



The Effect of Data Types' on the Performance of Machine Learning Algorithms for Cryptocurrency Prediction

Hulusi Mehmet Tanrikulu¹ · Hakan Pabuccu¹

Accepted: 26 February 2025 / Published online: 22 March 2025
© The Author(s) 2025

Abstract

Forecasting cryptocurrencies as a financial issue is crucial as it provides investors with possible financial benefits. A slight improvement in forecasting performance can lead to increased profitability; Therefore, obtaining a realistic forecast is very important for investors. Bitcoin, frequently mentioned in recent due to its volatility and chaotic behavior, has become an investment tool, especially during and after the COVID-19 pandemic. In this study, selected ML techniques were investigated for predicting cryptocurrency movements by using technical indicator-based data sets and measuring the applicability of the techniques to cryptocurrencies that do not have sufficient historical data. In order to measure the effect of data size, Bitcoin's last 1 year and 7 years of data were used. Following the related literature, Google trends and the number of tweets were used as input features, in addition to the most commonly used twelve technical indicators. Random Forest, K-Nearest Neighbors, Extreme Gradient Boosting (XGBoost-XGB), Support Vector Machine (SVM), Naive Bayes (NB), Artificial Neural Networks (ANN), and Long-Short-Term Memory (LSTM) network were optimized for best results. Accuracy, F1, and area under the ROC curve values were used to compare the model performance. For continuous data, ANN and SVM performed the best with the highest accuracy and outperformed the other ML models for complete and reduced sets. LSTM reached the best accuracy for trend data, but SVM, NB, and XGB models showed similar performance. The research shows that some indicators significantly affect prediction performance, and the data discretization process also improved the model's accuracy. While the number of samples affects the results of many ML models, correctly optimized and fine-tuned models may also give excellent results even with less data.

Keywords Financial prediction · Machine learning · Bitcoin · Continuous data · Trend data

1 Introduction

With the introduction of the internet into our lives in the past three decades, daily habits have undergone a rapid digital transformation. Also, the new virtual realities beyond physical spaces and borders have become a part of our lives. As part of this transformation, business models that interact with each other throughout the day have started to develop worldwide, independent of the concept of working hours. Parallel to the development of technology and the internet, the concept of money has also been digitized to a large extent and has become numbers connected to bank accounts. In addition to all these developments, in 2009, a completely digital currency called Bitcoin (BTC), which is independent of banking systems, not connected to any state or center, not physically produced, and can be obtained in a limited number by arranging digital blocks in a certain order has become as a new medium of exchange Delfabbro et al. (2021).

Cryptocurrency markets have rapidly expanded globally since the discovery of the BTC as seen in Fig 1. Cryptocurrencies are used for cross-border payments and financial investments, making them a valuable digital and safe haven asset Kim et al. (2021). Parallel to the development of the digital world, data analysis tools and models have also been developed in the same way. Researchers and investment experts try to develop better estimation tools by combining fundamental and technical analysis models and using traditional statistical and machine learning techniques for financial market prediction. The fact that Bitcoin has been frequently traded as an investment tool, especially with the COVID-19 pandemic, has increased the number of scientific and financial studies in this field Kim et al. (2020).

Predicting the price or movement of Bitcoin is a complex and challenging task due to the various technical and statistical challenges inherent in financial time series prediction. Over the years, numerous models have been developed to forecast stock prices or returns and cryptocurrency prices or returns. One of the earliest of these models is the classical regression analysis. Stock index data sets are pre-processed because they have a chaotic behavior and contain uncertainties and nonlinear relationships, resulting in information loss. Therefore, machine learning algorithms



Fig. 1 Bitcoin price volatility and volume from 2015 to 2021

have recently been more popular than conventional models (regression analysis, discriminant analysis, statistical methods, etc.) for financial time series forecasting.

One of the biggest challenges in ML predictions is model overfitting, where a model is fitted too closely to training data but cannot generalize well to unseen data. This problem is exacerbated by the high volatility as seen in Fig. 1 and the chaotic behavior of Bitcoin data, which makes it crucial to use the right regularisation techniques and model evaluation to avoid overfitting Khosravi (2023).

It was conducted a multi-faceted analysis to predict Bitcoin price movements. It was found that the number of data samples and data types significantly affect prediction performance. It was also investigated the performance of selected ML techniques by using the technical indicator-based continuous and trend data sets.

This article aims to make the following contributions:

The problem is formalized as a classification to predict the BTC price movements or trends. The proposed models investigated the effect of continuous and trend data sets on prediction performance and the effect of the amount of data samples. It also discussed the importance of indicators, Google trends, and tweet data on BTC price movements.

Our study aim is limited to investigating and improving the forecasting performance of machine learning models. Text classification and sentiment analysis were not included due to the main objective of this research. Therefore, models such as transformers, NLP, and LLM-based were not used in this research as well. To justify the selection of the topic, used methodology and problem type are also discussed at the end of the related work section with crucial references. The remainder of this paper is organized as follows: Sect. 2 presents the “Related works”. Section 3 explains the data sets and applied methodology for models. Section 4 presents the experimental results and the performance comparison of different prediction models. Finally, Sect. 5 finishes the discussions and conclusions.

2 Related Works

Financial time series prediction techniques are organized into four categories: statistical methods, machine learning, pattern recognition, and sentiment analysis by Shah et al. (2019). The financial time series prediction problems can also be categorized into classification and regression. In this section, statistical methods and machine learning models are considered within this research's scope. Statistical models simplify assumptions to obtain theoretical guarantees and thus have a simple structure and potentially lower performance than machine learning techniques. Machine learning models are more general, simplify assumptions less, and offer better model-fitting capabilities.

One of the earliest study on Bitcoin price movement prediction was published by Shah and Zhang (2014) using the Bayes Linear Regression model. Bitcoin price data from February to July 2014 were taken, and the k-mean clustering ML model was applied using more than 200 million raw data points. They try to propose a buy or sell strategy pattern. To do that, they only use the current price. Madan et al. (2015) applied machine learning techniques to predict BTC price movements using two

different datasets consisting of BTC daily closing prices with 10-second and 10-minute frequency. The authors applied the Generalized Linear Model (GLM), SVM, and RF models. Results were evaluated using sensitivity, specificity, precision, and accuracy. As a result of the study, the GLM and RF models provided better results than the SVM. Huang et al. (2019) used 124 financial technical indicators from the daily data of Bitcoin between 2012 and 2018 with 2168 samples. Classification and Regression Trees (CART) were used as a machine learning technique to forecast bitcoin prices. The proposed model has strong predictive power by using out-of-sample data set, for narrow ranges of Bitcoin's daily returns. The results highlighted that technical analysis is helpful in a market like cryptocurrency or stock, whose value is primarily driven by non-fundamental factors. In this study, the transformation of indicators also provided an idea of its importance. Mangla et al. (2019) used ARIMA, SVM, LR, and RNN models to forecast Bitcoin price movements. The dataset consists of Bitcoin one-hour intervals between 2015–2019. The performance of ML algorithms was tested, and they were tried to increase their performance. As a result of the study, among the four models, ARIMA performed well for next-day predictions but performed poorly for longer terms, like given the last few days' price predictions for the next 5–7 days' prices. The RNN performed consistently for up to 6 days. Where indicators were not used, accuracy rates were between 0.47–0.53. Chen et al. (2020) used two separate data sets. The first consists of Bitcoin daily closing price, and the other dataset consists of 5 min of Bitcoin price movements between 2017–2019. Block size, hash rate, excavation rate, number of transfers, mining, Google trend searches, transaction volume, and gold prices were used as input features. LR and Linear Discriminant Analysis (LDA), RF, XGB, Quadratic Discriminant Analysis, SVM, and LSTM were applied. Accuracy, precision, recall, and F1-Score were calculated to compare the performance of the selected algorithms. On the daily closing price dataset, LR gave the best results with 0.66 accuracy, while XGB remained at 0.48. For the second data set, the LSTM gave the best results with 0.67 while the LR was 0.55 and LDA 0.52 level of accuracy. These results led us to focus on the use of indicators for forecasting and the need to investigate the effect of trend data. Lahmiri and Bekiros (2020) implemented three different sets of algorithms: i. statistical machine learning, ii. algorithmic models, iii. artificial neural network topologies. 5-minute intervals from the beginning of 2016 to March 2018 were taken as datasets. The dataset was divided into two parts: 0.80 for training and 0.20 for testing. The optimal parameter values for the SVR, GRP, and KNN were found via Bayesian optimization. Based on diverse performance metrics, BRNN rendered outstanding accuracy in prediction, while its convergence was unhindered and remarkably fast. Pabuccu et al. (2020) used daily Bitcoin data between 2008–2019. The number of samples is 1935. The authors used an additional 100 data points after June 2020 as a validation set. The SVM, ANN, NB, and RF techniques were used, and LR was used for benchmarking. Accuracy, MAE, and RMSE were calculated to evaluate the model performance and conduct a t-test for statistical significance. While the RF showed the highest prediction performance for the continuous dataset, the NB was the lowest. The ANN and NB were the best and the worst prediction models for trend data sets, respectively. Furthermore, the trend datasets improved the overall prediction performance in all estimated models. Chen et al. (2021) used economic (gold, oil, SP500, exchange rates, etc.) and technological (Hasrate, average transaction fees, Tweet, Google trends, etc.) data as features

