International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Imbalanced Data Challenges and Their Resolution to Improve Fraud Detection in Credit Card Transactions

Vishwa Teja Manda[1], Kondapalli Dheeraj*[2], Y Charan[3] , Dr. N.M. Jyothi[4]

Submitted:11/03/2024 **Revised**: 26/04/2024 **Accepted**: 03/05/2024

Abstract: This project addresses the critical issue of fraud detection in credit card transactions, an imperative conand cern for both financial institutions and cardholders. With the increasing sophistication of fraudulent activities, accurate identification and prevention of fraudulent transactions have become paramount. The study focuses on a dataset comprising credit card transactions conducted by European cardholders in September 2013. Notably, the dataset exhibits a severe class imbalance, with fraudulent transactions accounting for a mere 0.172% of the total. The primary objective of this research is to develop a robust machine-learning model capable of effectively discerning between legitimate and fraudulent transactions. The project commences with an extensive exploration of the dataset, encompassing checks for data imbalance, feature visualization, and analysis of feature interrelationships. Subsequently, four predictive models, including Random Forest, AdaBoost, Cat Boost, and XG Boost, were employed and evaluated. The dataset was partitioned into three subsets: a training set, a validation set, and a test set. Initial results showcased promising performance, with the Random Forest model yielding an Area Under the Curve (AUC) the core of 0.85 on the test set. The AdaBoost model achieved a slightly lower AUC score of 0.83, while the Cat Boost model, following 500 iterations, attained an AUC score of 0.86. The XG Boost model demonstrated exceptional promise, achieving a validation score of 0.984, and subsequently producing an AUC score of 0.974 on the test set.
Further, the project introduced a Light GBM model, leveraging both train-validation split and cross-validation methods. The former yielded AUC scores of approximately 0.974 on the validation set and 0.946 on the test set. Cross-validation exhibited a similar effectiveness, culminating in an AUC score of 0.93 on the test predictions. This study not only underscores the efficacy of employing advanced machine learning techniques in fraud detection but also emphasizes the importance of model selection and evaluation in the context of imbalanced data. The findings provide valuable insights for financial institutions seeking to bolster their fraud detection capabilities, ultimately enhancing the security and trust of credit card transactions.

Keywords: *Fraud detection; Credit card transactions; Machine learning models; Imbalanced data; AUC score*

1. Introduction

In an era of expanding digital commerce, safeguarding financial transactions from fraudulent activities has become a paramount concern for both financial institutions and consumers. The sophistication of fraudulent tactics necessitates innovative approaches to detect and prevent unauthorized credit card transactions. This project delves into the realm of fraud detection, focusing on a dataset comprising credit card transactions by European cardholders in September 2013. Notably, the dataset presents a substantial class imbalance [10], with fraudulent transactions accounting for a mere fraction of a percent of the total. The primary goal of this research is to develop a robust machine-learning model capable of effectively discerning between legitimate and fraudulent transactions. By employing advanced algorithms and techniques, this project seeks to enhance the security and reliability of credit card transactions, contributing to the broader domain of financial security and trust in digital transactions.

2. Design/Methods/Modelling

2.1. Preprocessing the data

When compiling and analyzing credit card transaction data for fraud detection, there are several crucial steps involved. The first phases concentrate on resampling methods such as oversampling the minority class and under sampling the majority class in order to manage unbalanced data. Standardization and Min-Max scaling are two examples of feature scaling techniques that are used to guarantee uniform scaling across all features. The next step involves using Principal Component research (PCA) to minimize dimensionality while preserving important information for further research.

Important libraries for data processing, visualization, and modelling, such as pandas, NumPy, matplotlib, seaborn, and Plot, are imported first in the process. Changes include utilizing set option to set display options to help improve readability and control over Data Frame outputs. Essential constants and hyperparameters are defined; they include the

[1] *Department of Computer Science and Information Technology, Koneru Lakshmaiah Education Foundation, 522502, Andhra Pradesh, India 2000090122csit@gmail.com*
[2] *Department of Computer Science and Information Technology, Koneru Lakshmaiah Education Foundation, 522502, Andhra Pradesh, India 2000090010csit@gmail.com*
[3] *Department of Computer Science and Information Technology, Koneru Lakshmaiah Education Foundation, 522502, Andhra Pradesh, India 2000090008csit@gmail.com*
[4] *Department of Computer Science Engineering, Koneru Lakshmaiah Education Foundation, 522502, Andhra Pradesh, India jyothiarunkr@kluniversity.in*

Random Forest metrics, the number of estimators, the parallel jobs, the split sizes for validation and testing, and the K-fold values. they components are critical for further investigations.

