Check for updates

Received: 5 September 2022 | Revised: 21 October 2022 | Accepted: 8 November 2022

DOI: 10.1002/eng2.12599

RESEARCH ARTICLE

Engineering Reports Open Access WILEY

A boosting ensemble learning based hybrid light gradient boosting machine and extreme gradient boosting model for predicting house prices

Racheal Sibindi[1] | **Ronald Waweru Mwangi**[2] | **Anthony Gichuhi Waititu**[3]

[1]Department of Mathematics, Pan African University Institute for Basic Sciences, Technology and Innovation, Nairobi, Kenya

[2]School of Computing and Information Technology, Department of Computing, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya

[3]Statistics and Actuarial Sciences Department, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya

Correspondence
Racheal Sibindi, Department of Mathematics, Pan African University Institute for Basic Sciences, Technology and Innovation, Nairobi, Kenya.
Email: sibindi.racheal@students.jkuat.ac.ke

Abstract

The implementation of tree-ensemble models has become increasingly essential in solving classification and prediction problems. Boosting ensemble techniques have been widely used as individual machine learning algorithms in predicting house prices. One of the techniques is LGBM algorithm that employs leaf wise growth strategy, reduces loss and improves accuracy during training which results in overfitting. However, XGBoost algorithm uses level wise growth strategy which takes time to compute resulting in higher computation time. Nevertheless, XGBoost has a regularization parameter, implements column sampling and weight reduction on new trees which combats overfitting. This study focuses on developing a hybrid LGBM and XGBoost model in order to prevent overfitting through minimizing variance whilst improving accuracy. Bayesian hyperparameter optimization technique is implemented on the base learners in order to find the best combination of hyperparameters. This resulted in reduced variance (overfitting) in the hybrid model since the regularization parameter values were optimized. The hybrid model is compared to LGBM, XGBoost, Adaboost and GBM algorithms to evaluate its performance in giving accurate house price predictions using MSE, MAE and MAPE evaluation metrics. The hybrid LGBM and XGBoost model outperformed the other models with MSE, MAE and MAPE of 0.193, 0.285, and 0.156 respectively.

KEYWORDS

boosting ensemble learning, light gradient boosting machine, extreme gradient boosting

1 | INTRODUCTION

Different machine learning (ML) algorithms have been developed to address various classification and prediction problems. These algorithms have been implemented as individual ML models with different performance accuracy. Researchers have been working on integrating ML algorithms in order to improve performance through minimizing

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2022 The Authors. *Engineering Reports* published by John Wiley & Sons Ltd.

error and increasing accuracy. The use of various algorithms for ML problems is promoting diversity through combining independent techniques for improving accuracy. Ensemble learning is one of the techniques that involves combining multiple algorithms for performance improvement. Reference 1 defined ensemble learning as a ML technique that combines decision from multiple models in order to improve accuracy and reduce errors. Ensemble learning consists of a set of individual models called base learners or weak learners whose output are combined for prediction and classification problems. Bagging and Boosting techniques are the commonly known types of ensemble learning.[2] Boosting learners learn in a sequential manner in order to improve the errors from the previous learner.

There has been an increase in the use of boosting techniques in prediction and classification problems due to their performance. Ensemble learning models such as extreme gradient boosting (XGBoost) and light gradient boosting machine (LGBM) techniques have been implemented quite often because of their efficiency. Reference 3 proposed XGBoost and LGBM in cloud system for predicting malware attacks by comparing the boosting methods to evaluate their efficiency. Both models had almost the same accuracy with the XGBoost performing better than the LGBM model. However, these techniques have been implemented as individual ML methods and they have different accuracy in performance. XGBoost splits trees level or depth wise which results in higher computation time whilst LGBM splits trees leaf wise with less computation time. LGBM is prone to overfitting during model implementation due to leaf wise splitting but it can reduce loss and improve accuracy. XGBoost has a regularization parameter and implements column sampling and weight reduction on new trees which combats overfitting. The study propose a hybrid LGBM and XGboost model that will improve performance by taking into account the different properties that the individual models possess.

Prediction is a very important aspect for businesses and policy makers as it provides information for decision making purposes. Developing models that have higher predicting accuracy is essential in dealing with uncertainty. Price prediction helps consumers to make better decisions on what and when to purchase. Reference 4 argued that investors, financial institutions and industries rely on an efficient and precise model for forecasting in order to determine price fluctuations for better decision making. The study propose to develop a hybrid LGBM and XGBoost model for boosting base learners in ensemble learning. The integration of these boosting models aims to enhance performance of individual ML techniques in prediction problems. This reduces variance that is combatting overfitting through a reduction in loss. The base learners are optimized by implementing Bayesian hyperparameter optimization method to acquire the best sets of hyperparameters which minimizises error. Adaptive boosting (Adaboost), gradient boosting machine (GBM), LGBM, and XGBoost are compared to the proposed hybrid LGBM and XGBoost model to evaluate its performance. The hybrid model leads to enhanced performance with reduced error as compared to the individual models. The study is divided into the following sections: Section 2, which reviews the literature, Section 3, which outlines the technique used, Section 4, which discusses the results and Section 5 provides conclusions.

2 | LITERATURE REVIEW

2.1 | Predictive models

Numerous predictive models have been adopted in predicting house prices. These models have been limited to classical individual models with different performance accuracy. One of the most proposed technique is support vector machine (SVM) that has proved to attain better performance compared to other techniques. According to Reference 5 SVM model is mostly used because it is a powerful tool in improving accuracy through accurate predictions and works well with various kinds of data. Reference 5 implemented SVM, random forest (RF) and GBM algorithms in predicting house prices of property. The study showed that the algorithms were good techniques for researchers in predicting house prices. However, RF and GBM had better results despite their limitations which need to always be taken into account. Reference 6 suggested that if the main objective of the study is speed then SVM should be implemented to make forecasts. When predictive performance accuracy is the main goal then decision tree models should be taken into consideration. In many fields, including real estate price prediction, machine learning algorithms are now widely available as techniques that can be utilized to carry out prediction and classification tasks.[7]

Multiple linear regression techniques have also been implemented in house price prediction problems. They assess the relationship that exists between the dependent variable and independent variables. Reference 8 proposed 5 predictive models in predicting house prices which were ordinary least squares, lasso, ridge, SVM and XGBoost techniques.

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

The models were evaluated in order to compare their performance in predicting accurate results. The advanced algorithms that is SVM and XGBoost had better performance accuracy results with XGBoost outperforming the other models. Predicting house prices is a new field of study that calls for an understanding of various machine learning algorithms.[9] Reference 10 proposed house price prediction model using linear regression, RF, boosted regression and neural networks. The predictions from the models were integrated by the weighted mean resulting in accurate predictions compared to the individual model predictions. Classical methods are being combined with advanced ML algorithms (boosting algorithms) and deep learning methods which improves performance of the overall model.

There are some nonlinear relationships that exists between house price and factors influencing the outcome which traditional models might not be able to capture resulting in poor performance. Reference 11 argued that the factors influencing home prices are intricate and interconnected processes that typical statistics methods cannot address. Reference 12 proposed a hybrid model for forecasting prices of real estate. The hybrid model was a bagging ensemble learning model of SVM, classification and regression tree, linear regression and artificial neural network. Hybrid model outperformed the individual models having better forecasting results. Reference 11 proposed an autoregressive moving average using an Adam optimizer technique to train the model. The model was compared to SVM technique and the results showed that the proposed approach had better results than the SVM model in predicting house prices. The availability of numerous prediction models enables researchers to select the most appropriate one for house price prediction considering their history, requirements, and level of ML comprehension.

2.2 | Boosting ensemble learning

Ensemble learning is concerned with employing multiple base learners and combining their predictions.[13] The multiple learners can have different training sets and feature sets whose results are combined in order to enhance the overall performance accuracy of the prediction or classification. Reference 14 defined a base learner as a technique that takes into account a set of inputs as labeled examples to produce a model that can generalize those examples. The main idea behind ensemble learning is that through a combination of many models, one base learner's errors is likely to be compensated by the other base learners. Boosting ensemble learning deals with improving the performance of base learners to be stronger learners.[15] It is based on aggregating a group of models by averaging estimations or voting. Each model is an adaptive version of the previous one, giving incorrectly classified samples more weight in the next estimation.[16] There are 5 boosting techniques which have evolved throughout the years with the goal of improving their performance which are Adaboost, GBM, LGBM, XGBoost and catboost for categorical features.

Freund and Schapire introduced Adaboost technique in 1996 which has gained much recognition in data mining and ML for the past two decades.[17] The base learner uses the training set for training whereas the sample distribution for training is adjusted according to the base learner's performance. Reference 18 stated that the incorrect predictions of the samples by the previous base learner receives more attention during the follow up training process of the model. Adaboost is prone to overfitting and underfitting therefore when given new data set the algorithm will perform poorly.[19] Reference 20 proposed Adaboost-support vector regression model in predicting minimum miscibility pressure in carbon dioxide streams. The model outperformed the existing models by root mean square error of 0.9 and average absolute percentage relative error of 3.09%. The combination based on Adaboost and SVM algorithms in both classification and regression problems has shown greater performance accuracy. The GBM algorithm is an optimization technique which seeks for an additive model that deals with the minimization of the loss function.[21] Therefore GBM adds a new decision tree or weak learner iteratively which decreases the loss function at every step.

