Content list available at JournalPark

Turkish Journal of Forecasting

Bitcoin Trend Reversal Prediction with Tree-Based Ensemble Machine Learning

S. Urgenc[1,*], B. Asikgil[2]

[1] *Mimar Sinan Fine Arts University, Institute of Science, M.S. Program in Statistics, Istanbul, Turkey*
[2]*Mimar Sinan Fine Arts University, Faculty of Science and Letters, Department of Statistics, Istanbul, Turkey*

A R T I C L E I N F O

Article history:

Received: 15 November 2023
Revised: 01 March 2024
Accepted: 27 March 2024

Keywords:

Bitcoin trend prediction
Classification
Cryptocurrency price analysis
Machine learning
Tree-based algorithms

A B S T R A C T

In recent years, Bitcoin has become the most popular digital asset in the cryptocurrency market. Its prices are highly volatile due to rapidly increasing investor interest, making it difficult to predict price movements. Machine learning models can be developed to predict future price movements using time-based data and financial indicators. Bitcoin price movements can be influenced from many factors in the cryptocurrency market. Therefore, it is important to determine the accuracy and success of machine learning techniques. The aim of this study is to predict trend reversals in Bitcoin price movements by using tree-based ensemble machine learning techniques and compare the success rates of these techniques. In this paper, unlike other studies, the focus is not on the movement of the prices the next day, but rather on whether the trend will reverse or not. 'Buy', 'sell' and 'hold' classes are labelled depending on the trend reversals and the classes are balanced by undersampling method. Extreme Gradient Boosting, Random Forest and Random Trees models are developed. The results are evaluated by using precision, recall, specificity, F1 score and accuracy metrics. The study concludes that the Extreme Gradient Boosting model exhibits higher success compared to other models.

1. Introduction

Processing and analyzing data play a significant role in many industries today. Financial markets are one of these sectors, and with the increasing digitization in recent years, data analysis and machine learning techniques are being used more frequently in the management of financial markets. Cryptocurrencies are also a part of the digitization trend in financial markets. Bitcoin (BTC) has gained prominence among the increasingly popular cryptocurrencies in recent years. However, due to the highly volatile nature of BTC prices, investors are seeking various methods to predict BTC trend reversals [1].

When the literature is reviewed, it is evident that there is a growing research area focusing on cryptocurrencies. In the conducted research, deep learning models like Long Short-Term Memory (LSTM) and Recurrent Neural Network (RNN) are widely used. Classification studies for predicting the direction of BTC prices are much less compared to regression studies. Classification studies mainly focus on predicting the price direction for the next period and rarely cover trend directions and reversals.

İnce [2] made a study using by a total of 156 technical indicators, mathematical transformations, and financial patterns were used in the feature set with the aim of testing the 'buy,' 'sell,' and 'hold' classes created using the one-

[*] Corresponding author.
E-mail addresses: sergul.urgenc@gmail.com (Sergul Urgenc), baris.asikgil@msgsu.edu.tr (Baris Asikgil)