in their studies. Four different data sets were created to determine the important price change of Bitcoin between 2011 and 2018. The last 2 months of data were selected as the prediction set. The ANFIS, SVR, ARIMA, and LSTM machine-learning techniques were used as models. The LSTM model gave the best MAE, MAPE, and RMSE test results. The results demonstrated that using the economic and technology determinants effectively predicted the Bitcoin exchange rate. Sossi-Rojas et al. (2023) tried to predict Bitcoin price movements using the LSTM, BiLSTM, GRU, BiGRU, and LightGBM techniques. They created a data set consisting of hash rate, exchange rate, Google Trend, fear greed, and Bitcoin prices as features between 2013–2022. The RMSE, MSE, MAE, and accuracy were used to evaluate the model performance. BiLSTM showed the best performance in terms of accuracy. Goutte et al. (2023) obtained hourly Bitcoin data between 2017 and 2022 using Binance API. They used SMA, RSI, MACD, WR, STOC, and MFI as technical indicators. Seven machine learning algorithms were used: XGBoost, LightGBM, MLP, GRU, LSTM, and CNN. Unlike other studies, candlestick charts were converted into a discrete data structure and interpreted using it. Analyses were made using the GRU, CNN, LSTM, MLP, LGBM, and XGBoost models, and GRU showed the best performance. Rahaman et al. (2022) tried to develop a new indicator model to interpret bitcoin prices. They obtained data from various sources using the BeautifulSoup library, as we used in our research. They aimed to produce a buy-sell signal using daily tweets and conducting sentiment analysis. When related studies are examined overall, it can be said that similar models, data sets, and metrics were used for many studies. The OHLC dataset that belongs to financial assets was used as input features and did not apply advanced data generation and preprocessing techniques. Many studies have not provided statistical significance tests to compare the model's accuracy. Out-of-sample test predictions have not also been presented in many published papers. Presenting train and test set prediction is not enough to label ML models as accurate. From these points, the gaps mentioned in this study were tried by generating new continuous and trend data sets using technical indicators. A statistical significance t-test was applied to provide a clear justification of the model's predictions regarding accuracy levels. One of the most important things is to validate the model's accuracy by running them on out-of-sample data sets, as conducted in this study for two different ones. It is mandatory to investigate the performance of the estimated model for real-life simulations.

Classification approaches, as highlighted in the literature, offer distinct advantages in reducing information loss by directly predicting trends, which aligns with the needs of financial decision-making frameworks like buy-hold-sell strategies Shah and Zhang (2014). Overfitting, a common concern in financial modeling due to noisy and volatile data, is effectively mitigated by ensemble-based classification methods like RF and BRNN, as demonstrated in the studies by Madan et al. (2015) and Lahmiri and Bekiros (2020) where these models outperformed others in their respective datasets.

The selection of classification algorithms over regression in financial asset prediction is justified by their ability to provide actionable insights, such as buy-hold-sell signals, which are directly applicable to investment strategies. Classification focuses on categorizing outcomes (e.g., price increase or decrease), aligning with decision-making needs, and minimizing the complexity of interpreting continuous

values. However, regression and classification face challenges such as overfitting and information loss, which are critical for robust model performance. Thus, regardless of the choice between regression and classification, it is essential to implement robust techniques such as regularization, cross-validation, and feature engineering to mitigate overfitting and preserve critical information. This ensures that the model remains predictive and practical for real-world applications, such as producing reliable investment signals for financial assets.

The literature investigation highlights that classification algorithms are highly effective in predicting directional movements of financial assets, as they focus on identifying categories such as “price up” or “price down” rather than continuous price values. This capability aligns seamlessly with the need to produce actionable investment strategies, such as a buy-hold-sell signal, which are inherently categorical decisions. By utilizing classification models, investors can implement a strategic framework to optimize decision-making, enhancing the reliability of recommendations in volatile markets like cryptocurrencies. This approach aids in translating technical and market analysis into simplified, actionable guidance for investors.

3 Data Preparation

The scope of this research was limited to the Bitcoin price movement prediction problem. Other crypto assets are not included in this study. In this research, three different types of information were used from the original data.

i. Bitcoin historical prices contain daily values of OHLC (open, high, low, close, and trading volume in USD) dates between 17/09/2014–09/08/2021 were obtained by using the Python Yahoo Finance API.

ii. Following the literature from Dash and Dash (2016), Vargas et al. (2018), Julian et al. (2023); it is decided to use these technical indicators: Simple Moving Averages (SMA), Weighted Moving Average (WMA), Exponential Moving Average (EMA), Average True Range (ATR), Bollinger Bands (Boll), Momentum (Mom), Stochastic indicators (K and D), Relative Strength Index (RSI), Moving Average Convergence and Divergence (MACD), Larry Williams (LW), Accumulation Distribution Indicator (A/D). The indicators were calculated using the Python Technical Analysis Library (TA-Lib), which used the Bitcoin historical data we obtained in the first stage. These indicators are used as features for selected ML techniques.

iii. Following the literature from Matta et al. (2015), Abraham et al. (2018) and Linardatos (2020), daily tweet counts about Bitcoin and Google search trend data were obtained from <https://bitinfocharts.com/> using the Python BeautifulSoup library.

By using these three data sources, continuous and trend data sets were generated. Detailed information about the data sets is presented in the next section.

For investigating the data size effects, the complete dataset, which contained data between 17/09/2014–10/06/2021, and the reduced dataset, which contained data between 10/06/2020–10/06/2021, was generated. The validation set contains the data between 11/06/2021–09/08/2021 to validate the estimated models for 30–60 days, respectively.