Reference 14 argued that the goal is to build base learners in order for them to be correlated maximally with the loss function negative gradient which is in conjunction with the whole ensemble. When choosing the maximum number of trees in GBM, if you set them too high it results in overfitting and too low may lead to underfitting. According to Reference 14, the use of stochastic gradient boosting method in which trees are successively trained using small subsets taken from original dataset can reduce overfitting. Reference 22 presented a protein solubility predictor approach that implemented GBM technique. The main aim of the study was to approximate the sequence and feature structures of the protein then use GBM to predict the output. The proposed model outperformed the other ML algorithms by 9% more accuracy. The GBM model was better compared to SVM model since it captured the relationship between the outcome variable and the features which were nonlinear.

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Reference 23 defined LGBM as a fast gradient boosting technique implemented on decision trees. The technique is considered fast because it filters out a huge number of data instances with small gradients, leaving the ones with large gradients to compute information gain as well as bundling features that are mutually exclusive to reduce features dimension. LGBM adopts the tree leaf wise strategy which has a maximum depth limit and the histogram based technique in order to reduce memory consumption and increase training speed.[24] The tree leaf wise strategy splits on the same layer the leaf with the highest information gain which results in its efficiency. This technique may result in trees with high depth which causes overfitting and during tree growth a maximum depth limiting is applied.[24,25] LGBM implements four strategies which are gradient based one side sampling, histogram based algorithm, leaf wise growth strategy and exclusive feature bundling. According to Reference 26 these strategies makes the LGBM algorithm to be more superior in terms of computation efficiency and improving accuracy. Figure 1 shows the struture of the leaf wise strategy.

Reference 27 proposed LGBM and gated recurrent unit in predicting weekdays traffic congestion. The study's objective was to create a model that can improve the ability of expression of features which are limited by generating them using the proposed model. The model had better performance and improved accuracy compared to other algorithms. Reference 28 implemented LGBM model with Bayesian optimization in predicting free calcium oxide content in cement clinker using time series data. The Bayesian optimization technique was used to perform hyperparameter search and tuning in order to obtain better performance accuracy results. Hyperparameter optimization methods are very useful when dealing with algorithms that require extensive hyperparameter tuning.

XGBoost is a more advanced version of GBM which is a decision tree boosting algorithm that implements regularization to combat overfitting effectively.[29] The loss function falls along the direction of its gradient during the process of iteration which improves the robustness of the algorithm. XGBoost adopts the level or depth wise growth strategy which makes it computationally expensive. The XGBoost algorithm consists of an objective function that seeks to minimize the loss function and a regularization parameter that prevents overfitting.[24,30] XGBoost has two features that prevents overfitting in addition to the regularization parameter. Reference 31 argued that each new tree's weights can be reduced by a constant which reduces the impact on the final score of a single tree and leaves space for the next new trees for model improvement. Figure 2 shows the level-wise strategy.

Trees are built from the training dataset using column wise sampling commonly known as column sampling. XGboost purpose is in the simplification of the model provided by the algorithm and preventing overfitting.[32] Reference 33 proposed an XGBoost algorithm for predicting diabetes through a classification model. The model showed to have good stability, high running speed and high accuracy in predicting diabetes with an accuracy score of 80.2%.[34] developed a hybrid XGBoost and LGBM model where he built a meta learner for electrical load forecasting. The proposed model reduced error as compared to the other ML algorithms implemented. Reference 35 studied XGBoost and LGBM algorithms to optimize the hyperparameters in predicting product marketing. The goal was to examine the effect of hyperparameters on model development. The results showed that even though some of the hyperparameters are the

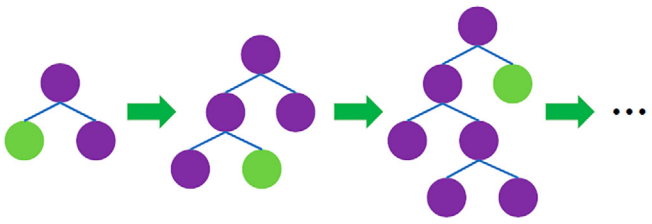


FIGURE 1 Leaf-wise tree growth strategy for LGBM algorithm

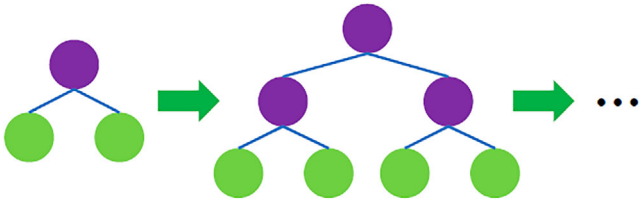


FIGURE 2 Level-wise tree growth strategy for XGBoost algorithm

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

same for both models such as learning rate, they do not pose the same effect on the models. In this paper, a super learner is built from out of fold predictions from the boosting algorithms to reduce error whilst improving performance. Bayesian hyperparameter optimization method is implemented which resulted in a powerful hybrid model being developed.

3 | MATERIALS AND METHODS

3.1 | Baseline methods

The study implemented Adaboost, GBM, LGBM, and XGBoost boosting models as baseline methods to validate the effectiveness of the proposed hybrid LGBM and XGBoost model. The following boosting techniques are discussed in this section.

3.1.1 | Adaptive boosting

Adaboost algorithm takes training set $S = \{(x_i, y_i)\}_{i=1}^{n}$ such that $x_i = X, y_i \in Y = \{-1, +1\}$. A weak or base learner is called repeatedly in iteration $t = 1, \cdots, T$ and a weak hypothesis $h_t : X \to \{-1, +1\}$ is found with weight distribution D_t on training examples. The error e_t measures the goodness of a weak hypothesis as

$$e_t = Pr_{i\sim D_t}\left[h_t(x_i) \neq y_i\right] = \sum_{i:h_t(x_i)\neq y_i} D_t(i). \tag{1}$$

Parameter λ_t is chosen where a weak learner h_t is received given by

$$\lambda_t = \frac{1}{2}\ln\left(\frac{1-e_t}{e_t}\right) \tag{2}$$

which measures the importance assigned to h_t. The distribution D_t is updated as

$$D_{t+1}(i) = \frac{D_t(i)\exp\left(-\lambda_t y_i h_t(x_i)\right)}{Z_t}, \tag{3}$$

where Z_t is the normalization factor to make D_{t+1} a distribution. This increases the weights of incorrectly predicted examples and decreases weight for correctly predicted examples. The final hypothesis H which is the voted by weighted majority of the T weak hypothesis, with h_t assigned λ_t as the weight is given by

$$\mathrm{H}(x) = sign\left(\sum_{t=1}^{T}\lambda_t h_t(x)\right). \tag{4}$$

Adaboost technique focuses more on incorrect predictions by assigning more weights in order to give them more focus in the next iteration.

3.1.2 | Gradient boosting machine

GBM algorithm seeks to combine weak learners to become a stronger learner in an iterative manner. Given a dataset $S = \{(x_i, y_i)\}_{i=1}^{n}$ with n observations, the aim is to find an estimation $\hat{f}(x)$ of the function $f^*(x)$ which maps x inputs to target values y. This is achieved through minimizing the loss function expected value $L(y, f(x))$. GBM constructs additive estimation of $f^*(x)$ as weighted sum of functions

$$f_t(x) = f_{t-1}(x) + \rho_t h_t(x), \tag{5}$$

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

where ρ_t is the weight of t^{th} base learner for t = $1, \cdots, T$. A constant estimation of $f^*(x)$ is obtained through iterative construction as

$$f_0(x) = \arg\min_{\alpha} \sum_{i=1}^{n} L(y_i, \alpha), \tag{6}$$

where $L(y_i, \alpha)$ is a differentiable loss function. The base learners seek to minimize

$$(\rho_t, h_t(x)) = \arg\min_{\rho,h} \sum_{i=1}^{n} L(y_i, f_{t-1}(x_i) + \rho h(x_i)). \tag{7}$$

Each base learner h_t in optimization of gradient descent for f^* is a greedy step. Therefore, a new dataset $S = \{(x_i, r_{ti})\}_{i=1}^{n}$ is trained on each model where psuedo residuals r_{ti} are computed as

$$r_{ti} = \left[\frac{\partial L(y_i, f(x))}{\partial f(x)}\right]_{f(x)=f_{t-1}(x)}. \tag{8}$$

The weight ρ_t is calculated by solving the line search optimization problem. GBM is more concerned with improving accuracy of the model through reduction of error or residuals.

3.1.3 | Light gradient boosting machine algorithm

LGBM algorithm is based on decision trees therefore the formulation of the model is as follows. Given a training data set $S = \{(x_i, y_i); i = 1, 2, \cdots, n; x_i \in R^m, y_i \in R\}$ where n is the samples with m features. To find the estimation, the decision trees predictions are combined as follows;

$$\hat{y}_i^{LG} = \sum_{p=1}^{P} f_p(x_i), \tag{9}$$

where the number of trees is p with f_p as the trees. The goal is to minimize the objective function below to obtain f_p.

$$f_p = \arg\min_{f_p} \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{LG(p)}\right) + \Omega(f_p). \tag{10}$$

The loss function is L and the regularization parameter Ω which is given by

$$\Omega(f_p) = \alpha \mathrm{T} + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2, \tag{11}$$

where α and λ are the penalty parameters for T leaves and weight of leaves w. Taking L as a loss function which is a squared error, then

$$\begin{aligned} L\left(y_i, \hat{y}_i^{LG(p-1)} + f_p(x)\right) &= \left(y_i - \hat{y}_i^{LG(p-1)} - f_p(x)\right)^2 \\ &= \left(r - f_p(x)\right)^2 \end{aligned} \tag{12}$$

the residual r is fitted to obtain f_p. The function for minimizing the objective function at iteration p is defined using a quadratic approximation as

$$\begin{aligned} f_p &\simeq \arg\min_{f_p} \sum_{i=1}^{n} \left[g_i f_p(x_i) + \frac{1}{2} h_i f_p^2(x_i)\right] + \Omega(f_p), \\ g_i &= \partial_{\hat{y}^{LG}(p-1)} L\left(y_i, \hat{y}_i^{LG(p-1)}\right), \\ h_i &= \partial^2_{\hat{y}^{LG}(p-1)} L\left(y_i, \hat{y}_i^{LG(p-1)}\right). \end{aligned} \tag{13}$$

