ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

OPTIMIZATION OF XGBOOST ALGORITHM USING PARAMETER TUNNING IN RETAIL SALES PREDICTION

Hendra Wijaya[1], Dandy Pramana Hostiadi[2], Evi Triandini[3]

[1] *Magister Program*, Department of Magister Information Systems, Institut Teknologi dan Bisnis STIKOM Bali, Indonesia
[2,3] *Department of Magister Information Systems*, Institut Teknologi dan Bisnis STIKOM Bali, Indonesia

email: 222012007@stikom-bali.ac.id[1] , dandy@stikom-bali.ac.id[2], evi@stikom-bali.ac.id[3]

Abstract

In retail companies, the owner needs sales analysis to make decisions in the company's business processes. Several previous studies have introduced forecasting techniques using regression analysis, and classification approaches that need optimization. This article proposes a new approach to sales prediction using XGBoost, which is optimized by comparing the best performance from three optimization methods: Random search, grid search, and Bayesian optimization. The aim is to obtain the best comparative analysis and increase prediction accuracy. The novelty of the proposed model is determining the best value for each optimization method using XGBoost. The results of the evaluation show that the best results were achieved by the grid search optimization technique in the XGBoost model with an increase in the evaluation value R^2 from 97.31 to 98.41. The results of the proposed model analysis can help retail business owners in accurate sales predictions to determine the development of business processes.

Keywords : Xgboost, Retail, Random Search, Grid Search, Bayesian Optimization

Received: 06-07-2024 | **Revised:** 29-08-2024 | **Accepted:** 08-09-2024
DOI: https://doi.org/10.23887/janapati.v13i3.82214

INTRODUCTION

In the past, humans relied on instinct and intuition to run their businesses. However, with technological advances, artificial intelligence plays an important role in various fields, including retail companies. Artificial intelligence is already changing various aspects of retail operations and customer experience and significantly impacting this sector's businesses and customers [1]. Sales in retail companies are a crucial factor, even determining whether the company will be successful or fail [2]. Retail is the main indicator for assessing the performance of leading macroeconomic indicators. One way is to observe retail sales prices and their developments, as well as various other indicators. These figures are significant amidst global uncertainty [3].

However, many retailers, including XYZ retail company, often need help making accurate sales predictions. Fortunately, machine learning has opened up new opportunities in this field [4]. This technique has been proven effective in various fields, including predicting specific market trends that are increasingly popular in different industries [5]. Machine learning allows systems to learn from data and make predictions or decisions without explicit programming [6]. Additionally, machine learning has been used for sales forecasting in the retail sector, demonstrating its ability to optimize inventory management and improve operational efficiency [7]. As new algorithms continue to develop and a large amount of data becomes available, demand prediction models are used and improved better than traditional methods and can overcome complex correlations in the retail chain [8].

Several studies have highlighted the effectiveness of machine learning algorithms in improving sales forecasting and optimizing inventory management. However, it is important to note that making sales predictions in the retail sector using machine learning with optimal performance is a significant challenge. This underscores the complexity of the retail sector and the need for advanced tools for decision-making and policy formulation.

Through historical data analysis, machine learning models can forecast future sales trends, fluctuations during certain events, and customer behavior, helping retailers make better decisions [9][10][11][12]. The retail sector, which has a lot of human work and low profit margins, is suitable for applying artificial intelligence and machine learning [13]. Importantly, industry players are increasingly recognizing the importance of using machine learning models to predict sales of their products [14]. This growing trend underscores the significant role of machine learning in the retail sector.

XGBoost, a popular machine-learning model in various industries, is known for its excellent reliability, precision, and portability [15]. It is widely used for predictions in various sectors, including property, geography, and medical fields [16][17][18][19][20]. The retail sector is an example of how the XGBoost model can adapt to various situations and conditions [21]. In the retail sector, XGBoost has been used effectively for sales, price, and demand forecasting tasks [22][23][24]. Its ability to manage complex data and capture intricate patterns makes it a reassuring and suitable choice for predicting retail trends and optimizing business strategies.

In the context of retail demand forecasting, XGBoost has been compared with other machine learning algorithms such as Random Forest, Gradient Boosting, and Artificial Neural Networks, consistently showing its superiority in achieving high forecast accuracy [23]. In [23] explores a hybrid model combining XGBoost, Random Forest, and Logistic Regression to improve sales predictions, addressing individual model weaknesses. In contrast, this study focuses solely on optimizing XGBoost model, the conclusion of [23] notes that hybrid models take longer to train and predict due to the need to run several models in parallel or sequentially. While the hybrid model offers potential performance gains, it is more complex and time-consuming. Focusing on a single model like XGBoost reduces complexity and is easier to manage. Studies also show that XGBoost outperforms traditional regression models in tasks such as price prediction [24]. This algorithm's ability to provide accurate predictions and compatibility with various data sets not only instills confidence but also empowers decision-makers in the retail industry.

Additionally, XGBoost has been applied to optimize e-commerce platforms to predict potential buyers, resulting in increased precision and more targeted marketing strategies [25]. Its efficiency in using memory and hardware resources improves algorithmic efficiency and model refinement, making it a preferred choice for multiclass classification tasks [26].

In [27], leadership attempted to project future sales using the XGBoost model. They used sales data from 2013 to 2017 at several stores in Favorita, Ecuador. The results show very accurate predictions, with a high % accuracy rate of 92% based on the R-squared value. The findings from this research indicate that the XGBoost model can provide scientific and accurate estimates for sales in large stores. In these studies, reliance on R-squared as the sole assessment metric has its drawbacks in that it only sometimes reflects the superiority of the model when there is the presence of heteroscedasticity or outliers. Additionally, the model performance assessment is limited by not accommodating other metrics such as MAE, MSE, or RMSE. Furthermore, the absence of parameter tuning or hyperparameter optimization can significantly impact model performance.

Pavlyshenko et al [28], employed the XGBoost algorithm in the model they developed for Forecasting Sales Time Series. The research results demonstrate that the model can yield superior results compared to the time series method, particularly when sales patterns deviate from clear historical trends. Similar research was conducted by Akande et al [29], who used XGBoost to forecast sales with data from 45 retail stores. The analysis results underscore the adaptability of XGBoost to different sales patterns, reassuring the audience about its effectiveness in sales prediction and its potential to aid sales managers in making product price decisions.

In [8], a hybrid model that combines XGBoost, RF (Random Forest), and LR (Linear Regression) is proposed to analyze sales data in real time. The research results show that the proposed hybrid model RF-XGBoost-LR has an R-squared score of 95.51%. However, the proposed hybrid model has limitations, such as the need for extensive training data size and decision integration. In [30], focuses on predicting insurance claims using XGBoost with hyperparameter grid search and Bayesian search on the Allstate and Porto Seguro datasets. Evaluation results show the performance using MAE and five iterations. The results show that XGBoost with grid search on Allstate has an average MAE of 1,151.3756, with Bayesian search 1,153.4370. In Porto Seguro, optimization techniques such as grid search and Bayesian search have been proven to significantly improve accuracy. However, the results may differ if used on datasets from other sectors, such as retail.

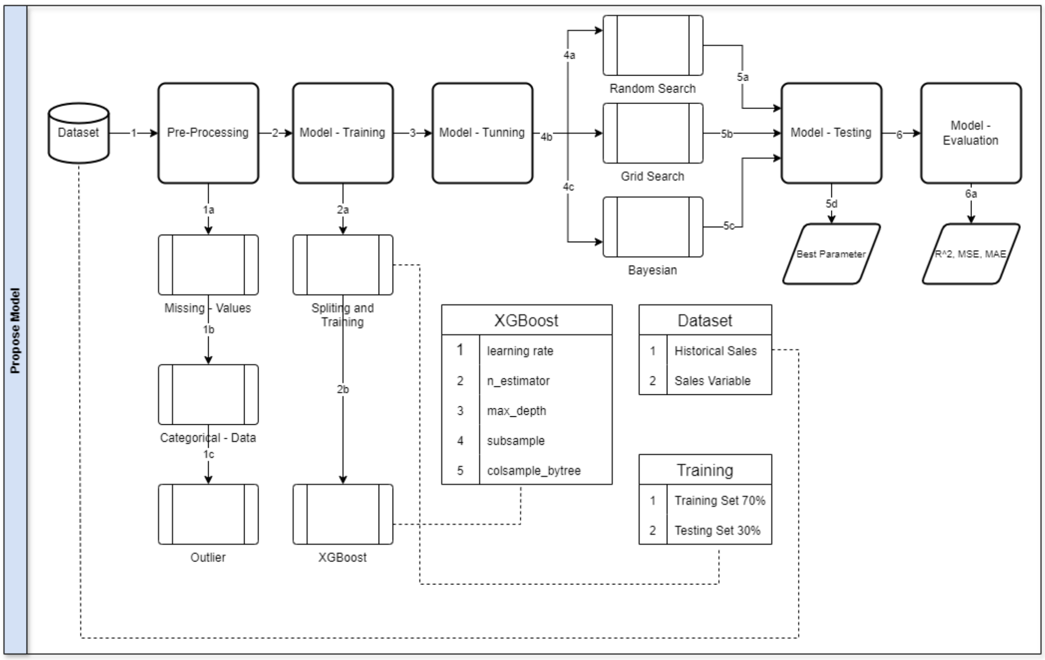
In [31] XGBoost and random search hyperparameter tuning are used to differentiate between phishing and non-phishing sites. Dataset from UCI Machine Learning Repository with 11,055 instances and 30 categorical features split into 80% for training and 20% for testing. Without hyperparameter tuning, XGBoost achieved 95.34% accuracy, 97.78%

recall, and 95.34% precision. Hyperparameter tuning increased accuracy to 97.69%, recall to 96.33%, and precision to 98.44%. However, the hyperparameters used were applied to phishing data and have not been tested using retail data.

