Academic Journal of Nawroz University (AJNU), Vol.12, No.2, 2023

Copyright ©2017. e-ISSN: 2520-789X

https://doi.org/10.25007/ajnu.v12n2a1612

Exploring the Power of eXtreme Gradient Boosting Algorithm in Machine Learning: a Review

Zeravan Arif Ali[1], Ziyad H. Abduljabbar[2], Hanan A. Taher[3], Amira Bibo Sallow[4], Saman M. Almufti[5]

[1]Department of Information Technology Management, Duhok Polytechnic, Kurdistan Region, Iraq
[2]Department of Information Technology Management, Duhok Polytechnic, Kurdistan Region, Iraq
[3]Department of Information Technology Management, Duhok Polytechnic, Kurdistan Region, Iraq
[4] Department of Information Technology Management, Duhok Polytechnic, Kurdistan Region, Iraq
[5]Department of Computer Science, College of Science, Nawroz University, Kurdistan Region, Iraq

ABSTRACT

The primary objective of machine learning is to extract valuable information from the vast amount of data generated every day, process it to learn from it, and take useful actions. Machine learning has diverse applications in various fields such as language processing, pattern detection, search engines, medical diagnostics, bioinformatics, and chemical informatics. Boost is a recently developed machine learning algorithm that has demonstrated exceptional capabilities in modeling complex systems. It is considered to be the superior machine learning algorithm in terms of prediction accuracy, interpretability, and classification versatility. Boost is an enhanced distributed scaling enhancement library that is built to be extremely powerful, adaptable, and portable. It uses augmented scaling to incorporate machine learning algorithms and is a parallel tree boost that can solve a wide range of data science problems quickly and accurately.Python continues to be the preferred language for scientific computing, data science, and machine learning. It boosts performance and productivity by allowing the use of clean low-level libraries and high-level APIs. This paper presents one of the most prominent supervised and semi-supervised learning (SSL) machine learning algorithms (XGBoost) in a Python environment.

KEY WORDS: Data Science, Machine learning, XGBoost Algorithm,semi-supervised learning, python.

1. INTRODUCTION

Data science is a broad field of study that focuses on the extraction of information and ideas from data by the use of computational techniques, computers, algorithms, and systems. It is based on deep learning, artificial intelligence, and tools for processing large amounts of data. it is a skill set using mathematics, statistics, programming, and business experience. Like any scientific method, it involves gathering data, defining the problem, forming a hypothesis, and running tests. More specifically, data scientists follow the process of data collection and cleaning (debate), investigation (analysis of exploratory data), building automation using machine learning (feature engineering, model development, publishing), view results (visualizations, reporting, storytelling), and maintenance. It is a wide field that encompasses the methods, theories, principles, instruments, and strategies used to review, analyze, and derive expertise and useful information from raw data. Data scientists work with large groups of this information to conclusions. For example, they may use financial data to forecast revenue generation seasonality or use application events (such as logins, clicks, or downloads) to detect security threats or fraud[1] as shown in figure 1.

Machine learning (ML) methods are commonly utilized in industrial business to extract valuable insights and solutions from large amounts of data. ML algorithms are an integral part of the market in a variety of very diverse areas, from medical labs to financial firms[2]. Academic literature on the philosophy and applications of ML techniques is increasing, indicating the increasing relevance of robust ML frameworks for current science and industrial applications. Besides the pure predictive accuracy of the model, most ML technologies in the industry need to be scalable, flexible and handle massive amounts of data in a distributed environment. The ability to interpret a model and explain the predictions it generates is fundamental to minimizing risk and requires an ethics of computational models. ML has three types: supervised, unsupervised, semi-supervised, and Reinforcement Learning[3][4].

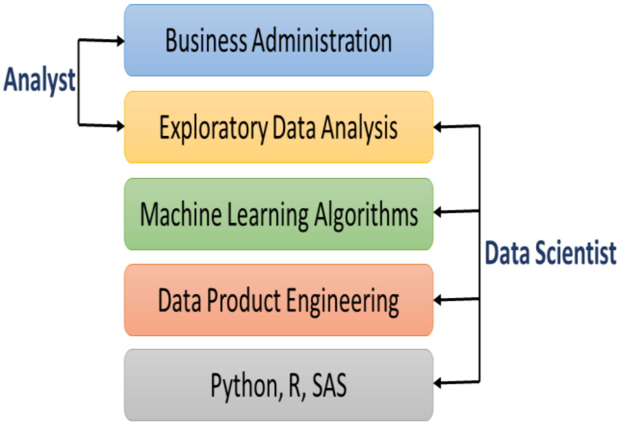


Fig. 1 Data Science

XGBoost is a distributed gradient-boosting library that has been developed to be extremely powerful, scalable, and portable. It utilizes Gradient Boosting to incorporate ML algorithms. XGBoost is a parallel tree-boosting algorithm (also known as GBDT, GBM) that is capable of quickly and accurately solving a large range of data science problems. And is often used by data scientists to achieve breakthrough results on a wide range of ML issues. XGBoost is an ensemble algorithm that focuses on decision- trees and utilizes gradient boosting. In prediction problems affecting unstructured data, artificial neural networks routinely outperform all other architectures or structures (images, text, etc.). However, Decision tree-based algorithms for small to medium-structured/tabular effects are currently considered the strongest in the class. Gradient boosting is a concept that is designed to improve or reinforce a single weak model by combining it with many other weak models to create a strong model together. XGBoost strengthens and improves poor learning mechanisms iteratively[5].

Python is a programming language used on both small and large scales as a general-purpose language for study, creativity, processing, and applications involving data science and ML. Through the efforts of its developers and the open-source group, a wide range of valuable libraries for scientific computing and ML have been created. Python was chosen for ML largely because it is a general-purpose programming language suitable for study, growth, and output both small and large. Python has a dynamic model structure and automated memory control, as well as a broad and robust set of mathematical computing and data processing. It includes many packages that include easy, scalable, and detailed data structures that make working with relational or specific data easier. The two main data models, the String (one-dimensional) and the (two-dimensional) data frame support the vast majority of common use cases in economics, mathematics, the social sciences, and a wide range of engineering disciplines[6].

Predictive analytics has gained a lot of interest in the present era. Statistical methods and ML techniques are applied to historical data to build a predictive model to forecast and recommend future events. Boosting is an ensemble ML technique that combines several low-accuracy models to create a high-accuracy model. It can be utilized in various domains for improving prediction such as credit, insurance, consumer behavior, medical diagnosis, and sales. This paper reviewed the use of XGBoost and Gradient Boosting ensemble ML techniques. The first section an introduction to the basic concepts of the research topic, section 2 provided ML techniques, section 3 submitted the XGBoost algorithm, section 4 reviewed more than 30 papers presented during the recent four years, and section 5 contains a table review for XGBoost algorithm, and section 6 concludes the paper review.