The output is defined as trend data, representing the up-down movements of the daily closing price. If any sample point is 0, it means a downward trend, and if it is 1, it means an upward trend. The output data was also used by dividing the training and testing sets at the same rate. For the classification problem, a binary output was calculated using (Eq. 1.). The up and down movement was coded as {0, 1} for the prediction models. This is a very common process that represents the trend of prices and returns, or in other words, economic profits or losses.

$$y_t = \begin{cases} 1, & \text{if } return_t > return_{t-1} \\ 0, & \text{else} \end{cases} \quad (1)$$

3.1 Continuous Data Preparation

The complete dataset, containing 17 features and 2455 samples using technical indicators and Google trend-tweet counts, was normalized to the 0-1 range, and a continuous data set was created. The reduced data set was created by trimming the last 365 days of the complete data set. The definitions and calculation procedures of the technical indicators used in the datasets are as follows in Table 1:

Table 1 Indicator formulas

Indicator	Formula
<i>A/D</i>	$A/D = \frac{H_t - A_{n-1}}{H_t - L_t}$
Bollinger Bands Upper	$UPPERBB = MA + D\sqrt{\frac{\sum_{i=1}^n (y_i - MA)^2}{n}}$
Bollinger Bands Low	$LOWERBB = MA - D\sqrt{\frac{\sum_{i=1}^n (y_i - MA)^2}{n}}$
EMA	$EMA = A_t * k + EMA_{t-1} * (1 - k)$
MACD	$MACD = EMA_{12} - EMA_{26}$
SMA	$SMA = \frac{(A_1 + A_2 + \dots + A_n)}{n}$
WMA	$WMA = \frac{\sum_{i=1}^n W_i * V_i}{\sum_{i=1}^n W_i}$
RSI	$RS = \frac{Average(PC)}{Average(NC)} \quad RSI = 100 - \frac{100}{1 + RS}$
ATR	$ATR = \frac{1}{n} \sum_{i=1}^n TR_i$
LW	$LW = \frac{H_n - A_n}{H_n - L_n} * 100$
MOM	$Mom = \frac{A_n}{A_{t-n}} * 100$
StoK - StoD	$StoK = \frac{A_n - L_n}{H_n - L_n} * 100 \quad StoD = \frac{H_n}{L_n} * 100$
<i>A</i> = Closing price	<i>t</i> = Day of calculation
<i>k</i> = 2÷(total time period + 1)	<i>n</i> = total time period
<i>V_t</i> = Actual price at time <i>t</i>	<i>W_t</i> = Weighting price at time <i>t</i>
<i>A_n</i> = Closing price at time <i>n</i>	<i>H_t</i> = Highest price at time <i>t</i>
<i>L_t</i> = Lowest price at time <i>t</i>	<i>TR_i</i> = Specified actual range
<i>H_n</i> = highest price in time interval <i>n</i>	<i>L_n</i> = Lowest price in time interval <i>n</i>

Table 2 Descriptive statistics

Model	Minimum	Maximum	Mean	SD
SMA	222.873	60050.151	7706.481	11654.864
WMA	219.543	60904.757	7738.658	11692.165
EMA	225.270	60397.746	7704.597	11626.316
ATR	3.556	5003.094	450.985	833.745
Momentum	-21229.875	14372.156	148.338	2094.090
Stochastic %K	0	100	56.147	29.923
Stochastic %D	2.714	99.199	56.148	28.172
Boll-Upper	211.193	66836.082	8288.583	12601.241
Boll-Middle	200.878	62382.740	7776.278	11759.605
Boll-Lower	161.955	60159.319	7263.974	10941.897
RSI	9.920	94.302	54.293	14.288
LW	-100	0	-43.853	29.923
A/D	-53660.135	119838.185	5269.312	14876.947
Tweet	7300	363566	37187.716	28701.176
Google Trend	5.268	616.867	48.825	60.356

In addition to technical indicators, the number of tweets and Google trend data were used. The purpose of using Google trends and the number of tweets is that both factors reflect public attention. However, there are differences between these data. While Google trends show the public's intention to learn about Bitcoin, tweets represent the public's intention to discuss Bitcoin. In this study, we observed the effect of the change in these data on the change in the direction of price movement.

Descriptive statistics are presented as follows in Table 2.

Table 2 provided the descriptive statistics of the features. Features were normalized before conducting the ML models to reach reliable and consistent results and predictions due to the magnitude of the standard deviation.

3.2 Trend Data Preparation

A discretization procedure is applied to construct a trend data set using the original continuous data set. The mentioned procedure was presented in Table 3. The trend data set was formed from the data coded as 1 for the uptrend and 0 for the downtrend. The last 365 samples of the trend and continuous data sets constitute the reduced sets to test the effect of sample size on prediction performance.

While constructing the trend data, a highly similar procedure is applied for each feature to calculate and code up-down movement. Since the momentum value indicates the sustainability of the trend, it is assigned a value of 1 if it is

Table 3 Discretization procedure for trend data

Data	Value 1 (Upward trend)	Value 0 (Downward trend)
SMA	If < Closing price	If > Closing price
WMA	If < Closing price	If > Closing price
EMA	If < Closing price	If > Closing price
ATR	If > previous day	If < previous day
Bollinger Bands	If Closing price closer to lower band	If Closing price closer to upper band
Momentum	If positive	If negative
Stochastic %K	If > previous day	If < previous day
Stochastic %D	If > previous day	If < previous day
Larry Williams	If > previous day	If < previous day
RSI	If < 30 or 30-70 and > previous day	If > 70 or 30-70 and < previous day
MACD	If > previous day	If < previous day
A/D	If > previous day	If < previous day
Tweet	If > previous day	If < previous day
Google Trend	If > previous day	If < previous day

positive and 0 if it is negative. Since the stochastic indicators %K, %D, and Larry Williams(LW) indicator show the rate of price change and overbought if it is higher than the previous day. It means the trend is upwards, and the value is 1; if it is lower, it means the trend is downwards, and the value of 0 is assigned. While interpreting the RSI, if the value is less than 30, the trend is upward, and value 1 is assigned. If it is between 30-70 and if it is greater than the previous day, value 1 is assigned; if not, value 0 is assigned. If the RSI is greater than 70, the trend goes downwards, and the value 0 is assigned. For MACD, if it is higher than the previous day's value of 1, if it is lower, a value of 0 is assigned. The A/D indicator is assigned in the same way, 1 if the Oscillator value is higher than the previous day's value, and 0 if it is lower. The number of tweets sent per day and Google Search Trend are also compared with the previous day and if higher value 1, and if lower value 0 assigned.

During the preparation of the 2455 samples data set, the first 13-day values for RSI, EMA, SMA, WMA, StoK, StoD, the first 9-day values for ADOSC, the first 3-day values for LW, and the first 33-day values for MACD could not be calculated due to the formulation. Only tweet data has 56 missing values. Those missing values are replaced with the mean of the related feature. Prediction mechanism are presented in Fig 2.

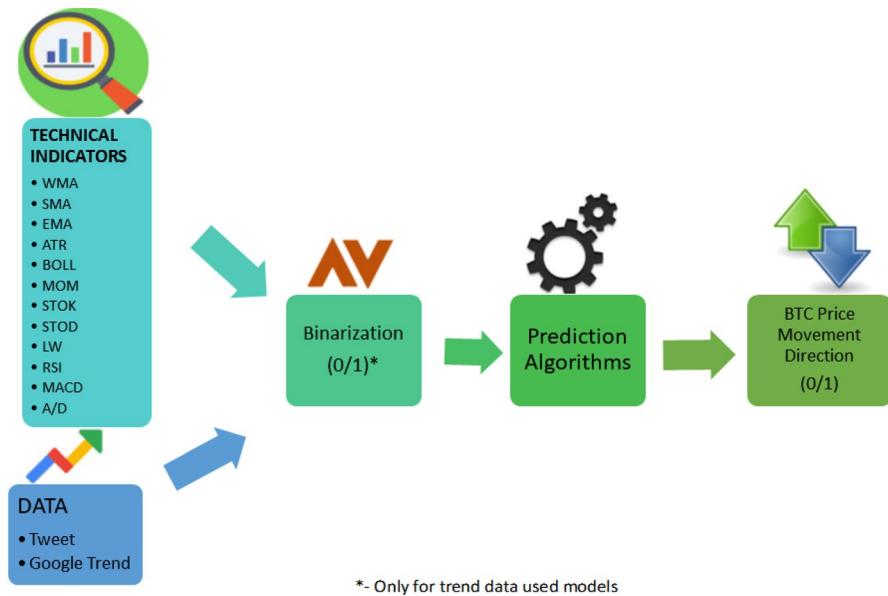


Fig. 2 Prediction mechanism

Table 4 Train-test decomposition accuracy change

Random state	Without indicators	With indicators
1	0.500000	0.726384
2	0.458514	0.754072
3	0.541444	0.728013
4	0.522855	0.721498
5	0.541486	0.714984
6	0.566864	0.708469
7	0.544412	0.703583
8	0.501256	0.716612
9	0.522684	0.726384
10	0.511264	0.714984
Mean	0.52110	0.72150