way transaction commission rate against opening, closing, highest, lowest, and volume data. In the study, LSTM, Deep Neural Network (DNN), and Gated Recurrent Unit (GRU) models were developed, and their performances were compared. It was observed that all deep learning models yielded more successful results when developed using technical indicators. LSTM emerged as the most successful one among the three models with an overall accuracy of 56.33%. In the study conducted by Qiang and Shen [3], classification algorithms were employed to make high-frequency trend predictions for BTC prices using minute-level technical indicators. This study aimed to create the most suitable BTC trading strategies for investors by capturing signals from historical data, so in addition to prediction accuracy, widely used financial metrics such as net asset value (NAV) and Sharpe ratio were used. In this context, a hybrid deep learning model incorporating Convolutional Neural Network (CNN) and LSTM were developed to predict price ups and downs. By calculating NAV and sharpe ratios with different hyperparameters, the hybrid model achieved an accuracy of 53%. Cavalli and Amoretti [4] aimed to predict the daily ups and downs of BTC prices by utilizing social media data, blockchain transaction data, and financial indicators. They developed a One-Dimensional Convolutional Neural Network (1D CNN) model, and the results were compared with CNN, LSTM, and standard machine learning algorithms. The 1D CNN model demonstrated high performance with an accuracy rate of 74.2%. A. Monsalve et al. [5] investigated the suitability of a CNN model in their study instead of Multilayer Perceptron (MLP). The experimental part of the study compared the performance of different neural network architectures in predicting whether the value of six popular cryptocurrencies (Bitcoin, Dash, Ethereum, Litecoin, Monero, and Ripple) against the United States Dollar (USD) would increase in the next minute using high-frequency technical analysis. CNN models developed with a total of 18 technical indicators like Relative Strength Index (RSI), Moving Average Convergence-Divergence (MACD), and Simple Moving Average (SMA) significantly outperformed MLP models. Livieris et al. [6] employed both regression and classification methods to predict the prices and price movements of Bitcoin, Ethereum, and Ripple. In the study, LSTM, Bidirectional LSTM (BiLSTM), and CNN models were combined with various commonly used ensemble learning strategies. The proposed ensemble models incorporate combinations of the deep learning models used. The results demonstrated that combining deep learning models with ensemble learning methods improved prediction accuracy compared to using a single deep learning model alone. Cohen [7] examined the capabilities of two different methods used to predict BTC price trends. Particle Swarm Optimization (PSO) was used to find the best prediction combinations. Linear Regression (LR) and Darvas Box method were used for prediction using measures such as minimum and maximum drawdown, the percentage of profitable trades, profit factor, and net profit. Based on this price prediction, price ups and downs were calculated. The results indicated that BTC price changes do not follow the efficient market hypothesis effectively and that both the Darvas Box and LR methods can be helpful in predicting BTC price trends. Akyıldırım et al. [8] aimed to predict the price ups and downs of cryptocurrencies on a daily basis in their study. They developed Support Vector Machine (SVM), LR, Artificial Neural Network (ANN), and Random Forest (RF) models and compared their performances. The predictability of the twelve most liquid cryptocurrencies was analyzed using historical price data and technical indicators with machine learning classification algorithms on both daily and minute-level data. The average classification accuracy of the four models is above the 50% threshold for all cryptocurrencies and timeframes. SVM provided the best and consistent results with an accuracy range of 55% to 65%. Atçeken [9] aimed to predict whether the BTC price would rise above different threshold values within 24 hours. Different percentage profit targets (%2, %3, %5, %10) were set as threshold values. Financial indicators, support and resistance levels, highest and lowest values, opening and closing prices were used for regression and classification studies. For regression, K Nearest Neighbor (KNN), SVM, RF, ANN, and Vector Autoregression (VAR) models were developed, while for classification, LR, Naïve Bayes (NB), KNN, RF, SVM, and ANN models were employed. RF outperformed the other models used in both regression and classification in this study. The most accurate performance was observed with a profit target of 5%. Valencia et al. [10] proposed the use of widely used machine learning algorithms and social media data to predict the price movement of cryptocurrency markets such as Bitcoin, Ethereum, Ripple and Litecoin. ANN, SVM and RF models were developed to predict price ups and downs. The ANN model outperformed the other models, and it was concluded that social media data alone can be used to predict certain cryptocurrencies. Ji et al. [11] utilised blockchain data to predict BTC price and compared whether the price will up or down to the previous day. DNN, LSTM, CNN, Res-Net models, and their combinations were developed using features such as average block size, blockchain size, estimated transaction volume, confirmation time and miners' income. LSTM for BTC price prediction and DNN model for classification prediction gave more successful results. In addition, the study showed that class prediction is more effective than price prediction in terms of profitability. Kwon et al. [12] used only opening, closing, highest price, lowest price, and volume variables in their study to predict the price direction of cryptocurrencies such as Bitcoin, Ethereum, Ripple, Bitcoin Cash, Litecoin, Dash and Ethereum. For this purpose, LSTM and Gradient boosting (GB) models were developed. As a result, a performance improvement of approximately 7% was obtained with the LSTM model compared to the GB model. Shintate and Pichl [13] proposed a random sampling method with a deep learning-based trend prediction classification structure for the non-stationary cryptocurrency series. The performance of two classical baseline approaches on unstable BTC prices is compared and the model is found to reduce the class imbalance problem. A three-classes forecasting study was conducted for up, down, and stable states in the minute price series. Profit rates based on the Response Surface Methodology (RSM) gave better results than those based on the LSTM model. It performs well in the domain where the stationarity assumption is quite favourable. Experiments were conducted with very small-scale models and the superiority of RSM was confirmed by comparing it with MLP and LSTM models to distinguish the impact of the method.