2. Background theory

2.1. Supervised Machine Learning Algorithms

ML Algorithms are described as the algorithms used to train models. They are classified into three types in ML, Supervised Learning (in which the datasets are labeled and techniques such as regression and classification are used), Unsupervised Learning (in which the datasets are not labeled and techniques such as dimension reduction and clustering are used), and Reinforcement Learning (in which the datasets are not labeled and techniques such as dimension reduction and clustering are used)[7], [8]. In Supervised Learning, the data set is named, and there is a corresponding goal data set for and function or independent variable that is used to train the model. ML algorithm challenges can be classified into two categories: classification and regression[9], [10], as shown in figure 2.

Classification**:** classifying a data point into a binary, multinomial, or ordinal class. These algorithms may be classified as linear, non-linear, or tree-based. Linear algorithms such as Linear Regression and Logistic Regression are often used where there is a linear association between the function and the goal variable, while tree-based approaches such as Decision Tree, Random Forest, Gradient Boosting, and others are often used where the data shows non-linear patterns[11][12].

Regression: this is a problem of predicting continuous values based on features of certain data, and is mainly used to predict certain patterns, trends, and trends. In other words, the answer is not just falling like 1 or 0 like a classification. The ML software must estimate – and comprehend – the relationships between variables while performing regression tasks. Regression analysis is especially useful for modeling and prediction since it relies on a single dependent variable and several other shifting variables[13][14].

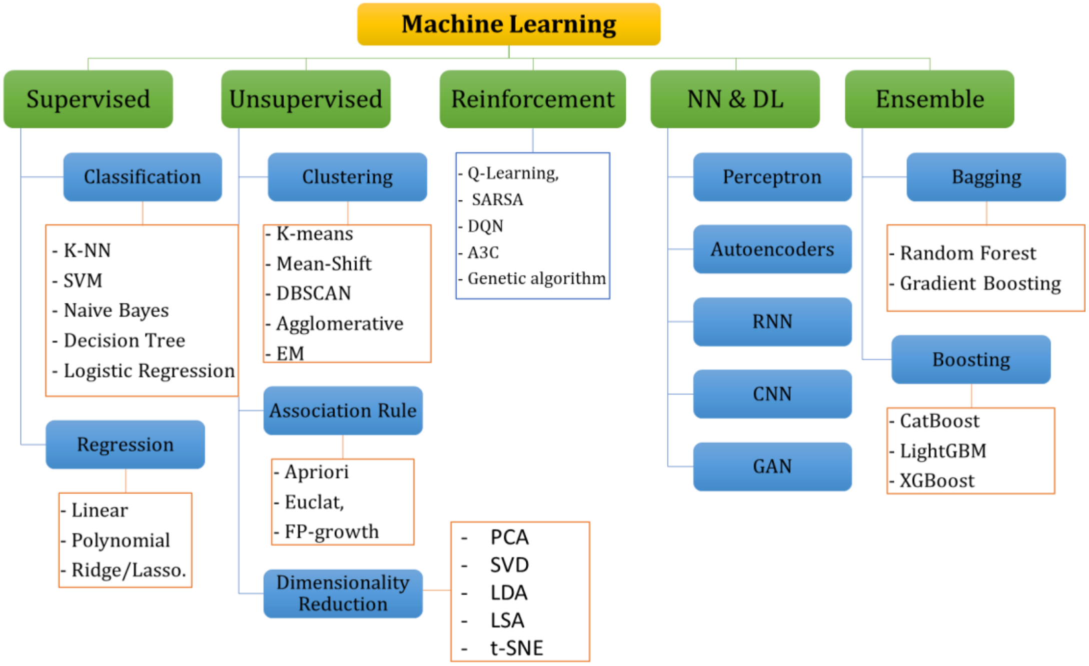


Fig.2 classification of ML Techniques[9]

The most supervised learning models that classify data are:

A. Support Vector Machines (SVM): are supervised learning models that work in conjunction with related learning algorithms to perform classification and regression analysis on results. Along with linear classification, SVMs can conduct effective non-linear classification using a technique called the kernel trick, which involves indirectly mapping their inputs into high-dimensional feature spaces. The SVM algorithm is often used in applications such as intrusion detection, handwriting recognition, protein structure prediction, and detecting steganography in digital images[15][16], as shown in figure 3-d.

B. Logistic Regression: a statistical model which uses a logistic function as its basic type, although there are several more complex extensions, to model a binary dependence variable. The logistic regression model itself simply models the probability of output in terms of input and does not perform statistical classification (it is not a classifier), but it may be used to create one, for example by selecting a cutoff value and classifying inputs with probability greater than or equal to the cutoff as one class and those with probability less than or equal to the cutoff as the other; this is a popular method for creating a binary classifier. It is used in fraud prevention, clinical testing, and other situations where the performance is binary[17], as shown in figure 3-a.

C. Naive Bayes Classifier is one of the most common learning algorithms for classifying data based on the calculation of conditional probability values. It applies the Bayes calculation theorem and class levels used that are interpreted as characteristic values or classification predictors. The Naive Bayes Algorithm is a fast rating algorithm. This algorithm fits quite well in the case of real-time forecasting, multiple class estimation, suggestion scheme, text classification, and sentiment analysis. The Gaussian, Multinomial, and Bernoulli distributions can be used to construct the Naive Bayes Algorithm. The algorithm for large data sets is flexible and simple to apply[18], as shown in figure 3-b.

D. K-Nearest Neighbors (k-NN): is classified as supervised learning's classification component. K Nearest Neighbors is a fundamental algorithm that stores all possible data and forecasts its classification based on a similarity calculation. When two parameters are plotted on a two-dimensional Cartesian device, the similarity measure is determined by measuring the distance between the points. The same holds here; since the KNN algorithm is based on the premise that related objects occur near, we may simply assume that similar objects remain close to one another[19], as shown in figure 3-c.

E. Decision Tree: is a kind of classification algorithm that often solves regression problems using classification rules (from the root to the leaf node). The structure is similar to a flowchart, with each internal node representing a test on a feature (e.g. if a random number is greater than a number or not), and each leaf node representing the solution to the regression problem. The deeper the tree and the more branches, the model is better[20], as shown in figure 3-e.