3.2.1 Verification Model Analyzes

Train-Test Split and Cross-Validation (CV) methods were tested to choose the most suitable validation model. In addition, tests were performed with only the (OHLC) data version of the continuous data set to measure the effect of the indicators. When it was tested the accuracy of the continuous data set with the KNN, an average accuracy of 0.72 was achieved, as presented in Table 4. It has been observed that the

Table 5 Train-test split classification Report

	Precision	Recall	F1-Score	N (Test)
0	0.68	0.72	0.70	282
1	0.77	0.73	0.75	332
Accuracy			0.72	
Makro Acc.	0.72	0.73	0.72	
Weighted avg.	0.73	0.73	0.73	

Table 6 10-fold cross validation classification report

	Precision	Recall	F Score	N(Test)
0	0.68	0.69	0.69	1120
1	0.74	0.73	0.73	1335
Accuracy			0.72	
Makro Acc.	0.72	0.72	0.72	
Weighted avg.	0.72	0.72	0.72	

accuracy rate changes with each constant in the data. This may mean we encounter different results each time the model is tested.

According to the data in the classification report in Table 5, when the Train-Test Split model is used with 25% split, 614 of the 2455 data set could be used for testing, and the average accuracy was 0.72.

In the CV model, the data set is divided by the number of layers determined. Each of these layers is used for training and validation. This model provides good results, especially in clustering problems Tarekegn (2020).

With the CV partitioning as seen in Table 6, it was achieved an accuracy of 0.72, as in the train-test method. Since the CV uses all data points, this method was used in suitable machine learning techniques such as the KNN, RF, XGB and SVM implementation. For the ANN and LSTM, training-test split method was used.

3.3 Problem and Methodology

Stock market or cryptocurrency price or price movement prediction as a financial time series problem is very important for investors or institutions due to the magnitude of the financial market capitalization. It is crucial to understand the behavior of the market and solve the mentioned problem in order to give some recommendations to investors and institutions. Fundamental analysis is a methodology for estimating the price of financial assets based on economic, social or political factors. Daily tweets, trend searches, hash rate, cost calculations, etc. can be combined with fundamental analysis to create a better dataset. For this reason, due to the claims of speculation in the Cryptocurrency exchange, daily tweet counts and Google trend data were included in the feature set and analyzed.

The main purpose of this study is to measure and compare the prediction performance of the selected ML algorithms using new data types. While running the

algorithms, continuous and trend data sets are constructed using the technical indicators, tweets, and Google trend data as features. Also, it is tried to test the effect of the number of samples for each data set on prediction performance. Therefore, reduced data sets were constructed using the last 1-year data for continuous and trend sets. The reason for using complete (2014-2021) and reduced (2020-2021) data sets with mentioned intervals is that they have similar volatility and behavior.

To avoid bias after tabulating calculations of technical indicators, it is essential to perform normalization to ensure the accuracy of model fitting. Recent studies have demonstrated the effectiveness of such data scaling models in enhancing model performance Ahsan et al. (2021). To mitigate the potential issue of unequal treatment of variables with different scales, the data sets were normalized before model fitting. These normalized data were used in the analysis of continuous data. Considering the data sizes, the attention was limited to machine learning techniques, and deep learning techniques were not used. ANN and recurrent neural networks have been added to machine learning models and interpreted by comparing training and estimation times, which are not frequently mentioned in other academic studies.

The methodology used in this paper is presented as a model training algorithm 1 as follows.

Algorithm 1 Model Training Algorithm

-
1. Load the historical data det
 2. Create indicators with Ta-Lib
 3. Get Tweet Counts and Google Trend Data with Beautiful Soup (Web Scraping)
 4. Combine data sets
 5. Binarize data sets to create trend data sets
 6. Test ML models for all datasets for best scores
-

Seven ML techniques were selected for this study, namely RF, KNN, XGB, SVM, NB, ANN, and LSTM, and these are presented below. During the study, parameter optimization was carried out according to the accuracy test results using trend and continuous data sets for each machine learning technique. Python scikit-learn, Keras, and Tensorflow libraries and their related plugins were used to implement the algorithms. Numpy library was used for mathematical calculations, and Matplotlib and Seaborn libraries were used for visualization and graphics. Previous research conducted in this area was evaluated and reviewed to determine the parameter levels that will be used for the most effective practices. To obtain the best model, the best parameter combination was found for each model separately. All machine learning models were tested using different parameters for each data set according to the best result of the train-test set; the same parameters were used to obtain validation set results.

The parameters that were tested in this study for each model are presented in Tables 7, 8, 9, 10, 11, 12 and 13.

Table 7 RF parameter levels

Parameters	Levels
Criterion	gini, entropy, log_loss
Estimators	10-500
Max Depth	None, 5, 10, 15
Cross Validation	10

Table 8 KNN parameter levels

Parameters	Levels
criterion	gini, entropy, log_loss
K Range	1-15
neighbors	None-5
algorithm	balltree, kdtree, brute
leaf size	30
Cross Validation	10

Table 9 XGB parameter levels

Parameters	Levels
Gradient	10-500
Max Depth	3,6,8,10
Eval. metric	logloss, rmse
Cross Validation	10

Table 10 SVM parameter levels

Parameters	Levels
Kernel	Poly, Linear
Degree	1-5
C range	0.025, 0.1, 0.5, 1, 10, 50, 100
Gamma	0.1, 0.5, 1, 2, 3, 4, 5
Cross Validation	10

Table 11 NB parameter levels

Parameters	Levels
Models	Gaussian, Bernoulli, Complement, Multi-nomial
Cross Validation	10

Table 12 ANN parameter levels

Parameters	Levels
Activation function	ReLU, tanh, sigmoid
Optimizer	SGD, ADAM, NADAM, ADAD, ADAG, FTRL, RMS
Loss	SCC, BC
Number of hidden layers	1, 2
Number of hidden neurons	10, 20, 30, 40, 50
Number of training epochs	20, 50, 100, 200, 500
Dropout rate	0.2, 0.3
Learning rate (η)	0.01–0.1
Momentum constant (c)	0.1–0.9

Table 13 LSTM parameter levels

Parameters -Levels	
Activation function	ReLU, tanh, sigmoid
Optimizer	SGD, ADAM, NADAM, ADAD, ADAG, FTRL, RMS
Number of hidden layers	1, 2
Number of hidden neurons	10, 20, 30, 40, 50
Number of training epochs	20,50,100,200,500
Dropout rate	0.2,0.3

3.3.1 Random Forest

RF algorithm is a bagging-based machine learning model Breiman (2001). The formulas used to measure at model were Gini impurity (Eq. 2), Entropy (Eq. 3), and Logloss (Eq. 4).

$$Gini = 1 - \sum_{i=1}^C (P_i)^2 \quad (2)$$

$$Entropy = \sum_{i=1}^C -P_i * \log_2(P_i) \quad (3)$$

$$Logloss = -\frac{1}{N} \sum_{i=1}^N [y_i \ln p_i + (1 - y_i) \ln(1 - p_i)] \quad (4)$$

3.3.2 K-Nearest Neighbors

The KNN classification model is obtained by measuring the distance in k of the newly entered data to the previously entered data for classification. The formulas were used to measure these distances are generally Euclidean (Eq. 5) distance, Minkowski (Eq. 6) distance, and Manhattan (Eq. 7) distance. The most important point is to determine the k value correctly Demir et al. (2019).

$$Euclidean = \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (5)$$

$$Minkowski = \left(\sum_{i=1}^k (|x_i - y_i|)^q \right)^{\frac{1}{q}} \quad (6)$$

$$Manhattan = \sum_{i=1}^k |x_i - y_i| \quad (7)$$

3.3.3 Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is an effective and optimized gradient tree boosting algorithm Sagi and Rokach (2021). XGBoost uses a weak learner to minimize the loss function and optimize the objective function (Luo et al., 2021). For any differentiable convex loss function $\iota = \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$, the objective function $\mathcal{O}(\phi)$ is defined as:

$$\mathcal{O}(\phi) = \sum_{i=1}^n l(\hat{y}_i, y_i) + \sum_k \Omega(f_k),$$

where $\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \|\omega\|^2$ is a regularization term, y_i are the labels, w is the weight of leaf, and T is the number of leaves of trees.

