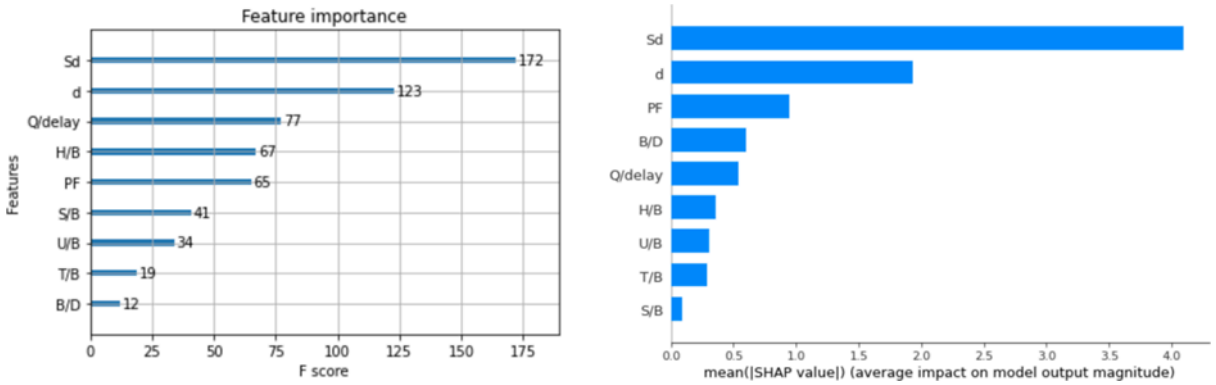

Figure 5: Importance of dimensions analysis.

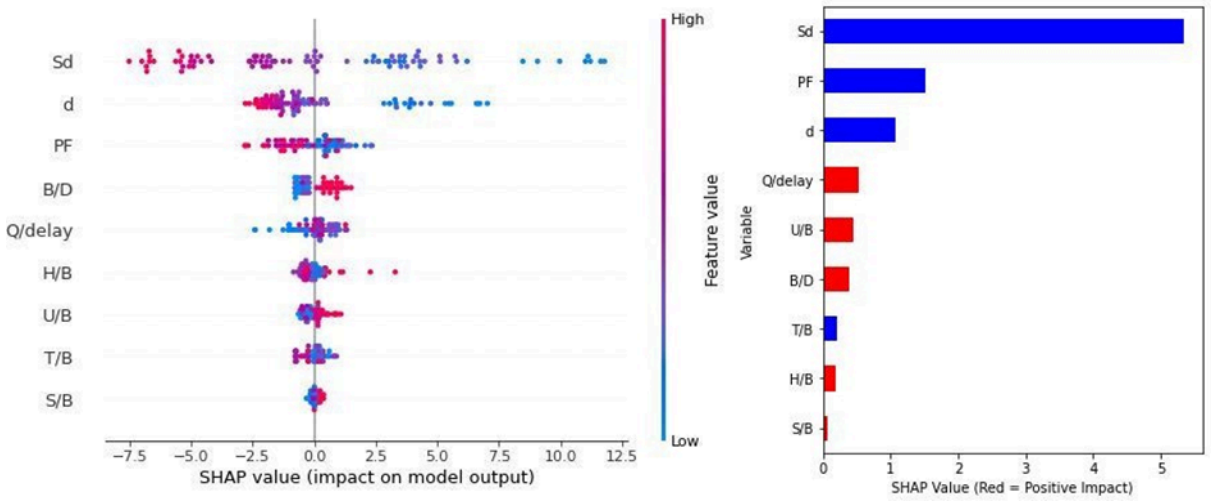

Figure 6: Contribution of dimensions analysis.

