Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: ijeeemi.org; Vol. 7, No. 1, pp. 167-180, February 2025; e-ISSN: 2656-8624.
https://doi.org/10.35882/ijeeemi.v7i1.48

RESEARCH PAPER

OPEN ACCES

Bitcoin Mining Hardware Profitability Prediction Using Categorical Boosting and Extreme Gradient Boosting Algorithms

Dimas Satria Prayoga, Anggraini Puspita Sari, dan Achmad Junaidi

Department of Informatic, Faculty of Computer Science, UPN "Veteran" Jawa Timur, Surabaya, Indonesia

ABSTRACT

Cryptocurrencies, especially Bitcoin, have gained global adoption, with mining activities being one of its most exciting aspects. This is especially important in a context where only a few types of bitcoin mining devices can operate profitably. On the other hand, in the field of machine learning, there are commonly used algorithms, namely Extreme Gradient Boosting (XGBoost), which is known for its performance and speed, and Categorical Boosting (CatBoost), which excels at handling categorical data. This study aims to compare and combine the performance of CatBoost and XGBoost algorithms using the Ridge Regression technique in predicting a case study that is not often encountered, namely predicting the profitability of Bitcoin mining hardware. The main steps include collecting data from reliable sources, pre-processing the data to ensure compatibility, selecting features to select the most relevant features, searching for the best combination of parameters, building a prediction model using a pre-processed dataset, and then training and testing both models to evaluate the accuracy of its predictions and visualization of the results. On the other hand, XGBoost showed an RMSE result of 0.352156 and a MAPE of 17.64%. Meanwhile, CatBoost showed a lower RMSE result of 0.223, with a lower MAPE of 13.94%. Finally, the combined CatBoost-XGBoost with the Ridge Regression technique showed a higher RMSE result of 0.264062 than CatBoost and a MAPE of 12.45%, which was the lowest of the previous two models.

Paper History:
Received Des. 010, 2024
Revised Feb. 10, 2025
Accepted Feb. 15, 2025
Published Feb. 25, 2025

Keywords:
Regression;
Bitcoin mining;
Extreme Gradient Enhancement;
Categorical Improvement.

Contact:
Dimas Satria Prayoga
20081010249@student.upnjatim.ac.id

Anggraini Puspita Sari
anggraini.puspita.if@upnjatim.ac.id

1. INTRODUCTION

The cryptocurrency industry is growing rapidly thanks to blockchain technology, which stores a history of Bitcoin transactions and allows full nodes to determine ownership without human intervention [1][2]. Cryptocurrencies such as Bitcoin are used for online financial transactions and replace trust-based systems with cryptographic proofs, allowing for intermediary transactions [3][4]. Bitcoin transactions are managed by miners, but mining activities are still debated regarding their environmental impact [5]. Mining pools emerged to coordinate the computing power of small-scale miners due to the increasing difficulty of mining [6]. In 2023, the Bitcoin mining network grew by 90% with the hashrate rising by 104% [7]. About 53% of its energy comes from renewable sources, but the exact number of miners is unknown, although it is estimated that there are around two million worldwide [7].

The XGBoost (Extreme Gradient Boosting) algorithm is a distributed gradient boosting library optimized for efficiency, flexibility, and portability [8]. This algorithm works by applying boosting techniques to improve the model's performance [9]. XGBoost builds a decision tree in stages, where each new tree attempts to correct the errors of the previous tree by minimizing the loss function using the gradient descent method. In other words, understanding the mathematical workings behind XGBoost can be challenging for users and researchers without a strong statistical background [9]. In addition to being quite complicated, XGBoost also has limitations because it needs to traverse the dataset during the process of splitting tree nodes, which takes up a lot of computer memory time [10].

On the other hand, the CatBoost Algorithm implements a gradient boosting model, which uses binary decision trees as predictors [11]. CatBoost itself tends to require more time for training and testing datasets that use more computing resources [33]. In addition, CatBoost also tends to require hyperparameter tuning such as iterations (n estimators) which tend to be more to achieve optimal results when compared to XGBoost. However, the effectiveness of the CatBoot algorithm has been shown through its performance which tends to be higher, when compared to Gradient Boosting-based algorithms [33]. This is inseparable from the built-in feature of catboost, namely the catboost encoding label which can easily convert features of categorical and numeric data types.

In previous research [21], a model comparison was conducted to calculate daily river flow in mountainous

Penulis Koresponden: Anggraini Puspita Sari, anggraini.puspita.if@upnjatim.ac.id, Department of Informatic, Faculty of Computer Science, UPN "Veteran" Jawa Timur, Surabaya, Indonesia

areas. In addition, the gradient boosting models used were CatBoost, XGBoost, and LightGBM. All models tested obtained RMSE results in the range of 6.8-7.8 m3s-1. To achieve such results, it would take at least 12 years of training series data Although XGBoost and LightGBM are not the best models for daily river flow prediction, despite their popularity, CatBoost with the default model parameters obtained the best results. But, LightGBM obtained the best RMSE result of 6.8 m3s-1 by optimizing for hyperparameters.

The method in this study begins with data collection and data preprocessing, including data transformation, missing value checking, data sharing based on time range, and label coding with CatBoost. Furthermore, data analysis was carried out through hypothesis tests, correlation analysis, and temporal features. The data was then divided based on the results of the best split train test using the XGBoost and CatBoost models, after which feature selection was carried out using RFE (Recursive Feature Elimination). In the implementation of the model, the best hyperparameters are determined through a manual grid search, followed by cross-validation on the CatBoost and XGBoost models, and comparing with other models. Then, model evaluation includes metric calculations, loss function analysis, and hypothesis testing. Then, the results are visualized through residual diagrams, plots of actual vs predicted values, and plots of prediction of Profitability results.

On the other hand, evaluation metrics such as RMSE are used to measure how much the average error in invariant mass prediction [13], so that RMSE is relevant in mitigating inaccurate predictions of profitability due to inaccurate estimates. Meanwhile, MAPE gives an idea of the magnitude of prediction error as a percentage of the actual value [14], especially when comparing hardware with different scale of profitability. This research combines the performance of CatBoost and XGBoost in predicting the profitability of Bitcoin mining devices. The goal is to determine which algorithm is the most optimal, and then compare it with other algorithms.

2. MATERIALS AND METHODS

A. Categorical Boosting

The Categorical Boosting Algorithm (CatBooost) starts with a dataset $(x_k,\ y_k)$, where, x_k is a vector with m features that represent the characteristics of the data, and y_k is a continuous target (regression). Each pair $(x_k,\ y_k)$ comes from an unknown and independent distribution. The purpose of learning is to acquire the function of F who predicts y By x by minimizing prediction errors, measured using the loss function $L(F)$. CatBoost has the following main processes [11]:

1) The boosting gradient builds the model incrementally by adjusting the predictions to make them more accurate. At every step F^{t-1}, The model is updated like equation (1).
$$F^t = F^{t-1} + ah^t \quad (1)$$
Where h^t is a base predictor (e.g. a decision tree that reduces prediction errors by minimizing function loss.
2) In each iteration, gradient boosting selects a base predictor h^t from the set H which minimizes the loss function as in equation (2).
$$h^t = \underset{h \in H}{\arg min}\, L(F^{t-1} + h) = \underset{h \in H}{\arg min}\, EL\big(y, F^{t-1}(x) + h(x)\big) \quad (2)$$
Equation (2) This function aims to find additional models h^t which, when added to the ensemble model F^{t-1}, will minimize prediction errors (loss function).
3) To minimize errors, Newton's method is used in equation (3) with a second-order approach or negative gradient steps. Both are forms of *functional gradient descent*.
$$g^t(x,y) := \frac{\partial L(y,s)}{\partial s}\Big|_{s=F^{t-1}(x)} \quad (3)$$
In this approach, gradient steps $h^t(x)$ selected to be close to $g^t(x,y)$, where $g^t(x,y)$ is a derivative of the loss function $L(y,s)$. s is the prediction value of the previous model, i.e. $F^{t-1}(x)$.
4) Equation (4) shows how to select an additional model h^t on each iteration in gradient boosting to reduce prediction errors.
$$h^t = \underset{h \in H}{\arg min}\, E(-g^t(x,y) - h(x))^2 \quad (4)$$
Negative gradient $-g^t(x,y)$ indicates the direction of model improvement. Base predictor h^t selected from the set H to approach the negative gradient and Minimizing the difference in squares, so that the model update is more effective.
5) Equation (5) shows how a decision tree can be represented as a mathematical function.
$$h(x) = \sum_{j=1}^{J} b_j\, 1\{x \in R_j\} \quad (5)$$
Decision tree $h(x)$ Divide the data into J Separate Regions R_j (leaf nodes), with each region having a prediction value b_j. Indicator functions $1\{x \in R_j\}$ Determining whether x is in R_j, where the value 1 indicates membership and 0 vice versa.

B. Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is a scalable machine learning system for tree upgrading. XGBoost has been widely recognized in a number of machine learning challenges. XGBoost has the following main processes [12]:

1) Equation (6) describes the decision tree ensemble model used in Gradient Boosting such as XGBoost.
$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), \qquad f_k \in F \quad (6)$$
Where is the prediction $\hat{y}_i$ obtained by adding up contributions K decision tree $f_k(x_i)$. Each tree comes from a function set F and gradually added to improve the accuracy of the model.
2) Equation (7) is an objective function in the XGBoost model, which optimizes the balance between accuracy and complexity of the model.
$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$

Penulis Koresponden: Anggraini Puspita Sari, anggraini.puspita.if@upnjatim.ac.id, Department of Informatic, Faculty of Computer Science, UPN "Veteran" Jawa Timur, Surabaya, Indonesia

Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: ijeeemi.org; Vol. 7, No. 1, pp. 167-180, February 2025; e-ISSN: 2656-8624.
https://doi.org/10.35882/ijeeemi.v7i1.48

$$where\ \Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2 \quad (7)$$

Loss Function $\mathcal{L}(\phi)$ measure the difference between predictions $\hat{y}_i$ and the actual value y_i, Meanwhile, the regulation $\Omega(f_k)$ prevent preventing overfitting with penalties on the number of tree nodes (γT) and model weight $\frac{1}{2}\lambda||w||^2$.

3) Equation (8) is a quadratic approach to the loss function in the gradient boosting algorithm in the iteration to-t.

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n}[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)] + \Omega(f_t) \quad (9)$$

The overall loss value is calculated based on previous losses. Function $f_t(x_i)$ The newly added one aims to improve predictions by reducing errors.

4) Equation (10) calculates the optimal value of leaf weight (w_j^*) in the Gradient Boosting algorithm, specifically for tree-based models such as XGBoost.

$$w_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda} \quad (10)$$

Where g_i, the gradient of the loss function against the previous prediction. h_i Hesian (the second derivative of the loss function) that measures the change in the gradient. I_j and the data index that falls into the leaves to-j.

5) Equation (11) calculates the gain or improvement of separation quality $(\mathcal{L}_{split})$ a node in a Gradient Boosting based decision tree.

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I_I} g_i)^2}{\sum_{i\in I_R} h_I + \lambda}\right] - \gamma \quad (11)$$

The first and second parts represent the increase in score after the separation into two leaves (left I_L dan right I_R), while the third part shows the score before the separation. If $\mathcal{L}_{split}$ Positive and considerable enough, the separation is considered beneficial.

C. Ridge Regression

Ridge regression (also known as L2 regularization) penalizes the function of the least squared by adding the sum of squares of the coefficients [20]. This penalty shrinks the coefficient, reduces variance, and addresses multicollinearity, especially when the predictors are correlated with each other [20]. Koefisien regresi ridge dihitung dengan persamaan (12).

$$\beta_k = [X'X + kI]^{-1} X'Y \quad (12)$$

When $k > 0$, The ridge regression model produces a refractive coefficient but is more stable than OLS. k Value It is usually selected using ridge traces, which are plots of standard ridge coefficients for various k (usually between 0 and 1), to determine the k optimal based on coefficient stability.

D. Root Mean Squared Error (RMSE)

The RMSE, which is the square root of the MSE, provides an error metric that is proportionate to the original scale. The model performs better when the RMSE value is less [34] (12).

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{n}} \quad (12)$$

Where n is the quantity of data (observation), y_t is the actual value (observation), and $\hat{y}_t$ is the regression model's forecast value [21].

E. Mean Absolute Percentage Error (MAPE)

The calculation of average absolute amount of the percentage error between the actual value and the forecast is known as the Mean Absolute Percentage Error, or MAPE [14]. The MAPE formula can be seen in equation (13).

$$MAPE(\%) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{|y_i - \hat{y}_i|}{y_i}\right) \times 100 \quad (13)$$

Where is the amount of data n, y_i is the actual value of the data to-i, dan $\hat{y}_i$ is the predicted value of the data to the MAPE value represented in positive numbers.

F. Proposed Methods

The proposed method is a roadmap that systematically explains how the research will be conducted. The stages of the research are seen in figure 1.

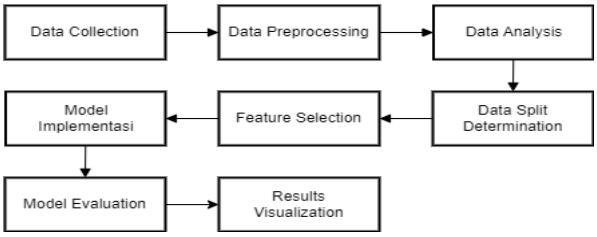

Figure 1. Proposed Methods

The research stage begins with data collection from valid sources, followed by data preprocessing such as handling lost data and normalization. Furthermore, data analysis is carried out to understand the characteristics of the dataset before determining the split data with appropriate proportions. Feature selection is applied to select variables that have an effect in the prediction. In the model implementation stage, the CatBoost and XGBoost algorithms are applied with hyperparameter tuning. The model is evaluated using metrics such as RMSE, MAPE, and R^2, and then the results are visualized to present the model's performance in an informative manner.

G. Data Collection

This study collects data systematically to ensure the quality and relevance of information. The dataset used includes 70 ASIC devices released between April 2020 and October 2024, representing the latest technology in bitcoin mining. Bitcoin mining device data [15], bitcoin market price data [16], and bitcoin network hashrate data [17] are obtained from reliable sources. Data collection was carried out in a span of 60 days, from October 1 to November 29, 2024, with a total of 4200 rows of data. This timeframe was chosen to reflect the latest market trends in bitcoin mining. The features used in this study include 'HardwareModel', 'Manufacturers', 'ReleaseYear', 'Hashrate', 'HashrateDev', 'Power', 'PowerDev', 'Efficiency', 'BitcoinPrice', 'BitcoinHashrate', 'Date',

Penulis Koresponden: Anggraini Puspita Sari, anggraini.puspita.if@upnjatim.ac.id, Department of Informatic, Faculty of Computer Science, UPN "Veteran" Jawa Timur, Surabaya, Indonesia

Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: ijeeemi.org; Vol. 7, No. 1, pp. 167-180, February 2025; e-ISSN: 2656-8624.
https://doi.org/10.35882/ijeeemi.v7i1.48

'Electricity', and 'Income'. An explanation of each feature and its data type will be presented in the data transformation section.

H. Data Pre-processing

The data processing stage is an important stage that is carried out after data collection. The goal is to ensure the quality and consistency of the dataset so that it is ready to be used in modeling. The stages of data preprocessing include data transformation, checking for missing values, sharing data based on time, and encoding labels with catboost.

1) Data Transformation

Data transformation is done by converting data of object or string type into numerical types such as integers or floats. In addition, date features such as those that are still objects are also converted to the datetime data type. The expected data types are as shown in table 1.

Table 1. Features Name, Data Type, and Explanation

Feature Names	Data Types	Explanation
HardwareModel	float64	Bitcoin mining machine name
Manufacturers	float64	Bitcoin mining machine manufacturing name
ReleaseYear	int64	Year of launch of mining device
Hashrate	Int64	Performance level of mining devices
HashrateDev	int64	How consistent the engine is in delivering the hashrate
Power	int64	The power consumed by the mining machine
PowerDev	int64	How consistent the engine is in delivering power
Efficiency	float64	Energy efficiency of Bitcoin mining devices
BitcoinPrice	float64	Bitcoin market price
BitcoinHashrate	float64	The level of competition on the Bitcoin network
Date	datetime64[ns]	Temporal features that represent time
Electricity	float64	Electricity costs measured in USD value
Income	float64	Gross revenue measured in USD value

2) Checking Missing Value

This stage begins with dealing with missing values, as incomplete data can affect the accuracy of the analysis and model. However, because each column in the dataset is complete, no additional steps such as data imputation or deletion are required.

3) Label Encoding (CatBoost)

CatBoost, which stands for "Categorical Boosting" is designed to handle categorical variables effectively. Unlike other methods that require encoders such as Label Encoder or One-Hot Encoder, CatBoost automatically encodes without the need for further preprocessing [18], for the formula as equation (14).

$$\frac{\sum_{j=1}^{P-1}[x_{\sigma p}=x_{\sigma j}]y_{\sigma j}+b\cdot k}{\sum_{j=1}^{P-1}[x_{\sigma p}=x_{\sigma j}]+b} \quad (14)$$

Where $x_{\sigma p}$ and $x_{\sigma j}$ is an independent feature, while $y_{\sigma j}$ is the target value. Indicators $[x_{\sigma p}=x_{\sigma j}]$ worth 1 if it's the same, and 0 if it's different. Variable b serves as a regularization to prevent zero division, with k as an additional constant. The features to be encoded are 'Manufacturers' and 'HardwareModel', here's an example of a manual calculation with the 'Manufacturers' feature, in table 2.