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Through minimizing the objective function, a new tree f_p is obtained. Each node with the biggest information gain is divided by the decision tree. The variance gain for a node that separates feature j at point s is given by

$$Z_{j|O}(s) = \frac{1}{n_O}\left\{\frac{\left(\sum_{\{x_i\in O: x_{ij}\le s\}} g_i\right)^2}{n^j_{l|O}(s)} + \frac{\left(\sum_{\{x_i\in O: x_{ij}> s\}} g_i\right)^2}{n^j_{r|O}(s)}\right\}, \tag{14}$$

O is samples on the decision tree fixed node. $n_O = \sum I\left[x_i \in O\right]$,
$n^j_{l|O}(s) = \sum I\left[x_i \in O : x_{ij} \le s\right]$ and $n^j_{r|O}(s) = \sum I\left[x_i \in O : x_{ij} > s\right]$. The decision tree selects $s^*_j = \arg\max_s Z_j(s)$ for each feature j and computes the highest gain $Z_j\left(s^*_j\right)$. The data will be split into the right and left nodes according to feature j^* at point s_{j^*}. All samples are scanned to find the optimal splitting point in order to calculate the information gain.

3.1.4 | Extreme gradient boosting algorithm

The XGBoost algorithm is implemented as follows. Given a dataset
$S = \{(x_i, y_i);\ i = 1, 2, \cdots, n;\ x_i \in R^m,\ y_i \in R\}$ such that number of observations is n, the number of features is m and the target variable y. Denote $\hat{y}_i^{XG}$ as the outcome from the model given by

$$\hat{y}_i^{XG} = \phi(x_i) = \sum_{p=1}^{P} f_p(x_i), \tag{15}$$

f_p is decision tree, $f_p(x_i)$ is the score for the p^{th} tree for the i^{th} observation. In order to choose the function f_p, the regularized objective function should be minimized which is given as

$$f_p = \sum_i L\left(y_i, \hat{y}_i^{XG}\right) + \sum_p \Omega\left(f_p\right) \tag{16}$$

with the loss function L and a penalty parameter Ω added in order to reduce model's complexity.

$$\Omega\left(f_p\right) = \alpha \mathrm{T} + \frac{1}{2}\lambda\|w\|^2, \tag{17}$$

where α and λ are the parameters that control penalty for leaves T and the leaf weight w respectively. $\Omega\left(f_p\right)$ prevents overfitting and also simplifies the models that are produced by the algorithm. To minimize the objective function, an iterative method is used that is Newton minimization method for the jth iteration, f_j is added to minimize the objective function

$$f_j = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{XG(j-1)} + f_j(x_i)\right) + \Omega\left(f_j\right). \tag{18}$$

The function is simplified using Taylor's expansion and the objective function will be as follows;

$$\begin{aligned} f_j &= \sum_i \left[L\left(y_i, \hat{y}_i^{XG(j-1)} + g_i f_j(x_i) + \frac{1}{2}h_i f_j^2(x_i)\right)\right] + \Omega\left(f_j\right) \\ g_i &= \partial_{\hat{y}XG(j-1)} L\left(y_i, \hat{y}_i^{XG(j-1)}\right) \\ h_i &= \partial^2_{\hat{y}XG(j-1)} L\left(y_i, \hat{y}_i^{XG(j-1)}\right). \end{aligned} \tag{19}$$

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

The loss reduction is derived by the formula after the tree split from a give node as

$$f_{split} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_l} g_i\right)^2}{\sum_{i\in I_l} h_i + \lambda} + \frac{\left(\sum_{i\in I_r} g_i\right)^2}{\sum_{i\in I_r} h_i + \lambda} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i + \lambda}\right] - \alpha, \tag{20}$$

I is current node's subset of observations, I_r and I_l are the subsets of observations available in the right and left nodes after splitting. The formula gives the optimal split at each node.

3.2 | Proposed hybrid LGBM and XGBoost model

The hybrid model was developed by building a super learner ensemble model through combining the individual models in a parallel manner. Given dataset D such that $D_i = \{(x_i, y_i); i = 1, \ldots, n\}$ and LGBM and XGBoost algorithms as base learners. The base learners are fitted on the dataset $D = \{D_i; i = 1, \ldots, n\}$ to estimate

$$\begin{aligned}\hat{Y}_{LG} &= \sum_{p=1}^{P} f_p(X)\\ \hat{Y}_{XG} &= \sum_{p=1}^{P} f_p(X).\end{aligned} \tag{21}$$

Training and testing sets are created for the dataset D, and K-fold cross validation is implemented on the training set with $k-1$ training sample and the remaining as validation sample. Define $T(k)$ and $V(k)$ to be the training and validation samples respectively.

$$T(k) = D \backslash V(k) \quad k = 1, \cdots, K. \tag{22}$$

For the k^{th} fold, fit the base learners on $T(k)$ and calculate the predictions of the corresponding validation sample given as

$$\hat{Y}_{LG,T(k)}(X_i), \quad \hat{Y}_{XG,T(k)}(X_i); \; D_i \in V(k) \quad \text{for} \quad k = 1, \ldots, K. \tag{23}$$

The out of fold predictions from each base learner are combined to have $n \times 2$ matrix meta data for the meta model.

$$Z = \left\{\hat{Y}_{LG,T(k)}\left(X_{V(k)}\right), \hat{Y}_{XG,T(k)}\left(X_{V(k)}\right)\right\}, \tag{24}$$

$X_{V(k)} = \{X_i : D_i \in V(k)\}$ for the covariate vectors of the validation sample. The combinations of the base learners are weighted by a weight vector

$$\begin{aligned} m(z|\alpha) &= \alpha_{LG}\hat{Y}_{LG,T(k)}\left(X_{V(k)}\right) + \alpha_{XG}\hat{Y}_{XG,T(k)}\left(X_{V(k)}\right)\\ &\alpha_{LG}, \alpha_{XG} > 0 \quad \text{and} \quad \alpha_{LG} + \alpha_{XG} = 1.\end{aligned} \tag{25}$$

The optimum value of α is selected from all α combinations that reduces risk on cross validation for base learners.

$$\hat{\alpha} = \arg\min_{\alpha} \sum_{i=1}^{n} (Y_i - m(z_i|\alpha))^2. \tag{26}$$

The optimal $\hat{\alpha}$ weight vector obtained from 26 is combined to construct the hybrid model.

$$\hat{Y}_{hybrid}(X) = \hat{\alpha}_{LG}\hat{Y}_{LG}(X) + \hat{\alpha}_{XG}\hat{Y}_{XG}(X). \tag{27}$$

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

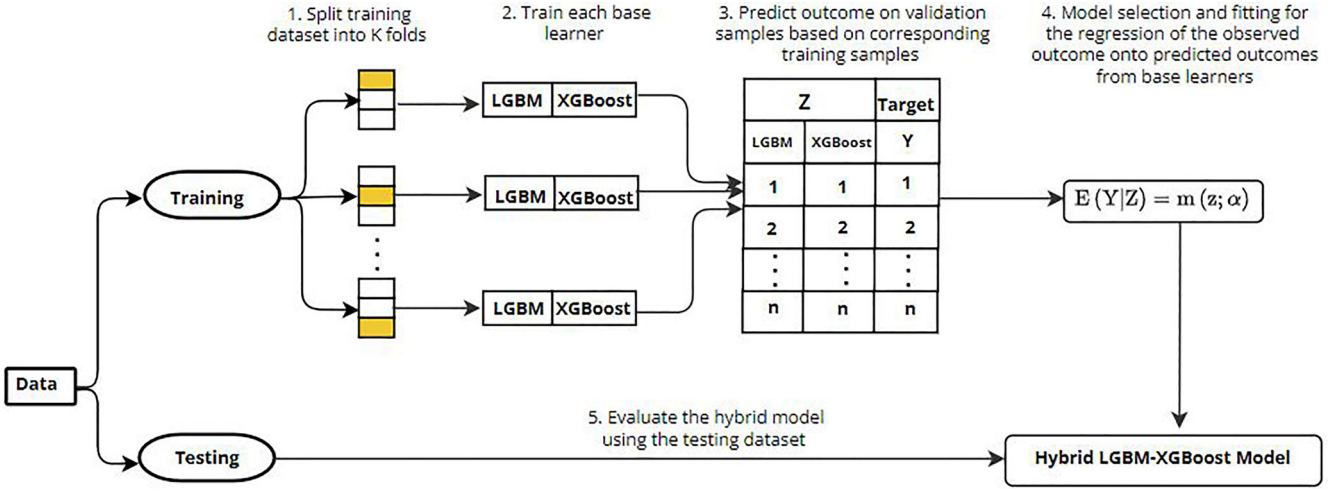


FIGURE 3 Flowchart which shows the implementation process of the proposed hybrid LGBM and XGBoost model

The parameter space for α is restricted to be a convex combination of the base learners which ensures that the final model predictions has greater stability. The restriction implies that if each base learner is bounded then the convex combination will be bounded as well. Figure 3 shows the flowchart for the hybrid model design.

3.3 | Data description and tools

Tabular data which is a subdivision of heterogeneous data with rows and columns has been efficiently implemented in tree-ensemble models. XGBoost has shown greater efficiency and accuracy using tabular datasets with different properties from domains such as e-commerce and finance. The well-known tabular dataset, California Housing Dataset[36] collected on block groups from 1990 census was used to develop the hybrid LGBM and XGboost model to predict house prices. It consists of 20,640 observations of 9 features including the target variable median house value illustrated in Figure 4.