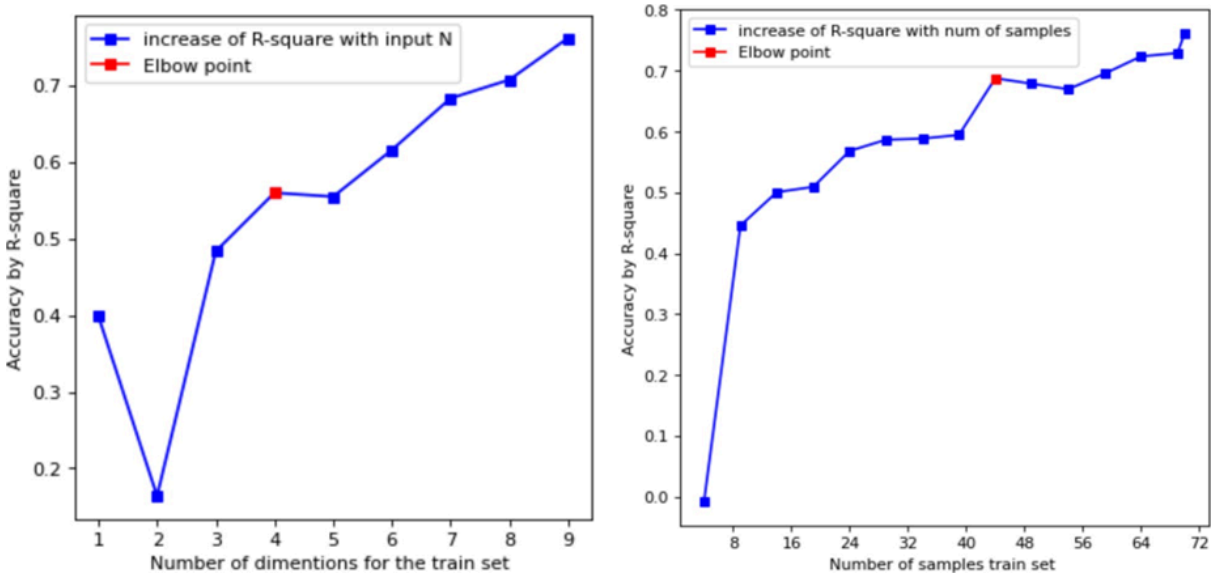

Figure 7: Impact of Sample size and dimensions on prediction accuracy

5.3 Optimisation

According to the prediction results and the analysis of training sample size, the model still has space for improvement. First, the sample size restricts XGboost from converging further. Second, because of imbalanced sample distributions, the Random Forest model performs moderately. In order to augment the data for the first point, a revised SMOTE data augmentation approach is applied to the XGboost model. For the second point, the augmented dataset is subjected to a stacking technique that integrates Random Forest and XGboost. The prediction results of the two optimised models and the three initial models in the test set are shown in the figure 8. It can be found that the improved SMOTE data augmentation algorithm can

improve the prediction accuracy R2 from 7.89 to 8.29, but the stacking model has no obvious improvement in prediction accuracy with R2 fluctuating around 8.10. The Discussion section presents potential reasons for this.

The figure 9 shows the predictions of the five models in the test set, and plots the average of the true and predicted values. It can be seen from the figure that even though the true value deviates from the predicted value in some samples, the mean values are overlapping. This demonstrates that the five models' overall quantitative analyses of the blasting predictions are compatible with the actual circumstance, demonstrating the feasibility of each of the five models.