While the XGBoost model has demonstrated its ability to predict regression and classification data in previous research, the suitability and analysis of using the best optimization techniques to produce optimal accuracy increases have not been fully explored. This research underscores the need for parameter tuning techniques with value adjustments for each optimization technique to improve the performance of accurate and optimal prediction models based on the characteristics of the data used in the retail dataset.

This paper proposes a prediction model using XGboost that is optimized by choosing one of three optimization approaches: Random search, Grid search, and Bayesian optimization, which produces the best performance in sales prediction on retail datasets. The aim is to obtain a comparative analysis of the best optimization method and the ideal value of the optimization method according to the characteristics of the dataset used. The novelty of this research is adjusting the weight values (Learning Rate, N Estimator, Max Deepth,Sub Sample, Sample Column) for each appropriate optimization technique for the XGboost method.

Optimizing hyperparameters like learning rate, n_estimators, max_depth, subsample, and colsample_bytree in XGBoost is key to improving model performance. Each parameter impacts the model's complexity and predictive accuracy. The learning rate (eta) controls the step size during each iteration. A lower rate improves convergence but requires more boosting rounds, increasing computation time [32][33]. A higher rate speeds up training but risks overshooting the optimal solution, leading to poor performance or overfitting [34][35]. The n_estimators parameter represents the number of trees in the ensemble. Increasing this number can capture complex data patterns but also raises the risk of overfitting, especially with deep trees or a high learning rate [36][37]. Max depth (max_depth) sets the maximum depth of each tree. Deeper trees can model complex relationships but are prone to overfitting, particularly in noisy or small datasets [38]. The subsample parameter determines the fraction of samples used for fitting individual learners. Values less than 1.0 add randomness, reducing overfitting and improving model robustness [39]. Colsample_bytree decides the fraction of features to be randomly sampled for each tree, helping to reduce overfitting and improve model diversity [40]. The optimization of the XGBoost model is anticipated to have a significant positive impact on supply chain efficiency, stock management, and retail company sales strategy planning. These implications are particularly relevant in the context of market uncertainty, highlighting the practical value of this research.



XGBoost	
1	learning rate
2	n_estimator
3	max_depth
4	subsample
5	colsample_bytree

Dataset	
1	Historical Sales
2	Sales Variable

Training	
1	Training Set 70%
2	Testing Set 30%

Figure 1. The Proposed Model

This model evaluated using performance metrics such as R^2, Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) [41]. The dataset used in this research comes from public data [42], which contains historical sales data and various variables that have the potential to influence sales, such as price, weather, holidays, and promotions.

METHOD

The following are the stages of the method propose in this paper, which are shown in Figure 1. This research begins with the preprocessing process on the dataset, which involves handling missing data, categorical data conversion, and outlier elimination. After that, the main model, namely XGBoost, then the model trained using the 30-70 validation scheme. The next step involves tuning the model using three different tuning methods, namely random search, grid search and Bayesian optimization. After the tuning process, the model was tested and evaluated using the R^2, MAE, and MSE metrics to measure the performance and accuracy of the sales prediction model. This evaluation aims to ensure that the XGBoost model has been optimized effectively and can provide accurate sales predictions. By using the three methods above, this research not only compares the results of the three tuning methods but also ensures that the best parameters are selected to increase the accuracy of sales predictions in the context of the retail industry, especially at Retail XYZ company.

Environment Requirement

In the research and experiments conducted to test the model and optimize its performance, the model needs specification to processing with specifications, M1 chip. This chip, with a total of eight cores, ensures robust performance, with four cores dedicated to efficiency and four to performance.

Table 1. Dataset Description

No	Feature Name	Features Description	Data Type
1	Store	Represents the number or unique identification of the retail store	Integer
2	Date	The date of the sale recorded in year-month-date format (YYYY-MM-DD)	String
	Type	Type Store with grades (A, B, C)	String
	Size	The size of the store	Integer
3	Weekly_Sales	The total weekly sales of the shop in question.	Float
4	IsHoliday	The binary variable (1 or 0) indicates whether the sales day falls on a holiday.	Integer
5	Temperature	The average temperature on the day of sale at the retail store location	Float
6	Fuel_Price	The average price of fuel on the day of sale	Float
7	CPI	Consumer Price Index(CPI) is the consumer price index that measures the level of inflation or changes in the average price of goods and services consumed by consumers.	Float
8	Unemployment	Represents the unemployment rate on the day of sale.	Float
9	Year	The year of sales.	Integer
10	Month	The month of sales.	Integer
11	Week	The week of sales.	Integer
12	Min	The minimum value of weekly sales data.	Float
13	Max	The maximum value of weekly sales data.	Float
14	Mean	The average value of weekly sales data.	Float
15	STD	The standard deviation value of weekly sales data	Float
16	Total_MarkDown	The total discount value from weekly sales data.	Float

The system also boasts 8 GB of memory, providing ample resources for research. The system firmware version is 8422.100.650, while the OS loader version used is 7459.141.1. This advanced hardware setup ensures the robustness and reliability of our research. Besides, this research leverages Spyder IDE Version 5.4.3, a popular and reliable software, and the widely used programming language Python 3.9.14 64-bit, Qt 5.15.2, Darwin 21.6.0. The use of these industry-standard tools, along with libraries such as scikit-learn for model implementation and optimization, and pandas and NumPy for data manipulation and analysis, ensures the reliability and credibility of our research.

Dataset Descriptions

This paper used public data in [42], which describes the retail sales dataset from the Kaggle site. The dataset file used as the source of this data is sales_dataset.csv, which has a file size of 51.7 MB. This dataset involves sales data from 45 stores from February 2019 to October 2021 and contains 37,427 data, providing a broad data framework. The information in the dataset covers various aspects of sales, enabling in-depth analysis of stores' performance and trends over significant periods. Thus, this dataset becomes a valuable resource for understanding sales dynamics. Details dataset description shows in Tabel 1.

The dataset feature above has 16 features covering various types of data. Six features use integer data types, which involve important aspects such as store information (Store), size (Size), and IsHoliday (whether the day is a holiday or not), as well as year, month (Month), and week (Week) information. Meanwhile, two features use the string data type to represent date and type information.

The other eight features use the float data type and include vital parameters in sales analysis, such as Weekly_sales (weekly sales), Temperature (temperature), Fuel_Price (fuel price), CPI (Consumer Price Index), Unemployment (unemployment rate), as well as mark statistics down (Min, Max, Mean, and STD), and Total_MarkDown. This dataset is not just a collection of numbers, but a practical tool for predicting sales, especially with features that represent minimum, maximum, and average weekly sales values.

The importance of external features such as CPI, Temperature, and IsHoliday becomes clear when using this dataset, enabling analysis of these factors in predicting sales behavior. For example, the IsHoliday feature provides insight into whether sales trend upward or downward during holiday periods, offering a deeper understanding of sales dynamics in the context of time. This dataset is a key to unlocking these insights.

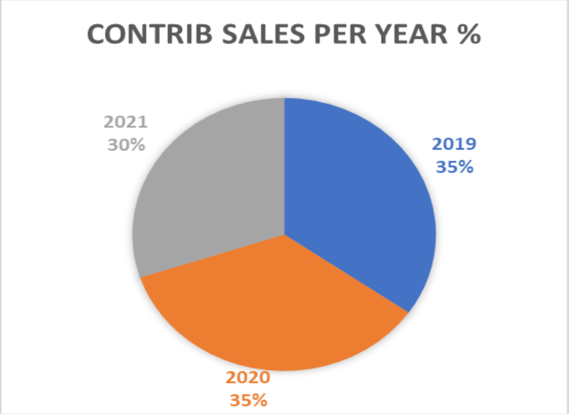


Figure 2. Contribution Sales Per Year (2019-2021)

Figure. 2 depicts sales for 2019, 2020, and 2021, revealing significant changes in sales performance. In 2019, sales represented approximately 34.70% of total sales. This percentage increased to around 35.20% in 2020, indicating a growth of 0.50% from the previous year. These changes underscore the importance of accurate sales predictions. This growth shows an increase in sales performance. However, 2021 shows that sales have decreased to around 30.10%, indicating a minus growth of 5.10% from 2020 to 2021. The increase in sales in 2020 can be attributed to the fact that the observation data for 2019 started in February, not January as usual, making the sales percentage appear higher overall. Conversely, the decrease in sales in 2021 is due to the fact that the observation data only covers the months up to October, not reaching December. Typically, year-end sales tend to increase, and the absence of this data explains the decrease in the sales percentage in 2021.