Bayesian hyperparameter optimization technique was implemented to the base learners in order to find the best set of hyperparameters which minimized the objective function, mean squared error. With its comprehensive ability to configure the fit, trial and error computation process of optimizing hyperparameters, Bayesian technique is a hyperparameter tuning optimization process developed in 2019.[37] The search process finds the best set or combination of hyperparameters that optimizes the objective function. For better understanding of the developed hybrid model, Shapley Additive Explanation (Shap) values were used. Shap values were used in economics field in cooperative game theory in order to explain each player's contribution that is in predicting what each player was bringing to the game.[38] This method computes how a variable contributes to the prediction and finds the average of the contributions. Therefore it assigns an importance value to each variable for the predictions to calculate the explanation. Python 3.9.12 programming software was used as a tool for implementing the proposed algorithm. Jupyter notebook which is an open source web application was used to implement the algorithms in python. Scikit-learn, optuna, Shap, lightgbm, and xgboost ML modules were used for hybrid model development. Matplotlib and optuna visualization modules were used in order to better understand the results from the implementation. The computer used was HP core i5 with 16gb random access memory and processing speed of 2.40ghz.

3.4 | Performance evaluation metrics

The error of estimation metrics were utilized for evaluating the perfomance accuracy of the proposed model. The metrics used were mean square error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). The equations for the evaluation metrics are as follows;

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

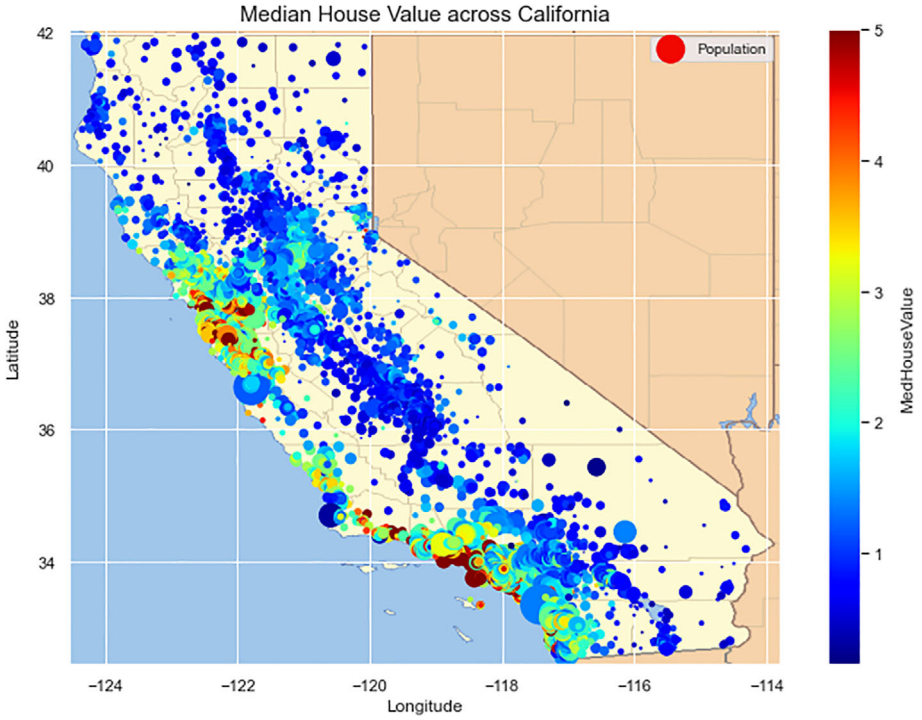


FIGURE 4 Median House Value (in 100000USD) across California. The distribution of the target variable according to the population based on their location on the California map

$$MSE = \frac{\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{n}, \tag{28}$$

$$MAE = \frac{\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right|}{n}, \tag{29}$$

$$MAPE = \frac{\sum_{i=1}^{n}\frac{\left|Y_i - \hat{Y}_i\right|}{Y_i}}{n} \times 100\% \tag{30}$$

with n being observations, Y_i as actual values and $\hat{Y}_i$ as forecasted values.

4 | RESULTS AND DISCUSSION

The objective function for the base learners was to minimize squared error whilst improving accuracy. It was optimized over 200 trials which gave the best combination of hyperparameters with minimum error. The hyperparameters were optimized over different sets of trials between 100 and 500. The best trial that gave the minimum MSE was 200 which had the best set of hyperparameters for the base learners. The experiment for the best trial is given in Figure 5.

4.1 | Hyperparameter optimization history

LGBM model progress over the 200 trials in optimizing the objective function is shown in Figure 6. The best value line shows the models that achieved the minimum objective value over the trials. The set of hyperparameters which minimized the objective value after 200 trials was achieved on trial 185. After the best trial, there was no improvement in the objective value therefore this model achieved the best results. The XGBoost model best objective value was attained on trial 187

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

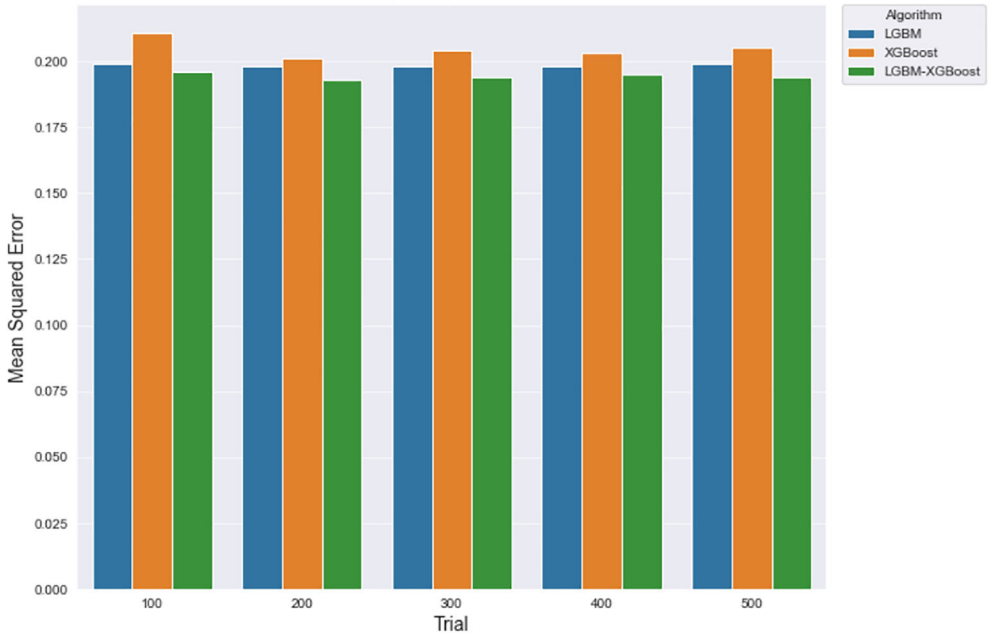


FIGURE 5 Hyperparameter optimization trials which gave the least MSE from the combination of hyperparameters for the base learners and the hybrid model

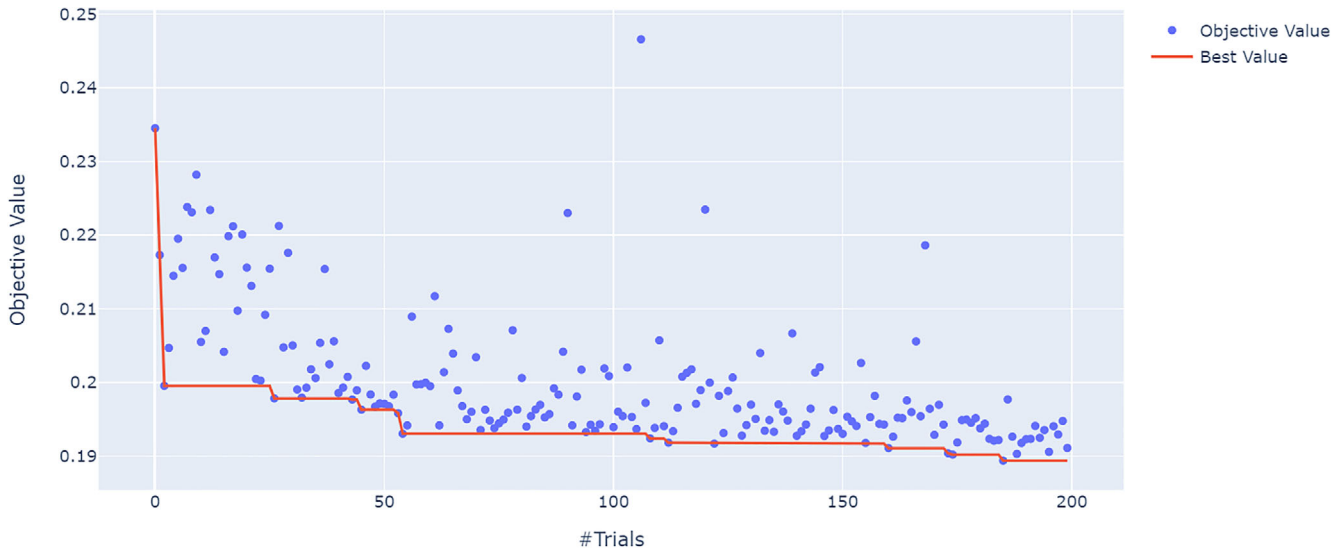


FIGURE 6 Optimization History for the LGBM Model over the 200 trials in minimizing the objective function

as illustrated in Figure 7. The model gave the best set of hyperparameters to achieve minimum error thereby increasing accuracy. LGBM model attained the best objective value of 0.189 as compared to 0.196 for XGBoost model.