3.3.4 Support Vector Machine

SVM, proposed by Cortes and Vapnik (1995), is a machine learning model based on types of vectors that find a decision region between the two classes that are furthest from any place in the training records. The formula for SVM can be formulated as follows in Eq. 8.

$$\sum_{i \in SV} (a_i - a_i^*) K(x_i, x) + b \quad (8)$$

3.3.5 Naïve Bayes

NB is a generative learning algorithm widely applied to text classification and sentiment analysis machine learning problems. The classifier is based on the Bayes theorem. This approach to text classification uses the initial feature vector format, where the appearance of a word is modeled by a Gaussian (Eq. 9), multinomial (Eq. 10), Bernoulli (Eq. 11), or complement (Eq. 12) distribution Colianni et al. (2015).

$$\text{Gaussian} = P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \quad (9)$$

$$\text{Multinomial} = \hat{\theta}_{yi} = \frac{N_{yi} + a}{N_y + an} \quad (10)$$

$$\text{Bernoulli} = P(x_i|y) = P(x_i = 1|y)x_i + (1 - P(x_i = 1|y))(1 - x_i) \quad (11)$$

$$\text{Complement} = w_{ci} = \frac{w_{ci}}{\sum_j |w_{cj}|} \quad (12)$$

3.3.6 Artificial Neural Network

ANN is an algorithm inspired by the human brain's information processing system Cahyani et al. (2021).

3.3.7 Long Short-Term Memory

LSTM is a ML technique based on RNN architecture and learns by remembering features at random intervals. LSTM can be formulated as follows in Eq. 13:

$$\begin{aligned} \mathbf{i}_t &= \sigma(\mathbf{W}_{xi}x_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i) \\ \mathbf{f}_t &= \sigma(\mathbf{W}_{xf}x_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f) \\ \mathbf{c}_t &= \mathbf{f}_t\mathbf{c}_{t-1} + \mathbf{i}_t \tanh(\mathbf{W}_{xc}x_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c) \\ \mathbf{o}_t &= \sigma(\mathbf{W}_{xo}x_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + \mathbf{b}_o) \\ \mathbf{h}_t &= \mathbf{o}_t \tanh(\mathbf{c}_t) \end{aligned} \quad (13)$$

where x_t and h_t are the input and respectively the hidden state at time t , W_j , W_i and W_0 are the weight matrices of the corresponding components, b_f , b_i , b_0 are the corresponding bias parameters, and σ is the activation function.

Fig. 3 The components of a standard confusion matrix

		Prediction Outcome		Total
		p	n	
Actual Value	p'	True Positive	False Negative	P'
	n'	False Positive	True Negative	N'
Total		P	N	

3.4 Evaluation Criteria

There are some common evaluation metrics to evaluate the models that use only two classes according to the Fawcett (2006). We used accuracy, F1 Score and AUC metrics to evaluate the models (Figs. 3, 4, 5, 6, 7).

4 Results

Seven different ML techniques were trained with continuous (complete and reduced) and trend (complete and reduced) data sets separately. Estimation models were evaluated on test sets, and then, to validate the models' performance, the 30-day and 60-day validation sets were used. The model performance was evaluated using the accuracy, F1 Score, and AUC metrics. Paired t-tests were used to compare the best result for each row (shown in bold with a star) with the other results. Results that were not significantly worse ($p > 0.05$) than the best were also shown in bold, representing the top-performing group. The experimental result and model comparisons are presented in this section with statistical significance. The separation of the data in terms of accuracy was presented in Appendix A. Variation of Loss and accuracy in the training plot for ANN and LSTM were presented in Appendix D and E.

Importance levels of the features were compared for each ML model. The tests have shown that different ML techniques assigned different weights to the same features. For complete continuous data, Stoch K and LW indicators were the most important feature, as seen in Fig. 8, while Stoch K, Stoch D, and LW for continuous reduced data set as seen in Fig. 9. Stoch K, RSI, and LW were the most important indicators for trend data as seen in Fig. 10 and 11. The ablation test for ANN showed that the Bollinger Band was a stronger indicator for the complete continuous data set, while the Momentum was for a trend data set as seen in Figs. 12 and 13. Detailed results for each model and dataset are presented as follows.

4.1 Continuous Datasets Results

Estimated model performance and critical comparisons are presented in this section for continuous data sets using accuracy, F1 score, and AUC values. Experimental results are presented in Tables 14, 15 and 16. Evaluations and comparisons were made for the test and “30–60” days validation data sets separately to provide a clear understanding of all estimated models.

In terms of accuracy, F1 score, and AUC, ANN significantly outperformed the other five estimated ML models while showing similar performance with the LSTM for complete test data. The same results were obtained for reduced data sets but with lower accuracy levels. It means more data supports the test prediction performance

Table 14 Evaluation metrics of Continuous Test Data Set for all ML Models

	RF	KNN	XGBoost	SVM	NB	ANN	LSTM
<i>Models trained using complete data set</i>							
Accuracy	0.665	0.738	0.773	0.794	0.611	0.804*	0.742
F1 Score	0.678	0.772	0.792	0.817	0.655	0.832*	0.790
AUC	0.667	0.730	0.771	0.789	0.604	0.795*	0.728
<i>Models trained using reduced data set</i>							
Accuracy	0.575	0.608	0.674	0.710	0.586	0.780*	0.681
F1 Score	0.605	0.646	0.697	0.744	0.619	0.800*	0.632
AUC	0.573	0.604	0.673	0.705	0.583	0.777*	0.676

Table 15 Evaluation metrics of 30-day Validation Continuous Data Set for all ML Models

	RF	KNN	XGBoost	SVM	NB	ANN	LSTM
<i>Models trained using complete data set</i>							
Accuracy	0.838	0.733	0.774	0.838	0.548	0.871*	0.838
F1 Score	0.838	0.733	0.774	0.848	0.708	0.882*	0.857
AUC	0.846	0.736	0.781	0.840	0.500	0.869*	0.843
<i>Models trained using reduced data set</i>							
Accuracy	0.645	0.613	0.741	0.806	0.646	0.838*	0.800
F1 Score	0.685	0.625	0.733	0.812	0.620	0.857*	0.833
AUC	0.638	0.615	0.752	0.810	0.657	0.834*	0.799

Table 16 Evaluation metrics of 60-day Validation Continuous Data Set for all ML Models

	RF	KNN	XGBoost	SVM	NB	ANN	LSTM
<i>Models trained using complete data set</i>							
Accuracy	0.833	0.667	0.833	0.867	0.567	0.833	0.883*
F1 Score	0.848	0.667	0.848	0.882	0.717	0.844	0.898*
AUC	0.840	0.686	0.840	0.869	0.491	0.846	0.882*
<i>Models trained using reduced data set</i>							
Accuracy	0.800	0.700	0.766	0.867	0.616	0.883*	0.750
F1 Score	0.823	0.743	0.788	0.882	0.657	0.904*	0.819
AUC	0.800	0.691	0.771	0.869	0.614	0.871*	0.706

of the models. The best accuracy level was 0.804 for complete and 0.780 for reduced data sets. The lowest accuracy level of 0.575 belongs to the RF model for the test set.