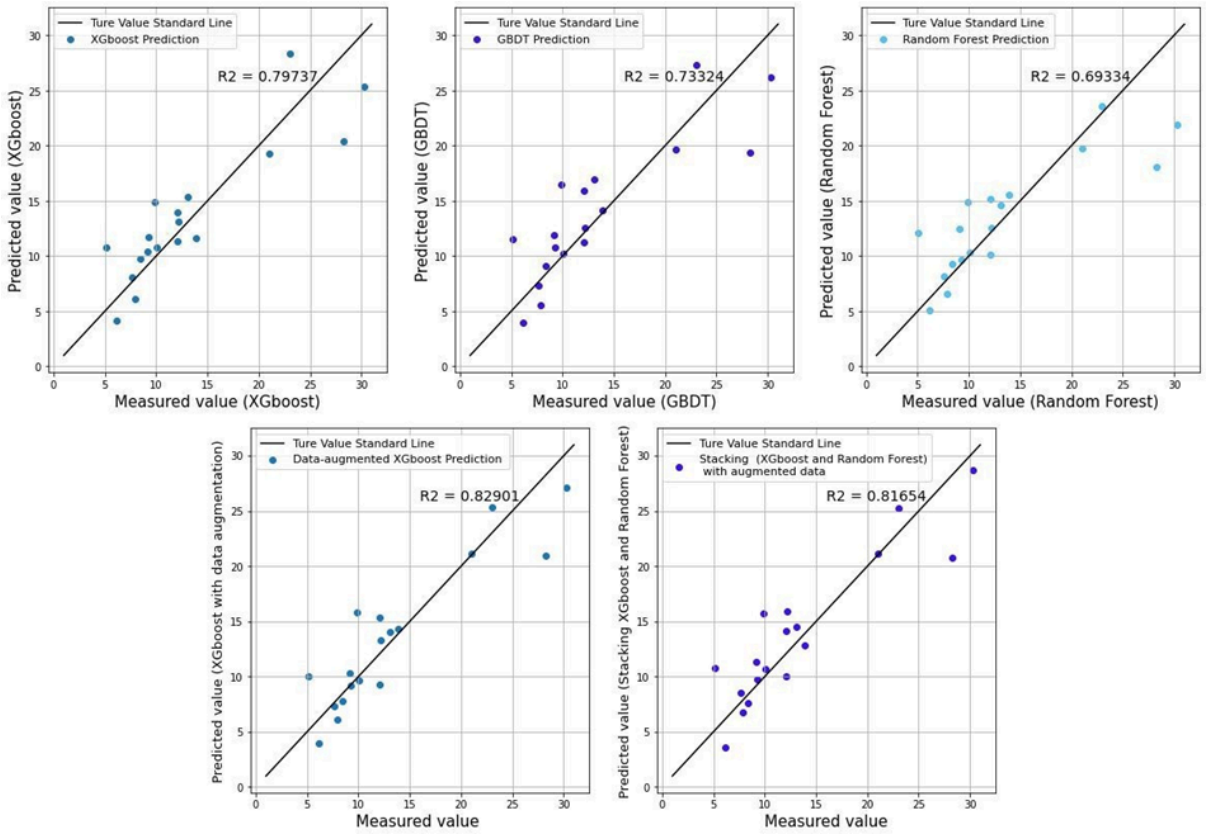

Figure 8: prediction accuracy performance of the models

Discussion

Prediction algorithm

The prediction performance of XGBoost is better than that of Random Forest in this project. For imbalanced datasets, XGBoost prunes the trees based on similarity. It takes the difference between the similarity of the node and the similarity of the child as the gain of the node. If the gain from the node is small, XGboost stops building the tree to greater depths, which prevents overfitting. The difference is that decision

trees in random forests are presented with highly similar samples, in which case Random Forests are likely to overfit the data.

The accuracy of XGboost and GBDT are equivalent during the first optimization stage of the PSO algorithm when the number of particles and cycles is kept low. However, when the number of particles or the number of PSO iterations increases, XGboost's convergence speed and prediction accuracy are higher than those of GBDT.

Traditional GBDT only employs first-order derivative, whereas XGboost uses both first and second- order derivative data simultaneously to implement Taylor expansion on the cost function. And XGboost multiplies the leaf node weights by the Shrinkage parameter after one iteration to balance the influence of each tree on the model. These implementations increase the speed of convergence and accuracy of the model.