The process of data loading entails obtaining the dataset of credit card transactions from a CSV file. Integrity checks ensure data quality by identifying and sorting missing values using the null function. Understanding the imbalance between fraudulent and non-fraudulent transactions—a crucial component of fraud detection tasks—is made easier by visualizing the class distribution using a bar plot. Before a density map is produced, separate data groups for fraudulent and non-fraudulent transactions are created using Time data. Thus, the resultant plot provides information on the temporal distribution of transactions between cases that are fraudulent and those that are not.

2.2. Preprocessing the data

Using a variety of classification techniques, including Random Forest, Gradient Boosting, Support Vector Machines (SVM), Logistic Regression, and Neural Networks, a fraud detection model is built and trained during the process.

Feature Selection: Class is the target variable, which stands for the label that indicates whether a transaction is fraudulent. characteristics used in the model training process are included in the Predictors list. These characteristics include Time, V1 through V28 (which are likely anonymized features), and Amount.

Train-Validation-Test Split: First, 'train_test_split' from a package such as Scikit-learn is used to split the dataset into training and test sets. The percentage of the dataset allotted for testing is determined by the 'test size' parameter; 'random state' fixes the random seed to ensure repeatability; and shuffle equates to true randomizes the data before splitting. Additional Splitting for Validation: After this split, the train test split is used once more to split the training set into a training subset and a validation subset. Here, the test size option indicates the percentage of the training set that is set aside for validation.

This process makes sure that the dataset is separated into three parts: a training subset that is used to train the model, a validation subset that is used to evaluate the model's performance during training, and a separate test set that is used to check the generalization skills of the final model. In order to iteratively construct and improve a fraud detection model that can reliably identify fraudulent transactions, the model building process divides the data into separate subsets for training, validation, and testing. It does this by utilizing a variety of classification methods [1].

3. MODEL SELECTION

3.1. Random Forest Classification

The fraud detection model makes use of the Random Forest Classifier. First, Random Forest Classifier function is used to initialize the classifier with a set of hyperparameters, including the number of jobs, the number of estimators (100 in this example), the criteria, the number of random states, and the verbosity [2].

The validation criterion is the GINI index, which is defined as '2 * (AUC) - 1', where AUC stands for the Receiver Operating Characteristic - Area Under Curve (ROC-AUC). In this case, the effectiveness of the model in differentiating between fraudulent and non-fraudulent transactions is measured using the GINI index. To maximize the model's processing performance, the design makes use of four simultaneous workloads and one hundred estimators (trees).

3.2. Ada Boost Classifier

One ensemble learning method designed specifically for classification challenges is the Ada Boost Classifier. It works by combining many poor learners one after the other until it produces a strong classifier that performs better than individual models. Its predictive power is shaped in part by important parameters like random state for reproducibility, algorithm, learning rate which determines the influence of each classifier, and n estimators which indicates the maximum number of weak learners used [3].

The fit function integrates information from the input features, the target variable for classification, and the predictors of training variables when fitting the model to training data. The model can discover the underlying correlations and patterns in the data thanks to this procedure.

The trained model is then used to leverage the predict () function to forecast the target variable of valid data frames. Here, these predictions are made using valid data variable within predictors as the input, which helps assess the model's performance on fresh, untested data.

Visualizing feature importance is a vital step after model training. The significance ratings for each feature are extracted from feature importance in this phase. Based on their relevance in predicting the target variable, these scores are ordered in descending order within a Data Frame with tmp. Lastly, this feature importance are displayed in a bar plot with Seaborn, highlighting the relevance (x-axis) and corresponding importance scores (y-axis). This visualization helps with feature selection or understanding the properties of the dataset by providing insight into which features contribute most substantially to the model's predictions.