This study focuses on the points where the trend reversals occur. By determining at which point the price direction will change, a three-classes estimation study is carried out as 'buy', 'sell' and 'hold'. The models are developed to decide the action to be taken for the next day by using today's closing time. The study investigates the effectiveness of machine learning techniques and technical indicators to predict BTC price trends. For this purpose, daily BTC data is used to examine the factors affecting BTC price and prediction models are developed by using tree-based ensemble learning algorithms. These models are trained to make forecasts with three-classes: 'buy', 'sell' and 'hold'. Financial indicators are calculated to create the data set. The data size is reduced, and the classes are balanced in different proportions. The performances of the trained models are evaluated by using various success metrics.

2. Methods

2.1. Extreme Gradient Boosting

Extreme Gradient Boosting (XGB), first presented by Chen and Guestrin [14], is a machine learning algorithm based on a group of decision trees and gradient boosting, is a machine learning algorithm. The first step in XGB is to create the initial prediction (base score). This prediction can be any number because the accurate result is obtained by approximating it through the operations in the next step. The default value for this prediction is 0.5. The quality of this prediction is examined by constructing a tree with the model's incorrect predictions in the next step. The classification problem for XGB is formulated as follows:

The similarity score for each branch of the tree is calculated to determine how well the data is grouped in the branches. The similarity score is given in equation (1):

$$Similarity\ Score = (\sum error)^2 / \left(\sum_i^N [P(1-P)] + \lambda\right) \tag{1}$$

where, N denotes the number of incorrect predictions, P denotes the probability percentage, and λ denotes the regularization parameter.

To understand which tree has better predictions, a gain score is calculated. While branches are evaluated with the similarity score, the entire tree is evaluated with the gain score. The gain score is given in equation (2):

$$Gain = S_{left} + S_{right} - S_{node} \tag{2}$$

where, S_{left} denotes the similarity score for the left branch, S_{right} denotes the similarity score for the right branch, and S_{node} is the similarity score of the previous node.

After deciding on the most successful tree, the pruning process begins. A cover value is calculated for each branch. If the cover value is lower than the gain score, the branch is pruned; otherwise, it continues to be split. The cover value is given in equation (3) and the output values of the model are calculated by using equation (4):

$$Cover\ Value = \sum_i^N [P(1-P)] \tag{3}$$

$$Output\ Value = \sum error / \left(\sum_i^N [P(1-P)] + \lambda\right) \tag{4}$$

where N is the number of incorrect predictions, P is the probability percentage and *error* are the number of incorrectly predicted records.

2.2. Random Forest

Random Forest (RF), first introduced by Breiman [15] is a technique that uses ensemble learning by combining multiple weak classifiers to solve complex problems. Each decision tree in a RF either votes for the predicted class or contributes to the average of the incoming predictions. The fundamental idea behind the RF algorithm is to leverage ensemble learning by including only a subset of features during tree development [16].

When selecting a feature for splitting in a dataset, the Gini index is used. The Gini index aims to make the split as pure as possible, where the lowest Gini value corresponds to the lowest impurity. Another metric used to measure impurity in a split is entropy. Mathematically, the Gini index and entropy are expressed as shown in equations (5) and (6), respectively:

$$Gini = 1 - \sum_i^N (p_i)^2 = 1 - [\,(p_{(+)})^2 + (p_{(-)})^2\,] \tag{5}$$

$$Entropy = -p_{(+)} \log p_{(+)} - p_{(-)} \log p_{(-)} \tag{6}$$

where, p_+ denotes the probability value of positive classes, and p_- denotes the probability value of negative classes.