Pre-processing

The cornerstone of sales data analysis is the pre-processing stage, a pivotal step in machine learning that significantly shapes the accuracy of prediction models [43][44]. Data pre-processing is a critical factor in enhancing the accuracy of various machine learning models across different domains [45][46], Numerous studies underscore the significance of feature selection and data pre-processing in refining the accuracy of prediction models, with a primary focus on ensuring data cleanliness and consistency. The initial step involves addressing missing values or incomplete data, as well as converting categorical data into numerical format, and the final step is eliminating outliers in the dataset.

Applying techniques such as feature engineering and data pre-processing has been proven to capture crucial sales trends over various periods, thereby increasing the accuracy of sales predictions [22]. In addition, pre-processing can simplify the computational process and reduce the feature space, ultimately improving performance and classification accuracy [47].

Model Training

eXtreme Gradient Boosting (XGBoost) is an ensemble learning algorithm that utilizes the boosting method to produce a robust and accurate model. Developed in 2014 by Tianqi Chen, XGBoost became a leading representative of the family of gradient-boosting algorithms [48]. The basic principle lies in focusing improvements on examples misclassified by previous models, allowing the formation of increasingly adaptive and accurate models with iteration. Ensembles allow XGBoost to combine the strengths of several weak models, particularly decision trees, to form an overall more robust model.

XGBoost has emerged as a preferred choice in the machine learning community, gaining particular fame in Kaggle competitions [18]. Its high efficiency and flexibility make it a versatile tool for a range of tasks, including classification, regression, and ranking. This model incorporates intelligent strategies for handling missing values, model regulation to prevent overfitting, and automatic feature selection by weighing the most informative features.

The objective function or formula optimized by XGBoost can be described as follows:

$$Obj = \sum_{i=1}^{N} loss(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \qquad (1)$$

where:

- N is the amount of training data.
- y_i is the actual label of the i^{th} data.
- $\hat{y}_i$ is the model prediction for the i^{th} data.
- loss$(y_i, \hat{y}_i)$ is a loss function that measures the error between the prediction and the actual value.
- K is the number of decision trees in the model.
- f_k is the k^{th} decision tree.
- $\Omega\,(f_k)$ is a regularization function that controls the decision tree's complexity.

An overview of the regression tree-based boosting algorithm is shown in Figure 3 [49]. In the initial step, the model learns the first tree using the training data$(features, Y)$, and the first estimation result (Y_i) is obtained. The next step involves a second tree that performs the learning process from the training data $(features, |Y - Y_1|)$, where $|Y - Y_1|$ shows the difference between the actual and predicted labels in the previous stage. The third tree follows suit by carrying out a learning process from the data $(features, |Y - Y_1 - Y_2|)$ and producing an estimate of (Y_3). Through this approach, effectiveness is achieved in reducing error values.

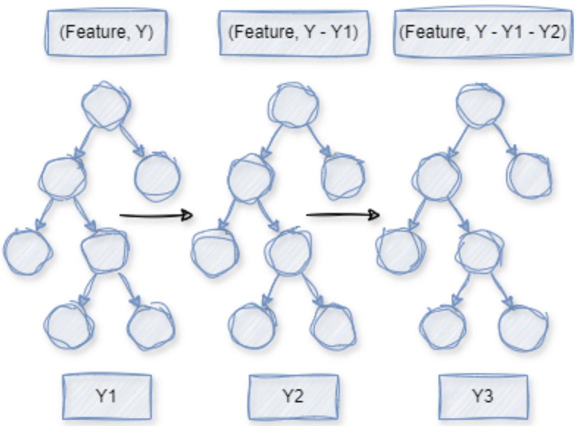


Figure 3. Regresion Tree

Model Tunning

The XGBoost model is implemented as the basis of the model from the default parameters above. Next, parameter tuning was carried out using three different methods: random search, grid search, and Bayesian optimization. Research conducted by Song Y et al. [50] and further research conducted by Xiong X et al. [51] recommend test range values for parameters that influence XGBoost performance, as shown in Table 2.

Table 2. Parameter Testing Range

Tunning Parameter	Testing Range
Learning Rate	0.001, 0.1, 0.2
Number of Trees (n_estimators)	100, 200, 300
Tree Depth (max_depth)	3, 5, 7,10
Subsample	0.8, 0.9, 1.0
Sample Column (colsample_bytree)	0.8, 0.9, 1.0

The parameters in Table 2 show the details of the combination in the model optimization process. First, the 'n_estimators' parameter determines the number of trees in the model, with 100, 200, and 300 value options. Then, 'learning_rate' indicates the extent to which the model learns from previous errors, with value options of 0.01, 0.1, and 0.2.

Next, the 'max_depth' parameter controls the maximum depth of each tree in the model and can be set as 3, 5, 7, or 10. 'Subsample' affects the proportion of samples taken to train each tree, with values of 0.8, 0.9, and 1.0. Finally, 'colsample_bytree' controls how many features each tree uses, with value options of 0.8, 0.9, and 1.0.

The model assessed based on its performance using a combination of these parameter values in the optimization process. Based on the results of the model evaluation, the best parameters selected form an optimal parameter configuration that can provide sales predictions with maximum accuracy and reliability.

Model Testing

This experiment evaluated and compared the performance of three optimization methods applied to the XGBoost model. This research process includes combining the XGBoost model and each tuning method, namely XGBoost with Random search, XGBoost with Grid search, and XGBoost with Bayesian optimization.

Model Evaluation

The next step is the evaluation process, which uses metrics such as R^2, Mean Absolute Error (MAE), and Mean Squared Error (MSE), the lowest value of which is considered the best result. The analysis was carried out by considering the highest R^2 accuracy value and the lowest Mean Squared Error (MSE) and Mean Absolute Error (MAE) values. The main focus of this experiment is to find the most accurate sales prediction model by identifying the optimal parameter configuration of the three optimization techniques to minimize prediction errors and increase model reliability.

In [52] conducted this study to analyze data from a local supermarket in Turkey to increase e-retail sales. In conclusion, this study uses MAE and RMSE metrics to evaluate the accuracy of product demand forecasting.

In Performance Analysis, the model used evaluation metrics such as R^2, MSE, and MAE to measure the quality and performance of the model. These metrics provide important information about the degree to which the model fits the data and how accurate the model predictions are in estimating actual values. The following is a detailed explanation of the use of these metrics:

R^2 (Coefficient of Determination) measures how well a regression model fits observational data. The R^2 value ranges from 0 to 1, where 1 indicates a model that perfectly fits the data. The model calculated R^2 using formula (2):

$$R^2 = 1 - \left(\frac{SSR}{SST}\right) \tag{2}$$

where SSR (Sum of Squares Residual) is the residual sum of squares (the difference between the predicted value and the actual value), and SST (Total Sum of Squares) is the total sum of squares (the difference between the actual value and the actual average). The model used R^2 to evaluate the extent to which the model can explain variation in the target variable based on existing features.

MSE (Mean Squared Error) measures the average of the squared differences between predicted values and actual values in regression. To calculate MSE the model used the formula (3) :

$$\text{MSE}= \frac{1}{\text{n}} \sum_{\text{i}=1}^{\text{n}} (y_i, \hat{y}_i)^2 \tag{3}$$

Where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of samples in the dataset. In the performance model analysis, the model uses MSE as an evaluation metric to determine how accurate the model predictions are in estimating actual values. The smaller the MSE value, the better the model performance in minimizing the average error.

MAE (Mean Absolute Error) measures the average of the absolute differences between predicted and actual values in regression. To calculate MAE, the model used formula (4):

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{4}$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of samples in the dataset. The model uses MAE as an evaluation metric to obtain information about the extent to which model predictions have an average error without considering the direction of the error. The smaller the MAE value, the better the model's performance in minimizing the average error.

This evaluation aims to ensure that the XGBoost model has been optimized effectively and can provide accurate sales predictions. In this paper, the model not only compares the results of the three tuning methods but also ensures that the best parameters are selected to increase the accuracy of sales predictions in the retail industry, especially in the XYZ Retail company.

RESULT AND DISCUSSION

This paper aims to predict retail product sales using XGBoost, which is optimized using the best optimization from three approaches: Random search, grid search, and Bayesian optimization. This research aims to obtain the best approach based on parameter value settings and increase prediction accuracy.

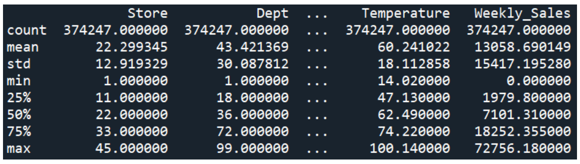
Experiment Result

Sales results are predicted using several steps, which have been explained in the methodological stages in Figure 1. The first step taken is pre-processing, namely ensuring the dataset is clean and consistent, which includes handling missing values or incomplete data and converting categorical data into numeric format. Then, the model processes the categorical data. There are three categorical columns, and only one is used in this research, which transforms into numerical data using the encoder technique. Two existing columns are not used and are deleted for reasons of relevance and duplication.

The goal of the pre-processing stage is to convert categorical variables into a form that can be processed by machine-learning models, which usually require input in numerical form. After the "Type" column is converted into numeric data using the label encoder technique, the next step is to delete several columns that are considered irrelevant. In this case, the "Date" and "Date1" columns are deleted because the information contained in them is already represented by the "Year", "Month", and "Week" columns. Removing irrelevant columns can simplify the dataset structure and reduce dimensions that are not needed in subsequent analysis, making the process more effective and efficient.