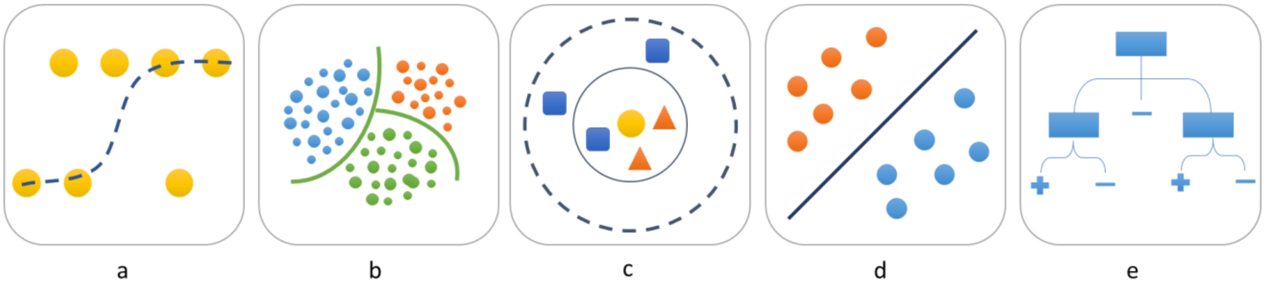


Fig.3 Classification Methods

2.2. Ensemble Learning:

To improve predictive precision, ensemble ML approaches employ several foundation learners. The fundamental insight is to increase the generality of individual classifiers by training them on several datasets (sampled from the initial dataset) and then averaging the effects. Averaging the outputs of several classifiers enables the variance factor and/or prejudice in classification to be reduced. Noise, variance, and bias are the primary causes of discrepancy between observed and expected values. Ensemble ML contributes to the reduction of variance and prejudice. Bagging and boosting are two commonly employed approaches for ensemble instruction. Bagging decreases variance while boosting reduces both variance and classification bias[21].

A. Bagging: The bagging method's goal is to reduce the model's large variance. The decision trees are poor in variation and have a low bias. Subsamples are taken from the large dataset. The multiple decision trees are constructed using the training data from each subsample. By slamming the subsampled data into the various decision trees, the risk of each decision tree becoming over-fit with training data is minimized. To increase the model's productivity, each of the individual decision trees is grown deep using subsampled training data. To comprehend the final forecast, the outcomes of each decision tree are aggregated. The volatility in aggregated results decreases. The precision of the model's estimation in the bagging process is proportional to the number of decision trees used. With substitution, the different sub-samples of a sample data set are randomly selected. Each tree's production is highly correlated[22].

B. Boosting: The trees are constructed sequentially, with each following tree attempting to minimize the errors of the preceding tree. Each tree builds on the knowledge gained by its predecessors and updates the residual errors. As a result, the tree that follows will benefit from a modified version of the residuals. In boosting, the foundation learners are weak learners with a strong bias and predictive ability that is only a tad stronger than random guessing. Each of these weak learners contributes critical knowledge for estimation, allowing the boosting technique to successfully combine these weak learners to create a strong learner. The final powerful learner reduces both bias and variation[23], as shown in figure 4.

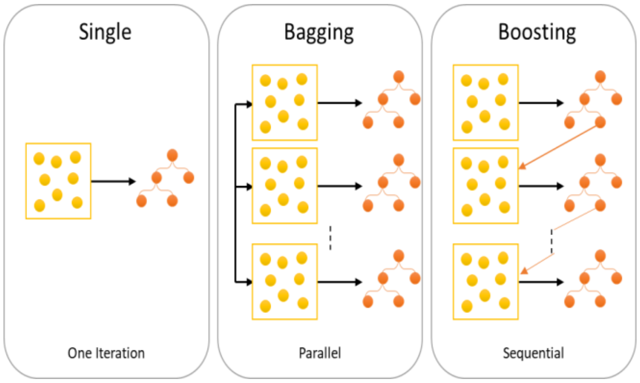
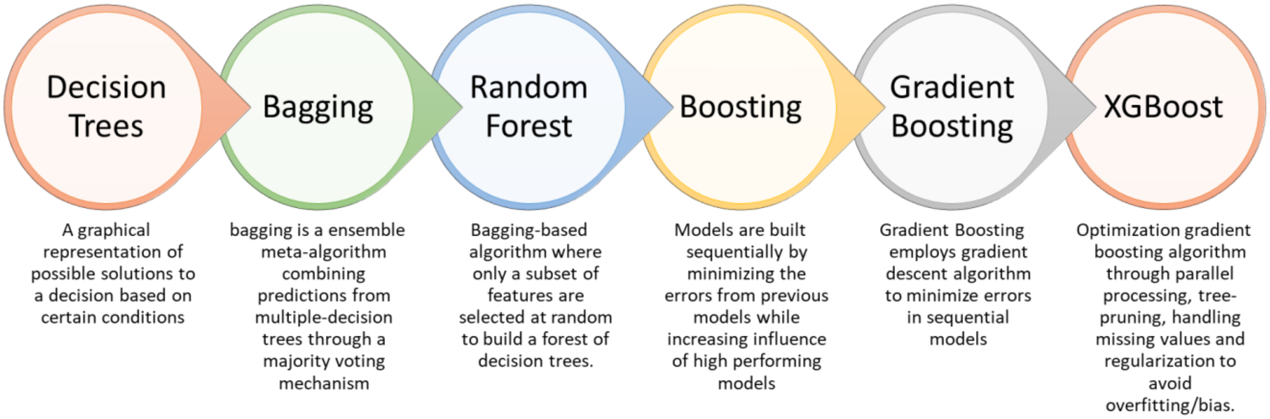


Fig.4 Ensemble Learning[24]

C. Gradient boosting: The gradient boost algorithm has been developed to have very high predictive. However, its adoption was very limited because the algorithm required generating one decision tree at a time to reduce errors of all previous trees in the model. So it took a long time to train even those small-sized models. Then came a new algorithm called eXtreme Gradient Boosting (XGBoost) that changed how gradient boosting was done. In XGBoost, individual trees are generated using multiple cores, and data are organized to reduce search times. This reduced the training time of the models which in turn increased the performance. This research study seeks to establish a quantitative comparison of the accuracy and speed of the XGBoost algorithm in multi-threaded single-system mode and enhance the gradation with different datasets[25], as shown in figure 5.



Decision Trees

A graphical representation of possible solutions to a decision based on certain conditions

Bagging

bagging is a ensemble meta-algorithm combining predictions from multiple-decision trees through a majority voting mechanism

Random Forest

Bagging-based algorithm where only a subset of features are selected at random to build a forest of decision trees.

Boosting

Models are built sequentially by minimizing the errors from previous models while increasing influence of high performing models

Gradient Boosting

Gradient Boosting employs gradient descent algorithm to minimize errors in sequential models

XGBoost

Optimization gradient boosting algorithm through parallel processing, tree-pruning, handling missing values and regularization to avoid overfitting/bias.