Table 2. Label Coding Features Manufacturers

Original Data (Manufacturers)	Target (Income)	Encoded Value
Bitmain	15.68	12.657
MicroBT	18.37	14.002
Canaan	9.68	9.657
iPollo	3.13	6.382
Ebang	1.31	5.472

4) Data Normalization

The normalization method is carried out, one of which is by using the Min-Max or Min-Max Normalization approach which helps in normalizing data with a data scale between 0 and 1 [22]. The formula is like equation (15).

$$x_{norm}=\frac{x_i-x_{min}}{x_{max}-x_{min}} \quad (15)$$

Where x_i into the range 0 until 1, x_{min} is the minimum value, and x_{max} is the maximum value in the dataset. Here's an example of an advanced manual calculation with the 'Manufacturers' feature from the previous catboost label coding, in table 3.

Table 3. Normalization of Manufacturers Feature Data

Original Data (Manufacturers)	Encoded Value	Normalization Value
Bitmain	12.657	0.8423
MicroBT	14.002	1
Canaan	9.657	0.4906
iPollo	6.382	0.1067
Ebang	5.472	0

I. Exploratory Data Analysis

One method for comprehending and analyzing data is exploratory data analysis, or EDA [23]. EDA aims to uncover patterns and information in data without relying on pre-existing theories [23]. In this study, EDA includes Univariate Graphics, Temporal Feature Analysis, and Correlation Matrix.

1) Univariate Graphics

This study uses the histogram method for univariate analysis. The histogram shows the distribution of data

Penulis Koresponden: Anggraini Puspita Sari, anggraini.puspita.if@upnjatim.ac.id, Department of Informatic, Faculty of Computer Science, UPN "Veteran" Jawa Timur, Surabaya, Indonesia

Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: ijeeemi.org; Vol. 7, No. 1, pp. 167-180, February 2025; e-ISSN: 2656-8624.
https://doi.org/10.35882/ijeeemi.v7i1.48

without gaps between bars or bins, where the bin height represents the frequency, and the extent is directly proportional to the number of cases in a given value range [23]. The features that will be displayed in the correlation matrix are all features except the 'HardwareModel' and 'Date' features.

2) Temporal Analysis
Temporal analysis is applied to data collected at regular intervals to identify trends, patterns, or changes over time [24]. In this study, the analysis was carried out on the 'BitcoinPrice', 'BitcoinHashrate', and 'Income' features.

3) Correlation Matrix
Correlation Matrix with Heatmap is a visual representation of a data, where the values in the matrix are represented as colors [25]. The features that will be displayed in the correlation matrix are all features except the 'Date' feature.

J. Data Split Determination

The dataset is divided by a certain proportion (e.g., 80:20) randomly [26]. However, this approach often ignores the complexity of data structures, such as temporal dependencies and data heterogeneity [26]. Therefore, in this study, data sharing was carried out by date, but still using test data from the same 70 mining devices in 60 different days. The division of data based on the time range can be seen in Table 4, shown by test size.

Table 4. Time-Based Data Sharing

Test Size	Start Train	End Train	Start Test	End Test
0.15	2024-10-01	2024-11-20	2024-11-21	2024-11-29
0.25	2024-10-01	2024-11-14	2024-11-15	2024-11-29
0.35	2024-10-01	2024-11-08	2024-11-09	2024-11-29

The data division based on split was carried out in three different scenarios, namely 0.15, 0.25, and 0.35. Every scenario displays the ratio of test data to all samples. In scenario 0.15, the training data consists of 3,570 samples, while the test data contains 630 samples. In scenario 0.25, the training data includes 3,150 samples, while the test data contains 1050 samples. Finally, in the 0.35 scenario, the training data had 2,730 samples, while the test data consisted of 1,470 samples.

K. Feature Selection

The feature selection method used is a wrapper, which evaluates the predictive performance of various feature combinations by training and testing the model [19]. One popular technique is Recursive Feature Elimination (RFE), which gradually removes less important features to reduce the dataset dimensions while retaining the most informative features [27]. This method also helps assess how accurate the model is in predicting the target 'income' feature and determine the number of features that need to be maintained out of a total of 11 available features.

L. Model Implementation

The model development process includes dataset separation, cross-validation implementation, and hyperparameter tuning to optimize performance. The main models used are CatBoost XGBoost, the Ridge Regression meta-model, and other boosting algorithms to compare prediction accuracy. After optimal training, the model is tested on test data.

1) K-fold Cross-Validation

This method predicts the model and evaluates its correctness in practical applications [28]. K-fold cross-validation is a popular cross-validation technique that divides data into K equal-size subsets [28]. In this study, the k-fold used ranged from 5 to minimize bias in the data and the number of folds that are commonly used in many studies. The K-fold Cross Validation technique can be seen in Figure 2.

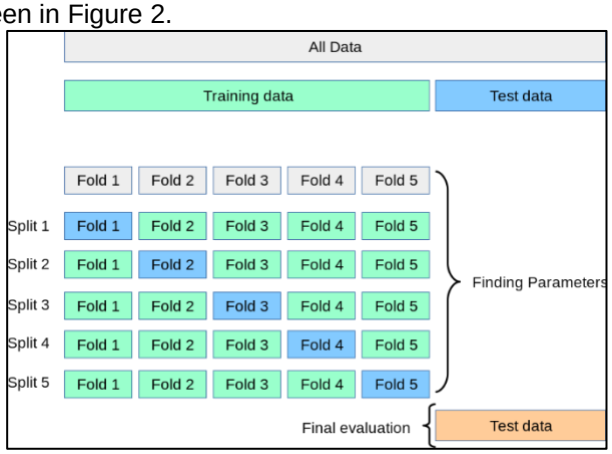

Fig 2. K-fold Cross Validation

2) Tuning Hyperparameter

This study applies hyperparameter tuning using the Grid Search method. Grid search works by trying all combinations of parameters in the grid to find the best configuration [29]. Numerous hyperparameters for XGBoost and CatBoost can impact the model's performance. These hyperparameters consist of the following:

a. Learning rate, the shrinkage of each tree's coefficient to avoid overfitting.
b. Max depth, the maximum value of each tree's depth.
c. N estimators, the number of estimators/iterative computations.

Then various combinations of values from each parameter were systematically tested, and their performance was measured using RMSE and MAPE. The best hyperparameters are selected based on the combination that results in the lowest RMSE.

G. Evaluasi Model

The purpose of model evaluation is to evaluate the model's ability to predict data that has never been seen before [30]. Calculation of evaluation metrics such as RMSE and MAPE, residue analysis, and hypothesis testing are examples of model evaluations that can be used. Statisticians often use the $p\text{-}value$ method to measure the significance of hypothesis test results, with a

Penulis Koresponden: Anggraini Puspita Sari, anggraini.puspita.if@upnjatim.ac.id, Department of Informatic, Faculty of Computer Science, UPN "Veteran" Jawa Timur, Surabaya, Indonesia

Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: ijeeemi.org; Vol. 7, No. 1, pp. 167-180, February 2025; e-ISSN: 2656-8624.
https://doi.org/10.35882/ijeeemi.v7i1.48

general significance level of 0.05. This level of significance depends on the severity of type I errors [31]. The decision rules in the use of $p\text{-}value$ are as follows: if the $p\text{-}value \leq \alpha$, then reject the null hypothesis (H_0), while if the $p\text{-}value > a$, then accept the null hypothesis (H_0) [31]. Specifically, the null hypothesis is accepted, If $p\text{-}value > 0.05$, which shows that the difference is not significant.

3. RESULT

This chapter presents the findings of the study, covering the entire analysis process from Exploratory Data Analysis (EDA) to the Statistical Significance Test with P-value. The results are systematically organized to provide a clear understanding of the data patterns, relationships, and statistical significance observed in the study. The results described in Chapter 2 are revisited and analyzed in greater detail. However, the broader implications, interpretation of the findings, and comparisons with previous research will be explored further in Chapter 4, Discussion.

A. Exploratory Data Analysis

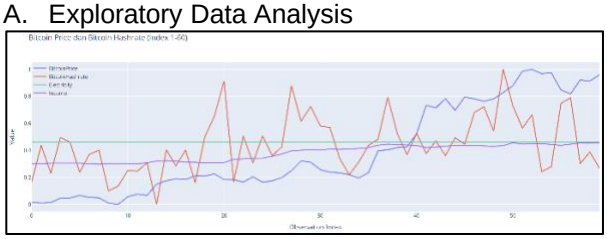

Fig 3. Temporal Features Analysis

Figure 3 shows data on the normalized 'Bitcoin Price', 'Bitcoin Hashrate', 'Electricity', and 'Income' features showing a trend where there is an increase in the 'Bitcoin Price' feature, the 'Bitcoin Hashrate' feature fluctuates, the 'Income' feature rises steadily, and the 'Electricity' feature is constant. The data from these four features is data from the Canaan Avalon A1566 hardware model over a span of 60 days. This analysis was carried out to determine the correlation of different temporal features.