Then, further handling of outliers is carried out. Outliers are significant values that are far from other values in the dataset. Research conducted by Alabrah et al. [53], confirms this, highlighting the use of the Interquartile Range (IQR) method in detecting credit card fraud (CCF) in online / e-commerce transactions. By cleaning and normalizing outliers using IQR and selecting significant features, this experiment successfully resolved class imbalance in the dataset, significantly impacting the data distribution and representation.

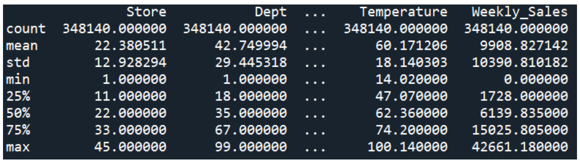
In Figure 4, the "Weekly_Sales" column depicts store sales data. From the available data, the average weekly sales in these stores are around 13,058.69. This figure reflects the average number of sales achieved by stores in one week. Additionally, the standard deviation of the "Weekly_Sales" column is approximately 15,417.19. Standard deviation is a statistical measure that indicates how far weekly sales data is spread from the average. With a relatively high standard deviation, weekly sales data at these stores show significant variations. This process implies a large difference between the highest and lowest weekly sales.



```
               Store           Dept  ...    Temperature   Weekly_Sales
count  374247.000000  374247.000000  ...  374247.000000  374247.000000
mean       22.299345      43.421369  ...      60.241022   13058.690149
std        12.919329      30.087812  ...      18.112858   15417.195280
min         1.000000       1.000000  ...      14.020000       0.000000
25%        11.000000      18.000000  ...      47.130000    1979.800000
50%        22.000000      36.000000  ...      62.490000    7101.310000
75%        33.000000      72.000000  ...      74.220000   18252.355000
max        45.000000      99.000000  ...     100.140000   72756.180000
```

Figure 4. Dataset Description Before Outlier Proses

One step to reduce the standard deviation (std) in the dataset above is to handle outliers. Outliers such as the mean and std can greatly impact statistical calculations. By removing or properly handling outliers, model can reduce the extreme variations that affect the std. Before the outlier technique was carried out for the Weekly_Sales column, it reached 15,417.19; after the outlier technique was carried out, the standard deviation of the Weekly_Sales column was 10,390.81. In this context, the reduction in the mean after applying the standard deviation outlier technique can be interpreted as a decrease in the influence of the outliers on the overall statistics of the Weekly_Sales column. By eliminating outliers, the data becomes more representative, and the distribution of Weekly_Sales values becomes more stable, allowing for more accurate analysis and models. The results can be seen in Figure 5.



```
               Store           Dept  ...    Temperature   Weekly_Sales
count  348140.000000  348140.000000  ...  348140.000000  348140.000000
mean       22.380511      42.749994  ...      60.171206    9908.827142
std        12.928294      29.445318  ...      18.140303   10390.810182
min         1.000000       1.000000  ...      14.020000       0.000000
25%        11.000000      18.000000  ...      47.070000    1728.000000
50%        22.000000      35.000000  ...      62.360000    6139.835000
75%        33.000000      67.000000  ...      74.200000   15025.805000
max        45.000000      99.000000  ...     100.140000   42661.180000
```

Figure 5. Dataset Description After Outlier Process

After the pre-processing process, the dataset is divided proportionally into two main groups: training data and test data, with a ratio of 70:30. As much as 70% of the data is used as a training set to train the model. Besides, the remaining 30% is used as a testing set to evaluate model performance.

The XGBoost model used as a training basis has default values for the main parameters, as shown in Table 3.

Table 3. Default Parameter XGBoost

Parameter	Default Value
Learning Rate	0.3
Number of Trees (n_estimators)	100
Tree Depth (max_depth)	6
Minimum Sample Split (min_child_weight)	1
Subsample	1
Sample Column (colsample_bytree)	1
Gamma	0

In the classification phase, the XGBoost algorithm used parameter values, namely Learning Rate of 0.3, Number of Trees set is n_estimators with default value 100, Tree Depth used max_depth set as default value at 6, Minimum Sample Split use min_child_weight with set as default of 1, Subsample set as default with value at 1, Sample Column set as colsample_bytree with default at 1, and Gamma set as default at 0. Learning Rate indicates how much the model remembers the influence of previous trees, while Number of Trees determines the number of decision trees to build. Tree Depth sets the maximum depth of each tree, and Minimum Split Samples control the number of samples required to create a split on a node. Subsample and Sample Columns affect the proportion of data and features used in each iteration. Gamma is a parameter that controls whether a node splited based on the profit from the split. The training results using the XGBoost basic model are shown in Table 4.

Table 4. XGBoost Result

Model	*R^2*	*MSE*	*MAE*
XGBoost	97.31	6,328,529.29	1,403.40

The XGBoost results table in Table 4 shows quite good performance with an R^2 value of 97.31%, which means this model can correctly predict test data up to 97.31% of the data variability. MSE (Mean Squared Error) of 6,328,529.29 shows the average squared error of the model predictions, which shows how far the model predictions are from the actual value. In addition, the MAE (Mean Absolute Error) of 1,403.40 shows the average absolute error between the prediction and the actual value, indicating how big the model prediction error is in general. Besides, this model is entirely accurate in making predictions. After training with the XGBoost basic model and the R^2 results reaching a value of 97.31%, the next step is parameter tuning. The aim is to optimize the XGBoost basic model so it can produce more accurate R^2 values, as well as decreasing MSE and MAE values. Tuning parameters use the test range, which using range value.

The model has tested various values of each parameter within a predetermined range to obtain the most optimal XGBoost prediction results. Through this process, the model evaluates multiple combinations of parameters to determine the configuration that provides the best performance. The ten best combinations of parameter values were identified and presented in Table 5, which includes all tested combinations.

Table 5 Combination of Parameter Testing

Optimization	Rank	Learning Rate	N Estimator	Max Deepth	Sub Sample	Sample Coloumnt	R^2 %
RS	10	0.01	100	3	0.8	0.9	79.43
RS	9	0.01	100	5	0.8	0.8	80.00
RS	8	0.01	100	7	0.8	0.9	80.57
RS	7	0.01	100	10	1	0.9	81.65
RS	6	0.01	200	3	0.8	1	90.58
RS	5	0.1	100	3	0.8	0.9	93.12
RS	4	0.2	100	3	0.8	0.9	93.57
RS	3	0.01	300	7	0.8	0.9	93.79
RS	2	0.1	200	5	1	1	95.44
RS	**1**	**0.2**	**300**	**5**	**0.9**	**0.9**	**97.23**
GS	10	0.2	100	7	0.8	1	96.77
GS	9	0.2	100	7	0.9	1	96.84
GS	8	0.2	100	7	0.8	0.8	96.85
GS	7	0.2	100	7	0.8	0.9	96.90
GS	6	0.1	200	7	1	0.8	97.04

Optimization	Rank	Learning Rate	N Estimator	Max Deepth	Sub Sample	Sample Coloumnt	R^2 %
GS	5	0.2	200	7	0.8	0.8	97.41
GS	4	0.2	200	7	0.8	0.9	97.41
GS	3	0.1	300	10	1	0.9	98.11
GS	2	0.2	300	10	0.9	1	98.15
GS	**1**	**0.2**	**300**	**10**	**1**	**1**	**98.41**
BO	10	0.2	100	3	1	1	93.63
BO	9	0.01	300	7	0.9	1	93.76
BO	8	0.2	200	3	0.8	0.9	94.22
BO	7	0.2	300	5	0.8	0.9	96.72
BO	6	0.2	300	5	0.9	0.9	96.74
BO	5	0.2	300	5	1	1	96.77
BO	4	0.2	100	7	0.9	0.8	96.89
BO	3	0.1	300	7	1	1	97.24
BO	2	0.2	100	10	1	0.9	97.76
BO	**1**	**0.2**	**200**	**10**	**0.9**	**1**	**98.35**

RS = Random Search, GS = Grid Search, BO = Bayesian Optimization

The combination of Random Search optimization with a learning rate of 0.01, 100 estimators with a depth of 3, and subsample and colsample_bytree with values of 0.8 and 0.9, respectively, resulted in an R^2 value of 79.43%. This result indicates that the model performance is not good. In the following combination, when the learning rate is increased to 0.1, but other parameters remain the same, the model performance increases significantly with an R^2 of 93.12%. The experiment result shows that a higher learning rate can improve model accuracy. Furthermore, using a learning rate of 0.2, 300 estimators, and a depth of 5, the best optimization combination produces the best performance with an R^2 of 97.23%. The experiment result shows that the parameter combination is optimal. A higher learning rate has been shown to improve model accuracy compared to a lower one. This optimal parameter combination has a high learning rate, many estimators, and an appropriate depth, producing the best model performance. In Grid Search optimization, with a learning rate of 0.2, 100 estimators, depth 7, and colsample_bytree of 1, the model produces an R^2 of 96.77%. When the learning rate is fixed at 0.2, the number of estimators is increased to 200, and colsample_bytree is slightly reduced to 0.8, the model performs very well with an R^2 of 97.41%. In the last combination, with the highest learning rate of 0.2, the largest number of estimators of 300, depth 10, and the maximum values for subsample and colsample_bytree, the model produces the best results with the highest R^2 of 98.41%. In the last optimization, namely Bayesian Optimization, with a learning rate of 0.2 and 100 estimators, and subsample and colsample_bytree each with a value of 1, the model produces lower performance with an R^2 of 93.63% compared to the combination using a learning rate of 0.2, 300 estimators, depth 5, and subsample and colsample_bytree values of 1 each, which shows better performance with an R^2 of 96.77%. When the learning rate is fixed at 0.2, the number of estimators is increased to 200, depth 10, and colsample_bytree is 1. This optimization performs best with the highest R^2 of 98.35%, indicating that these parameters are optimal for this combination. In Grid Search optimization, the model achieves the best performance, combining the highest learning rate, the largest number of estimators, and deep depth. However, in Bayesian Optimization, even with a high learning rate and other optimal parameters, the model performance is not as good as the best parameter combination of Grid Search. With the right configuration in Grid Search, the model produces the best results, indicating that the parameter combination is the most effective. The explanation above can be seen in graphical form in Figure 6 for a clearer picture.