Fig.5 tree-based algorithms[26]

3. XGBoost Algorithm

A. *XGBoost is an ensemble ML algorithm that is dependent on decision trees and employs a gradient-boosting method. The XGBoost algorithm was created at the University of Washington as part of a research project. It was introduced in 2016 by Tianqi Chen and Carlos Guestrin. Not only has this algorithm been credited with winning several Kaggle tournaments, but it has also been credited with being the guiding force behind many cutting-edge industry applications. As a consequence, a sizable group of data scientists contributes to the XGBoost open-source project's GitHub commits[27].*

B. *XGBoost is an optimized gradient tree boosting method that generates sequential decision trees. It is capable of performing related calculations very quickly in all computational environments. As a result, XGBoost is* *commonly used for its ability to model newer attributes and classify marks. The XGBoost algorithm's use has exploded in popularity due to its implementations in tabular and structured datasets[28].*

C. *The XGBoost algorithm evolved from a decision tree-based method in which graphical representations of alternative decision solutions are computed based on some parameters. Then, using a majoritarian voting strategy, an ensemble meta-algorithm called 'bagging' was developed for aggregating predictions from different decision trees. By randomly choosing features, this bagging technique was developed to create a forest or aggregation of decision trees[29].*

D. *The performance of the models was improved by minimizing errors associated with the construction of sequential models. Additionally, the gradient descent algorithm was used to minimize errors in the sequential model. Finally, the XGBoost algorithm was described as a beneficial approach for optimizing the gradient boosting algorithm via the removal of missing values and the elimination of overfitting issues through parallel processing. The XGBoost algorithm optimizes the system by the use of parallelization, tree pruning, and hardware optimization. The algorithm supports three types of gradient boosting: ML-based gradient boosting; stochastic gradient boosting through subsampling; and regularized gradient boosting via L1 and L2 regularizations[30] as shown in figure 6.*

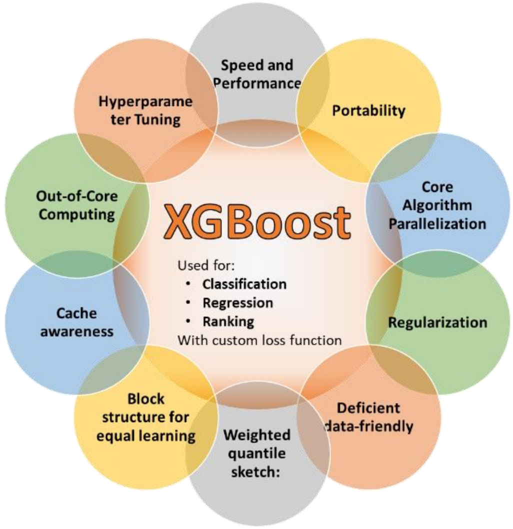


Fig.6 XGBoost features[31]

3.1. Algorithm Optimization:

Sparsity Aware Split Finding: XGBoost deals with missing data by assigning it to the default orientation and determining the best computation value to mitigate training error. The improvement here is that the algorithm visits only the missed values, which allows it 50 times quicker than the naïve approach[31].

Regularization: It corrects more complicated models to avoid overfitting by applying both the LASSO (also known as L1) and Ridge regularization (also called L2)[32], as shown in figure 7.

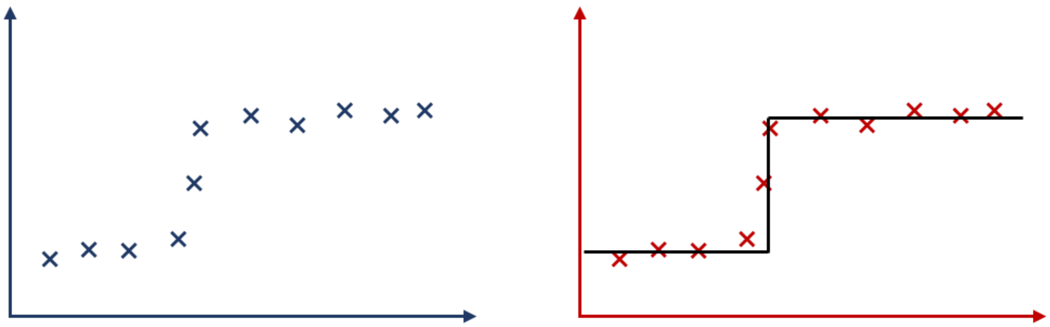


Fig.7 good balance[33]

Cross-Validation: It has built-in cross-validation attributes that are introduced during each model development iteration. This eliminates the need to compute the amount of boosting iterations needed[34].

Distributed Weighted Quantile Sketch: It employs a distributed weighted quantile sketch to determine the optimum number of breakpoints between weighted datasets[35].

3.2. System improvements:

Tree Pruning: Pruning is the ML method for shrinking regression trees by removing nodes that do not lead to leaf classification improvement. Pruning a regression tree is used to avoid overfitting the training results. The most effective pruning approach is Cost Complexity or Weakest Link Pruning, which utilizes mean square error, k-fold cross-validation, and a learning rate internally. XGBoost generates nodes (also known as splits) up to the stated max depth and then begins pruning backward until the loss is less than a specified threshold[36].

Parallelization: Tree learning requires ordered results. To reduce sorting expenses, data is compacted into blocks (each column with a corresponding feature value). XGBoost sorts each block in parallel, using all usable CPU cores/threads. This optimization is advantageous since a tree always generates a huge number of nodes. Simply put, XGBoost parallelizes the sequential method of tree generation[37].

Cache Aware: stored gradient statistics (direction and value) for each split node in an internal buffer of each thread and accumulate them in a mini-batch manner using cache-aware optimization. This contributes to the reduction of the time overhead associated with immediate read/write operations and also helps avoid cache misses. Cache recognition is accomplished by selecting the optimal block size[26].

Execution Speed: XGBoost was almost always quicker than the other implementations benchmarked, and it continues to be faster as opposed to other algorithms[38].

Model Performance: On classification and regression predictive modeling problems, XGBoost dominates hierarchical or tabular datasets[39].

3.3. XGBoost Parameters

XGBoost defines three categories of parameters: general, booster, and task parameters.

- General parameters specify the type of booster we can use to do the boosting, which is usually a tree or linear model[40].
- Booster parameters vary according to the booster selected.
- Acquaintance with task parameters chooses a learning situation. For instance, regression tasks and ranking tasks can need different parameters[41], as shown in figure 8.