4.2 | Hyperparameter importance

The set of hyperparameters which minimized the objective function had different impact on the performance of the models. Maximum depth of the tree for LGBM model had the highest impact on minimizing error contributing 38% to the learning process followed by learning rate with 24% contribution as shown in Figure 8. The other hyperparameters had the lowest impact contributing less than 16% in improving performance of the model. Gamma which is the minimum split

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

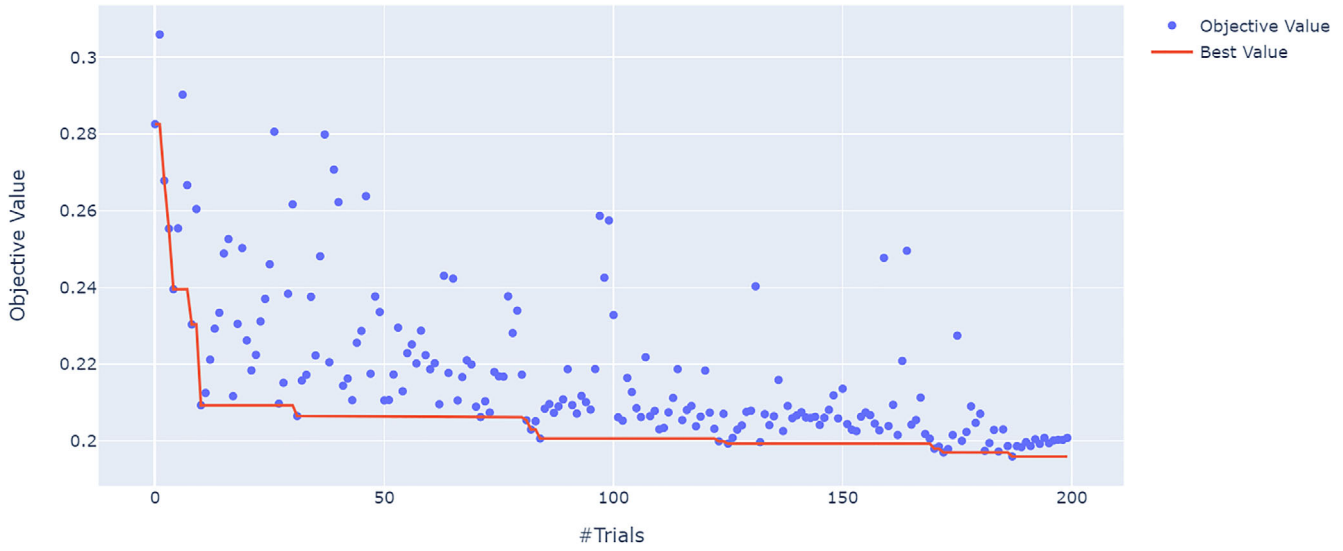


FIGURE 7 Optimization History for the XGboost Model over the 200 trials in minimizing the objective function

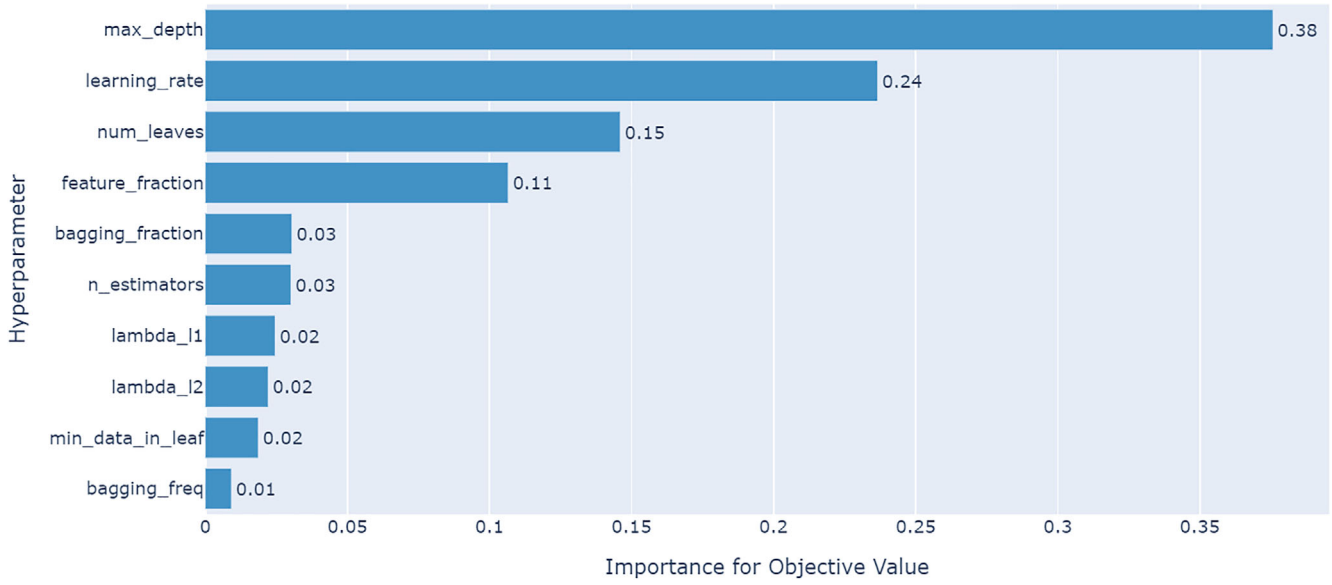


FIGURE 8 Hyperparameter importance in minimizing error for LGBM model

loss for the nodes had the highest impact in optimizing the objective function for XGBoost model with a contribution of 73%. The other hyperparameters contributed below 9% having the lowest impact on the model's performance as illustrated in Figure 9.

The base learners use different hyperparameters with some which are the same since they are decision tree algorithms such as learning rate, maximum depth, number of estimators, α (L1 regularization), λ (L2 regularization), subsample (bagging fraction) and column sample by tree (feature fraction). The low impact that some of the hyperparameters had on the model does not mean that they did not have an impact on improving performance. The small contribution was necessary in optimizing the objective function with the hyperparameters that had higher impact.

4.3 | Hyperparameter parallel coordinate

Figure 10 shows the association between different hyperparameter values in optimizing the objective function value for the LGBM model. Low objective values were associated with high bagging fraction, high bagging