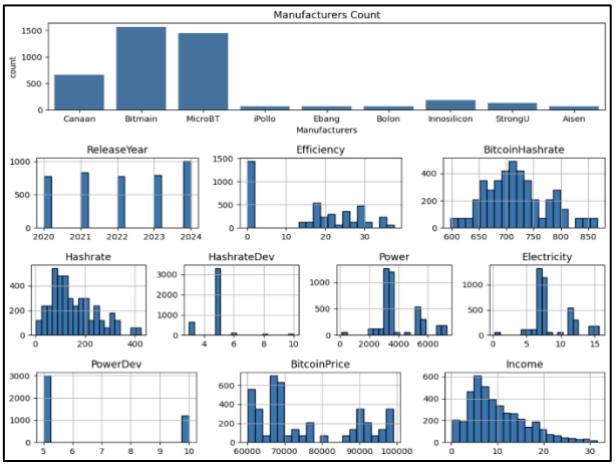

Fig 4. Exploratory Data Analysis

The diagram in figure 4 shows the distribution of various variables related to the bitcoin mining hardware dataset. Manufacturers indicates from which manufacturer a mining device is manufactured. 'ReleaseYear' describes the number of devices released per year, while 'Efficiency' and BitcoinHashrate have varying distributions. 'Hashrate', 'Power', and 'Electricity' show distribution patterns with multiple dominant peaks. 'HashrateDev' and 'PowerDev' have two striking groups of values, while 'BitcoinPrice' and 'Income' show an uneven distribution with a tendency to spike in certain values.

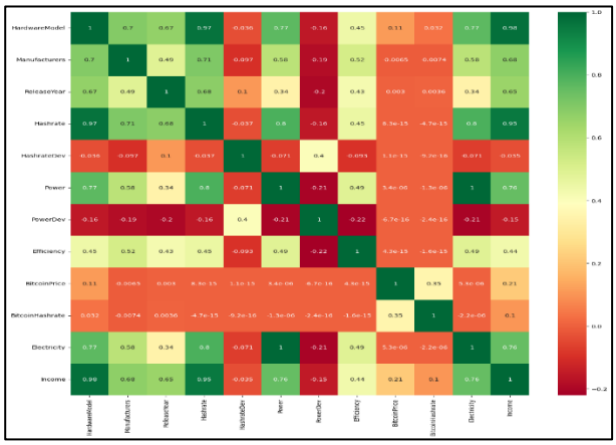

Fig 5. Correlation Matrix Heatmap

In the context of Bitcoin mining devices, a lot of the data presented shows a strong relationship between various factors. The HardwareModel (0.98) and Hashrate (0.95) variables show a very high correlation with Income, indicating that a particular hardware model and high hashrate value have a significant influence on mining revenue and performance. On the other hand, the HashrateDev (-0.035) and Power Dev (-0.15) variables show a weak or even negative correlation with the Income variable. This implies that deviations or variations in hashrate or power consumption do not contribute positively to mining revenue.

B. Data Split Parameter Testing

In the training and testing stage, data sharing is an important factor in determining optimal results according to the methods used. Using three different data sharing schemes, namely 85:15, 75:25, and 65:35. The selection of the scheme aims to evaluate how the proportion of training and testing data affects model performance. So it is hoped that the most optimal proportion of data sharing can be found.

1) Split Data on CatBoost

CatBoost prediction results along with RMSE and MAPE. This process uses the CatBoost model, which was tested with the default parameters of learning rate 0.03, n estimator 1000, and maximum depth of 6.

Table 5. Test results with catboost split data

Model	Split Data	RMSE	MAPE(%)
CatBoost	**85:15**	**0.252939**	**2.13**

Penulis Koresponden: Anggraini Puspita Sari, anggraini.puspita.if@upnjatim.ac.id, Department of Informatic, Faculty of Computer Science, UPN "Veteran" Jawa Timur, Surabaya, Indonesia

CatBoost	75:25	0.390028	2.71
CatBoost	65:35	0.572969	4.61

The results of the metric evaluation in table 5 using the CatBoost model for 85:15 data division show an RMSE value of 0.252939 and a MAPE of 2.13%. Meanwhile, the 75:25 data division yielded RMSE of 0.390028 and MAPE of 2.71%. Meanwhile, the 65:35 data division recorded RMSE of 0.572969 and MAPE of 4.61%. In other words, the best data sharing is the 85:15 scenario for the CatBoost model. This is because the larger the data train, the better the model's performance, and vice versa.

2) Split Data on XGBoost

With the same data sharing scheme, the XGBoost model was tested using the default parameters of learning rate 0.3, n estimator 100, and max depth 6. The default parameters used by XGBoost are like larger learning rates but use smaller n estimators.

Table 6. Test results with xgboost split data

Model	Pembagian Data	RMSE	MAPE(%)
XGBoost	**85:15**	**0.389336**	**2.27**
XGBoost	75:25	0.396582	2.12
CatBoost	65:35	0.733161	4.39

The results of the metric evaluation in table 6 using the CatBoost model for 85:15 data division show an RMSE value of 0.389336 and a MAPE of 2.27%. Meanwhile, the 75:15 data division yielded an RMSE of 0.396582 and a MAPE of 2.12%. Meanwhile, the 65:35 data division recorded RMSE of 0.733161 and MAPE of 4.39%. In other words, the best data sharing is the 85:15 scenario for the XGBoost model. The implications for the XGBoost model are also similar to those for the CatBoost model

C. Feature Selection with RFE

Feature selection with Recursive Feature Elimination (RFE) is a technique to select the best features in a dataframe by eliminating less important features gradually and recursively. In the split data testing stage, CatBoost showed better performance than XGBoost based on RMSE values, so feature selection was carried out using CatBoost. The selection process begins by removing the lowest-ranked features first until the highest-ranked features are reached, as shown in Table 7.

Table 7. Feature Selection with RFE

Ranking Score	RMSE	MAPE(%)
1	0.802325	8.22
2	0.772188	8.66
3	0.287987	2.77
4	0.287522	2.62
5	0.275881	2.54
6	0.276013	2.61
7	0.302102	2.75
8	0.261702	2.55
9	**0.248486**	2.30
10	0.270254	2.12
11	0.252533	2.14

On the other hand, Table 3 shows that the features with ranking scores 1 to 9 are retained because they produce the best RMSE of 0.248486, while the features with ranking scores 10 and 11 namely 'HashrateDev' and 'HahrateDev' are eliminated. The feature importance scores of the nine features maintained based on Figure 6 are 'HardwareModel' (44.72), 'Hashrate' (23.82), 'BitcoinPrice' (14.76), 'Efficiency' (5.77), 'Electricity' (2.60), 'Manufacturers' (2.46), 'ReleaseYear' (2.42), 'BitcoinHashrate' (1.76), and 'Power' (1.68).

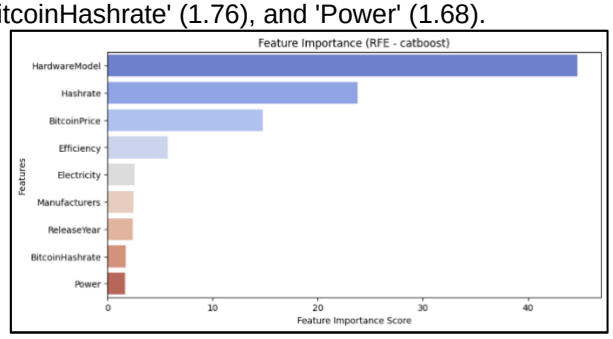

Fig 6. RFE Feature Selection with 9 Features

D. Tuning Hyperparameter

Hyperparameter tuning is carried out using the Manual Grid Search technique using the CatBoost and XGBoost models, with hyperparameters and values namely learning rate, max depth, and n estimators. Each combination of parameters used is calculated individually to obtain the RMSE results shown in Table 8.

Table 8. CatBoost learning rate parameter tuning results

Learning Rate	Max Depth	N Estimators	RMSE	MAPE(%)
0.03	6	1000	0.248503	2.30
0.04	6	1000	0.245979	2.25
0.05	6	1000	0.269071	2.98
0.06	**6**	**1000**	**0.234417**	**1.57**

In table 8, the results of tuning hyperparameters show that the combination of learning rate values affects the performance of the model. Testing with learning rates of 0.03, 0.04, 0.05, and 0.06 resulted in RMSE values of 0.248503, 0.245979, 0.269071, and 0.234417, respectively. The best results were obtained at a learning rate of 0.06, because it obtained the lowest RMSE results and the lowest MAPE. Furthermore, the max depth parameter is adjusted for further optimization.