3.3. Cat Boost Classifier

The Cat Boost Classifier configuration's hyperparameters

are crucial in determining how the model behaves and performs during training. The number of boosting iterations or trees built throughout the training phase is specified by the parameter iterations=600. This affects the final model's resilience and complexity. The step size, or the pace at which the model adjusts to the training data, is controlled by learning rate=0.02 [4]. While a higher learning rate may hasten convergence at the expense of potentially overshooting ideal values, a lesser learning rate frequently results in more precise parameter modifications for the model.

The greatest depth that individual trees within the ensemble are permitted to be is indicated by depth variable to twelve. This parameter has a big impact on how complicated the model is and how well it can identify complex patterns in the data. While a lesser depth may limit the model's ability to understand intricate correlations, a larger depth may result in overfitting.

The evaluation metric selection, AUC centers on evaluating the model's training performance using the Area Under the ROC Curve (AUC). Higher values suggest better predictive performance. This statistic assesses the model's capacity to distinguish across classes. By fixing the random seed, promotes reproducibility, ensuring that the outcomes hold true after several executions.

Bagging intensity, a method of reducing overfitting by modifying the choice of data samples utilized to build each tree, is controlled by bagging temperature, which is set to 0.2. Lower values may lessen overfitting by limiting the effect of random sampling. Based on the amount of iterations, an early stopping strategy is defined by the option 'Iter'. If the model does not show progress after a certain number of iterations, this approach stops training. This minimizes computing resources and helps avoid overfitting by ending training early if the model's performance starts to decline. The frequency at which evaluation metrics are displayed during training is set by the metric period to VERBOSE_EVAL, which offers information about the model's performance at particular intervals.

3.4. Light GBM

Model training, assessment, and prediction using Light GBM, a gradient boosting technique. The Light GBM model's parameters are set up in the params dictionary and boosting type, goal, and measurement Name the boosting type, evaluation measure, and optimization aim, number leaves, maximum depth, minimum child samples, maximum bin, sub sample, sub sample frequency, col sample by tree, minimum children weight, minimum split gain, n thread, verbose, and scale position weight are among the variables evaluated. control the regularization, tree structure, learning rate, and other hyperparameters with the goal of improving model performance, reducing overfitting, and addressing class imbalance.

The Dataset method is used to prepare the model by building the necessary Dataset data structures for training and validation. The training and validation sets are used to create dataset train and dataset valid, which include feature columns and label data, respectively [5].

Using the given parameters (params), training dataset, and validation dataset, the model training process is initiated using the train method. To avoid overfitting, it includes early halting based on validation results.

Model Evaluation and Variable Importance: The training procedure yields verbose output that periodically shows details about validation performance and iteration rounds. After a certain number of cycles with no progress in validation scores, early halting is initiated. Using plot importance method, variable importance is displayed to show the role of each feature in the model's decision-making process. Finally, the trained model is utilized to predict the target variable for the test data.

4. FEATURE IMPORTANCE

4.1. Random Forest Classifier

The most important features are V17, V12, V14, V10, V11, V16.