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

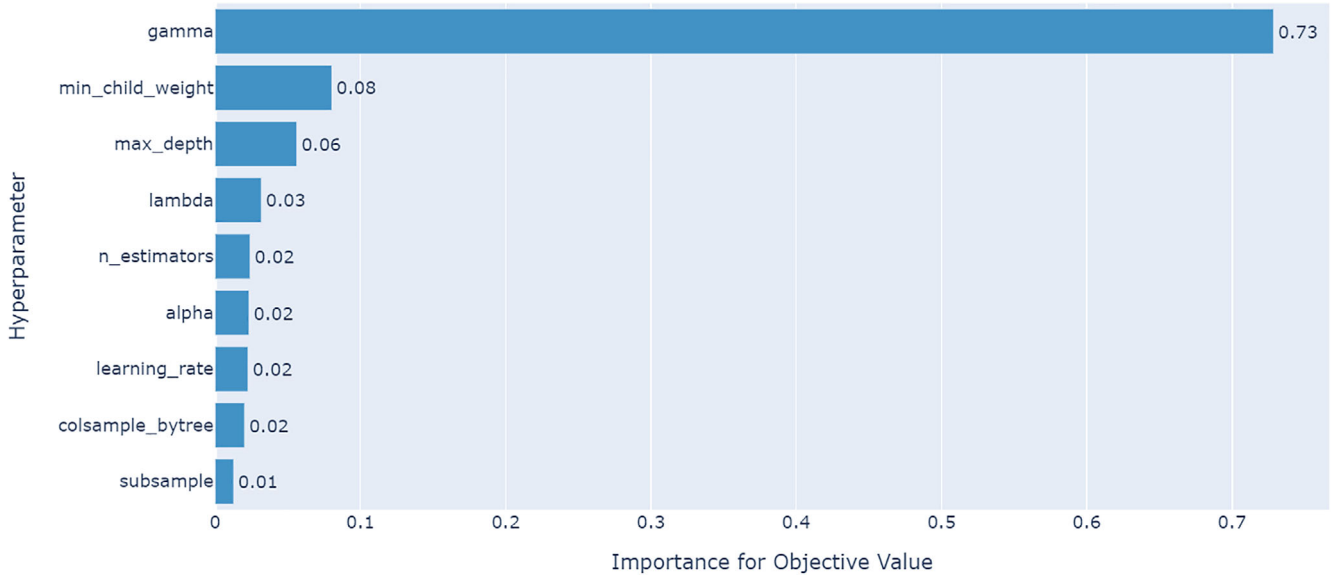


FIGURE 9 Hyperparameter importance in minimizing error for XGBoost model

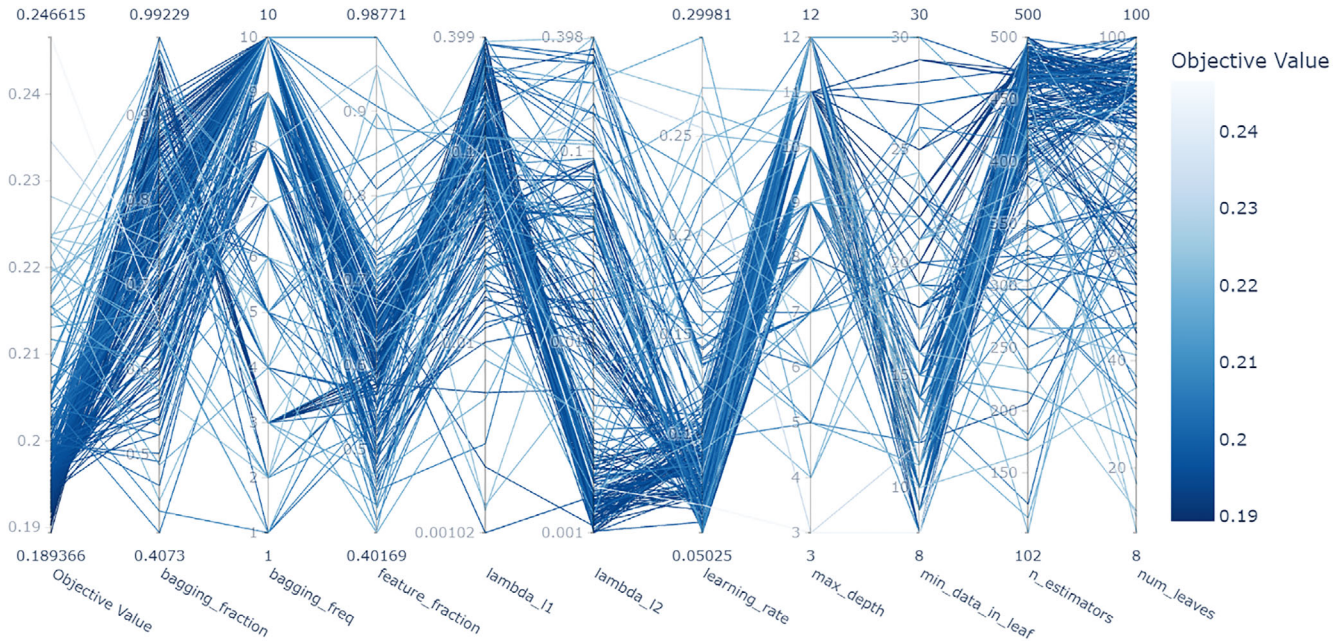


FIGURE 10 Parallel coordinate for the different hyperparameter values and the objective function values for LGBM model

frequency, low feature fraction, high lambda, low alpha, low learning rate, high maximum depth, low minimum child sample, low minimum data in leaf, high number of estimators and high number of leaves values for different trials.

Figure 11 shows XGBoost model low objective values were associated with high alpha, medium column sample by tree, low gammma, high lambda, high learning rates, high maximum depth, low minimun child weight, high number of estimators and low subsamples. Both base learners had low objective value being associated with high maximum depth and high number of estimators in improving performance.

Through the experiment of finding the best combination of hyperparameters for the base learners, a single hyperparameter value is searched at the same time testing it on different values of another hyperparameter.

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

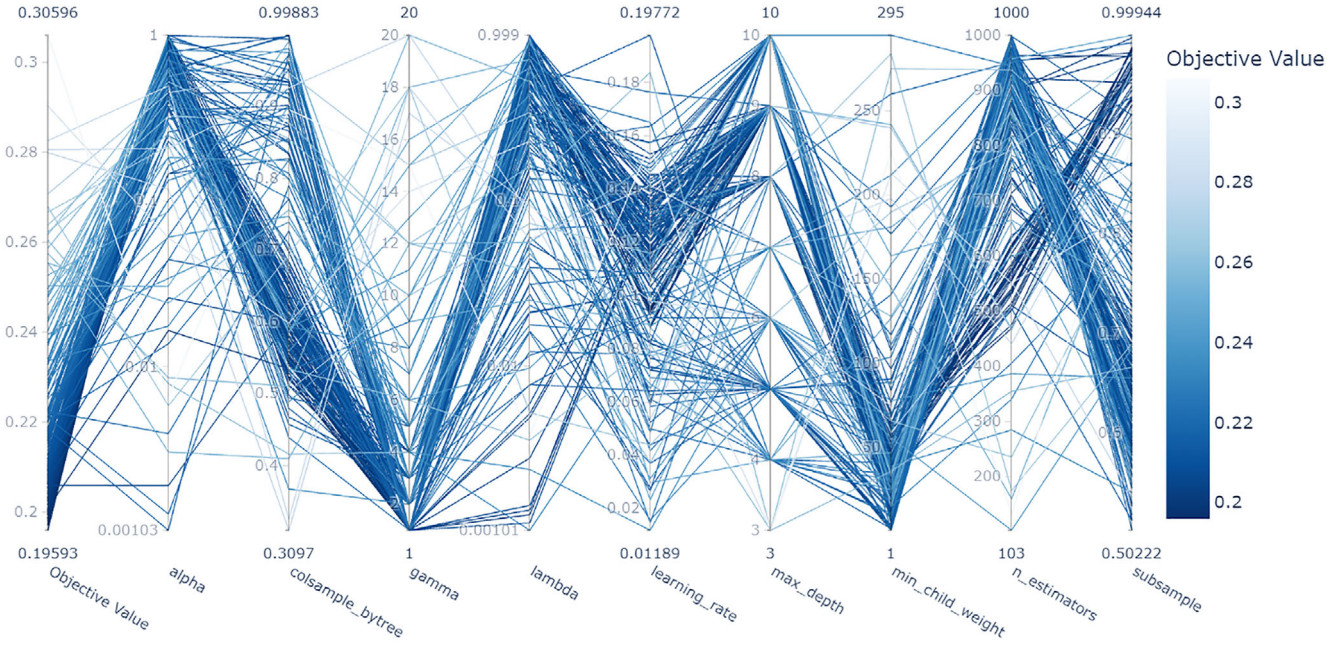


FIGURE 11 Parallel coordinate for the different hyperparameter values and the objective function values for XGBoost model

TABLE 1 Optimal hyperparameter values for the base learners.

Hyperparameter	LGBM	XGBoost
Number of Estimators	487	555
Learning rate	0.0876	0.1069
Maximum tree depth	11	9
Subsample/Bagging fraction	0.8787	0.9849
Column sample by tree/Feature fraction	0.6054	0.5463
α (L1 Reularization)	0.3968	0.9115
λ (L2 Reularization)	0.0013	0.8607

4.3.1 | Summary of base learners hyperparameters

Table 1 summarizes the best set of the same hyperparameters that achieved the minimum error through optimizing the objective function for the base learners. LGBM model hyperparameter values for number of leaves, bagging frequency and minimum data in leaf were 93, 10, and 19 respectively. XGBoost model hyperparameter values for gamma and minimun child weight were 1 and 40 respectively.

4.4 | Hybrid LGBM and XGBoost model interpretation

Shap method was used to compute the Shap values for the contribution of each base learners to the hybrid model's output. Figure 12 illustrates the importance of the features in descending order and also their distribution and influence in predicting house prices. Low median income values resulted in a low house price being predicted whilst high median income resulted in predicting high house price. Households that earn a higher income had a higher predicted house price because the more you earn the more you will find a house that is within your income. Households that earn a lower income had a lower predicted house price as shown in Figure 12. Longitude, average occupants and population had the majority of samples with a shap value of zero which resulted in less to no impact to the model's output.

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

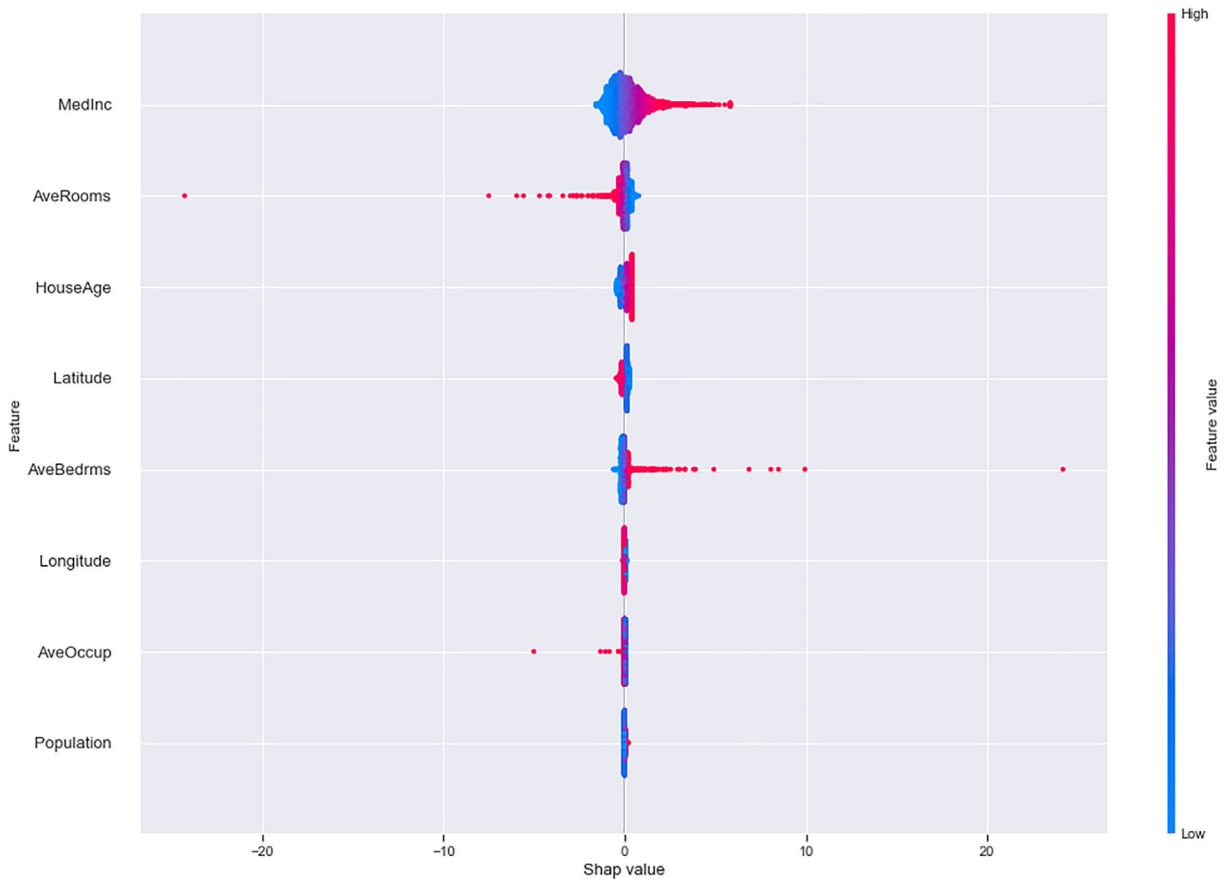


FIGURE 12 Hybrid Model Feature Importance Summary. The distribution and influence of the features in predicting house prices

Figure 13 shows the contribution from each base learner to the overall output of the hybrid model based on how accurate the predictions were compared to the target values. The hybrid model had an almost equal contribution from both the base learners as shown by the length of the bars. The percentage contribution for LGBM and XGBoost models to the hybrid model was 49.8% and 50.2% respectively. There is a 0.2% difference between the base learners contribution. Therefore, both models had almost the same influence or impact to the hybrid model's output which resulted in improved performance accuracy.