Table 9. CatBoost max depth parameter tuning results

Learning Rate	Max Depth	N Estimators	RMSE	MAPE(%)
0.06	5	1000	0.240906	1.90
0.06	**6**	**1000**	**0.234417**	**1.57**

Penulis Koresponden: Anggraini Puspita Sari, anggraini.puspita.if@upnjatim.ac.id, Department of Informatic, Faculty of Computer Science, UPN "Veteran" Jawa Timur, Surabaya, Indonesia

0.06	7	1000	0.277363	2.81
0.06	8	1000	0.284938	2.34

In table 9, the results of tuning hyperparameters show that the combination of max depth values in this study has no effect on the performance of the model. Tests with a max depth of 5, 6, 7, and 8 resulted in RMSE values of 0.240906, 0.234417, 0.277363, and 0.284938, respectively. The best results were obtained at max deph 6, This shows that selecting a max depth lower than 6 results in less than optimal performance, while a higher max depth also does not provide a significant performance improvement. Furthermore, the n estimators parameter is adjusted for further optimization.

Table 10. CatBoost n estimators parameter tuning results

Learning Rate	Max Depth	N Estimators	RMSE	MAPE(%)
0.06	6	1000	0.234417	1.57
0.06	6	3000	0.224524	1.43
0.06	6	5000	0.223711	1.40
0.06	**6**	**7000**	**0.223477**	**1.39**

In Table 10, the results of hyperparameter tuning show that the combination of n estimators in this study has no effect on the performance of the model. Testing with n estimators of 1000, 3000, 5000, and 7000 yielded RMSE values of 0.234417, 0.224524, 0.223711, and 0.223477, respectively. The best results were obtained on n 7000 estimators. This is due to the improvement in model performance due to the large number of estimators.

Furthermore, hyperparameters are adjusted with a default learning rate of 0.3 and a max depth of 6, as well as four n estimator scenarios. Adjustments to the number of estimators need to be made because the learning rate and max depth parameters are optimal, even though using the default values, this is expected to affect the performance of the XGBoost model.

Tabel 11. Results of XGBoost parameter tuning n estimators

Learning Rate	Max Depth	N Estimators	RMSE	MAPE(%)
0.3	6	100	0.353223	1.95
0.3	6	200	0.352885	1.93
0.3	6	300	0.352810	1.91
0.3	**6**	**400**	**0.352765**	**1.91**

In table 11, the results of XGBoost hyperparameter tuning show that the combination of n estimators in this study has effect on the performance of the model. Testing with n estimators of 100, 200, 300, and 400 yielded RMSE values of 0.353223, 0.352885, 0.352810, and 0.352765, respectively. The best results were obtained on n 400 estimators. This happens because the more number of estimators, the more the performance of the model also increases.

Moreover, there are several other boosting models that can be used as a comparison besides CatBoost and XGBoost. These models include Adaptive Boosting (AdaBoost) which focuses on the most difficult mistakes. Then, the Histogram-Based Gradient Boosting Regressor (HistGBR) Model excels in its fast processing with Binning Histogram. Finally, the Light Gradient Boosting Machine (LightGBM) model uses a Leaf-wise Growth approach for Maximum Efficiency.

Tabel 12. Comparison model parameter tuning results

Model	Learning Rate	Max Depth	N Estimators
AdaBoost	0.09	9	100
HistogramGBR	0.5	15	400
LightGBM	0.04	5	100

In table 12, the results of this hyperparameter tuning show that each model has its own characteristics and needs to achieve optimal performance.

1. AdaBoost
   Low learning rate (0.09), medium max depth (9), and moderate estimator (100). This configuration makes AdaBoost effective for handling fairly complex data sets while maintaining a balance between bias and variance.
2. HistogramGBR
   High learning rate (0.5), max depth in (15), and many estimators (400). These hyperparameters allow the model to learn efficiently from large data sets.
3. LightGBM
   Low learning rate (0.04), medium max depth (5), and moderate estimator (100). This setup allows LightGBM to provide high-speed training and prediction while maintaining competitive accuracy.

The results of the evaluation of metrics such as RMSE and MAPE will be explained further in chapter 4 of the discussion.

E. Model Validation

After adjusting the hyperparameters on the CatBoost model, it was found that the 85:15 data separation was the best. Through the feature selection process using RFE, 9 features were selected to be maintained. Furthermore, the best hyperparameter tuning results for CatBoost result in a combination of a learning rate of 0.06, a max depth of 6, and n estimators of 7000. As for XGBoost, it produces a combination of learning rate 0.3, max depth 6, and n estimators of 400. The combination of the hyperparameters of the 2 models will also be applied to the CatBoost-XGBoost model for validation. The next step is to test and evaluate the models for the three models, namely CatBoost, XGBoost, CatBoost-XGBoost. The model was then cross-validated using the K-Fold Cross Validation technique with 5 folds, then the model performance was measured using two metrics, namely (RMSE) and (MAPE) and then averaged.

Table 13. CatBoost Model Validation Results

K-Fold	RMSE	MAPE (%)
1	0.128326	0.88%

Penulis Koresponden: Anggraini Puspita Sari, anggraini.puspita.if@upnjatim.ac.id, Department of Informatic, Faculty of Computer Science, UPN "Veteran" Jawa Timur, Surabaya, Indonesia

Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: ijeeemi.org; Vol. 7, No. 1, pp. 167-180, February 2025; e-ISSN: 2656-8624.
https://doi.org/10.35882/ijeeemi.v7i1.48

2	**0.071044**	**0.65%**
3	0.109537	0.88%
4	0.110837	2.41%
5	0.112889	1.01%
Average	**0.106527**	**1.16%**

The results of the CatBoost model validation can be seen in table 13. The evaluation results showed an average RMSE of 0.1065, which indicates how much the model error in the lowest original data unit and MAPE was 1.16%, The best Fold based on RMSE and MAPE was the 2nd Fold (RMSE = 0.071) and MAPE 0.65, while the 1st Fold showed the highest RMSE (0.1283) and the 4th Fold showed the highest MAPE (2.41%). Overall, the CatBoost model shows good performance in terms of RMSE, and the MAPE results are worse than CatBoost.

Table 14. XGBoost Model Validation Results

K-Fold	RMSE	MAPE (%)
1	0.250425	1.10%
2	0.171480	1.13%
3	0.232923	1.33%
4	0.174162	1.82%
5	**0.162346**	**1.01%**
Average	**0.198267**	**1.28%**

Furthermore, the results of the validation of the XGBoost model can be seen in table 14. The evaluation results showed an average RMSE of 0.1983 and a MAPE of 1.28%, which indicates a relatively lower percentage of model error rate than CatBoost. The best fold based on RMSE is the 5th Fold (RMSE = 0.1623) and MAPE by 1.01%, while the 1st Fold shows the highest RMSE (0.2504) and the 4th Fold shows the highest MAPE (1.82%). Overall, the XGBoost model shows worse performance compared to CatBoost, both in terms of RMSE and MAPE averages.

Table 15. CatBoost-XGBoost Model Validation Results

K-Fold	RMSE	MAPE (%)
1	0.102795	0.89%
2	0.112224	1.22%
3	**0.095204**	0.86%
4	0.146725	1.00%
5	0.145977	**0.67%**
Average	**0.120585**	**0.93%**

Furthermore, the results of the validation of the CatBoost-XGBoost combined model can be seen in table 15. The evaluation results showed an average RMSE of 0.120585 and a MAPE of 0.93%, which indicates the lowest percentage of model error rates compared to CatBoost and XGBoost. The best fold based on RMSE is the 3rd Fold (RMSE = 0.095204) and the best Fold based on MAPE is the 5th Fold 0.67%, while the 5th Fold actually shows the highest RMSE (0.145977) and the 2nd Fold showed the highest MAPE (1.22%). Overall, the CatBoost-XGBoost model shows better performance than CatBoost and XGBoost in terms of MAPE average.

F. Model Testing

After passing the model validation stage, the next step is the model testing stage which uses all train data and test data in the dataset. Then, the hyperparameter tuning is obtained from the best scenario obtained from tables 10 and 11 to determine the comparison of its performance with the previously validated models of CatBoost, XGBoost, and CatBoost-XGBoost. The evaluation metrics used are RMSE and MAPE, coupled with execution time. The results of this test will be the basis for determining the most optimal model to predict the 'income' of bitcoin mining devices.