2.3. Random Trees

Random Trees (RT), first presented by Ho [17], is an algorithm for constructing a decision tree. The algorithm iteratively divides the data into smaller subsets based on a random feature and continues until a stopping criterion is reached (e.g., a minimum number of data points in a subset or a maximum tree depth). Each leaf node of the tree generates a prediction based on the class distribution in the subset it represents [18].

In measuring the success of a tree, rule accuracy, tree accuracy, and interestingness measure are used. The interestingness measure is given in equation (7):

$$I_{index(t)} = P(A_t) * P(B_t) * [P(B_t / A_t) + P(\bar{B}_t / \bar{A}_t)] \tag{7}$$

where, $P(A_t)$ denotes the tree accuracy, $P(B_t)$ denotes the rule accuracy, $P(B_t | A_t)$ denotes the correct predictions made by both trees and node and $P(\bar{B}_t / \bar{A}_t)$ denotes the incorrect predictions made by both trees and node.

2.4. Likelihood Ratio

In this study, likelihood ratio statistics is utilized for feature selection Likelihood is a probability function that expresses how observations in a dataset are explained by a certain parameter set. This function calculates the probability of the dataset under a specific set of parameter values [19]. Likelihood ratio, first presented by Fisher [20], is the ratio of two hypotheses and is given in equation (8):

$$Likelihood\ Ratio = -2ln \frac{Likelihood\ of\ Alternative\ Hypothesis}{Likelihood\ of\ Null\ Hypothesis} \tag{8}$$

2.5. Principal Component Analysis

Principal Component Analysis (PCA), first introduced by Pearson [21], is one of the multivariate statistical analysis methods used to reduce the dimensionality of a dataset and understand relationships between variables. It aims to make variables independent from each other, especially in datasets with multicollinearity issues.

During PCA, a process called rotation is typically performed to enhance the interpretation and better understanding of the obtained components. Rotation modifies the orientation of the components and their relationships with variables, making the components more easily interpretable. The two most used types of rotation are Varimax and Promax. Varimax rotation aims to reduce the correlation between components and variables while increasing the correlation among components. Promax rotation, on the other hand, maintains the correlation between components while altering their relationships with variables [22].

The PCA computation consists of two steps: the calculation of the covariance matrix and the calculation of the eigenvalues. The mathematical expressions for the covariance matrix and eigenvalue calculation are as follows, respectively, as given in equations (9) and (10):

$$cov\,(X,Y) = \frac{1}{n}\sum_i^n (x-\bar{x})(y-\bar{y}) \tag{9}$$

where, *cov (X, Y)* denotes the covariance between variables *X* and *Y*, n denotes the number of observations, $\bar{x}$, detones the mean of variable *X*, and ve $\bar{y}$ denotes the mean of variable *Y*.

$$det(A\text{-}\lambda I) = 0 \tag{10}$$

where, *A* denotes the covariance matrix, *I* denotes the identity matrix, and λ denotes the eigenvalue.

3. Application

3.1. Data Generation

Daily data obtained from 'binance.com' [23] is examined in this study. The current dataset, consisting of approximately 2000 observations, covers the period from September 1, 2017, to April 1, 2023. It includes data on the opening, closing, highest price, lowest price, and volume of BTC. Using this information, trend, momentum, volatility, and volume indicators are calculated. Alongside these generated indicators, a total of 90 features are employed in the analysis and IBM SPSS Modeler version 18.2 is used in the study [1].

The highest and lowest prices of the day are called as "up bar" if they are higher than the previous day, and "down bar" if they are lower. Three-day trends are determined based on the positions of price bars. If an observation is not in an upward trend (1) or a downward trend (-1), it is considered as a stable trend (0). In this context, the rules for the dependent variable created for buy, sell, and hold decisions are given in Table 1.

Table 1. Rules of the dependent variable.