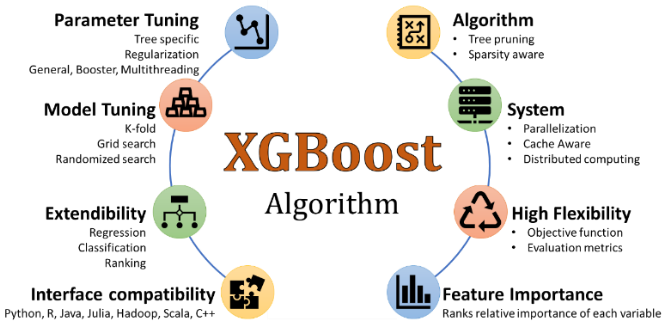


Fig. 8 XGBoost algorithm[30]

4. Related Work

This paper will review the previous research on the applications of the XGBoost algorithm in the areas of prediction, classification, and system optimization.

4.1 Prediction with XGBoost

B. Yu et al. (2020) [42], suggested a novel approach for protein submitochondrial localization estimation, SubMito-XGBoost. and the ReliefF algorithm is used to pick features. On the two training datasets M317 and

M983, the SubMito-XGBoost system achieved a prediction accuracy of 97.7 percent and 98.9 percent, respectively, which is 2.8-12.5 percent and 3.8-9.9 percent higher than other approaches.

C. Midroni, & et al. (2018) [43], used XGBoost to estimate blood glucose levels at a 30-minute horizon in the OhioT1DM dataset for the 2018 BGLP Challenge. The studies demonstrated that, as opposed to a prior study, XGBoost may be a competitive indicator of blood glucose levels, and that signals from several sources lead in differing degrees to XGBoost's enhanced predictive capacity.

X. Ma & et al (2018) [44], cleaned the data using a 'multi observation' and 'multi-dimensional approach and applied the current ML algorithms LightGBM in Asia at the end of 2016 and XGboost, which are focused on actual P2P transaction data from Lending Club. The platform's default risk is strongly and innovatively expected. And the findings indicate that the LightGBM algorithm, which is built on several empirical data sets, produces the best classification prediction results. The overall success rate of the Lending Club platform's historical transaction data increased by 1.28 percentage points, resulting in a reduction of approximately $117 million in loan defaults.

Y. Liang et al. (2019) [45], used the XGboost and LightGBM algorithms to do a statistical study of market volume in the commodity sales data collection. The XGboost and LightGBM algorithms are studied in detail, as are the expected artifacts and environments, as well as the algorithm parameters and data set characteristics. The optimal parameters were determined, and the revenue amount from January to October 2015 was predicted using the optimal parameters, as well as the root mean square error (RMSE) values for the two algorithms. Statistical research reveals that there is no statistically important discrepancy between the two algorithms' optimum prediction outcomes when their parameters are modified.

Y. Song et al. (2019) [46], proposed a prediction model for double-high diseases based on the LightGBM and XGBoost algorithms, after analyzing the effects of various features and their weights on double-high diseases using ML. The prediction model is constructed using data from the user's physical examination and five biochemical indicators such as systolic blood pressure, diastolic blood pressure, serum triglycerides, and serum cholesterol. Finally, model fusion is achieved using the model's output. The mean square error (MSE) between the expected and real values is used as the estimation criterion, and the likelihood that consumers have the double-high disease is predicted.

4.2 Classification with XGBoost

Z. Chen & et al. (2018) [47], used (XGBoost), used the POX as an SDN controller, Mininet to create an SDN topology, and Hyenae to simulate a real DDoS attack area. The XGBoost classifier contrasts itself with other classifiers using the flow packet data collection collected by TcpDump for DDoS identification. The detection results demonstrate that our system is more accurate, has a lower false-positive rate, is fast, and is scalable.

L. Chao & et al. (2019) [48], built the star/galaxy classification model using the XGBoost algorithm and obtained the full photometric data set from the SDSS-DR7, which he separated into a bright source set and a dark source set based on the star magnitude. Then, using the grid scan and other tools, the XGBoost parameters are modified. The experimental findings indicate that the XGBoost algorithm increases the classification precision of galaxies in the dark source classification by nearly 10% when compared to the function tree algorithm, and nearly 5% when compared to the function tree algorithm for sources with the darkest magnitudes in the dark source range. XGBoost often exhibits varying degrees of development as compared to more standard ML algorithms and deep neural networks.

C. Wang & et al. (2020) [49], assessed Imbalance-success XGBoost's on a Parkinson's disease classification dataset thoroughly, and various competitive performances are provided using the Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves. To demonstrate the methods' supremacy, success on four additional benchmark datasets from the UCI ML repository is also recorded. Given XGBoost's scalability, it has the potential to be widely applicable to real-world binary classification tasks, which are often large-scale and label-imbalanced.

N. Manju & et al. (2019) [50], suggested an Internet traffic classification XGboost algorithm network control and control. Because internet traffic detection is a multiclass challenge, a large number of emerging ML models perform poorly. The findings show that using just eight chosen elements, we were able to identify dataset01 and dataset02 with an accuracy of 98.51 percent and 93.54 percent, respectively. Additionally, reducing the number of features reduces the computing overhead.

S. Thongsuwan & et al. [51], defined the new model of profound learning — Convolutionary eXtreme Gradient Boosting (ConvXGB) to the problem of classification centered on neural convolutions networks and XGBoost. Besides the picture info, with a data preprocessing module, ConvXGB supports general classification issues. ConvXGB consists of various stacked coevolutionary layers for learning the features of the input, preceded by the last layer of XGBoost for the prediction of the class marks. Experiments have shown that the grouping issues of both data sets were much improved for our model than CNN and XGBoost alone and were often considerably improved.

4.3 System Optimization with XGBoost

K. Song & et al. (2019) [52], suggested an optimization paradigm for continuous multivariable optimization that combined the XGBoost algorithm with an enhanced PSO. The purpose of this project is to evaluate mapping functions and their influences between stress and plasticity based on a diversity of models such as linear regression, SVM, XGBoost, etc. After assessing the efficiency of these models, we pick the most accurate XGBoost model as the mapping feature which in previous studies was not performed. In addition, the mapping function calculated acts as the health value for optimization of the particle swarm, which allows optimization of tensile strength and plasticity for several variables. Lastly, the effects are technically evaluated and shown to be effective and consistent.

Y. Wang (2019) [53], researched market risk classification models using the severe gradient boosting (XGBoost) technique. During model preparation, both feature selection (FS) algorithms and hyper-parameter optimization are regarded concurrently. The output of XGBoost is compared to that of the more commonly used logistic regression (LR) model in terms of classification precision, region under the curve (AUC), recall, and the F1 score obtained from 10-fold cross-validation. The results indicate that hierarchical clustering is the optimum FS approach for LR, while weight by Chi-square performs the best in XG-Boost. The model analysis is aided by the rating of function value dependent on XGBoost. Thus, XGBoost combined with Bayesian TPE hyper-parameter optimization provides a practical and efficient solution to market risk modeling.