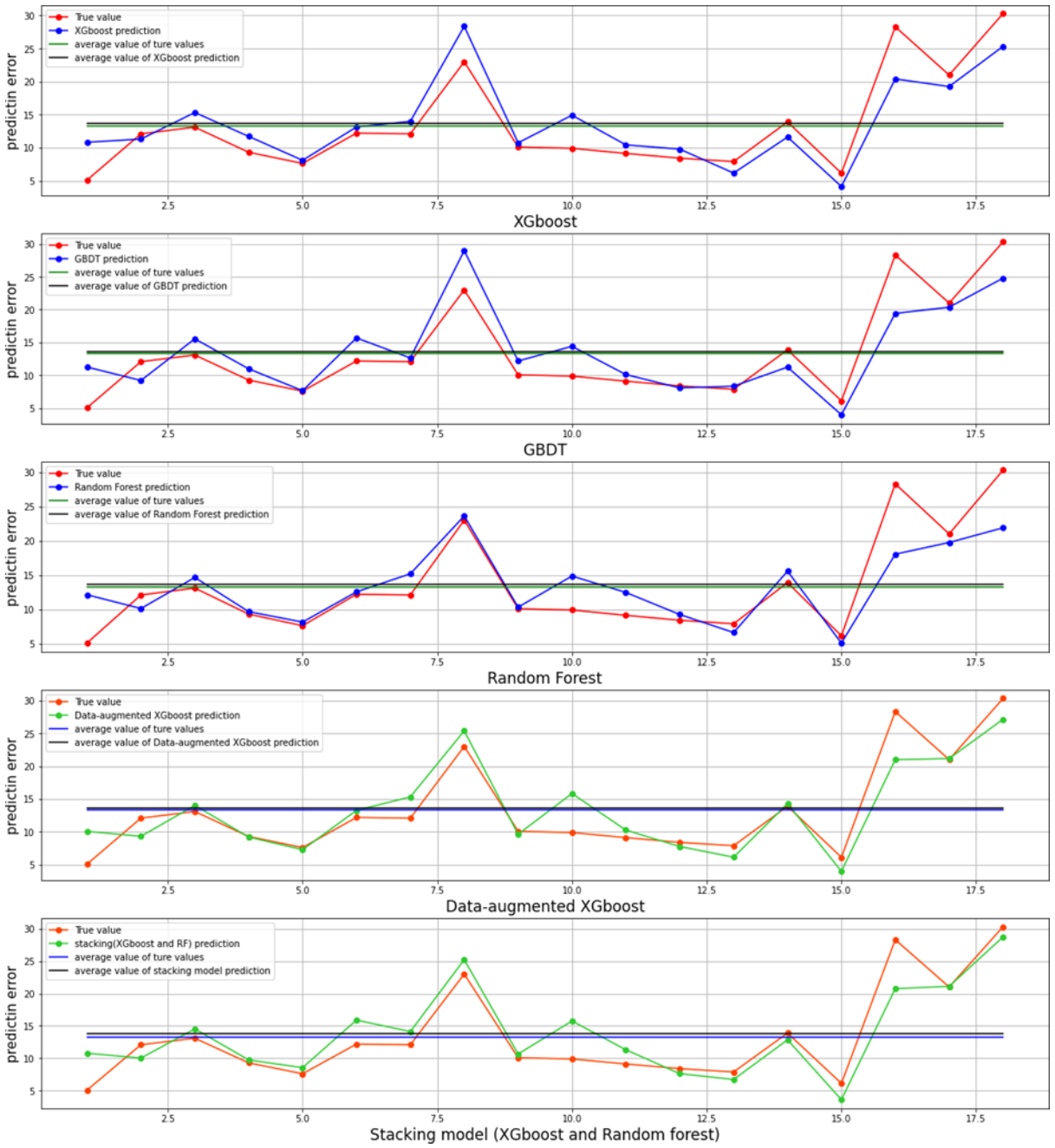

Figure 9: predicted value and measured value analysis figure

However, XGboost's algorithm structure is more intricate than GBDT, and therefore the latter requires more cost in tuning hyper-parameters. So, only during the initial PSO iteration, there is no discernible difference between the two methods' accuracy and rate of convergence.

In conclusion, gradient boosting methods such as GBDT and XGboost are preferable over bagging models represented by Random Forest in the practical application of blasting parameter design where the sample size of blasting is small and the sample distribution is imbalanced. Researchers may use the GBDT algorithm to perform

qualitative analysis, identify the range of blasting parameters, and then apply the XGboost algorithm to further forecast the ground vibration brought on by blasting.

Interpretability Analysis

The two evaluation approaches may roughly divide the dimensions into two groups based on the findings of the dimension important analysis, although there is a slight deviation in the ranking. The first group includes maximum charge per delay Q(kg), power factor(PF), distance(d) and scaled distance (Sd). The remaining five parameters are included in the second group. Both importance analyses reveal that in the predicted PPV models, the first set of parameters is vastly more important than the second set, which means that in the blasting design, after satisfying the requirements of rock blasting, the amount of explosives should be decreased and PF should be increased as much as possible, rather than attempting to optimize the second set of parameters to significantly reduce vibration impact. After determining Q and PF, priority should be given to the design of sub-drilling

(U) and bench height (H) because they have a more significant effect on the model. It can also be seen from the figure 7 that the accuracy of R2 is higher than 0.5 when only using the first set of parameters for prediction. With the addition of the second set of parameters, the curve reaches an elbow point, and as the input number of parameters gradually increases, R2 gradually rises to around 0.8 .

After normalization and contribution analysis, the contributions of design parameters to the model are qualified. the increase of PF can significantly improve the overuse of explosives, and the stem

(T) should be increased in the design while other design parameters are minimized. The analysis also demonstrates the shortcomings of conventional empirical formulas, which lacked direction for parameter design because they only took the contributions of one design parameter into consideration and ignored those of other design parameters.