Low predicted values from both base learners had a low impact on the hybrid model's predicted house price therefore low values resulted in lower house prices being predicted as shown in Figure 14. Majority of the samples in both learners had lower values with very few with moderate values and higher values. As the value increased drastically it resulted in a high impact in the model's output that is a higher house price being predicted. Both base learners were positively skewed as the distribution of the samples were shifted to the left with negative Shap values. Equal distribution of samples from both models as they contributed almost the same amount to the model's output.

Figure 15 shows how the base learners contributed to pushing the hybrid model first prediction from the base value (the average output from the training set) to the targeted model output value. Both base learners pushed the prediction higher to 4.88 from a base value of 2.11 close to its targeted output with both models contributing equal amount to this prediction as shown by the length of the base learners.

4.5 | Hybrid model summary

The hybrid model showed a greater performance in optimizing the objective function that is minimizing error whilst improving performance. Table 2 shows the MSE for the hybrid model before and after implementing Bayesian hyperparameter optimization to the base learners. The hybrid model's MSE decreased from 0.214 using the default hyperparameters to 0.193 with the best combination of hyperparameters.

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

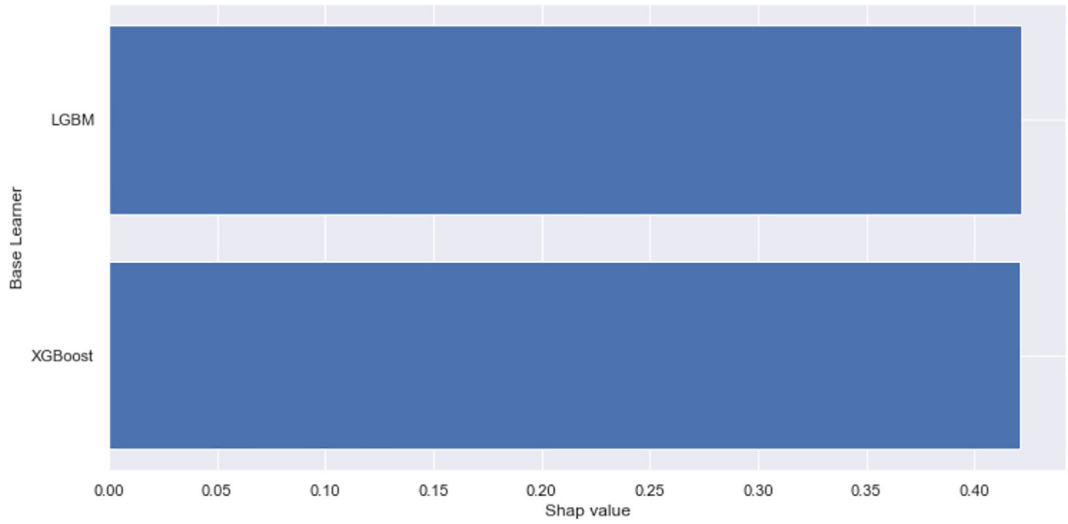


FIGURE 13 Contribution of each base learner to the overall output of the hybrid model in house price prediction

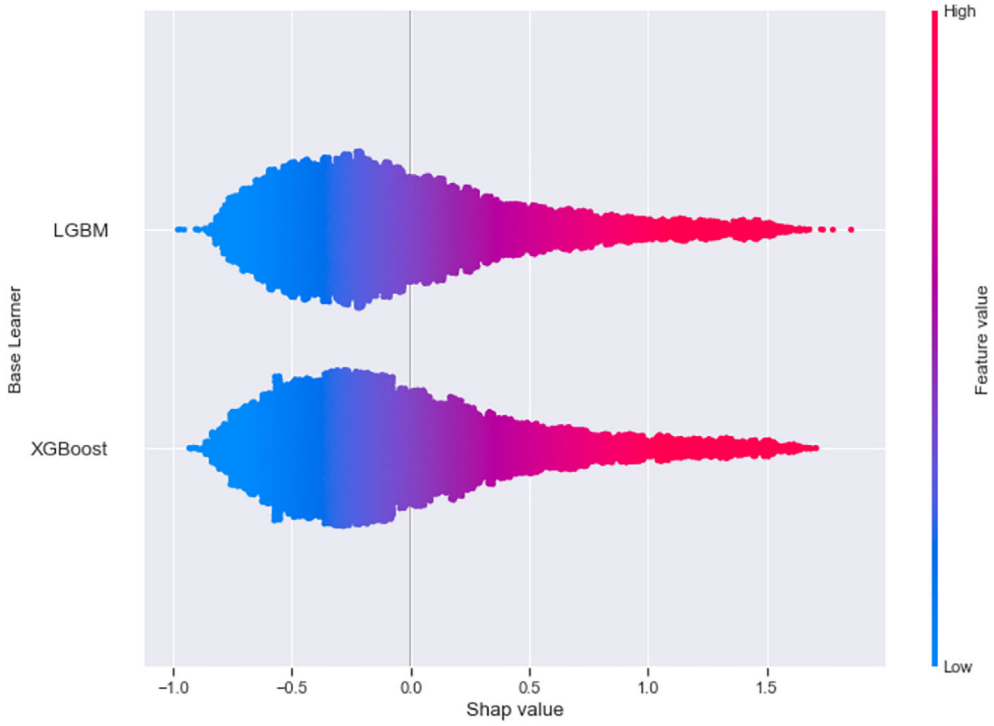


FIGURE 14 The distribution and influence of each base learners to the hybrid model's output

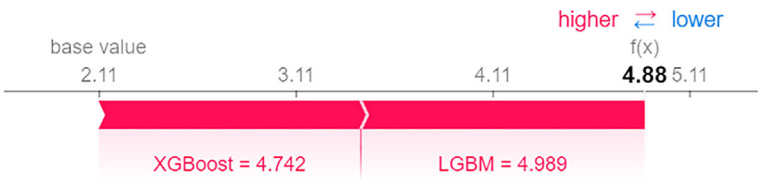


FIGURE 15 Hybrid model individual prediction explanation. The contribution of each base learner to the individual prediction

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 2 Performance evaluation of the hybrid LGBM and XGBoost based on MSE.

Model	MSE
LGBM-XGBoost without Bayesian	0.214
LGBM-XGBoost with Bayesian	**0.193**

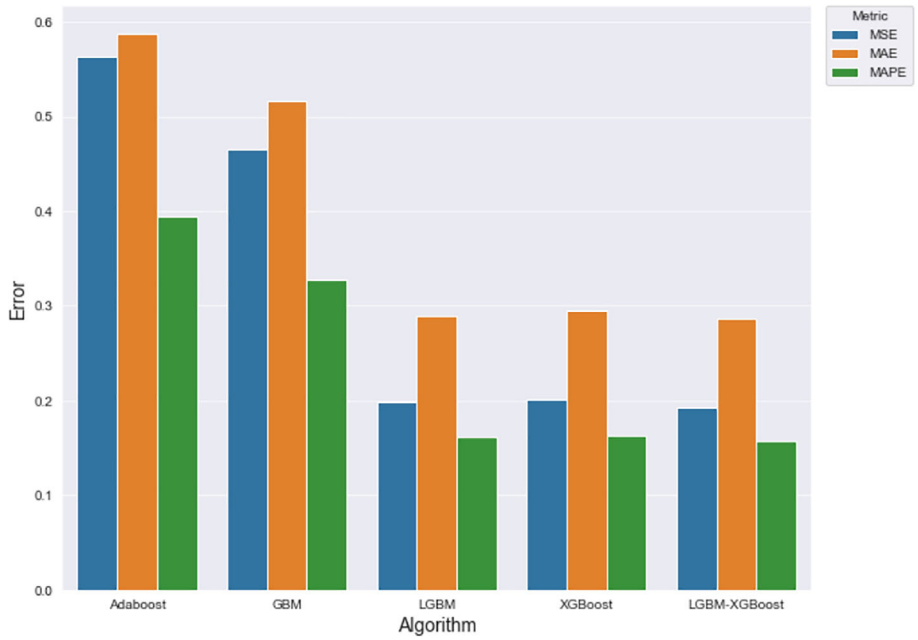


FIGURE 16 Performance evaluation of the proposed hybrid model. The hybrid LGBM and XGBoost model compared to Adaboost, GBM, LGBM, and XGBoost models

TABLE 3 Performance evaluation summary

Algorithm	MSE	MAE	MAPE	Time complexity (seconds)
Adaboost	0.564	0.588	0.395	1.791
GBM	0.466	0.517	0.328	4.914
LGBM	0.198	0.290	0.161	1.043
XGBoost	0.201	0.295	0.163	7.005
LGBM-XGBoost	**0.193**	**0.285**	**0.156**	**58.631**

4.6 | Hybrid model performance evaluation

Based on the baseline methods that is the base learners, LGBM and XGBoost models and tree based algorithms, Adaboost and GBM models, the hybrid model's performance was evaluated. The hybrid LGBM and XGBoost model had the lowest MSE, MAE and MAPE of 0.193, 0.285, and 0.156 outperforming the other algorithms as shown in Figure 16. The hybrid model performed better with minimum error thereby increasing performance in predicting house prices. The computation time was higher due to computation of out of fold predictions from base learners, hyperparameter processing and size of the dataset. The hybrid model's performance evaluation in predicting the output is summarized in Table 3.