When examining the 30-day validation data set estimations, the RF, SVM, ANN, and LSTM were in the top-performing group, while the ANN provided the best accuracy level of 0.871 for complete data. The ANN showed the best prediction performance for the reduced data set with 0.838 accuracy, which is lower than the accuracy of the complete data. The NB showed the lowest performance with 0.548 accuracy for complete data however, 0.646 accuracy for reduced data. This means that the NB may perform better estimations using less data, contrary to the LSTM.

The results obtained from the 60-day validation data models were slightly different from the 30-day validation estimated models. The ANN was not in the top-performing group for the complete data set, as seen in Table 16. The LSTM showed the best performance with the highest accuracy level of 0.883, but it was not significantly better than the SVM. The ANN and SVM were in the top-performing group for reduced data set estimation. This means that the LSTM model needs more data samples to provide more accurate predictions due to the complex mathematical structure. All selected techniques were trained using the same data sets, but as can be seen in Tables 15 and 16, out-of-sample prediction performance is different for these models. This means that additional analysis or predictions are necessary to validate the model's accuracy, reliability, and consistency.

Three different evaluation metrics were used to compare the model performance. The best model selected was generally the same for these three metrics. For example, the ANN model outperformed other models for the test data, as seen in Table 14 in terms of accuracy, F1 Score, and AUC values.

It is possible to make some general comments by examining the results of the analysis. The sample size significantly affects the prediction performance by decreasing the accuracy level. When the estimated models are examined, it can be easily seen that the accuracy level decreases in reduced models. One of the most important results obtained from this study is the ANN model showed significantly better results for small samples. The results, especially the 30-day and 60-day validation data prediction, showed that the estimated ML models did not overfit, and these models can be used for the selected problem properly.

4.2 Trend Datasets

Estimated model performance and critical comparisons are presented in this section for continuous data sets using accuracy, F1 score, and AUC values. Experimental results are presented in Tables 17, 18 and 19. Evaluations and comparisons were made for the test and "30-60" days validation data sets separately to provide a clear understanding of all estimated models.

While examining the test data estimations in terms of accuracy in Table 17, the LSTM showed the best performance with 0.959 accuracy. The RF, XGBoost, SVM, and ANN models were in the top-performing group with similar performance for the complete data set. The accuracy levels for each model were very

Table 17 Evaluation metrics of Trend Test Data Set for all ML Model

	RF	KNN	XGBoost	SVM	NB	ANN	LSTM
<i>Models trained using complete data set</i>							
Accuracy	0.954	0.944	0.954	0.957	0.928	0.951	0.959*
F1 Score	0.958	0.948	0.959	0.960	0.935	0.954	0.962*
AUC	0.954	0.945	0.953	0.957	0.927	0.954	0.960*
<i>Models trained using reduced data set</i>							
Accuracy	0.933	0.911	0.937	0.942*	0.900	0.890	0.923
F1 Score	0.938	0.917	0.940	0.945*	0.908	0.903	0.925
AUC	0.934	0.912	0.938	0.944*	0.901	0.884	0.922

Table 18 Evaluation metrics of 30-day Validation Trend Data Set for all ML Models

	RF	KNN	XGBoost	SVM	NB	ANN	LSTM
<i>Models trained using complete data set</i>							
Accuracy	0.965	0.965	0.960	0.966	0.959	0.961	0.967*
F1 Score	0.965	0.966	0.964	0.967	0.960	0.959	0.969*
AUC	0.965	0.966	0.964	0.966	0.960	0.959	0.970*
<i>Models trained using reduced data set</i>							
Accuracy	0.967*	0.935	0.967*	0.935	0.903	0.903	0.967*
F1 Score	0.969*	0.937	0.969*	0.937	0.903	0.909	0.969*
AUC	0.970*	0.941	0.970*	0.941	0.912	0.905	0.970*

Table 19 Evaluation metrics of 60-day Validation Trend Data Set for all ML Models

	RF	KNN	XGBoost	SVM	NB	ANN	LSTM
<i>Models trained using complete data set</i>							
Accuracy	0.950	0.967*	0.950	0.950	0.933	0.950	0.950
F1 Score	0.957	0.978*	0.957	0.957	0.938	0.955	0.957
AUC	0.951	0.978*	0.951	0.951	0.943	0.957	0.951
<i>Models trained using reduced data set</i>							
Accuracy	0.950*	0.917	0.950*	0.900	0.883	0.933	0.951*
F1 Score	0.957*	0.923	0.957*	0.909	0.888	0.939	0.959*
AUC	0.951*	0.929	0.951*	0.908	0.900	0.942	0.959*

close to each other; however, the NB model provided the lowest performance with 0.9283 accuracy. The results obtained from the reduced data are different in terms of ANN and the LSTM. These two models are not in the top-performing group, while the SVM showed the best performance with 0.942 accuracy. The RF and XGBoost were in the same group as the SVM. Surprisingly the ANN model showed the worst performance with 0.890 accuracy. The LSTM could not also placed in the top-performing group like ANN. The test data estimations highlighted that the discretization process of the input features improved the

prediction performance of selected ML techniques, increasing the accuracy level. Also, it highlighted that the number of samples significantly affects the prediction performance. It is believed that the discretization process of the input features made the problem easier.

For 30-day validation predictions, all estimated ML models provided no significant differences with almost equal accuracy, F1 Score, and AUC for the models trained using the complete data sets. The LSTM model showed the best performance with a 0.967 level of accuracy. However, the results were completely different in terms of reduced data set estimations. Reducing the number of samples decreased the accuracy level for the KNN, SVM NB, and ANN models. No significant fluctuations were detected in terms of the RF, XGBoost and LSTM. As can be seen in Table 19, the estimations were interestingly different. The most accurate model was the KNN, with a 0.967 accuracy for complete data, and outperformed the other six estimated models. Reducing the data samples did not cause significant changes in accuracy for the LSTM, XGBoost, and RF but decreased the accuracy of KNN estimations. The LSTM, XGBoost, and RF were in the top-performing group in reduced data-trained models.

4.3 Model Comparison for Training and Testing

Model accuracy is crucial, but it is not the only metric for choosing the best one to make more profit. More time-consuming models would not be suitable for financial investors many times because sometimes even seconds gain great importance in stock market or cryptocurrency trading transactions. For this reason, the prediction model must be chosen correctly to make the right decision, in other words, to make more money or profit.

As seen in Tables 20 and 21, calculated elapsed time values may look like they are not important due to their magnitude, but if a system uses a big data set containing thousands of features and billions of samples may need a lot of time to provide accurate predictions. The most important thing at this point is to find more suitable models regarding accuracy and elapsed time.