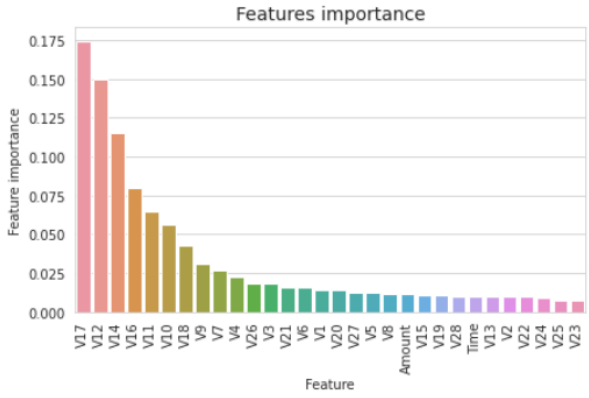

Fig. 1. feature importance of Random forest

4.2. Ada Boost Classifier

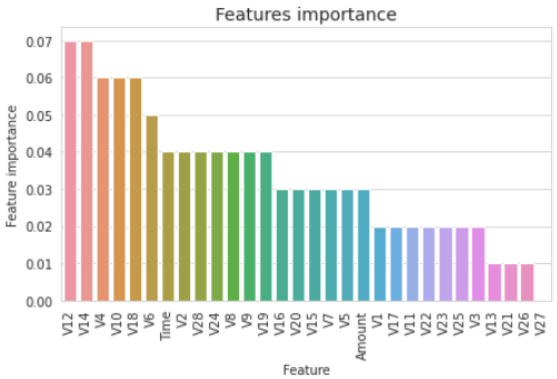

Fig. 2. feature importance of Ada Boost

4.3. Cat Boost Classifier

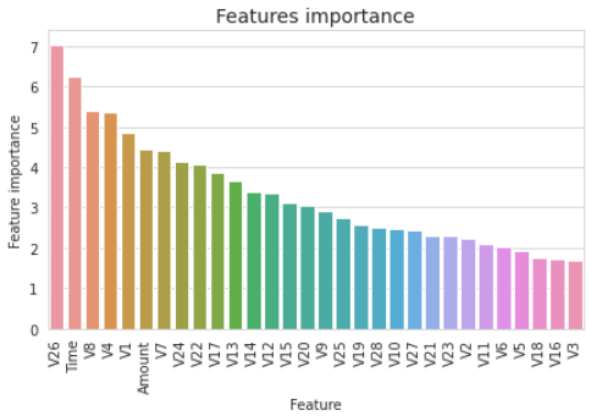

Fig. 3. feature importance of Cat Boost

4.4. Light GBM

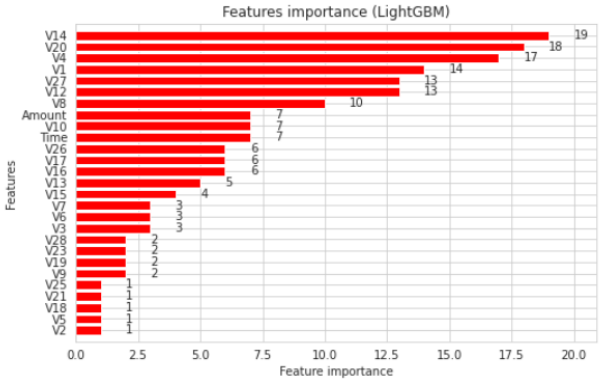

Fig. 4. feature importance of Cat Boost

5. RESULTS

5.1. Random Forest Classifier

5.1.1. Confusion Matrix

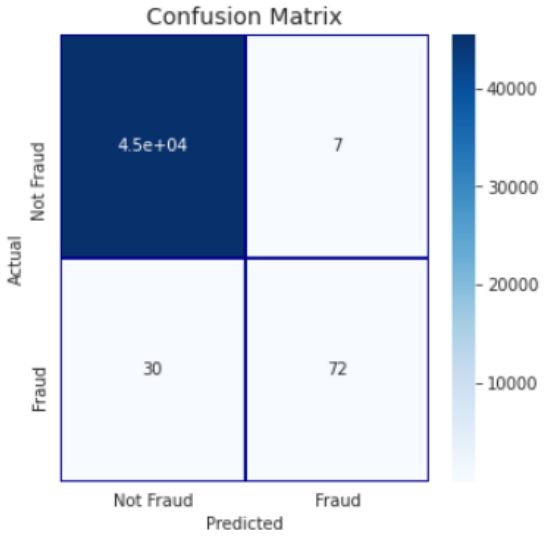

Fig. 5. Confusion Matrix for Random Forest classifier

5.1.2. ROC-AUC

The Receiver Operating Characteristic - Area Under the Curve, or ROC-AUC, score is a commonly used statistic to evaluate a binary classification model's performance [7]. It measures how well the model can distinguish between the positive and negative classes at different thresholds. The Random Forest Classifier, the model in this instance, appears to do a respectable job of differentiating between the positive and negative classes, as indicated by its ROC-AUC score of 0.8511. The score ranges from 0 to 1, with 1 denoting perfect classification and 0.5 denoting random chance (no capacity to discriminate). The model's excellent discrimination ability, as indicated by a ROC-AUC score of 0.8511, suggests that it is generally effective in rating positive instances higher than negative examples.