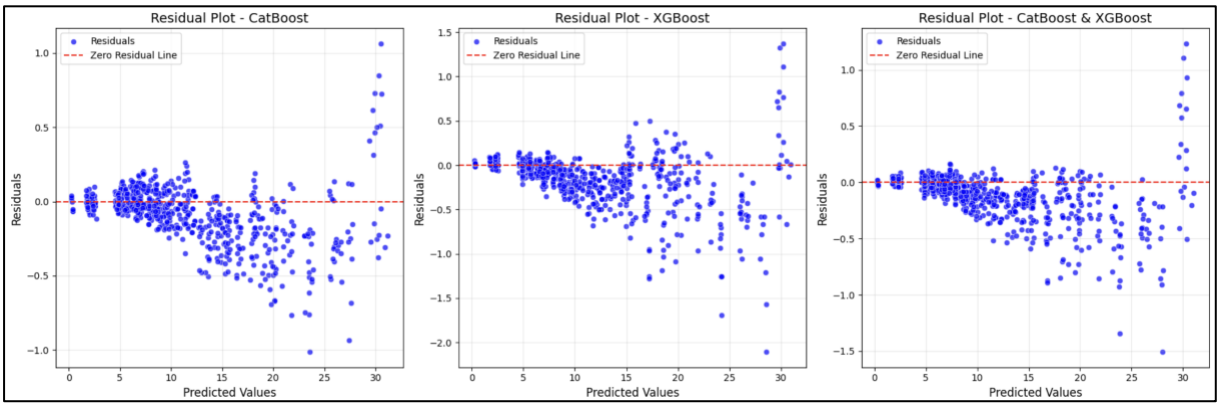

Fig 8. Residual Plot Diagrams for CatBoost, XGBoost, and CatBoost & XGBoost models

Table 16. Test Results of the Main Model

Model	RMSE	MAPE (%)	Runtime (s)
CatBoost	**0.223475**	1.39	14.12
XGBoost	0.352765	1.91	**2.82**
CatBoost-XGBoost	0.264659	**1.37**	24.77

The test results in Table 16 show that the CatBoost model has the best performance in terms of RMSE, with a value of 0.223475. This is due to the adjustment of the

Penulis Koresponden: Anggraini Puspita Sari, anggraini.puspita.if@upnjatim.ac.id, Department of Informatic, Faculty of Computer Science, UPN "Veteran" Jawa Timur, Surabaya, Indonesia

Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: ijeeemi.org; Vol. 7, No. 1, pp. 167-180, February 2025; e-ISSN: 2656-8624.
https://doi.org/10.35882/ijeeemi.v7i1.48

hyperparameter learning rate, max depth, and the number of estimators that are already optimal. The CatBoost-XGBoost model ranks second with an RMSE of 0.264659, while XGBoost comes in third with an RMSE of 0.352765. In other words, CatBoost is the best choice for high accuracy (especially in RMSE) with acceptable execution times. On the other hand, the CatBoost-XGBoost model obtained the lowest MAPE value, which is 1.37%, which shows its superiority in absolute percentage error rate compared to CatBoost and XGBoost, which have MAPE values of 1.39% and 1.91%. This is due to the setting of the alpha parameter of 200, which serves as a regularization to control the complexity of the model in the final estimator of Ridge. CatBoost-XGBoost is able to offer the best accuracy (especially in MAPE) but with longer execution times, suitable for tasks that prioritize precision. Respectively. Finally, the XGBoost model shows the fastest execution time, which is 2.82 seconds, making it the most efficient model in terms of processing time, This is due to the use of a relatively small number of estimators in the XGBoost model, which is as many as 400. XGBoost is ideal for applications that require speed and efficiency.

G. Plotting Results

The first is to display the loss diagram of the two main models, namely CatBoost and XGBoost, which can be seen in figure 7.

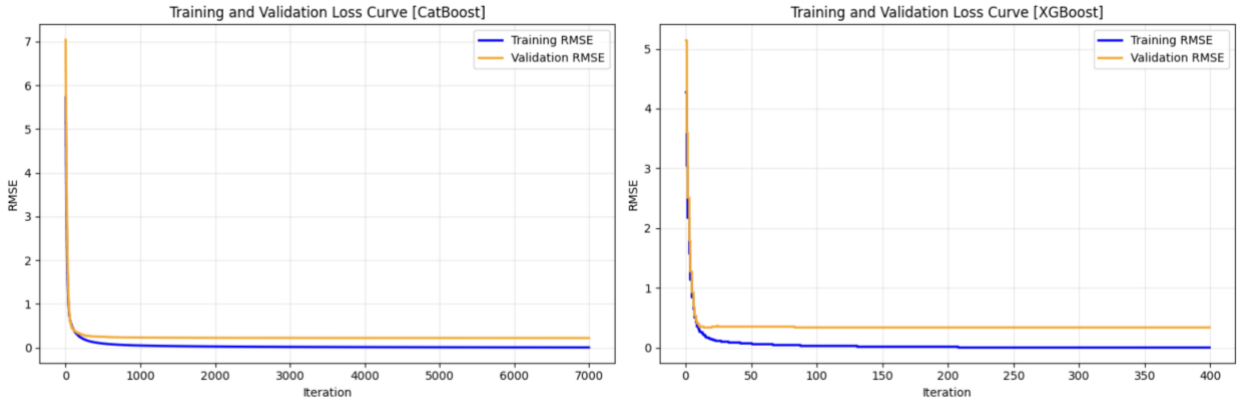

Fig 7. Training and Validation Loss Curve for CatBoost & XGBoost models

The first loss curve, CatBoost, shows that the RMSE on training and validation has decreased significantly in the iteration range below 500. After that, the decline becomes less significant, so the curve begins to slope. The second loss curve, XGBoost, which represents the other models, shows a similar pattern to CatBoost, where RMSE experienced a sharp decline in the initial iteration. However, after reaching the 50th iteration, both RMSE training and validation began to slope.

The next is to display the residual diagrams of the three models that have been tested to find out the distribution of the residue along with the predicted value with. The residual plot as shown in figure 8 of the CatBoost model, XGBoost, and a combination of both show slightly different characteristics in error propagation. In the CatBoost model, the residual tends to be scattered around the zero value line, but there is a certain pattern at predicted values greater than 20 that indicate a residual value greater than 1 and greater than -1. Meanwhile, the XGBoost plot residuals show a more concentrated residual distribution around the zero line, although there is a tendency to overestimate the smaller prediction values. The combination of CatBoost and XGBoost results in a more even distribution of residual than each model individually, although there are still some extreme residuals at high prediction values. Next, the results are plotted to display the actual and predicted values of the three models.

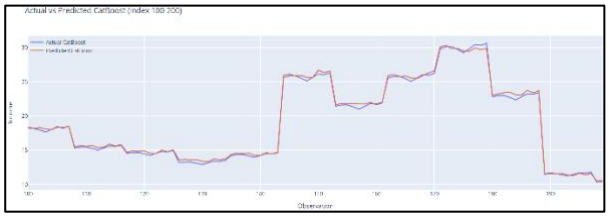

Fig 9. CatBoost Model Prediction Chart

The CatBoost model performs well in Figure 9, with predictions that closely match the actual values, especially at significant turning points. The model's prediction line almost always intersects with the actual data line, showing that the model does a good job of capturing complex patterns in the data. This shows that CatBoost has an advantage in understanding variations and trends in data, thus providing more accurate and stable prediction results.

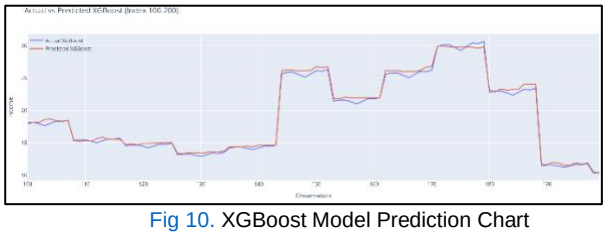

Fig 10. XGBoost Model Prediction Chart

Meanwhile, in Fig. 10, the XGBoost model also shows quite accurate predictions with a pattern that is almost similar to the CatBoost model. However, it can be seen that at some points, XGBoost's predictions deviate slightly from the actual values, especially on sharp changes in the data.

Penulis Koresponden: Anggraini Puspita Sari, anggraini.puspita.if@upnjatim.ac.id, Department of Informatic, Faculty of Computer Science, UPN "Veteran" Jawa Timur, Surabaya, Indonesia

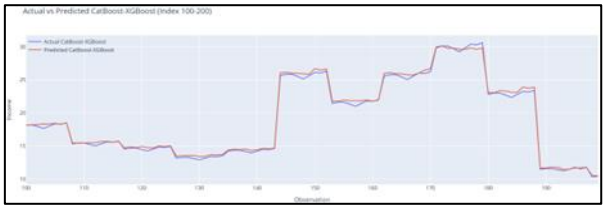

Fig 11. CatBoost-XGBoost Model Prediction Chart

Based on the plot of the prediction results in figure 11 with the CatBoost-XGBoost model, It demonstrates that the XGBoost model's prediction results do not differ much. Overall, both models provide good performance, but CatBoost seems to have a slight edge in capturing more complex patterns of change in the data.

H. Statistical Significance Test with p-value

Dalam analisis ini, pendekatan statistik digunakan untuk mengevaluasi kinerja model secara objektif. This process involves calculating the p-value based on the results of statistical tests applied to previous model performance evaluation metrics as shown in table 17.