The LSTM and ANN models need more time for training for both continuous and trend data sets. These models need more data points for training because of the

Table 20 Elapsed time (Continuous Data Set)

	Complete data set		Reduced data set	
	Training time	Prediction time	Training time	Prediction time
LSTM	33,972134	3,253587	49,599901	2,938089
ANN	7,369106	0,169317	14,201830	0,464278
RF	4,980866	0,072236	0,056011	0,008095
SVM	0,427818	0,007975	0,024007	0,008005
XGBoost	0,120009	0,008000	0,032001	0,008003
KNN	0,008016	0,015995	0,008002	0,016004
NB	0,008000	0,008002	0,008004	0,005010

Table 21 Elapsed time (Trend Data Set)

	Complete data set		Reduced data set	
	Training time	Prediction time	Training time	Prediction time
LSTM	32,262309	0,120025	32,331675	2,616832
ANN	7,879686	0,194058	3,611537	0,162802
RF	0,415293	0,032011	0,413021	0,030233
SVM	0,360201	0,008007	0,008004	0,008002
XGBoost	0,402129	0,016006	0,032004	0,008001
KNN	0,032311	0,008003	0,008003	0,008005
NB	0,008005	0,007996	0,800000	0,008001

mathematical structure and complexity. As can be seen in Table 20, the LSTM and ANN need more time for training using reduced data sets due to the model structure and problem complexity in continuous data sets. A small sample size is not suitable for these models. The LSTM and ANN techniques are mentioned as deep learning models that use big data in many recent studies, such as Nita (2016), Wang et al. (2020), Sismanoglu et al. (2019). Using small samples for training and testing may cause overfitting for this kind of technique. It is possible to train the KNN and NB in a short time, but the accuracy level of estimations may not satisfy the user or researcher. However, the XGBoost model can be a better option for accurate prediction in a shorter time for trend data sets, in other words, for easier problems to solve.

5 Conclusions and Discussion

In the world of trading, price movement prediction has historically been important. Thanks to its trading volume, Bitcoin, the first cryptocurrency, is frequently analyzed by researchers and investors. This paper investigated whether to improve the prediction performance of ML techniques using different types of data, namely continuous and trend. Also, it was investigated whether the prediction performance may have been affected by the data size. To this aim, seven ML techniques were trained and tested using different data types and sizes. Daily Google Search Trends and daily tweets on Bitcoin were used, in addition to the most commonly used technical indicators as input features. To obtain the best parameter set for each ML technique, specific parameter levels were used as presented in Sect. 3.3. The optimized settings for trend and continuous data were almost different for each data set. Therefore, the importance of optimizing the model according to the structure of the data instead of a general setting has once again emerged in the process.

While making critical comparisons to evaluate the model performance, significant differences were found between the prediction models. The ANN and LSTM models consistently outperformed the others in terms of accuracy, F1-score, and AUC for both complete and reduced datasets. The ANN, which was a top performer in the 30-day validation, did not maintain its position in the 60-day validation for complete datasets. This could be due to overfitting or sensitivity to data distribution, although it was checked. The LSTM's performance was comparable to the ANN, suggesting its

suitability for time-series or sequential data. The RF and NB models appear to be more sensitive to data volume. RF's performance dropped significantly with reduced data, while NB's performance improved. More data generally improves model performance. This is evident in the higher accuracy levels achieved with complete datasets compared to reduced ones. The RF model's performance was highly dependent on data volume. Its accuracy dropped significantly with reduced data, suggesting it might be more susceptible to overfitting. The NB model showed an inverse relationship between data volume and performance. It performed better with reduced data, possibly due to its simplicity and reduced susceptibility to overfitting with smaller datasets.

For this specific prediction task, the ANN or LSTM would be the preferred choices due to their consistently high performance across different data volumes. The LSTM's superior performance with 60-day validation highlights its ability to benefit from larger datasets, likely due to its complex mathematical structure. The impact of data volume on model performance should be carefully considered when designing experiments. More data generally leads to better results, but there might be exceptions like the NB model. The sensitivity of different models to data volume should be evaluated to understand their limitations and potential biases.

Analyzing the importance of different features could provide insights into which factors are most influential in the predictions. Optimizing hyperparameters for each model could potentially improve their performance further. Combining multiple models (e.g., using ensemble models) might yield even better results. It is thought that models using pipelines can provide better performance than models using static singular prediction. Dong et al. (2020). Overall, the results highlight the importance of thorough model validation and the consideration of factors like data volume and model complexity when selecting a suitable model.

Since cryptocurrency exchanges stream continuously on a 7-24 basis, instant trading is more important than daily closing prices. It would be useful to investigate the effects of the models that yielded efficient results in this study on different frequency data sets such as hours and minutes. The information obtained from the estimated models can help investors take short/long-term positions, especially in future markets, and develop novel structures for buy-hold-sell signal generation. Since deep learning models give better results with larger data, they may provide better predictions and bring the studies to a higher level. Text classification, natural language processing, transformers, attention mechanisms, and large language models may also provide better results. It is also thought that these suggestions can be considered as limitations of the research. With the approval of spot Bitcoin Exchange Traded Funds by the US Securities and Exchange Commission on January 10, 2024, it is thought that studies in this field will increase rapidly.

Appendices

Appendix A: Separation of Data According to Accuracy

See Figs. 4, 5, 6, 7.