As shown in Figure figure 7, the sample analysis can be observed as three stages . When the sample size is 10, the accuracy curve reaches the first elbow point. When the sample size is about 45, the accuracy curve reaches the second elbow point and then the accuracy increases as the sample size increases. Therefore, in the initial stage of blasting design, 10 samples can be enough for empirical model analysis of the entire mining environment with R2 of 0.45. The ML models with acceptable accuracy with R2 of 0.7 can be established around 50 samples. After that, the model accuracy increases with the sample size.

Optimisation of the Model

The predicted accuracy of R2 of the XGboost model can be increased by roughly 0.3 after employing the SMOTE augmented-data algorithm. This is due to the fact that SMOTE creates new data through interpolation, which expands the generalization of the training data, therefore reducing overfitting and improving the robustness of the model.

After integrating Random Forest into XGboost, the prediction accuracy of the stacking model does not improve significantly, and even tends to decline. There are two potential explanations. The decision trees in the Random Forest have a high degree of similarity when the sample distribution is imbalanced, which causes the model to be overfit. The second is that the training data has reached the upper utilization. The accuracy cannot be further improved by optimizing the model due to the lack of acquisition parameters related to topography.

Conclusion

Three hybrid models PSO - XGboost, PSO - XGBT, and PSO - Random Forest are established in this project. The proper hyperparameters of PSO are found using the grid search approach. PSO- XGboost performs the best of the bunch, with a R2 score of 0.7974. After a series of interpretability analysis, two optimisation strategies are developed and implemented. The first is to use SMOTE augmented-data algorithm in the model. The second is to use the stacking method to fuse the random forest and the XGboost model. Finally, a R2 value of 0.829 confirms the first optimization strategy's superiority. The test set of samples demonstrates the feasibility of all the five models developed in this project.

In future studies, in order to generalize the model and reduce the significant demand for sample size in model training, the topography of the mining area will be quantified and used as a set of hyper-parameters to train the model since it has a non-negligible impact on PPV (Raina et al. - 2014).

References

[1] V. F. Navarro Torres, Leandro G.C. Silveira, Paulo F.T. Lopes, and Hernani M. de Lima. Assessing and controlling of bench blasting-induced vibrations to minimize impacts to a neighboring community. Journal of Cleaner Production, 187:514–524, 6 2018.

[2] A. K. Raina, A. Haldar, A. K. Chakraborty, P. B. Choudhury, M. Ramulu, and C. Bandyopadhyay. Human response to blast-induced vibration and air-overpressure: An Indian scenario. Bulletin of Engineering Geology and the Environment, 63:209–214, 8 2004.

[3] V. F. Navarro Torres, Leandro G.C. Silveira, Paulo F.T. Lopes, and Hernani M. de Lima. Assessing and controlling of bench blasting-induced vibrations to minimize impacts to a neighboring community. Journal of Cleaner Production, 187:514–524, 6 2018.

[4] Manoj Khandelwal and T. N. Singh. Prediction of blast-induced ground vibration using artificial neural networks. International Journal of Rock Mechanics and Mining Sciences, 46:1214–1222, 2009.

[5] P. H.S.W. Kulatilake, W. Qiong, T. Hudaverdi, and C. Kuzu. Mean particle size prediction in rock blast fragmentation using neural networks. Engineering Geology, 114:298–311, 8 2010.

[6] Mohammad Esmaeili, Alireaza Salimi, Carsten Drebenstedt, Maliheh Abbaszadeh, and Ab- bas Aghajani Bazzazi. Application of pca, svr, and ANFIS for modeling of rock fragmentation. Arabian Journal of Geosciences, 8:6881–6893, 9 2015.

[7] Danial Jahed Armaghani, Ehsan Momeni, Seyed Vahid Alavi Nezhad Khalil Abad, and Manoj Khandelwal. Feasibility of ANFIS model for prediction of ground vibrations resulting from quarry blasting. Environmental Earth Sciences, 74:2845–2860, 8 2015.

[8] Combination of neural network and ant colony optimization algorithms for prediction and optimization of flyrock and back-break induced by blasting. Engineering with Computers, 32:255–266, 4 2016.