Trend of the previous day	Trend of today	Dependent variable
0	1	Buy
-1	1	Buy
0	-1	Sell
1	-1	Sell
1	0	Hold
-1	0	Hold
0	0	Hold
1	1	Hold
-1	-1	Hold

As seen in Table 1, a data labeling process is conducted in such a way that when the trend turned upward, a buy decision is made, when it turned downward, a sell decision is made, and when it remained stable, no action is taken. As a result of feature selection applied using likelihood ratio statistics, it is decided to use 40 independent variables.

3.2. Principal Component Analysis

To mitigate the issue of multicollinearity due to a large number of independent variables, the dataset's dimensionality is reduced using PCA. Since the components can't fully separate the variables, a varimax rotation is applied. The analysis results indicate that the first 10 components explain approximately 81% of the total variance in the independent variables. Total explained variance is given in Figure 1.

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	9.148	23.456	23.456	9.148	23.456	23.456	7.900	20.255	20.255
2	7.928	20.328	43.784	7.928	20.328	43.784	4.324	11.088	31.343
3	3.226	8.273	52.057	3.226	8.273	52.057	3.771	9.669	41.013
4	2.543	6.520	58.577	2.543	6.520	58.577	3.681	9.437	50.450
5	2.006	5.142	63.719	2.006	5.142	63.719	2.896	7.427	57.877
6	1.666	4.272	67.991	1.666	4.272	67.991	2.642	6.775	64.651
7	1.570	4.026	72.018	1.570	4.026	72.018	2.050	5.257	69.908
8	1.297	3.326	75.344	1.297	3.326	75.344	1.518	3.892	73.800
9	1.234	3.163	78.507	1.234	3.163	78.507	1.462	3.750	77.549
10	1.042	2.671	81.178	1.042	2.671	81.178	1.415	3.629	81.178
11	.967	2.479	83.657						
12	.820	2.103	85.760						
13	.786	2.015	87.775						
14	.702	1.799	89.575						
15	.646	1.657	91.232						

Figure 1. PCA Variance Summary

3.3. Resampling

The imbalance among the classes of the dependent variable is resolved using the random undersampling method. The class distribution of the dependent variable is given in Table 2.

Table 2. Class distribution of the dependent variable.

Class	Number of Observation	Ratio of Observation
Buy	192	9.47%
Sell	187	9.23%
Hold	1648	81.3%

As shown in Table 2, the proportions of the 'buy' and 'sell' classes in the dataset are approximately 9%, while the 'hold' class constitutes about 81.3% of the dataset. To solve this issue, two different undersampling studies are conducted, and the results are compared. The class distributions of the first and second datasets are given in Table 3 and Table 4, respectively.

Table 3. Distribution where all classes are equal.

Class	Number of Observation	Ratio of Observation
Buy	192	33.51%
Sell	187	32.64%
Hold	194	33.85%

Table 4. Distribution where the hold class is high.

Class	Number of Observation	Ratio of Observation
Buy	192	26.82%
Sell	187	26.12%
Hold	337	47.06%

The first dataset (Table 3) is obtained as a result of an undersampling study where the record counts of all classes are very close to each other. However, the models trained on the first dataset can't sufficiently learn the 'hold' class, so a resampling study is performed at different rates.

In the second resampling study, the dataset is prepared in a way that the 'hold' class has a higher number of observations compared to the other classes, similar to its proportion in the original dataset. In the second dataset (Table 4), the proportions of the 'buy' and 'sell' classes are approximately 26%, while the 'hold' class constitutes about 47.06% of this dataset. Consequently, two different datasets are obtained.

3.4. Model Training and Results

The dataset is randomly divided into two subsets: an 80% training set and a 20% test set. RF, XGB, and RT models are trained on both datasets with the same hyperparameters. Model results are evaluated using 5-fold cross-validation. The hyperparameters used for XGB, RF and RT models are demonstrated in Tables 5, 6, and 7, respectively.

Table 5. Hyperparameters of the XGB model.

Hyperparameter	Value
tree_method	Auto
number_boost_round	10
max_depth	6
min_child_weight	1
max_delta_step	0
objective	Multi
stopping_rounds	10
evaluation_data_ratio	0.3
sub_sample	1
eta	0.3
gamma	0
colsample_by_tree	1
colsample_by_level	1
lambda	1
alpha	0
scale_pos_weight	1

Table 6. Hyperparameters of the RF model.