Table 17. Significance Test Results of the Three Models

Model1	Model2	T-test Statistic	P-value
CatBoost	XGBoost	-0.197083	0.843794
CatBoost	CaBoost-XGBoost	-0.094447	0.924769
XGBoost	XGBoost-CatBoost	0.102719	0.918203

The results of the t-test on XGBoost, CatBoost, and a comparison of the two (XGBoost-CatBoost) showed that there was no significant evidence that the residuals of these models differed from zero. XGBoost dan CatBoost memberikan hasil yang sangat mirip, karena nilai T-test Statistic sangat kecil.

The P-values obtained for all three analyses (0.843794 for XGBoost, 0.924769 for CatBoost, and 0.918203 for XGBoost-CatBoost) is greater than 0.05, which means all three fail to reject the null hypothesis. In other words, there is no significant prediction bias in these models, and they both show fairly similar performance without any noticeable differences in terms of prediction.

4. DISCUSSION

In this discussion part, first, there will be a comparison of the results of the evaluation of the metric with several previous studies whose topic or discussion is close to this research. Then the test results were compared with other models such as AdaBoost, GradientBR, HistGBR, and LightGBM. Finally, predict the profitability value with the help of the Income feature that gets the best results, namely using the catboost model.

A. Comparison of Previous Research

Understanding how the current findings compare to previous research [32] who predicted price movements for 2 simple types of crypto assets, namely BTC and ETH using the XGBoost and CatBoost models, and then compared to this researcher who used an 'income' case study. The study's findings in Table 18 demonstrate the model's benefits over earlier studies.

Table 18. Comparison of MAPE Previous Research

Reference	Case Study	Model	MAPE (%)
Rosa et al. [32]	BTC	XGBoost	16
Rosa et al. [32]	ETH	XGBoost	7
Rosa et al. [32]	BTC	CatBoost	15
Rosa et al. [32]	ETH	CatBoost	7
This work	Income	CatBoost	1.39
This work	Income	XGBoost	1.91
This work	Income	CatBoost - XGBoost	**1.37**

When compared to Rosa et al. [32's research], the XGBoost model in predicting BTC revenue has a MAPE of 16%, while for ETH it is 7%. Meanwhile, the CatBoost model registers a MAPE of 15% for BTC and 7% for ETH. These results show that the model applied in this study has a lower prediction error rate compared to the previous approach.

B. Results of the Evaluation of the Income

The discussion about model evaluation was carried out by comparing seven models that have gone through hyperparameter tuning in Chapter 3, hyperparameter tuning section. Comparison of model performance is carried out using RMSE, MAPE, and runtime metrics.

Table 19. Results of the Evaluation of the Income Feature Model

Model	RMSE	MAPE (%)	Runtime (s)
AdaBoost	0.359283	1.72	2.60
HistGBR	0.344610	2.39	1.90
LightGBM	0.350972	2.56	**0.49**
CatBoost	**0.223475**	1.39	14.12
XGBoost	0.352765	1.91	2.82
CatBoost-XGBoost	0.264659	**1.37**	24.77

Considering the model evaluation findings displayed in Table 19, in terms of Root Mean Squared Error (RMSE), the CatBoost model has the best performance with a value of 0.223475, which indicates that the prediction of this model is more accurate in approaching the actual value than other models. The CatBoost-XGBoost model comes in second with an RMSE of 0.264659, while XGBoost has the highest RMSE among the boosting models used, which is 0.352765.

Penulis Koresponden: Anggraini Puspita Sari, anggraini.puspita.if@upnjatim.ac.id, Department of Informatic, Faculty of Computer Science, UPN "Veteran" Jawa Timur, Surabaya, Indonesia

Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: ijeeemi.org; Vol. 7, No. 1, pp. 167-180, February 2025; e-ISSN: 2656-8624.
https://doi.org/10.35882/ijeeemi.v7i1.48

However, with a MAPE of 1.37%, the CatBoost-XGBoost Model performs best in terms of absolute percentage error rate. This model outperforms CatBoost and XGBoost which have a MAPE of 1.39% and 1.91%, respectively. Meanwhile, the Gradient Boosting Regressor (GradientBR) model has the highest MAPE value, which is 3.39%, which indicates that this model is less accurate than other models. In addition, the AdaBoost, HistGBR, and LightGBM models recorded MAPE values of 1.69%, 2.39%, and 2.52%, respectively, which are still higher than CatBoost, XGBoost, and CatBoost-XGBoost.
However, in terms of execution time efficiency, LightGBM became the fastest model with just 0.12 seconds, followed by XGBoost which took 0.79 seconds. In contrast, the CatBoost and CatBoost-XGBoost models require longer execution times, 14.56 seconds and 16.07 seconds, respectively. Thus, the CatBoost model provides the best balance between accuracy and prediction error, although it requires a higher processing time than other models.

C. Profitability Feature Prediction

In this section, after predicting the target feature 'Income', the next step is to calculate the prediction of the 'Profitability' feature. This calculation is carried out by subtracting the prediction value of 'Income' from the value of the 'ElectricityCost' feature, so that a new target feature is obtained, namely 'ProfitabilityPredict'. The results of this prediction are then compared with the actual target data for the 'Profitability' feature. Next, a metric evaluation was performed to measure the performance of the previous three main models as seen in table 20.

Tabel 20. Profitability Feature Prediction

Model	RMSE	MAPE (%)
CatBoost	**0.223**	13.94
XGBoost	0.352156	17.64
CatBoost-XGBoost	0.264062	**12.45**

The test results in Table 20 show that the profitability prediction of the CatBoost model has the best performance in terms of RMSE, with a value of 0.223. The CatBoost-XGBoost model ranks second with an RMSE of 0.264062, while XGBoost ranks third with an RMSE of 0.352156. On the other hand, the CatBoost-XGBoost model obtained the lowest MAPE value, which is 12.45%, which shows its superiority in absolute percentage error rate compared to CatBoost and XGBoost, which have MAPE values of 13.94% and 17.64%, respectively. In this part, MAPE is higher because the data range of the profitability feature tends to be smaller than the data range of the 'Income' feature. The three comparator models (AdaBoost, HistGBR, and LightGBM) in the evaluation of the target income feature were not included in the profitability prediction because they were not part of the three main models. The three models tested in table 20 can be said to have entered the good forecasting criteria because they are in the range of 10%-20%.

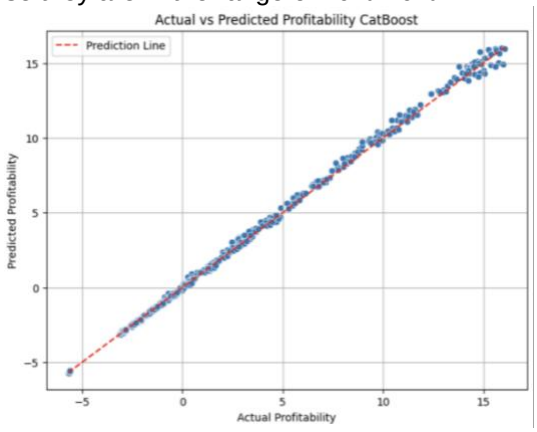

Figure 12. CatBoost Actual Graphs and Profitability Predictions

In figure 12, the pattern of points parallel and intersecting with the red line indicates that the CatBoost model has excellent prediction performance, with minimal error. The spread of points very close to the line indicates that the model has low prediction error and can accurately capture data patterns.

5. CONCLUSION

This study evaluates the performance of the CatBoost and XGBoost models in predicting the income of Bitcoin mining devices using various parameters, such as learning rate, max depth, and n estimators. The test results show that CatBoost with a learning rate of 0.06, max depth 6, and 7000 estimators provides the best performance with an RMSE of 0.223477 and a MAPE of 1.39%. Meanwhile, XGBoost with a learning rate of 0.3, max depth of 6, and 400 estimators has a higher RMSE, which is 0.352765, with a MAPE of 1.91%. In the profitability prediction, the CatBoost-XGBoost model shows the best results with an RMSE of 0.264062 and a MAPE of 12.45%, followed by CatBoost with an RMSE of 0.223 and a MAPE of 13.94%, and an XGBoost with an RMSE of 0.352156 and a MAPE of 17.64%. These findings demonstrate that both the CatBoost and CatBoost-XGBoost combination models can produce predictions that are more accurate than those of the XGBoost model by itself. In addition, the study showed improved accuracy compared to previous studies, such as the study of Rosa et al. [32] who obtained a 15% MAPE for BTC price prediction using CatBoost and 16% with XGBoost. Despite this study's findings demonstrating the benefits of the model the, there are some limitations, especially in the limited amount of data on 70 mining devices over 60 days. Therefore, for further research, it is recommended to expand the scope of the data by using the latest mining tools as well as trying other models outside the algorithms of the ensemble learning group. In

Penulis Koresponden: Anggraini Puspita Sari, anggraini.puspita.if@upnjatim.ac.id, Department of Informatic, Faculty of Computer Science, UPN "Veteran" Jawa Timur, Surabaya, Indonesia

addition, it is also necessary to test the model under different market conditions to assess the stability and reliability of the model in the face of Bitcoin price volatility as well as changes in operational costs, such as electricity prices and mining difficulties.