The summarized results of the test range parameters for the three optimization techniques, which obtain the best parameter values Random Search (RS), Grid Search (GS), and Bayesian Optimization (BO), can be seen in Table 6.

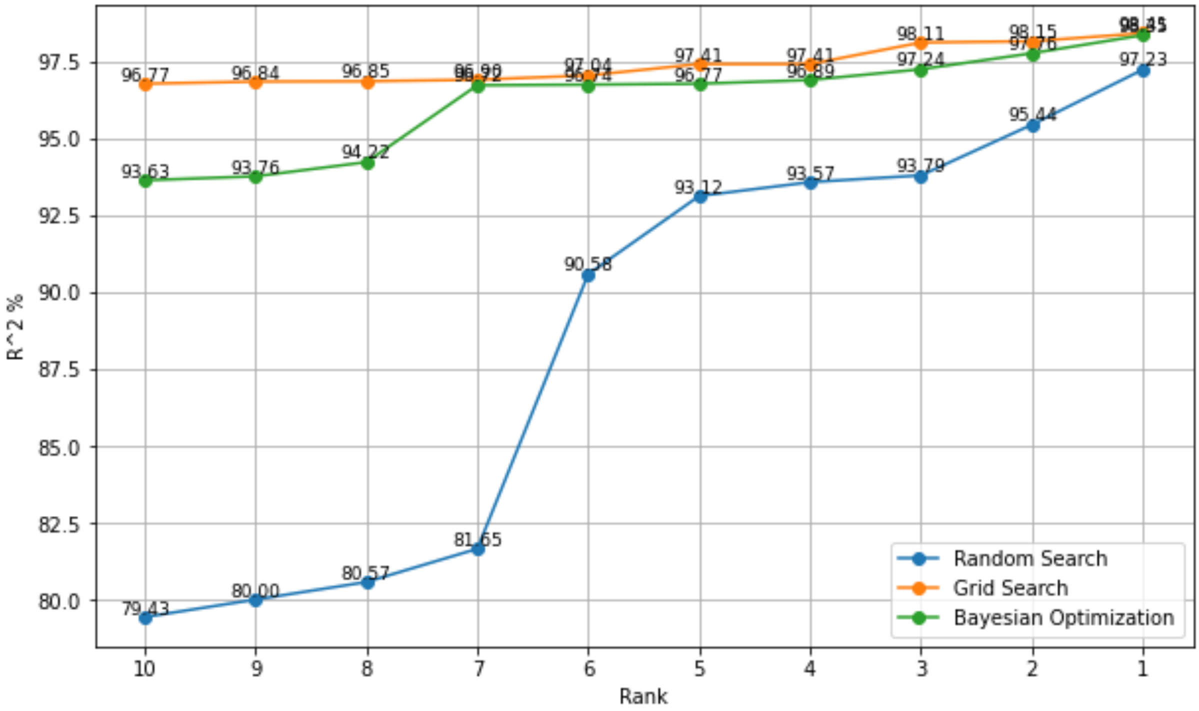


Figure 6. Grapich R^2 Each Combination

Table 6. Best Parameter Values

Tunning Parameter	RS	GS	BO
Learning Rate	0.2	0.2	0.2
Number of Trees (n_estimators)	300	300	200
Tree Depth (max_depth)	5	10	10
Subsample	0.9	1.0	0.9
Sample Column (colsample_bytree)	0.9	1.0	1.0

GS = Random Search, GS = Grind Search, BO = Bayesian Optimazation

First, using the Random search technique, the best value for 'Learning Rate' is 0.2, 'Number of Trees (n_estimators)' reaches 300, 'Tree Depth (max_depth)' is 5, 'Subsample' is 0.9, and 'Sample Column (colsample_bytree) ' reached 0.9. Meanwhile, Grid search produces a slightly different best parameter configuration, with 'Learning Rate' and 'Number of Trees' remaining at 0.2, 300, but 'Tree Depth' getting a value of 10. However, 'Subsample' and 'Sample Column' reach optimal values of 1.0 and 1.0, respectively. In the Bayesian optimization approach, the best parameter for 'Learning Rate' is 0.2, 'Number of Trees' and 'Tree Depth' with values of 200 and 10, 'Subsample' reaches a value of 0.9, and 'Sample Column' approaches the optimal value with 1.0. In the 'Learning Rate' optimization, three algorithms, Random search, Grid search, and Bayesian optimization, they obtained similar set values at 0.2. This consistency indicates that a value of 0.2 may have an advantage in balancing the learning rate of the model. However, in the 'Number of Trees (n_estimators)' parameter, Random and Grid search tend to select 300 trees, while more Bayesian optimization selects 200 trees. This difference indicates a trade-off between several trees and model performance, with greater emphasis on tree diversity by Random search and Grid search to improve accuracy. In addition, 'Tree Depth (max_depth)' is maintained at a value of 10 by both methods, namely Random search and Grid search; this depth selection can positively impact the quality of sales predictions. Besides, differences appear in the 'Subsample' parameter, where Random search and Bayesian optimization select a value of 0.9, indicating that this sample provides optimal results. In the experiment, the Grid search prefers a value of 1.0, reflecting the diversity in preferences between techniques. Finally, in 'Column Sample (colsample_bytree)', Grid search and Bayesian optimization choose the optimal value of 1.0.

In contrast, Random search chooses 0.9, which indicates a focus on different feature choices between optimization techniques. After tuning the parameters using three techniques and getting the best parameter results, its used as basis for optimization in the XGBoost basic model, the next step is to test the best parameters in the basic XGBoost model. The results of this testing can be seen in Table 6.

The XGBoost base model without initial optimization shows an R^2 of 97.31%, an MSE of 6,328,529.29, and an MAE of 1,403.4. Through a series of optimizations that have been carried out previously, three optimization techniques are applied: Random search, Grid search, and Bayesian optimization.

By using Random search, the XGBoost model's performance experienced a slight decrease, with R2 being 97.23%, MSE increasing to 6,507,225.66, and MAE being 1,430.8. Next is Grid search, which increased R2 to 98.41%, reducing MSE to 3,733,646.67 and MAE to 1,028.27. The final technique applied was Bayesian optimization, where the XGBoost model achieved an R2 of 98.35%, an MSE of 3,863,754.15, and an MAE of 1,054.30. The results generally show that Grid search and Bayesian optimization can increase the accuracy and reliability of sales predictions for the XGBoost model.

In contrast, Random search does not provide an improvement but instead reduces prediction accuracy. Grid search can increase the accuracy and reliability of sales predictions of the XGBoost model, with an increase in the R^2 value from 97.31% to 98.41%. These findings confirm that the use of Grid search and Bayesian optimization is effective and can produce more accurate and reliable sales predictions.

Model Performance Analysis

The performance evaluation results of the sales prediction model show significant progress through a series of optimization steps. Its shows the performance of the XGBoost model in predicting data with different R^2 values based on the optimization technique used. The XGBoost model without special optimization produces an R^2 value of 97.31%. When optimization was carried out with Random Search to optimize the model, the R^2 value dropped slightly to 97.23%. R^2 comparation result shows in Figure 7.

Figure 8 explained evaluation data regarding MSE for four variants of the sales prediction model reflects the performance comparison between these models. The XGBoost model without optimization has an MSE of 6,328,529.29. When using Random Search to optimize the model, the MSE increased slightly to 6,507,225.66, which means the model predictions became slightly less accurate.

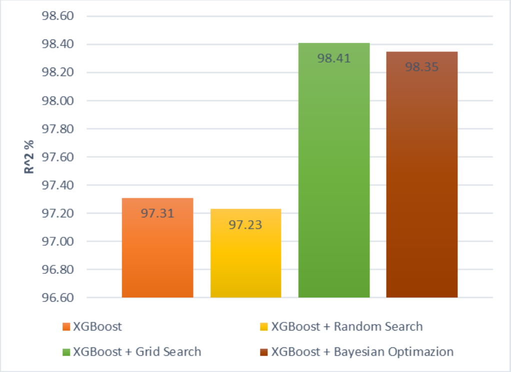


Figure 8. Grapich R^2 Comparation

However, the model shows a significant improvement with an R2 value of 98.41% when using the grid search optimization technique. Bayesian optimization techniques also provide excellent results, with an R2 value of 98.35%. It shows that Grid Search and Bayesian Optimization techniques can substantially improve the accuracy of the XGBoost model compared to the basic method or Random Search.