The baseline boosting methods for validating the effectiveness of the proposed hybrid LGBM and XGBoost model were outperformed by the proposed model which shows that it is the best model in predicting house prices with reduced variance and improved accuracy.

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

5 | CONCLUSIONS

The optimized hybrid LGBM and XGBoost model had better performance results with a lower MSE, MAE and MAPE as compared to the individual baseline ML algorithms. However the computation time for the hybrid model was higher due to processing of hyperparameters for the base learners, obtaining out of fold predictions and the size of the dataset. The implementation of Bayesian hyperparameter optimization method was important to the hybrid model since it minimized the error which improved accuracy of the model. Therefore, hyperparameter tuning for the base learners should be implemented for better improved performance. By optimizing the hyperparameters, the values for regularization parameters reduced variance thereby combating overfitting in the hybrid model. For further research, other various datasets can be used on this model to check its performance based on accuracy and complexity of time. Experiments on the different hyperparameter optimization techniques for the base learners for better results.

AUTHOR CONTRIBUTIONS

Racheal Sibindi: conceptualization (equal); formal analysis (equal); methodology (equal); software (equal); validation (equal); visualization (equal); writing – original draft (equal). **Ronald Waweru Mwangi:** conceptualization (equal); methodology (equal); supervision (equal); writing – review and editing (equal). **Anthony Gichuhi Waititu:** conceptualization (equal); methodology (equal); supervision (equal); writing – review and editing (equal).

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1002/eng2.12599.

DATA AVAILABILITY STATEMENT

Data available at https://inria.github.io/scikit-learn-mooc/python_scripts/datasets_california_housing.html

ORCID

Racheal Sibindi https://orcid.org/0000-0003-0367-9875

REFERENCES

1. Kabari LG, Onwuka UC. Comparison of bagging and voting ensemble machine learning algorithm as a classifier. *Int J Adv Res Comput Sci Software Eng*. 2019;9(3):19-23.
2. Bühlmann P. *Bagging, boosting and ensemble methods*. Springer; 2012:985-1022.
3. Patel V, Choe S, Halabi T. Predicting future malware attacks on cloud systems using machine learning. *IEEE*. 2020;151-156.
4. Jabeur SB, Mefteh-Wali S, Viviani JL. Forecasting gold price with the XGBoost algorithm and SHAP interaction values. *Ann Operat Res*. 2021;1-21.
5. Ho WK, Tang BS, Wong SW. Predicting property prices with machine learning algorithms. *J Property Res*. 2021;38(1):48-70.
6. Zulkifley NH, Rahman SA, Ubaidullah NH, Ibrahim I. House price prediction using a machine learning model: a survey of literature. *Int J Modern Edu Comput Sci*. 2020;12(6):46-54.
7. Heidari M, Zad S, Rafatirad S. Ensemble of supervised and unsupervised learning models to predict a profitable business decision. *IEEE*. 2021;1-6.
8. Manasa J, Gupta R, Narahari N. Machine learning based predicting house prices using regression techniques. *IEEE*. 2020;624-630.
9. Rawool AG, Rogye DV, Rane SG. House price prediction using machine learning. *Int J Res Appl Sci Eng Technol*. 2021;9:686-692.
10. Varma A, Sarma A, Doshi S, Nair R. House price prediction using machine learning and neural networks. *IEEE*. 2018;1936-1939.
11. Wang F, Zou Y, Zhang H, Shi H. House price prediction approach based on deep learning and ARIMA model. *IEEE*. 2019;303-307.
12. Chou JS, Fleshman DB, Truong DN. Comparison of machine learning models to provide preliminary forecasts of real estate prices. *J Housing Built Environ*. 2022;1-36.
13. Sewell M. Ensemble learning. *RN*. 2008;11(02):1-34.
14. Sagi O, Rokach L. Ensemble learning: A survey. *Wiley Interdisciplin Rev Data Mining Knowledge Discovery*. 2018;8(4):e1249.
15. Basaran K, Özçift A, Kılınç D. A new approach for prediction of solar radiation with using ensemble learning algorithm. *Arabian J Sci Eng*. 2019;44(8):7159-7171.
16. Abou Omar KB. XGBoost and LGBM for Porto Seguro's Kaggle challenge: A comparison. *Preprint Sem Project*. 2018;1-16.

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

17. Cui S, Yin Y, Wang D, Li Z, Wang Y. A stacking-based ensemble learning method for earthquake casualty prediction. *Appl Soft Comput*. 2021;101:107038.
18. Ferreira AJ, Figueiredo MA. Boosting algorithms: A review of methods, theory, and applications. *Ensemble Mach Learn*. 2012;35-85.
19. Mayr A, Binder H, Gefeller O, Schmid M. The evolution of boosting algorithms. *Methods Informat Med*. 2014;53(06):419-427.
20. Dargahi-Zarandi A, Hemmati-Sarapardeh A, Shateri M, Menad NA, Ahmadi M. Modeling minimum miscibility pressure of pure/impure CO_2-crude oil systems using adaptive boosting support vector regression: Application to gas injection processes. *J Petrol Sci Eng*. 2020;184:106499.
21. Touzani S, Granderson J, Fernandes S. Gradient boosting machine for modeling the energy consumption of commercial buildings. *Energy Buildings*. 2018;158:1533-1543.
22. Rawi R, Mall R, Kunji K, Shen CH, Kwong PD, Chuang GY. PaRSnIP: sequence-based protein solubility prediction using gradient boosting machine. *Bioinformatics*. 2018;34(7):1092-1098.
23. Nalluru G, Pandey R, Purohit H. Relevancy classification of multimodal social media streams for emergency services. *IEEE*. 2019;121-125.
24. Liang W, Luo S, Zhao G, Wu H. Predicting hard rock pillar stability using GBDT, XGBoost, and LightGBM algorithms. *Mathematics*. 2020;8(5):765.
25. Machado MR, Karray S, Sousa IT. LightGBM: An effective decision tree gradient boosting method to predict customer loyalty in the finance industry. *IEEE*. 2019;1111-1116.
26. Chen P, Deng Y, Zhang X, et al. Degradation trend prediction of pumped storage unit based on MIC-LGBM and VMD-GRU combined model. *Energies*. 2022;15(2):605.
27. Cheng W, Jl L, Xiao HC, Ln J. Combination predicting model of traffic congestion index in weekdays based on LightGBM-GRU. *Sci Rep*. 2022;12(1):1-13.
28. Hao X, Zhang Z, Xu Q, Huang G, Wang K. Prediction of f-CaO content in cement clinker: a novel prediction method based on LightGBM and Bayesian optimization. *Chemom Intel Lab Syst*. 2022;220:104461.
29. Liang M, Chang Z, Wan Z, Gan Y, Schlangen E, Šavija B. Interpretable ensemble-machine-learning models for predicting creep behavior of concrete. *Cem Concr Compos*. 2022;125:104295.
30. Li K, Xu H, Liu X. Analysis and visualization of accidents severity based on LightGBM-TPE. *Chaos Solitons Fractals*. 2022;157:111987.
31. Luckner M, Topolski B, Mazurek M. *Application of XGBoost Algorithm in Fingerprinting Localisation Task*. Springer International Publishing; 2017:661-671.
32. Asselman A, Khaldi M, Aammou S. Enhancing the prediction of student performance based on the machine learning XGBoost algorithm. *Interactive Learn Environ*. 2021;1-20.
33. Li M, Fu X, Li D. *Diabetes prediction based on XGBoost algorithm*. Vol 768. IOP Publishing; 2020:072093.
34. Nziyumva E, Hu R, Hsu CY, Niyogisubizo J. *Electrical load forecasting using hybrid of extreme gradient boosting and light gradient boosting machine*. Springer; 2022:1083-1093.
35. Liang Y, Wu J, Wang W, et al. Product marketing prediction based on XGboost and LightGBM algorithm. 2019;150-153.
36. Pace RK, Barry R. Sparse spatial autoregressions. *Stat Probab Lett*. 1997;33(3):291-297.
37. Ekundayo I. *OPTUNA Optimization Based CNN-LSTM Model for Predicting Electric Power Consumption*PhD thesis, Dublin. National College of Ireland; 2020.
38. Matthews S, Hartman B. mshap: Shap values for two-part models. *Risks*. 2021;10(1):3.

How to cite this article: Sibindi R, Mwangi RW, Waititu AG. A boosting ensemble learning based hybrid light gradient boosting machine and extreme gradient boosting model for predicting house prices. *Engineering Reports*. 2023;5(4):e12599. doi: 10.1002/eng2.12599

25778196, 2023, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/eng2.12599 by RH-net, Wiley Online Library on [22/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License