5.2. Ada Boost Classifier

5.2.1. Confusion Matrix

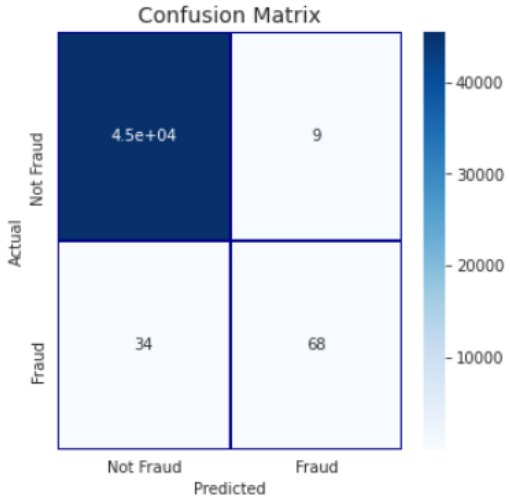

Fig. 6. Confusion Matrix for Ada Boost classifier

5.2.2. ROC-AUC

The Ada Boost Classifier works rather well in differentiating the positive and negative classes, as seen by its ROC-AUC score of 0.8223. This score, which ranges from 0 to 1, indicates that the classifier has strong discrimination capacity and can, for the most part, appropriately rank positive instances higher than negative examples. In contrast, this score (ROC-AUC of 0.5) represents the Ada Boost Classifier's performance on the validation set, demonstrating its ability to make precise predictions and perform well in classification when compared to chance.

5.3. Cat Boost Classifier

5.3.1. Confusion Matrix

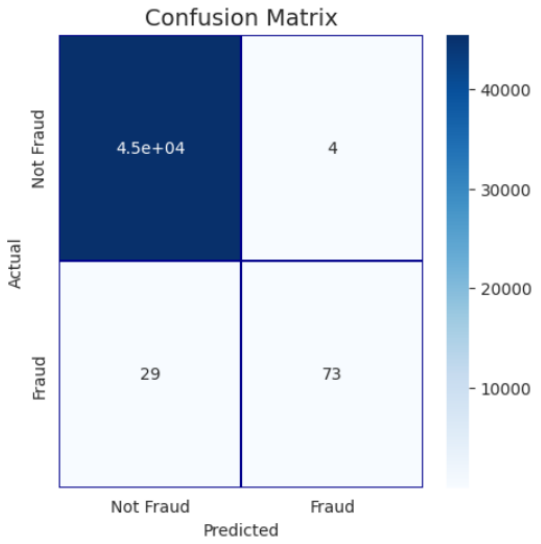

Fig. 7. Confusion Matrix for Cat Boost classifier

5.3.2. ROC-AUC

As indicated by the Cat Boost Classifier's powerful discriminating capacity in differentiating between the positive and negative classes, the ROC-AUC score of 0.8578 is obtained. Based on a range of criteria, this score, which falls between 0 and 1, shows a high ability to correctly rate positive instances over negative examples. Based on the validation set, this score evaluates how well the Cat Boost Classifier performs.

5.4. Light GBM

5.4.1. ROC-AUC

The model works exceptionally well on test data that has not yet been seen, as indicated by its ROC-AUC score of 0.9473 during training and validation. This number, which ranges from 0 to 1, indicates how well the model can accurately rank positive instances over negative examples at different thresholds. When the model's score is near to 1, it indicates that it is highly successful at differentiating between the classes and producing precise predictions on fresh, untested data.

5.4.2. Cross Validation

By training and evaluating the Light GBM model over many subsets of the training data, the cross-validation approach offers a reliable evaluation of the model's performance [6]. It assesses how well the model predicts outcomes and is consistent, giving information on how well it can generalize to new data. The model's performance in discriminating across classes and its discriminative capacity are highlighted by the reported AUC values. AUC values are 0.98765 for every fold [12]. These scores show how well the model performed throughout each fold on distinct validation sets. On the training set, the total AUC score across all folds is 0.934130. The test set predictions' AUC score, which is derived from the average predictions made over the course of five folds, is 0.9332.

TABLE I. RESULTS OF VARIOUS MODELS

Models	***Accuracy***	***ROC-AUC***
Random Forest Classifier	0.85	0.8511
Ada Boost Classifier	0.83	0.8223
Cat Boost Classifier	0.91	0.8578
Light GBM	0.89	0.9473

Conclusion

The experiment demonstrates how machine learning may be used to solve the critical issue of fraud detection, which might have a significant influence on cardholders as well as financial institutions. We assessed the model's performance with a validation set. Its efficacy was evaluated using metrics such as the ROC AUC score, F1 score, log loss, precision, recall, and accuracy. The model's predictions were thoroughly broken out in the confusion matrix, which made it easier to comprehend the true positives, true negatives, false positives, and false negatives. The model performed admirably, displaying good recall, accuracy, and F1 score. This suggests that in identifying fraudulent transactions, it successfully balanced false positives and false negatives. Even if the model performed well on the validation set, it's important to recognize that real-world performance may vary for reasons not included in the dataset.