REFERENCE

[1] A. M. Antonopoulos and D. A. Harding, *Mastering Bitcoin*. "O'Reilly Media, Inc.," 2023.
[2] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," Business & Information Systems Engineering, vol. 59, no. 3, pp. 183–187, Mar. 2017.
[3] F. Adzim, A. Harakan, M. Z. U. Haq, and A. Syakur, "Pendampingan Proses Penambangan Mata Uang Digital Untuk Pemuda di Kota Makassar," *www.pusdig.my.id*, Dec. 2021.
[4] C. Tahir, G. Airlangga, dan K. Hendrawan, Bitcoin: Sebuah Sistem Uang Tunai Elektronik Peer-to-Peer.
[5] M. Roeck and T. Drennen, "Life cycle assessment of behind-the-meter Bitcoin mining at US power plant," The International Journal of Life Cycle Assessment, vol. 27, no. 3, pp. 355–365, Feb. 2022.
[6] I. Eyal and E. G. Sirer, "Majority is not enough," Communications of the ACM, vol. 61, no. 7, pp. 95–102, Jun. 2018.
[7] CoinShares, "The Halving and its impact on hash rate and miners' cost structures," 2024 Mining Report, 2024. [Online]. Available: https://coinshares.com/research/2024-mining-report. [Accessed: Jan. 17, 2025].
[8] "XGBoost Documentation," XGBoost, 2025. [Online]. Available:https://xgboost.readthedocs.io/en/latest/index.html. [Accessed: Mar. 10, 2025].
[9] S. Ajgaonkar, Practical Automated machine learning using H2O.ai: Discover the power of automated machine learning, from experimentation through to deployment to production. Packt Publishing Ltd, 2022.
[10] D. Saputro and D. W. Utomo, "Rekomendasi Produk E-commerce Berbasis Klasifikasi Ulasan Menggunakan Ensemble Random Forest dan Teknik Boosting," *Infotekmesin*, vol. 15, no. 2, pp. 390–396, 2024.
[11] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," Neural Information Processing Systems, vol. 31, pp. 6639–6649, Dec. 2018.
[12] T. Chen and C. Guestrin, "XGBoost," XGBoost: A Scalable Tree Boosting System, pp. 785–794, Aug. 2016, doi: 10.1145/2939672.2939785.
[13] M. S. W. S PhD, Fisika Komputasi Berbasis Machine Learning dengan Pemrograman Python. Bolabot, 2024.
[14] L. F. M. H. S. Kom. MKom, A. P. S. St. MT, Y. A. St. Mmsi, and Noviyanti. P. S. Kom. MKom, EKSPLORASI MACHINE LEARNING DENGAN SCIKIT-LEARN Strategi belajar Machine Learning. Uwais Inspirasi Indonesia, 2024.
[15] ASIC Miner Value. (n.d.). ASIC Miner Value. [Online]. Available: https://www.asicminervalue.com/ [Accessed: Mar. 2, 2025].
[16] CoinMarketCap. (n.d.). Historical Data. [Online]. Available: https://coinmarketcap.com/historical/ [Accessed: Mar. 2, 2025].
[17] CoinWarz. (n.d.). Bitcoin Hashrate Chart. [Online]. Available: https://www.coinwarz.com/mining/bitcoin/hashrate-chart [Accessed: Mar. 2, 2025].
[18] K. K. Singh, M. K. Bajpai, and A. S. Akbari, Machine vision and augmented Intelligence: Select Proceedings of MAI 2022. Springer Nature, 2023.
[19] I. A. E. Zaeni, Penerapan Machine Learning Pada Embedded System (Machine Learning untuk Teknik Elektronika. Media Nusa Creative (MNC Publishing), 2025.
[20] Educohack Press. (2025). Introduction to Robotics. [n.p.].
[21] R. Szczepanek, "Daily streamflow forecasting in mountainous catchment using XGBoost, LightGBM and CatBoost," Hydrology, vol. 9, no. 12, p. 226, Dec. 2022, doi: 10.3390/hydrology9120226.
[22] B. A. D. S. Fairuz, Panduan praktis Machine Learning klasifikasi menggunakan Python: Diandra Kreatif. Diandra Kreatif, 2024.
[23] F. Elfaladonna, I. G. T. Isa, D. Sartika, Yusniarti, and A. M. Putra, Buku ajar Dasar Exploratory Data Analysis (EDA). Penerbit NEM, 2024.
[24] F. Sarie et al., Metodelogi penelitian. Cendikia Mulia Mandiri, 2023.
[25] M. Al-Husaini, P. A. Saputra, M. Renaldi, and R. A. Maulana, Prediksi tsunami dengan metode Ensemble Machine learning. PT. Sonpedia Publishing Indonesia, 2024.
[26] J. Tan, J. Yang, S. Wu, G. Chen, and J. Zhao, "A critical look at the current train/test split in machine learning," arXiv (Cornell University), Jan. 2021, doi: 10.48550/arxiv.2106.04525.
[27] N. Sutarman, R. Siringoringo, D. Arisandi, E. Kurniawan, and E. B. Nababan, "Model klasifikasi dengan logistic regression dan recursive feature elimination pada data tidak seimbang," Jurnal Teknologi Informasi Dan Ilmu Komputer, vol. 11, no. 4, pp. 735–742, Aug. 2024, doi: 10.25126/jtiik.1148198.
[28] H. Azis, P. Purnawansyah, F. Fattah, and I. P. Putri, "Performa Klasifikasi K-NN dan Cross Validation Pada Data Pasien Pengidap Penyakit Jantung," ILKOM Jurnal Ilmiah, vol. 12, no. 2, pp. 81–86, Aug. 2020, doi: 10.33096/ilkom.v12i2.507.81-86.
[29] M. S. W. S PhD, Algoritma Lebah untuk Bidang Robotika Berbasis Pemrograman Python. Bolabot, 2024.
[30] L. Afuan and R. Isnanto, Machine learning. Zahira Media Publisher, 2024.
[31] P. Pandriadi et al., *Statistika Dasar*. Penerbit Widina, 2023.
[32] P. De Rosa, P. Felber, and V. Schiavoni, "Practical forecasting of cryptocoins timeseries using correlation patterns," in *Proceedings of the 17th ACM International Conference on Distributed and Event-Based Systems*, 2023, pp. 80–90.
[33] M. R. Supriadi and R. Andarsyah, DETEKSI HALAMAN WEBSITE PHISHING MENGGUNAKAN ALGORITMA MACHINE LEARNING GRADIENT BOOSTING CLASSIFIER. Penerbit Buku Pedia, 2023.
[34] A. A. A. G. H. Wadjdi Achmad Farid, Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) Aplikasi Penginderaan Jauh untuk Kelautan & Perikanan Laut Tangkap. CV Jejak (Jejak Publisher), 2024.

BIOGRAFI PENULIS

Dimas Satria Prayoga is a student of the Faculty of Computer Science, Informatics Study Program, National Development University "Veteran" East Java, Surabaya, Indonesia, with an interest in machine learning (ML) and Database Systems.

Anggraini Puspita Sari received her B.E.and M.E. degrees from Universitas Brawijaya, Malang, Indonesia in 2009 and 2012 respectively. She received Dr. Eng. degree from Tokushima University, Tokushima, Japan in 2021. She is a Assistant Professor in Informatics, Universitas Pembangunan Nasional Veteran Jawa Timur, Surabaya, Indonesia. Her current research interest includes forecasting, wind power, artificial intelligent, microelectronic, and Electrical Engineering. She is a member of the Electrical Engineering Education Forum Indonesia (Fortei Indonesia), a member of the Institute of Electrical and Electronics Engineers (IEEE), and a member of the Institution of Engineers Indonesia.

Penulis Koresponden: Anggraini Puspita Sari, anggraini.puspita.if@upnjatim.ac.id, Department of Informatic, Faculty of Computer Science, UPN "Veteran" Jawa Timur, Surabaya, Indonesia

Achmad Junaidi meraih gelar Bachelor of Computer Science, Institute Technology Sepuluh Nopember Surabaya. Beliau juga meraih gelar Master of Computer Science, di STMIK Eresha Jakarta. Beliau merupakan Asisten Ahli di bidang Informatika, Universitas Pembangunan Nasional Veteran Jawa Timur, Surabaya, Indonesia. Minat penelitiannya saat ini meliputi Computer Network, Network Security, Human Computer Interaction, and Digital Image Processing. Beliau melakukan riset Design of Computer Network Infrastructure In UPN "Veteran" Jatim Using Dynamic Routing OSPF.

Penulis Koresponden: Anggraini Puspita Sari, anggraini.puspita.if@upnjatim.ac.id, Department of Informatic, Faculty of Computer Science, UPN "Veteran" Jawa Timur, Surabaya, Indonesia