[9] Aminaton Marto, Mohsen Hajihassani, Danial Jahed Armaghani, Edy Tonnizam Mohamad, and Ahmad Mahir Makhtar. A novel approach for blast-induced flyrock prediction based on imperialist competitive algorithm and artificial neural network. Scientific World Journal, 2014, 2014.

[10] Turker Hudaverdi. Application of multivariate analysis for prediction of blast-induced ground vibrations. Soil Dynamics and Earthquake Engineering, 43:300–308, 12 2012.

[11] Jialin Zhang, Da Xu, Kaijing Hao, Yusen Zhang, Wei Chen, Jiaguo Liu, Rui Gao, Chuanyan Wu, and Yang De Marinis. Fs-gbdt: Identification multi cancer risk module via a feature selection algorithm by integrating fisher score and GBDT. Briefings in Bioinformatics, 22, 5 2021.

[12] Zhendong Zhang and Cheolkon Jung. Gbdt-mo: Gradient-boosted decision trees for multiple outputs. IEEE Transactions on Neural Networks and Learning Systems, 32:3156–3167, 7 2021.

[13] Turker Hudaverdi and Ozge Akyildiz. Investigation of the site-specific character of blast vibration prediction. Environmental Earth Sciences, 76, 2 2017.

[14] Justin M. Calabrese, Grégoire Certain, Casper Kraan, and Carsten F. Dormann. Stacking species distribution models and adjusting bias by linking them to macroecological models. Global Ecology and Biogeography, 23:99–112, 2014.

[15] Rui Yin, Viet Hung Tran, Xinrui Zhou, Jie Zheng, and Chee Keong Kwoh. Predicting antigenic variants of h1n1 influenza virus based on epidemics and pandemics using a stacking model. PLoS ONE, 13, 12 2018.

[16] João Nobre and Rui Ferreira Neves. Combining principal component analysis, discrete wavelet transform and XGBoost to trade in the financial markets. Expert Systems with Applications, 125:181–194, 7 2019.

[17] Meihong Ma, Gang Zhao, Bingshun He, Qing Li, Haoyue Dong, Shenggang Wang, and Zhongliang Wang. Xgboost-based method for flash flood risk assessment. Journal of Hydrology, 598, 7 2021.

[18] Hardik Raipal, Madalina Sas, Chris Lockwood, Rebecca Joakim, Nicholas S. Peters, and Max Falkenberg. Interpretable XGBoost based classification of 12-lead ecgs applying information theory measures from neuroscience. volume 2020-September. IEEE Computer Society, 9 2020.

[19] Omer Sagi and Lior Rokach. Approximating XGBoost with an interpretable decision tree. Information Sciences, 572:522–542, 9 2021.

[20] Yingui Qiu, Jian Zhou, Manoj Khandelwal, Haitao Yang, Peixi Yang, and Chuanqi Li. Performance evaluation of hybrid woa-xgboost, gwo-xgboost and bo-xgboost models to predict blast-induced ground vibration. Engineering with Computers, 2021.

[21] Mohammad Parsa. A data augmentation approach to xgboost-based mineral potential mapping: An example of carbonate-hosted zn–pb mineral systems of western iran. Journal of Geochemical Exploration, 228, 9 2021.

[22] Mohsen Hajihassani, Danial Jahed Armaghani, Masoud Monjezi, Edy Tonnizam Mohamad, and Aminaton Marto. Blast-induced air and ground vibration prediction: a particle swarm optimization-based artificial neural network approach. Environmental Earth Sciences, 74:2799– 2817, 8 2015.

[23] D. Jahed Armaghani, M. Hajihassani, E. Tonnizam 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 Journal of Geosciences, 7:5383–5396, 11 2014.

[24] Chen Wang, Chengyuan Deng, and Suzhen Wang. Imbalance-xgboost: leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost. Pattern Recognition Letters, 136:190–197, 8 2020.

[25] Na Ke, Guoqing Shi, and Ying Zhou. Stacking model for optimizing subjective well-being predictions based on the CGSS database. Sustainability (Switzerland), 13, 11 2021.

[26] Georgios Douzas and Fernando Bacao. Geometric smote a geometrically enhanced drop-in replacement for smote. Information Sciences, 501:118–135, 10 2019.

[27] A. K. Raina, V. M.S.R. Murthy, and A. K. Soni. Flyrock in bench blasting: a comprehensive review. Bulletin of Engineering Geology and the Environment, 73:1199–1209, 10 2014.