Hyperparameter	Value
bootstrap	TRUE
ccp_alpha	0
criterion	gini
max_depth	10
max_features	auto
min_impurity_decrease	0
min_samples_leaf	1
min_samples_split	2
min_weight_fraction_leaf	0
n_estimators	10
oob_score	FALSE
verbose	0
warm_start	FALSE
number_of_trees_to_build	10
learning_rate	0.01
max_iteration	1000
max_evaluation	300

Table 7. Hyperparameters of the RT model.

Hyperparameter	Value
number_of_models_to_build	100
sample_size	1
max_number_of_nodes	10000
max_tree_depth	10
min_child_node_size	5
max_percentage_of_missing_values	70
exclude_ fields__with_a_single_category_majority_over (%)	95
max_number_of_field_categories	49
min_field_variation	0.05
number_of_bins	10
number_of_interesting_rules_to_report	50

The results of the models trained with the first and second datasets by using the above hyperparameters are given in Figure 2 and Figure 3, respectively.

XGT											
TEST	Precision	Recall	Specifity	F1	Accuracy	TRAIN	Precision	Recall	Specifity	F1	Accuracy
HOLD	0.74	0.43	0.93	0.55	0.77	HOLD	0.81	0.52	0.94	0.63	0.80
SELL	0.77	0.87	0.89	0.82	0.88	SELL	0.74	0.90	0.86	0.82	0.87
BUY	0.73	0.92	0.85	0.81	0.87	BUY	0.75	0.91	0.84	0.82	0.87
RF											
TEST	Precision	Recall	Specifity	F1	Accuracy	TRAIN	Precision	Recall	Specifity	F1	Accuracy
HOLD	0.83	0.51	0.94	0.63	0.77	HOLD	0.78	0.49	0.93	0.60	0.78
SELL	0.76	0.91	0.87	0.83	0.88	SELL	0.78	0.89	0.87	0.83	0.88
BUY	0.79	0.94	0.88	0.85	0.90	BUY	0.77	0.92	0.85	0.83	0.87
RT											
TEST	Precision	Recall	Specifity	F1	Accuracy	TRAIN	Precision	Recall	Specifity	F1	Accuracy
HOLD	0.77	0.40	0.94	0.52	0.76	HOLD	0.70	0.39	0.91	0.50	0.73
SELL	0.72	0.91	0.80	0.80	0.84	SELL	0.70	0.88	0.81	0.78	0.84
BUY	0.75	0.82	0.86	0.78	0.85	BUY	0.70	0.85	0.83	0.77	0.83

Figure 2: Training and test results of the first dataset.

When examining the XGB model in Figure 2, it can be observed that all success metrics for the 'buy' and 'sell' classes are above 70% for both the training and test data sets. However, the recall rate of the 'hold' class in the test set which equal to 43% indicates that the model doesn't learn this class as effectively as the others. The F1 score of 55% on the test set suggests that the 'hold' class is not predicted accurately. Similar results are obtained when examining the training set. Similarly, when examining the training and test results of the RF and RT models, it is evident that the 'hold' class is not predicted well enough.

XGT											
TEST	Precision	Recall	Specifity	F1	Accuracy	TRAIN	Precision	Recall	Specifity	F1	Accuracy
HOLD	0.79	0.78	0.85	0.78	0.82	HOLD	0.82	0.78	0.85	0.80	0.82
SELL	0.74	0.83	0.89	0.78	0.87	SELL	0.76	0.86	0.90	0.80	0.89
BUY	0.79	0.78	0.91	0.78	0.87	BUY	0.82	0.80	0.93	0.81	0.90
RF											
TEST	Precision	Recall	Specifity	F1	Accuracy	TRAIN	Precision	Recall	Specifity	F1	Accuracy
HOLD	0.81	0.66	0.87	0.72	0.77	HOLD	0.88	0.69	0.92	0.77	0.81
SELL	0.71	0.91	0.87	0.80	0.88	SELL	0.77	0.92	0.91	0.84	0.91
BUY	0.75	0.83	0.90	0.79	0.88	BUY	0.75	0.88	0.88	0.81	0.88
RT											
TEST	Precision	Recall	Specifity	F1	Accuracy	TRAIN	Precision	Recall	Specifity	F1	Accuracy
HOLD	0.82	0.61	0.88	0.70	0.75	HOLD	0.86	0.61	0.91	0.71	0.77
SELL	0.62	0.86	0.83	0.73	0.84	SELL	0.68	0.85	0.85	0.75	0.85
BUY	0.60	0.85	0.83	0.70	0.84	BUY	0.71	0.84	0.86	0.77	0.85