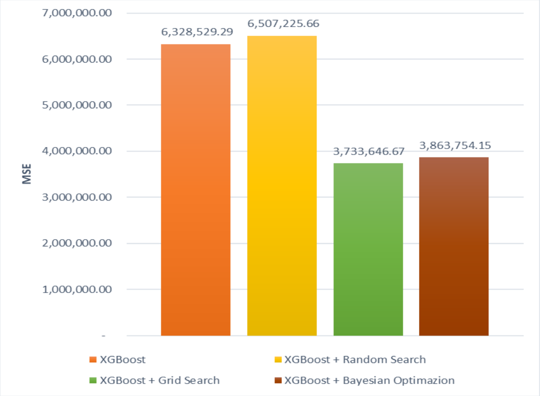


Figure 9. Grapich MSE Comparation

Table 7. Optimization Testing Result

Model	R^2 (%)	MSE	MAE
XGBoost	97.31	6,328,529.29	1,403.40
XGBoost + Random Search	97.23	6,507,225.66	1,430.80
XGBoost + Grid Search	**98.41**	**3,733,646.67**	**1,028.27**
XGBoost + Bayesian Optimazion	98.35	3,863,754.15	1,054.30

However, using Grid Search, the MSE decreased drastically to 3,733,646.67, indicating that the model predictions became much more accurate. The Bayesian optimization technique also reduces the MSE to 3,863,754.15, showing a significant increase in accuracy, although not as good as Grid Search. The percentage difference between the MSE of the initial XGBoost model and after optimization with Grid Search reaches approximately 40.98%. It shows a significant improvement in the model's prediction accuracy after applying these optimization techniques. The increase in MSE in Random Search can be caused by inaccurate random parameter selection, which could be more optimal. At the same time, the decrease in MSE in Grid Search and Bayesian Optimization occurs because these two techniques are more systematic in finding the optimal combination of parameters, thereby increasing the accuracy of model predictions.

Furthermore, Figure 10 explains that data evaluating the performance of the sales prediction model using the Mean Absolute Error (MAE) metric provides a clear picture of the model's progress through a series of optimizations. MAE is a measure that describes the average absolute differences between predicted and actual values. The XGBoost model without optimization has an MAE of 1,403.4. When Random Search optimized the model, the MAE increased to 1,430.8. It could be caused by a non-optimal random parameter search, which does not always produce the best configuration for the model.

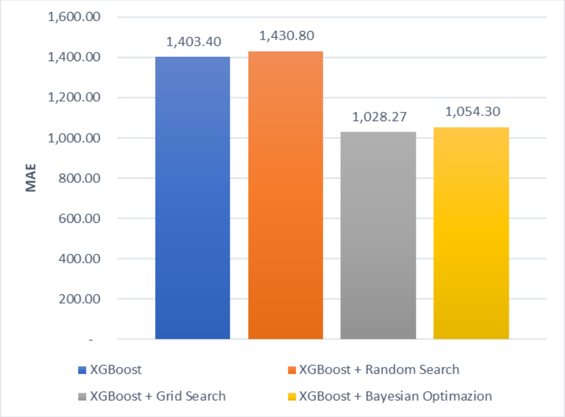


Figure 10. Grapich MAE Comparation

However, when Grid Search optimized the model, the MAE decreased drastically to 1,028.27, indicating a significant improvement in prediction accuracy. This decrease in MAE reflects that the model is getting better at estimating the actual value more closely. Likewise, with Bayesian Optimization, although it is not as optimal as Grid Search, it still produces a significant reduction in MAE to 1,054.30. Thus, increases and decreases in MAE values reflect changes in the model's prediction accuracy.

Table 8. Comparation Using Other Dataset

No	Researcher	Model	Dataset	R^2 (%)	MSE	MAE
1	Yao etc [54]	Decision Tree	Walmart	90.50	49,832,386.62	2,377.97
		Random Forest		93.70	32,993,323.63	1,937.81
		K Neighbors		59.40	213,246,328.55	8,199.39
2	Akande etc [29]	XGBoost	Walmart	97.62	12,090,252.08	1,317.65
3	Catal etc [55]	Bayesian Linear	Walmart	96.00	-	2,469.54
		Linear Regression		96.00	-	2,480.12
		Neural Network		0.00	-	14,951.09
		Boosted Decision Tree		97.00	-	1,669.10
4	Purposed	XGBoost	Walmart	93.29	34,495,589.09	3,275.88
		XGBoost + Random Search		97.99	10,343,565.92	1,576.02
		XGBoost + Grid Search		**98.11**	**9,698,788.83**	**1,493.62**
		XGBoost + Bayesian Optimazion		97.94	10,575,010.23	1,538.20

Comparation Using Other Dataset

To show that the proposed model can be applied to retail datasets, especially sales datasets, it conducted trials on another dataset, namely the Walmart dataset [54]. This dataset is historical data covering sales from February 5, 2010, to November 1, 2012, with 16 main features: Store, Date, IsHoliday, Dept, Weekly_Sales, Temperature, Fuel_Price, MarkDown1, MarkDown2, MarkDown3, MarkDown4, MarkDown5, CPI, Unemployment, Type, Size, with a total of 421,570 rows of data. Several researchers have carried out trials on this dataset [54] [29] [55].

Table 7 data shows the performance of various models as measured by R^2 from several studies. Researchers [54] tested several models, with Decision Tree achieving R^2 of 90.50%, Random Forest of 93.70%, and K Neighbors only 59.40%. Researchers [29] found that the XGBoost model had the best performance with an R^2 of 97.62%. Researchers [55] reported similar results for Bayesian Linear Regression and Linear Regression with R^2 of 96.00%. In comparison, Neural Network Regression did not give very bad results with R^2 of 0.00% but Boosted Decision Tree Regression produced R^2 of 97.00%.

The proposed model in this paper focuses on optimizing the XGBoost model, with R^2 results for the base model of 93.29%. After optimization using Random Search, Grid Search, and Bayesian Optimization, the results increased to 97.99%, 98.11%, and 97.94% respectively.

Research conducted by [29] and by us show significant differences in results in the use and optimization of the XGBoost model. Researchers [29] used the XGBoost model without additional optimization and achieved a coefficient of determination R2 of 97.62%. This shows that the XGBoost model they used is quite strong and can explain around 97.62% of the data variation. On the other hand, our research uses the basic XGBoost model, which achieves an R^2 of 93.29% and applies various optimization techniques to improve its performance. These optimization techniques include Random Search, Grid Search, and Bayesian Optimization, which yield R^2 of 97.99%, 98.11%, and 97.94%, respectively. By using these techniques, the proposed model succeeded in increasing the accuracy of the XGBoost model, with Grid Search providing the best results with an R^2 of 98.11%. The main difference between these two studies is that researchers [29] achieved high results with the basic XGBoost model. In contrast, our research achieved higher results by optimizing the XGBoost model through the Grid Search optimization technique.

Discussion

The use of engineering features and variable selection in this research indicates that several problems need attention. One of them is that the number of variables/columns deleted is still limited (3 columns deleted), so if the dataset has a larger number, the execution time required longer. It shows the need to consider the complexity of the dataset in selecting variables so the results can be more representative and accurate. Besides, the use of outlier techniques in this research is still limited to the interquartile range (IQR) method, and other methods that may be more effective have yet to be tried. Using alternative outlier methods can provide additional insight into extreme data and assist in cleaning up unusual or unrepresentative data.

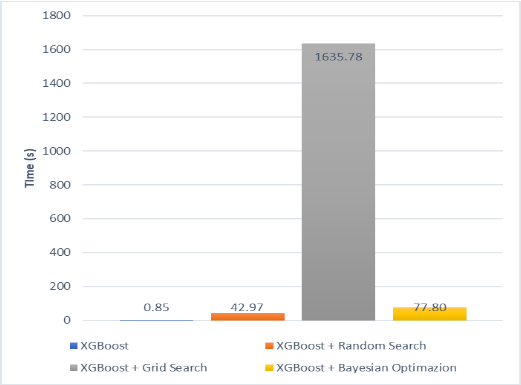


Figure 11. Grapich Time(s) Execution Comparation

From the execution time graph, which can be seen in Figure 10, execution time is a separate focus in evaluating the performance of four sales prediction models. The data presents execution times in seconds for four variations of the XGBoost model. The base XGBoost model completes the process relatively quickly, taking just 0.85 seconds. However, when the Random Search technique was applied to optimize the model, the execution time increased significantly to 42.97 seconds. This additional time is caused by the random parameter search process, which requires a longer time to find the optimal configuration.