K. Budholiya (2020) [54], introduced a diagnostic device that predicts cardiac disease using an optimized XGBoost classifier and Bayesian optimization, a highly efficient approach for hyperparameter optimization. Additionally, we used the One-Hot (OH) encoding strategy to increase prediction accuracy by encoding categorical features in the dataset. On the Cleveland heart disease dataset, the suggested model's effectiveness is tested and contrasted to that of Random Forest (RF) and Extra Tree (ET) classifiers. Five separate output measures were used to evaluate performance: precision, sensitivity, specificity, F1-score, and AUC (area under the curve) of ROC charts. The experimental findings established the relevance and effectiveness of the model in predicting cardiovascular disease. Additionally, the suggested model achieves a 91.8 percent estimation precision. These findings suggest that the proposed procedure may be used to forecast cardiac attacks in the clinic with reasonable accuracy.

J. Guo et al. (2019)[55], suggested a classification module that makes use of an (XGBoost) algorithm to classify each teenager's physical fitness level from preprocessed PPG signals, with hyper-parameters adaptively calibrated using Bayesian optimization. Not only do the experimental findings show that the proposed model outperforms current reference models in terms of evaluation precision, but they also provide a promising approach for possible physical health evaluations of adolescents using intelligent computation rather than conventional observational model-based manual estimation.

J. Zhou & et al. (2020) [56], suggested improving prediction model accuracy by modeling the TBM AR using a hybrid model of (XGBoost) and Bayesian optimization (BO). The results indicate that the BO algorithm is more effective at capturing hyper-parameters for the XGBoost prediction model than the default XGBoost model is. The robustness and generalizability of the BO-XGBoost model resulted in significant RMSE and R2 values. The findings validated the proposed BO-XGBoost concept. Additionally, vector importance experiments were used to interpret the XGBoost concept, demonstrating that system parameters have a greater effect than rock mass and material properties.

5. Discussion

XGBoost is a faster algorithm when compared to other algorithms because of its parallel and distributed computing. XGBoost was designed with a strong emphasis on device optimization and ML concepts. The objective of this library is to push the extremes of the machinery to have a versatile, compact, and precise library[44], [48], [49].

The XGBoost algorithm now includes power over the model's difficulty. Random selection of samples and features during testing reduces the likelihood of the learned model overfitting, which increases the model's generalizability, and ultimately reduces the predictive errors for the validation and test sets substantially. Additionally, XGBoost places a higher premium on model interpretability, allowing us to determine which hallmark genes have a stronger effect on the expression value of each target gene[53].

Simultaneously, although the XGBoost algorithm maintains a serial arrangement across trees, the same-level nodes may be parallelized and the CPU's multi-threading is automatically used for parallel computation, resulting in a quicker XGBoost model than conventional tree models and a higher functional benefit[54].

6. Conclusion

Machine learning has found applications in diverse fields such as language analysis, pattern recognition, medical diagnostics, bioinformatics, and chemical computer technology. Among the various machine learning algorithms, XGBoost stands out for its exceptional capability in modeling complex systems, superior prediction accuracy, interpretability, and classification versatility. XGBoost is a powerful and adaptable distributed scaling enhancement library that uses augmented scaling to incorporate machine learning algorithms. It is a parallel tree boost that addresses various data science problems accurately and quickly. With the increasing amount and diversity of data in the production world, consumers value XGBoost's functionality, scalability, and robustness. As an active open-source community, XGBoost continues to evolve and remain one of the most prominent supervised and semi-supervised learning machine learning algorithms in the Python environment.

Reference

[1] M. Khalaf *et al.*, "A Data Science Methodology Based on Machine Learning Algorithms for Flood Severity Prediction," Sep. 2018, doi: 10.1109/CEC.2018.8477904.

[2] "CLASSIFICATION BASED ON SEMI-SUPERVISED LEARNING: A REVIEW | Iraqi Journal for Computers and Informatics." http://ijci.uoitc.edu.iq/index.php/ijci/article/view/277 (accessed May 20, 2021).

[3] A. Dey, "Machine Learning Algorithms: A Review," *Int. J. Comput. Sci. Inf. Technol.*, vol. 7, no. 3, pp. 1174–1179, 2016, [Online]. Available: www.ijcsit.com.

[4] N. M. Abdulkareem and A. M. Abdulazeez, "Machine Learning Classification Based on Radom Forest Algorithm: A Review," *Int. J. Sci. Bus.*, vol. 5, no. 2, pp. 128–142, 2021, Accessed: May 20, 2021. [Online]. Available: https://ideas.repec.org/a/aif/journl/v5y2021i2p128-142.html.

[5] J. Pesantez-Narvaez, M. Guillen, and M. Alcañiz, "Predicting motor insurance claims using telematics data—XGboost versus logistic regression," *Risks*, vol. 7, no. 2, 2019, doi: 10.3390/risks7020070.

[6] S. Raschka, J. Patterson, and C. Nolet, "Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence," *Inf.*, vol. 11, no. 4, 2020, doi: 10.3390/info11040193.

[7] "(2) Machine Learning Supervised Algorithms of Gene Selection: A Review | Request PDF." https://www.researchgate.net/publication/341119469_Machine_Learning_Supervised_Algorithms_of_Gene_Selection_A_Review (accessed May 20, 2021).

[8] O. Ahmed and A. Brifcani, "Gene Expression Classification Based on Deep Learning," in *4th Scientific International Conference Najaf, SICN 2019*, Apr. 2019, pp. 145–149, doi: 10.1109/SICN47020.2019.9019357.

[9] R. N. Behera, K. Das, B. Tech, and A. Professor, "A Survey on Machine Learning: Concept, Algorithms and Applications Machine Learning View project International Journal of Innovative Research in Computer and Communication Engineering A Survey on Machine Learning: Concept, Algorithms and Applications," *Artic. Int. J. Innov. Res. Comput.*, 2017, doi: 10.15680/IJIRCCE.2017.

[10] S. Athmaja, M. Hanumanthappa, and V. Kavitha, "A survey of machine learning algorithms for big data analytics," in *Proceedings of 2017 International Conference on Innovations in Information, Embedded and Communication Systems, ICIIECS 2017*, Jan. 2018, vol. 2018-Janua, pp. 1–4, doi: 10.1109/ICIIECS.2017.8276028.