Figure 3: Training and test results of the second dataset.

When examining Figure 3, all success metrics of the XGB model are above 70%. The results for the test and training sets are very close to each other. This indicates that the model is consistently trained. The recall rates for the 'buy,' 'sell,' and 'hold' classes of the XGB model in the test set are 78%, 83%, and 78%, respectively. When examining the specificity rates, they are 91%, 89%, and 85%, respectively. These results indicate that the model is successfully trained for all classes. The F1 score of 78% for all classes suggests that the imbalance problem between classes is resolved in the model's predictions. When examining the RF model, it is observed that the recall rate for the 'hold' class is 66% in the test set and 69% in the training set. However, the F1 score which is equal to 72% and 77% in the test and training sets, respectively, indicates that it is able to strike a balance between precision and recall. Similarly, although the precision rate for the 'hold' class is low of the RT model, the F1 score is at an acceptable level. When looking at the accuracy rates of the XGB model, they are generally higher than the accuracy rates of the other models, with 87%, 87%, and 82% for the test set and 90%, 89%, and 82% for the training set. If all these results are compared, it can be concluded that the XGB model has higher performance than the RF and RT models.

4. Conclusion

In this paper, classification algorithms are utilized to predict trend reversals of the BTC cryptocurrency. Daily updated data spanning two and a half years, obtained from 'binance.com', is used in the study along with derived new features from this data set. Using by principal component analysis, the size of the data set is reduced, and the multicollinearity problem is prevented. Two different datasets with varying class ratios are created by using undersampling methods. The datasets are split into random 80% training and 20% test sets. Three-class prediction models, 'buy,' 'sell,' and 'hold,' are developed by using the XGB, RF, and RT algorithms. The results are compared based on precision, recall, specificity, F1 score, and accuracy metrics. It is observed that the models trained with the second dataset, which contains a higher number of observations for the 'hold' class compared to the other classes, yield better results. When comparing the success metrics in the test and train sets, it is concluded that the XGB model outperformed the RF and RT models in terms of higher performance.

According to the model results, it is observed that the most important factors affecting BTC price trend changes are variables derived from the position of price bars, RSI, exponential moving average (EMA), SMA, and volume information. However, these variables can't adequately explain situations where the trend is stable. This problem is resolved by increasing the number of observations for the 'hold' class in the resampling process. As a result, all classes can be predicted with sufficient accuracy by using the available variables. When the results are compared with the literature, ensemble learning models such as RF and XGB give a high performance like in the other studies. However, this study focuses solely on trend reversals and has developed models with accuracies of 80% and above.

References

[1] S. Ürgenç, Predicting Bitcoin Trends Reversals With Machine Learning Methods (Makine Öğrenmesi Yöntemleri ile Bitcoin Trend Dönüşlerinin Tahmin Edilmesi), (2023). Master Thesis, Mimar Sinan Fine Arts University, Istanbul.

[2] N.T. İnce, Predicting The Bitcoin Trend Using Technical Indicators For Deep Learning Algorithmic Features, (2019). Master Thesis, Boğaziçi University, Istanbul.

[3] Z. Qiang, J. Shen, Bitcoin High-Frequency Trend Prediction with Convolutional and Recurrent Neural Networks, Comput. Sci. (2021).

[4] S. Cavalli, M. Amoretti, CNN-based multivariate data analysis for bitcoin trend prediction, Appl. Soft Comput. 101 (2021) 107065. doi: 10.1016/J.ASOC.2020.107065.