References

[1] Correa Bahnsen, A., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. Expert Systems with Applications, 51, 134–142. https://doi.org/10.1016/j.eswa.2015.12.030.

[2] Anjali S. More, Dipti P. Rana, An Experimental Assessment of Random Forest Classification Performance Improvisation with Sampling and Stage Wise Success Rate Calculation, Procedia Computer Science, Volume 167, 2020, Pages 1711-1721, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.03.381.

[3] Song, Jie & Lu, Xiaoling & Wu, Xizhi. (2009). An Improved AdaBoost Algorithm for Unbalanced Classification Data. 109-113. 10.1109/FSKD.2009.608.

[4] Oyedeji, Joseph. (2022). Comparative Assessment of Radom Forest, SVC and Cat Boost Performances as Property Price Forecasting Models. 1283-1289.

[5] Sivanandam, Chakaravarthi & Perumal, Vaishnnave & Mohan, Jagadeesh. (2023). A novel light GBM-optimized long short-term memory for enhancing quality and security in web service recommendation system. The Journal of Supercomputing. 1-33. 10.1007/s11227-023-05552-1.

[6] Zhou, Hong. (2023). Cross-Validation and ROC. 10.1007/978-1-4842-9771-1_5.

[7] Tafvizi, Arya & Avci, Besim & Sundararajan, Mukund. (2022). Attributing AUC-ROC to Analyze Binary Classifier Performance. 10.48550/arXiv.2205.11781.

[8] Karimi, Zohreh. (2021). Confusion Matrix.

[9] Ehigie, Aimienrovbiye & Isenmilia, P. & Omoye, A.. (2023). Fraud Pentagon: Detection of Financial Statement Fraud in a Firm. Mediterranean Journal of Social Sciences. 14. 102. 10.36941/mjss-2023-0040.

[10] Zhou, Qingwei & Qi, Yongjun & Tang, Hailin & Wu, Peng. (2023). Machine learning-based processing of unbalanced data sets for computer algorithms. Open Computer Science. 13. 10.1515/comp-2022-0273.

[11] Nápoles, Gonzalo & Griffioen, Niels & Khoshrou, Samaneh & Guven, Cicek. (2023). Feature Importance for Clustering. 10.1007/978-3-031-49018-7_3.

[12] Ferreira, Artur & Figueiredo, Mário. (2023). Leveraging Explainability with K-Fold Feature Selection. 10.5220/0011744400003411.

[13] Oyedele, Opeoluwa. (2023). Determining the optimal number of folds to use in a K-fold cross-validation: A neural network classification experiment. Research in Mathematics. 10. 10.1080/27684830.2023.2201015.

[14] Salian, Prof. (2023). Credit Card Fraudulent Transaction Detection and Prevention. International Journal for Research in Applied Science and Engineering Technology. 11. 3255-3260. 10.22214/ijraset.2023.50849.

[15] Reite, Endre J. & Oust, Are & Bang, Rebecca & Maurstad, Stine. (2023). Changes in credit score, transaction volume, customer characteristics, and the probability of detecting suspicious transactions. Journal of Money Laundering Control. 26. 10.1108/JMLC-06-2022-0087.

[16] Werdiningsih, Indah & Purwanti, Endah & Aditya, Gede & Hidayat, Auliya & Athallah, R. & Sahar, Virda & Wibisono, Tio & Somba, Darren. (2023). Identifying Credit Card Fraud in Illegal Transactions Using Random Forest and Decision Tree Algorithms. Jurnal Sisfokom (Sistem Informasi dan Komputer). 12. 477-484. 10.32736/sisfokom.v12i3.1730.

[17] Ghorbani, Ebrahim & Adoko, Amoussou & Yagiz, Saffet. (2023). Estimation of TBM Penetration rate using Gradient Boosting-based Algorithms.