[11] M. Iqbal, I. Muhammad, and Z. Yan, "SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY machine learning, SEO, Virtual Reality View project Content Management System View project SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY," *Artic. Int. J. Soft Comput.*, 2015, doi: 10.21917/ijsc.2015.0133.

[12] H. Yahia, A. A.-I. J. of S. and, and undefined 2021, "Medical Text Classification Based on Convolutional Neural Network: A Review," *ideas.repec.org*, Accessed: May 06, 2021. [Online]. Available: https://ideas.repec.org/a/aif/journl/v5y2021i3p27-41.html.

[13] P. C. Sen, M. Hajra, and M. Ghosh, "Supervised Classification Algorithms in Machine Learning: A Survey and Review," in *Advances in Intelligent Systems and Computing*, 2020, vol. 937, pp. 99–111, doi: 10.1007/978-981-13-7403-6_11.

[14] A. Mohsin Abdulazeez, D. Zeebaree, D. M. Abdulqader, and D. Q. Zeebaree, "Machine Learning Supervised Algorithms of Gene Selection: A Review Machine Learning View project How To Choose A Performance Metric View project Machine Learning Supervised Algorithms of Gene Selection: A Review," 2020. Accessed: May 06, 2021.

[Online]. Available: https://www.researchgate.net/publication/341119469.

[15] S. C. Dharmadhikari, M. Ingle, and P. Kulkarni, "Empirical Studies on Machine Learning Based Text Classification Algorithms," *Adv. Comput. An Int. J. (ACIJ)*, vol. 2, no. 6, 2011, doi: 10.5121/acij.2011.2615.

[16] D. Mustafa Abdullah and A. Mohsin Abdulazeez, "Machine Learning Applications based on SVM Classification A Review," *Qubahan Acad. J.*, vol. 1, no. 2, pp. 81–90, Apr. 2021, doi: 10.48161/qaj.v1n2a50.

[17] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: A review of classification and combining techniques," *Artif. Intell. Rev.*, vol. 26, no. 3, pp. 159–190, Nov. 2006, doi: 10.1007/s10462-007-9052-3.

[18] P. Strecht, L. Cruz, C. Soares, J. Mendes-Moreira, and R. Abreu, "A Comparative Study of Classification and Regression Algorithms for Modelling Students' Academic Performance," International Educational Data Mining Society. e-mail: admin@educationaldatamining.org; Web site: http://www.educationaldatamining.org, Jun. 2015.

[19] H. Q. Tran and C. Ha, "Improved visible light-based indoor positioning system using machine learning classification and regression," *Appl. Sci.*, vol. 9, no. 6, p. 1048, Mar. 2019, doi: 10.3390/app9061048.

[20] J. Alzubi, A. Nayyar, and A. Kumar, "Machine Learning from Theory to Algorithms: An Overview," in *Journal of Physics: Conference Series*, Nov. 2018, vol. 1142, no. 1, p. 12012, doi: 10.1088/1742-6596/1142/1/012012.

[21] B. Choubin, E. Moradi, M. Golshan, J. Adamowski, F. Sajedi-Hosseini, and A. Mosavi, "An ensemble prediction of flood susceptibility using multivariate discriminant analysis, classification and regression trees, and support vector machines," *Sci. Total Environ.*, vol. 651, pp. 2087–2096, Feb. 2019, doi: 10.1016/j.scitotenv.2018.10.064.

[22] D. P. P. Mesquita, J. P. P. Gomes, and A. H. Souza Junior, "Ensemble of Efficient Minimal Learning Machines for Classification and Regression," *Neural Process. Lett.*, vol. 46, no. 3, pp. 751–766, Dec. 2017, doi: 10.1007/s11063-017-9587-5.

[23] R. Costache *et al.*, "Novel hybrid models between bivariate statistics, artificial neural networks and boosting algorithms for flood susceptibility assessment," *J. Environ. Manage.*, vol. 265, p. 110485, Jul. 2020, doi: 10.1016/j.jenvman.2020.110485.

[24] G. Biau, B. Cadre, and L. Rouvière, "Accelerated gradient boosting," *Mach. Learn.*, vol. 108, no. 6, pp. 971–992, Jun. 2019, doi: 10.1007/s10994-019-05787-1.

[25] F. Sigrist, "Gradient and Newton boosting for classification and regression," *Expert Syst. Appl.*, vol. 167, p. 114080, Apr. 2021, doi: 10.1016/j.eswa.2020.114080.

[26] R. Mitchell, A. Adinets, T. Rao, and E. Frank, "XGBoost: Scalable GPU accelerated learning," *arXiv*, pp. 1–5, 2018.

[27] W. Niu, T. Li, X. Zhang, T. Hu, T. Jiang, and H. Wu, "Using XGBoost to Discover Infected Hosts Based on HTTP Traffic," *Secur. Commun. Networks*, vol. 2019, 2019, doi: 10.1155/2019/2182615.

[28] D. Uenoyama, H. Yoshiura, and M. Ichino, "Personal authentication of iris and periocular recognition using XGBoost," *2019 IEEE 8th Glob. Conf. Consum. Electron. GCCE 2019*, pp. 186–187, 2019, doi: 10.1109/GCCE46687.2019.9015469.

[29] S. Zhao *et al.*, "Mutation grey wolf elite PSO balanced XGBoost for radar emitter individual identification based on measured signals," *Meas. J. Int. Meas. Confed.*, vol. 159, p. 107777, 2020, doi: 10.1016/j.measurement.2020.107777.

[30] C. Zopluoglu, "Detecting Examinees With Item Preknowledge in Large-Scale Testing Using Extreme Gradient Boosting (XGBoost)," *Educ. Psychol. Meas.*, vol. 79, no. 5, pp. 931–961, 2019, doi: 10.1177/0013164419839439.

[31] D. Bhulakshmi and G. Gandhi, "The Prediction of Diabetes in Pima Indian Women Mellitus Based on XGBOOST Ensemble Modeling Using Data Science The Prediction of Diabetes in Pima Indian women Mellitus Based on XGBOOST Ensemble Modeling using data science," 2020.

[32] M. A. Fauzan and H. Murfi, "The accuracy of XGBoost for insurance claim prediction," *Int. J. Adv. Soft Comput. its Appl.*, vol. 10, no. 2, pp. 159–171, 2018.

[33] A. Pathy, S. Meher, and B. P, "Predicting algal biochar yield using eXtreme Gradient Boosting (XGB) algorithm of machine learning methods," *Algal Res.*, vol. 50, no. April, p. 102006, 2020, doi: 10.1016/j.algal.2020.102006.