Furthermore, when the model is optimized using Grid Search to obtain the best parameters, the execution time increases to 1635.78 seconds. This result depends on parameter combinations, which can take exponentially more time, especially if the parameter space is large. The Bayesian Optimization is more efficient than Grid Search but still takes longer than Random Search, with an execution time of around 77.80 seconds. Thus, although optimization techniques such as Grid Search and Bayesian Optimization can improve model accuracy, they also require significant time in the optimization process.

The significant increase in execution time in optimization techniques such as Grid Search and Bayesian Optimization needs to be considered as a factor in future research. Although this technique can improve model accuracy, the time required for a long optimization process can be an obstacle in practical applications, especially if there are strict time constraints in real-time data processing. Therefore, future research could lead to the development of more computationally efficient optimization methods.

This research has several limitations that need to be considered. It is limited to using only the XGBoost machine learning model and three optimizations, and the dataset is limited to retail companies. The following is an explanation of these limitations. This research is limited to the use of the XGBoost machine learning model. Although these models are commonly used in predictive analytics, many other machine learning models can be used to solve similar problems. In addition, the models may need help to capture the complexity contained in the data or provide an optimal level of accuracy.

Further research using other models may provide valuable additional insights. The limitations of optimization in this research lie in the exclusive focus on three optimization methods: random search, grid search, and Bayesian. While these three approaches have contributed significantly to improving the performance of sales prediction models, future research can expand their scope by exploring other optimization methods. The potential use of techniques such as simulated annealing, particle swarm optimization, or other evolutionary optimization methods could be an exciting step to explore. By expanding the variety of optimization methods, research can provide more comprehensive insight into the best options for improving model accuracy and optimization process efficiency in the context of sales prediction.

This research only uses a limited dataset on retail company XYZ. Due to the specific characteristics of the data, this may result in limited generalization. The results of this research may only be directly applicable to some industrial or sector contexts. Using broader and more representative datasets from various sectors or different data sources can broaden the generalization of research results.

This research only focuses on using machine learning models in the context of retail companies. Therefore, the results of this study may not be directly applicable to other industries or different business situations. Each industry or business context has unique characteristics and factors that must be considered when developing predictive models. This limitation needs to be acknowledged so the research results are not considered a universal solution.

CONCLUSION

In the context of sales prediction at XYZ retail, the XGBoost model and Grid Search optimization are superior to other optimizations such as Random search and Bayesian Optimization. The results show that the Grid Search optimization technique in the XGBoost model achieves the best performance, with the R^2 evaluation value increasing from 97.31% to 98.41%. The improvement difference between standard XGBoost and XGBoost after optimization with Grid Search is 1.10 points. This optimization was carried out with the parameters 'Learning Rate' of 0.2, 'Number of Trees' of 300, 'Tree Depth' of 10, and 'Subsample' and 'Sample Column' each of 1.0. However, it is important to note that Grid Search execution time is longer than both existing optimizations. Testing on other datasets, such as Walmart, also got better results than existing models; previeus model found that Decision Tree, Random Forest, and K Neighbors had R^2 of 90.50% and 93.70%, respectively, and 59.40%. Hybird model showed XGBoost as the best model with an R^2 of 97.62%. Other model showed R^2 of 96.00% for Bayesian Linear Regression and Linear Regression, while Neural Network Regression was only 0.00%, and Boosted Decision Tree Regression was 97.00%. The proposed model focuses on XGBoost optimization, with a base model R^2 of 93.29%, which increases to 97.99% with Random Search, 98.11% with Grid Search, and 97.94% with Bayesian Optimization. Grid Search optimization significantly improves performance compared to hybird and other model. So, the proposed model can be implemented on similar retail datasets.

Overall, these results underscore the importance of further development in various aspects of research, including the selection of other models, engineering features, variable selection, use of outlier techniques, and variations in optimization techniques. The results of this research also have limitations, involving an exclusive machine learning model on XGBoost and a dataset limited to retail companies only. In future research, further exploration must be carried out to increase the validity and generalization of research results.

ACKNOWLEDGMENT

I would like to extend my sincere gratitude to my advisor and everyone who contributed to the completion of this journal publication as part of my studies in the Magister Program, Department of Magister Information System, Institut Teknologi dan Bisnis STIKOM Bali.

REFERENCES

[1] S. Sharma, N. Islam, G. Singh, and A. Dhir, "Why Do Retail Customers Adopt Artificial Intelligence (AI) Based Autonomous Decision-Making Systems?," *IEEE Trans Eng Manag*, vol. 71, pp. 1846–1861, 2024, doi: 10.1109/TEM.2022.3157976.

[2] X. Dairu and Z. Shilong, "Machine Learning Model for Sales Forecasting by Using XGBoost," *2021 IEEE International Conference on Consumer Electronics and Computer Engineering, ICCECE 2021*, pp. 480–483, Jan. 2021, doi: 10.1109/ICCECE51280.2021.9342304.

[3] H. Martinus, "ANALISIS INDUSTRI RETAIL NASIONAL".

[4] A. Schmidt, M. W. U. Kabir, and M. T. Hoque, "Machine Learning Based Restaurant Sales Forecasting," *Mach Learn Knowl Extr*, vol. 4, no. 1, 2022, doi: 10.3390/make4010006.

[5] D. R. Pradiptyo, I. H. Sahid, I. Budi, A. B. Santoso, and P. K. Putra, "Incorporating Stock Prices and Social Media Sentiment for Stock Market Prediction: A Case of Indonesian Banking Company," *Jurnal Nasional Pendidikan Teknik Informatika : JANAPATI*, vol. 13, no. 1, pp. 156–165, Mar. 2024, doi: 10.23887/JANAPATI.V13I1.74486.

[6] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, Sep. 2021, doi: 10.1007/S12525-021-00475-2/TABLES/2.

[7] "Air Conditioner Sales Prediction Using CTGAN, XGBoost and SHAP – IJSREM." Accessed: Jun. 26, 2024. [Online]. Available: https://ijsrem.com/download/air-conditioner-sales-prediction-using-ctgan-xgboost-and-shap/

[8] A. Mitra, A. Jain, A. Kishore, and P. Kumar, "A Comparative Study of Demand Forecasting Models for a Multi-Channel Retail Company: A Novel Hybrid Machine Learning Approach," *Operations Research Forum*, vol. 3, no. 4, Dec. 2022, doi: 10.1007/S43069-022-00166-4.

[9] X. Lu, "A Comparative Study of Machine Learning-Based Regression Models for Supply Chain Management," *Applied and Computational Engineering*, 2024, doi: 10.54254/2755-2721/53/20241233.

[10] K. Seethapathy, "Unlocking Inventory Efficiency: Harnessing Machine Learning for Sales Surge Prediction," *International Journal of Supply Chain and Logistics*, 2024, doi: 10.47941/ijscl.1863.

[11] D. Liu, "Enterprise Digital Retail Business Data Analysis and Forecasting Based on Time Series Analysis," *Advances in Economics Management and Political Sciences*, 2024, doi: 10.54254/2754-1169/77/20241678.

[12] H. Alparslan, "Utilizing Logistic Regression for Analyzing Customer Behavior in an E-Retail Company," *Financial Engineering*, 2024, doi: 10.37394/232032.2024.2.10.

[13] F. Weber and R. Schütte, "A domain-oriented analysis of the impact of machine learning—the case of retailing," *Big Data and Cognitive Computing*, vol. 3, no. 1, 2019, doi: 10.3390/bdcc3010011.

[14] E. Martins and N. V. Galegale, "RETAIL SALES FORECASTING INFORMATION SYSTEMS: COMPARISON BETWEEN TRADITIONAL METHODS AND MACHINE LEARNING ALGORITHMS," in *Proceedings of the 15th IADIS International Conference Information Systems 2022, IS 2022*, 2022. doi: 10.33965/is2022_202201l004.

[15] N. Wu, "Mathematically Improved XGBoost Algorithm for Truck Hoisting Detection in Container Unloading," *Sensors*, 2024, doi: 10.3390/s24030839.

[16] K. Xu, "Predicting housing prices and analyzing real estate markets in the Chicago suburbs using machine learning," *Journal of Student Research*, vol. 11, no. 3, p. undefined-undefined, Aug. 2022, doi: 10.47611/JSRHS.V11I3.3459.

[17] C. Çılgın and H. Gökçen, "Machine learning methods for prediction real estate sales prices in Turkey," *Revista de la Construccion*, vol. 22, no. 1, pp. 163–177, 2023, doi: 10.7764/RDLC.22.1.163.

[18] W. Zhang, C. Wu, H. Zhong, Y. Li, and L. Wang, "Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization," *Geoscience Frontiers*, vol. 12, no. 1, pp. 469–477, Jan. 2021, doi:

10.1016/J.GSF.2020.03.007/PREDICTION_OF_UNDRAINED_SHEAR_STRENGTH_USING_EXTREME_GRADIENT_BOOSTING_AND_RANDOM_FOREST_BASED_ON_BAYESIAN_OPTIMIZATION.PDF.

[19] A. M. Abdi, "Land cover and land use classification performance of machine learning algorithms in a boreal landscape using Sentinel-2 data," *GIsci Remote Sens*, vol. 57, no. 1, pp. 1–20, Jan. 2020, doi: 10.1080/15481603.2019.1650447.