[5] S. Alonso-Monsalve, A.L. Suárez-Cetrulo, A. Cervantes, D. Quintana, Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators, Expert Syst. Appl. 149 (2020) 113250. doi: 10.1016/J.ESWA.2020.113250.

[6] I.E. Livieris, E. Pintelas, S. Stavroyiannis, P. Pintelas, Ensemble Deep Learning Models for Forecasting Cryptocurrency Time-Series, Algorithms 2020, Vol. 13, Page 121. 13 (2020) 121. doi:10.3390/A13050121.

[7] G. Cohen, Forecasting Bitcoin Trends Using Algorithmic Learning Systems, Entropy 2020, Vol. 22, Page 838. 22 (2020) 838. doi:10.3390/E22080838.

[8] E. Akyildirim, A. Goncu, A. Sensoy, Prediction of cryptocurrency returns using machine learning, Ann. Oper. Res. 297 (2021) 3–36. doi:10.1007/S10479-020-03575-Y/TABLES/18.

[9] M.A. Atçeken, Trading Strategy Based Classification On Cryptocurrency Price Prediction, (2021). Master Thesis, TED University, Ankara.

[10] F. Valencia, A. Gómez-Espinosa, B. Valdés-Aguirre, Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning, Entropy 2019, Vol. 21, Page 589. 21 (2019) 589. doi:10.3390/E21060589.

[11] S. Ji, J. Kim, H. Im, A Comparative Study of Bitcoin Price Prediction Using Deep Learning, Math. 2019, Vol. 7, Page 898. 7 (2019) 898. doi:10.3390/MATH7100898.

[12] D.H. Kwon, J.B. Kim, J.S. Heo, C.M. Kim, Y.H. Han, Time Series Classification of Cryptocurrency Price Trend Based on a Recurrent LSTM Neural Network, J. Inf. Process. Syst. 15 (2019) 694–706. doi:10.3745/JIPS.03.0120.

[13] T. Shintate, L. Pichl, Trend Prediction Classification for High Frequency Bitcoin Time Series with Deep Learning, J. Risk Financ. Manag. 2019, Vol. 12, Page 17. 12 (2019) 17. doi:10.3390/JRFM12010017.

[14] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. 13-17-August-2016 (2016) 785–794. doi:10.1145/2939672.2939785.

[15] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32. doi:10.1023/A:1010933404324/METRICS.

[16] C.R. Sekhar, Minal, E. Madhu, Mode Choice Analysis Using Random Forrest Decision Trees, Transp. Res. Procedia. 17 (2016) 644–652. doi: 10.1016/J.TRPRO.2016.11.119.

[17] T.K. Ho, Random decision forests, Proc. Int. Conf. Doc. Anal. Recognition, ICDAR. 1 (1995) 278–282. doi:10.1109/ICDAR.1995.598994.

[18] N. Boodhun, · Manoj Jayabalan, Risk prediction in life insurance industry using supervised learning algorithms, Complex Intell. Syst. 2018 42. 4 (2018) 145–154. doi:10.1007/S40747-018-0072-1.

[19] J. Fan, C. Zhang, J. Zhang, Generalized Likelihood Ratio Statistics and Wilks Phenomenon. 29 (2001) 153–193. doi:10.1214/AOS/996986505.

[20] R. A., Fisher, The Conditions Under Which χ2 Measures the Discrepancey Between Observation and Hypothesis, Journal of the Royal Statistical Society (1924) 442-450.

[21] K. Pearson, LIII. On lines and planes of closest fit to systems of points in space, London, Edinburgh, Dublin Philos. Mag. J. Sci. 2 (1901) 559–572. doi:10.1080/14786440109462720.

[22] H. Abdi, L.J. Williams, Principal component analysis, Wiley Interdiscip. Rev. Comput. Stat. 2 (2010) 433–459. doi:10.1002/WICS.101.

[23] Binance - Bitcoin, Ethereum ve Altcoin'ler için Kripto Para Borsası, (n.d.). https://www.binance.com/tr (accessed October 12, 2023).