[34] R. Zhong, R. Johnson, and Z. Chen, "Generating pseudo

density log from drilling and logging-while-drilling data using extreme gradient boosting (XGBoost)," *Int. J. Coal Geol.*, vol. 220, no. July 2019, p. 103416, 2020, doi: 10.1016/j.coal.2020.103416.

[35] R. Santhanam, N. Uzir, S. Raman, and S. Banerjee, "Experimenting XGBoost Algorithm for Prediction and Classification of Different Ramraj S , Nishant Uzir , Sunil R and Shatadeep Banerjee Experimenting XGBoost Algorithm for Prediction and Classi fi cation of Different Datasets," *Int. J. Control Theory Appl.*, vol. 9, no. March, pp. 651–662, 2017.

[36] R. Sundaram, "An End-to-End Guide to Understand the Math behind XGBoost," *Anal. Vidhja*, 2018, [Online]. Available: https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/.

[37] R. Zhang, B. Li, and B. Jiao, "Application of XGboost Algorithm in Bearing Fault Diagnosis," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 490, no. 7, 2019, doi: 10.1088/1757-899X/490/7/072062.

[38] "XGBFEMF: An XGBoost-Based Framework for Essential Protein Prediction | IEEE Journals & Magazine | IEEE Xplore." https://ieeexplore.ieee.org/abstract/document/8370098 (accessed May 20, 2021).

[39] Y. Qiu, J. Zhou, M. Khandelwal, H. Yang, P. Yang, and C. Li, "Performance evaluation of hybrid WOA-XGBoost, GWO-XGBoost and BO-XGBoost models to predict blast-induced ground vibration," *Eng. Comput.*, pp. 1–18, Apr. 2021, doi: 10.1007/s00366-021-01393-9.

[40] X. Ji, W. Tong, Z. Liu, and T. Shi, "Five-feature model for developing the classifier for synergistic vs. Antagonistic drug combinations built by XGboost," *Front. Genet.*, vol. 10, no. JUL, p. 600, Jul. 2019, doi: 10.3389/fgene.2019.00600.

[41] X. Liao, N. Cao, M. Li, and X. Kang, "Research on Short-Term Load Forecasting Using XGBoost Based on Similar Days," in *Proceedings - 2019 International Conference on Intelligent Transportation, Big Data and Smart City, ICITBS 2019*, Mar. 2019, pp. 675–678, doi: 10.1109/ICITBS.2019.00167.

[42] B. Yu *et al.*, "SubMito-XGBoost: Predicting protein submitochondrial localization by fusing multiple feature information and eXtreme gradient boosting," *Bioinformatics*, vol. 36, no. 4, pp. 1074–1081, 2020, doi: 10.1093/bioinformatics/btz734.

[43] C. Midroni, P. J. Leimbigler, G. Baruah, M. Kolla, A. J. Whitehead, and Y. Fossat, "Predicting glycemia in type 1 diabetes patients: Experiments with XGBoost," *CEUR Workshop Proc.*, vol. 2148, pp. 79–84, 2018.

[44] X. Ma, J. Sha, D. Wang, Y. Yu, Q. Yang, and X. Niu, "Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGboost algorithms according to different high dimensional data cleaning," *Electron. Commer. Res. Appl.*, vol. 31, pp. 24–39, 2018, doi: 10.1016/j.elerap.2018.08.002.

[45] Y. Liang *et al.*, "Product marketing prediction based on XGboost and LightGBM algorithm," *ACM Int. Conf. Proceeding Ser.*, no. 1, pp. 150–153, 2019, doi: 10.1145/3357254.3357290.

[46] Y. Song *et al.*, "Prediction of double-high biochemical indicators based on lightGBM and XGBoost," *ACM Int. Conf. Proceeding Ser.*, pp. 189–193, 2019, doi: 10.1145/3349341.3349400.

[47] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu, and J. Peng, "XGBoost Classifier for DDoS Attack Detection and Analysis in SDN-Based Cloud," *Proc. - 2018 IEEE Int. Conf. Big Data Smart Comput. BigComp 2018*, pp. 251–256, 2018, doi: 10.1109/BigComp.2018.00044.

[48] L. Chao, Z. Wen-hui, and L. Ji-ming, "Study of Star/Galaxy Classification Based on the XGBoost Algorithm," *Chinese Astron. Astrophys.*, vol. 43, no. 4, pp. 539–548, 2019, doi: 10.1016/j.chinastron.2019.11.005.

[49] C. Wang, C. Deng, and S. Wang, "Imbalance-XGBoost: leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost," *Pattern Recognit. Lett.*, vol. 136, pp. 190–197, 2020, doi: 10.1016/j.patrec.2020.05.035.

[50] N. Manju, B. S. Harish, and V. Prajwal, "Ensemble Feature Selection and Classification of Internet Traffic using XGBoost Classifier," *Int. J. Comput. Netw. Inf. Secur.*, vol. 11, no. 7, pp. 37–44, 2019, doi: 10.5815/ijcnis.2019.07.06.

[51] S. Thongsuwan, S. Jaiyen, A. Padcharoen, and P. Agarwal, "ConvXGB: A new deep learning model for classification problems based on CNN and XGBoost," *Nucl. Eng. Technol.*, vol. 53, no. 2, pp. 522–531, 2021, doi: 10.1016/j.net.2020.04.008.

[52] K. Song, F. Yan, T. Ding, L. Gao, and S. Lu, "A steel property optimization model based on the XGBoost algorithm and improved PSO," *Comput. Mater. Sci.*, vol. 174, no. December 2019, p. 109472, 2020, doi: 10.1016/j.commatsci.2019.109472.

[53] Y. Wang, "a Xgb Oost R Isk M Odel Via F Eature S Election and B Ayesian H Yper -P Arameter O Ptimization," vol. 11, no. 1, pp. 1–17, 2019.

[54] K. Budholiya, S. K. Shrivastava, and V. Sharma, "An optimized XGBoost based diagnostic system for effective prediction of heart disease," *J. King Saud Univ. - Comput. Inf. Sci.*, no. xxxx, 2020, doi: 10.1016/j.jksuci.2020.10.013.

[55] J. Guo *et al.*, "An XGBoost-based physical fitness evaluation model using advanced feature selection and Bayesian hyper-parameter optimization for wearable running monitoring," *Comput. Networks*, vol. 151, pp. 166–180, 2019, doi: 10.1016/j.comnet.2019.01.026.

[56] J. Zhou, Y. Qiu, S. Zhu, D. J. Armaghani, M. Khandelwal, and E. T. Mohamad, "Estimation of the TBM advance rate under hard rock conditions using XGBoost and Bayesian optimization," *Undergr. Sp.*, 2020, doi: 10.1016/j.undsp.2020.05.008.