[20] N. Hou *et al.*, "Predicting 30-days mortality for MIMIC-III patients with sepsis-3: a machine learning approach using XGboost," *J Transl Med*, vol. 18, no. 1, Dec. 2020, doi: 10.1186/S12967-020-02620-5.

[21] K. Matuszelański and K. Kopczewska, "Customer Churn in Retail E-Commerce Business: Spatial and Machine Learning Approach," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 17, no. 1, 2022, doi: 10.3390/jtaer17010009.

[22] V. Redkar, "Air Conditioner Sales Prediction Using CTGAN, XGBoost and SHAP," *Interantional Journal of Scientific Research in Engineering and Management*, 2024, doi: 10.55041/ijsrem32201.

[23] A. Mitra, A. Jain, A. Kishore, and P. Kumar, "A Comparative Study of Demand Forecasting Models for a Multi-Channel Retail Company: A Novel Hybrid Machine Learning Approach," *Operations Research Forum*, vol. 3, no. 4, 2022, doi: 10.1007/s43069-022-00166-4.

[24] S. Guo, "Revolutionizing the Used Car Market: Predicting Prices With XGBoost," *Applied and Computational Engineering*, 2024, doi: 10.54254/2755-2721/48/20241349.

[25] F. N. Fitrah Insani, "Optimizing E-Commerce in Indonesia: Ensemble Learning for Predicting Potential Buyers," *Indonesian Journal of Computer Science*, 2024, doi: 10.33022/ijcs.v13i1.3690.

[26] S. Soni, "Performance Evaluation of Multiclass Classification Models for ToN-IoT Network Device Datasets," *Indonesian Journal of Electrical Engineering and Computer Science*, 2024, doi: 10.11591/ijeecs.v35.i1.pp485-493.

[27] K. Li, "A Sales Prediction Method Based on XGBoost Algorithm Model," *BCP Business & Management*, vol. 36, pp. 367–371, Jan. 2023, doi: 10.54691/BCPBM.V36I.3487.

[28] B. M. Pavlyshenko, "Machine-learning models for sales time series forecasting," *Data (Basel)*, vol. 4, no. 1, 2019, doi: 10.3390/data4010015.

[29] Y. F. Akande, J. Idowu, A. Misra, S. Misra, O. N. Akande, and R. Ahuja, "Application of XGBoost Algorithm for Sales Forecasting Using Walmart Dataset," *Lecture Notes in Electrical Engineering*, vol. 881, pp. 147–159, 2022, doi: 10.1007/978-981-19-1111-8_13.

[30] "Optimasi hyperparameter XGBoost-studi kasus prediksi klaim asuransi = Hyperparameter optimization in XGBoost-case study of insurance claim prediction." Accessed: Jan. 09, 2024. [Online]. Available: https://lib.ui.ac.id/detail?id=20509606&lokasi=lokal

[31] M. Ryan Afrizal, R. Adi Nugroho, D. Kartini, R. Herteno, J. Ahmad Yani Km, and K. Selatan, "XGBOOST DENGAN RANDOM SEARCH HYPER-PARAMETER TUNING UNTUK KLASIFIKASI SITUS PHISING".

[32] S. E. Herni Yulianti, O. Soesanto, and Y. Sukmawaty, "Penerapan Metode Extreme Gradient Boosting (XGBOOST) Pada Klasifikasi Nasabah Kartu Kredit," *Journal of Mathematics Theory and Application*, 2022, doi: 10.31605/jomta.v4i1.1792.

[33] A. Laios *et al.*, "Factors Predicting Surgical Effort Using Explainable Artificial Intelligence in Advanced Stage Epithelial Ovarian Cancer," *Cancers (Basel)*, 2022, doi: 10.3390/cancers14143447.

[34] W. Chimphlee, "Hyperparameters Optimization XGBoost for Network Intrusion Detection Using CSE-CIC-IDS 2018 Dataset," *Iaes International Journal of Artificial Intelligence (Ij-Ai)*, 2024, doi: 10.11591/ijai.v13.i1.pp817-826.

[35] J. Xu, "NSGA–III–XGBoost-Based Stochastic Reliability Analysis of Deep Soft Rock Tunnel," *Applied Sciences*, 2024, doi: 10.3390/app14052127.

[36] A. Ardana, "Performance Analysis of XGBoost Algorithm to Determine the Most Optimal Parameters and Features in Predicting Stock Price Movement," *Telematika*, 2023, doi: 10.31315/telematika.v20i1.9329.

[37] O. M. Katipoğlu, "Data Division Effect on Machine Learning Performance for Prediction of Streamflow," *Dümf*

Mühendislik Dergisi, 2022, doi: 10.24012/dumf.1158748.

[38] Q. Wang, "Comparison of Machine Learning Methods for Estimating Leaf Area Index and Aboveground Biomass of Cinnamomum Camphora Based on UAV Multispectral Remote Sensing Data," *Forests*, 2023, doi: 10.3390/f14081688.

[39] L. Gou, "State Reliability of Wind Turbines Based on XGBoost–LSTM and Their Application in Northeast China," *Sustainability*, 2024, doi: 10.3390/su16104099.

[40] Y. Duan, "Forecasting Carbon Price Using Signal Processing Technology and Extreme Gradient Boosting Optimized by the Whale Optimization Algorithm," *Energy Sci Eng*, 2024, doi: 10.1002/ese3.1655.

[41] Y. Ensafi, S. H. Amin, G. Zhang, and B. Shah, "Time-series forecasting of seasonal items sales using machine learning – A comparative analysis," *International Journal of Information Management Data Insights*, vol. 2, no. 1, 2022, doi: 10.1016/j.jjimei.2022.100058.

[42] "Amazon uk SalesForecasting 2019-2021 | Kaggle." Accessed: Jun. 10, 2023. [Online]. Available: https://www.kaggle.com/datasets/revanthkrishnakomali/amazon-uk-salesforecasting-20192021

[43] P. Chowdhury, "Analytical Detection of ' Smart Stock Trading System' Utilizing AI-model," *Interantional Journal of Scientific Research in Engineering and Management*, 2024, doi: 10.55041/ijsrem34829.

[44] K. Abnoosian, "Prediction of Diabetes Disease Using an Ensemble of Machine Learning Multi-Classifier Models," *BMC Bioinformatics*, 2023, doi: 10.1186/s12859-023-05465-z.

[45] Y. Wang, "Research on Space Image Fast Classification Based on Big Data," *Scalable Computing Practice and Experience*, 2023, doi: 10.12694/scpe.v24i3.2423.

[46] M. Miteva, "Preprocessing Techniques for Brain Mri Scans: A Comparative Analysis for Radiogenomics Applications," *Ann. Sofia Univ. Fac. Math. Informat.*, 2023, doi: 10.60063/gsu.fmi.110.111-125.

[47] O. G. Horsa, "Aspect-Based Sentiment Analysis for Afaan Oromoo Movie Reviews Using Machine Learning Techniques," *Applied Computational Intelligence and Soft Computing*, 2023, doi: 10.1155/2023/3462691.

[48] J. M. Ayu, S. Dachi, and P. Sitompul, "Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit," *Jurnal Riset Rumpun Matematika dan Ilmu Pengetahuan Alam (JURRIMIPA)*, vol. 2, no. 2, 2023, doi: 10.55606/jurrimipa.v2i2.1336.

[49] R. Siringoringo, R. Perangin-angin, and M. J. Purba, "SEGMENTASI DAN PERAMALAN PASAR RETAIL MENGGUNAKAN XGBOOST DAN PRINCIPAL COMPONENT ANALYSIS," *METHOMIKA Jurnal Manajemen Informatika dan Komputerisasi Akuntansi*, vol. 5, no. 1, pp. 42–47, Apr. 2021, doi: 10.46880/jmika.Vol5No1.pp42-47.

[50] Y. Song *et al.*, "Spatial prediction of PM2.5 concentration using hyper-parameter optimization XGBoost model in China," *Environ Technol Innov*, vol. 32, Nov. 2023, doi: 10.1016/j.eti.2023.103272.

[51] X. Xiong, X. Guo, P. Zeng, R. Zou, and X. Wang, "A Short-Term Wind Power Forecast Method via XGBoost Hyper-Parameters Optimization," *Front Energy Res*, vol. 10, May 2022, doi: 10.3389/fenrg.2022.905155.

[52] M. Aci and G. A. Doğansoy, "Demand forecasting for e-retail sector using machine learning and deep learning methods," *Journal of the Faculty of Engineering and Architecture of Gazi University*, vol. 37, no. 3, 2022, doi: 10.17341/gazimmfd.944081.

[53] A. Alabrah, "An Improved CCF Detector to Handle the Problem of Class Imbalance with Outlier Normalization Using IQR Method," *Sensors*, vol. 23, no. 9, May 2023, doi: 10.3390/s23094406.

[54] B. Yao, "Walmart Sales Prediction Based on Decision Tree, Random Forest, and K Neighbors Regressor," 2023.

[55] C. CATAL, K. ECE, B. ARSLAN, and A. AKBULUT, "Benchmarking of Regression Algorithms and Time Series Analysis Techniques for Sales Forecasting," *Balkan Journal of Electrical and Computer Engineering*, vol. 7, no. 1, pp. 20–26, Jan. 2019, doi: 10.17694/bajece.494920.