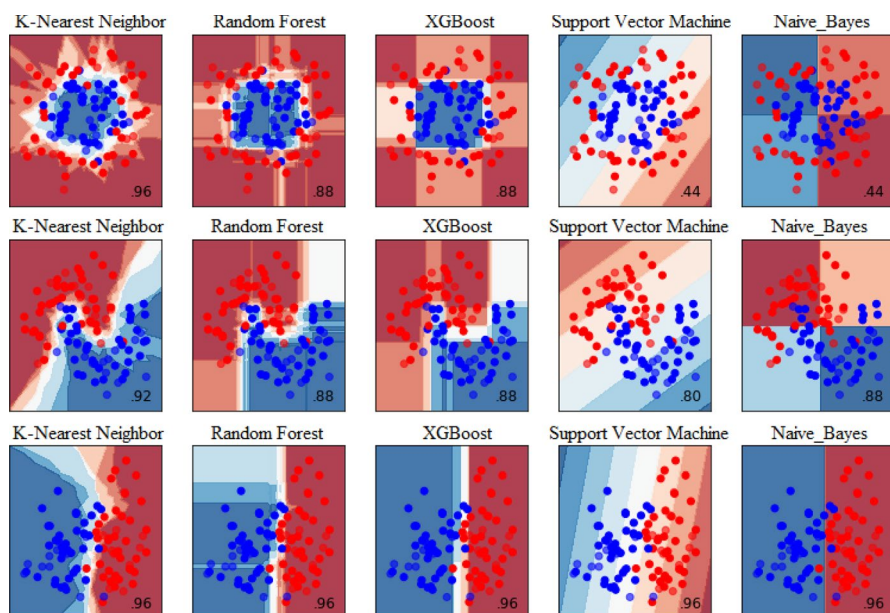


Fig. 4 Separation of all continuous data according to accuracy

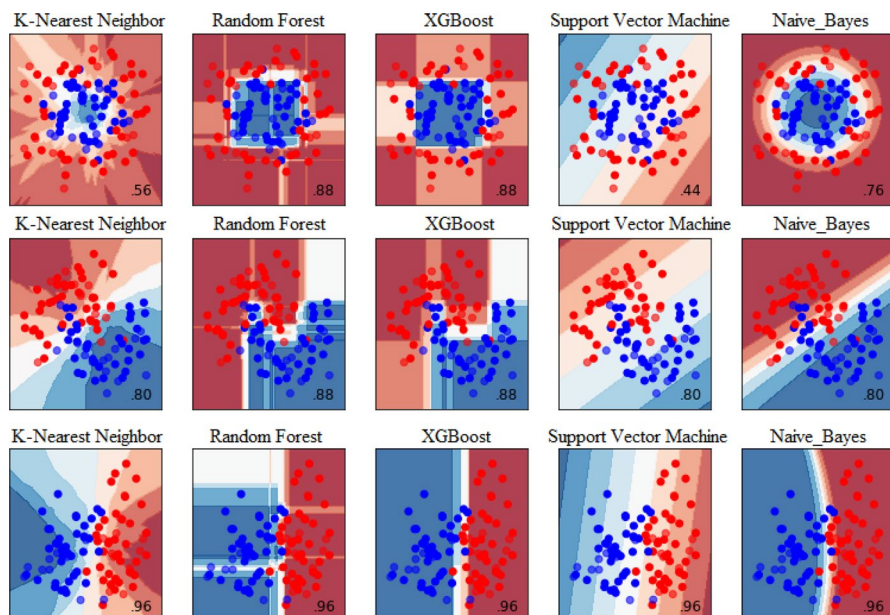


Fig. 5 Separation of 1 year continuous data according to accuracy

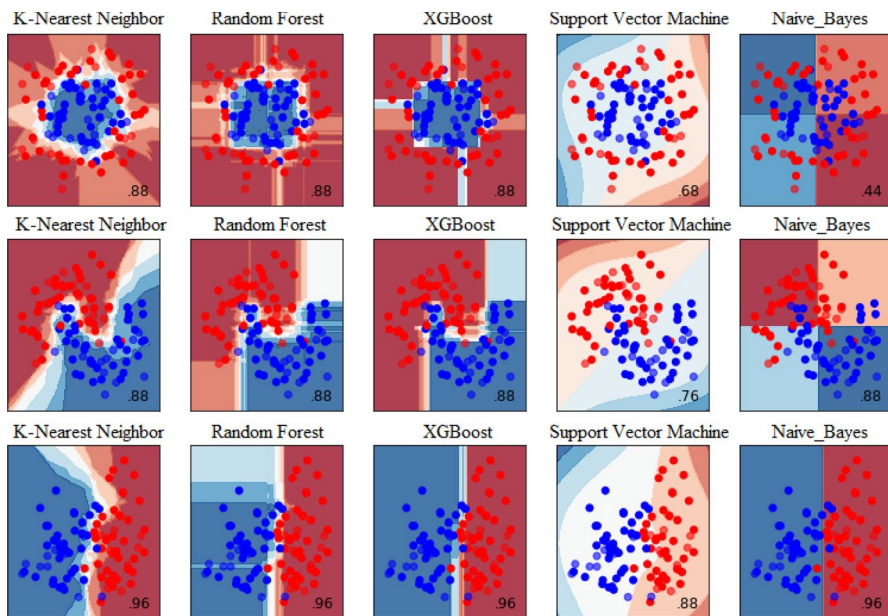


Fig. 6 Separation of all trend data according to accuracy

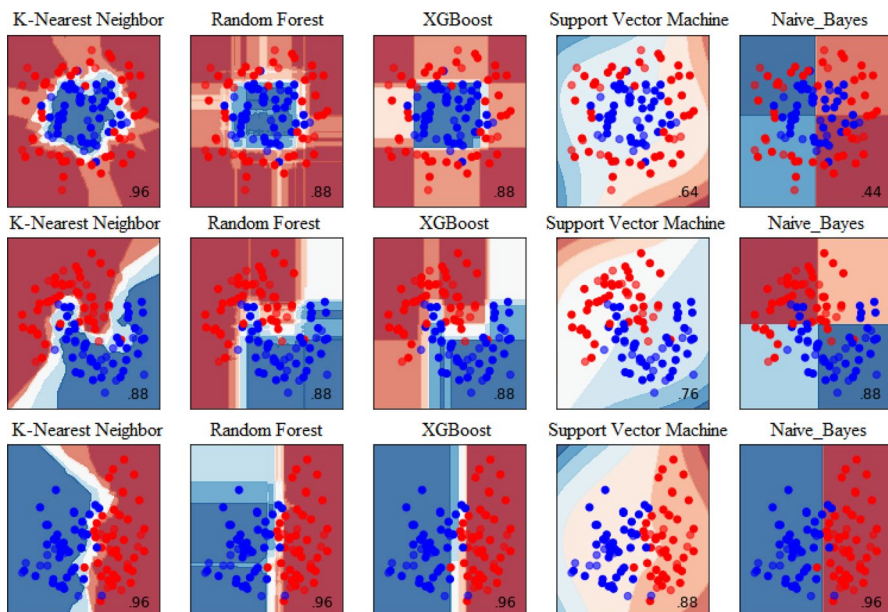


Fig. 7 Separation of 1 year trend data according to accuracy

Appendix B: Feature Importance

See Figs. 8, 9, 10 and 11.

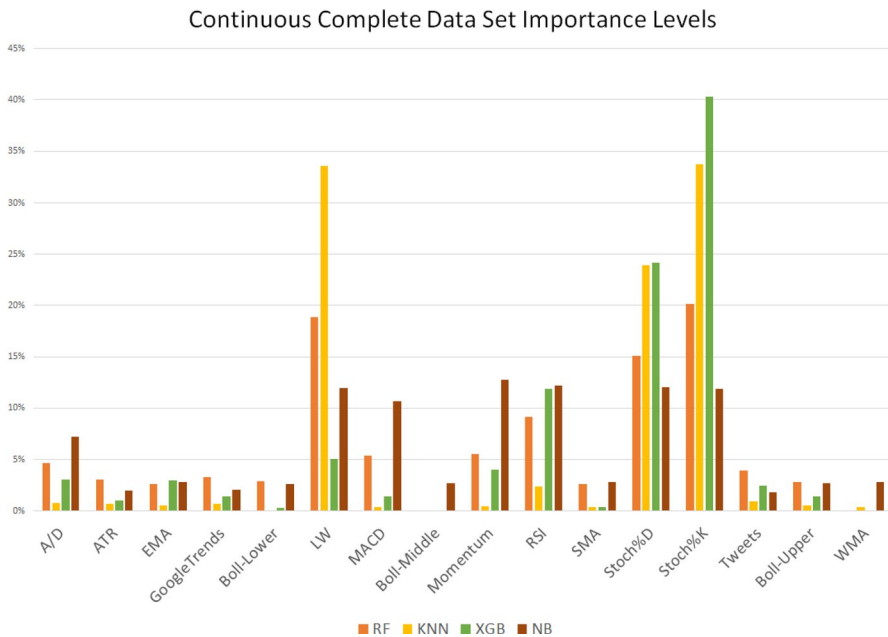


Fig. 8 Importance Continuous Complete Data Set



Fig. 9 Continuous Reduced Data Set Importance Levels

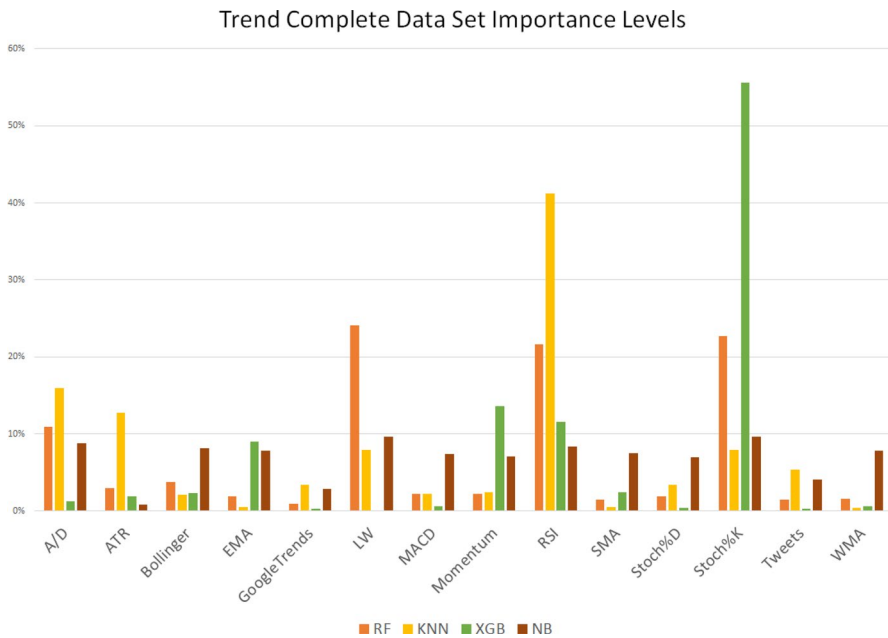


Fig. 10 Trend Complete Data Set Importance Levels

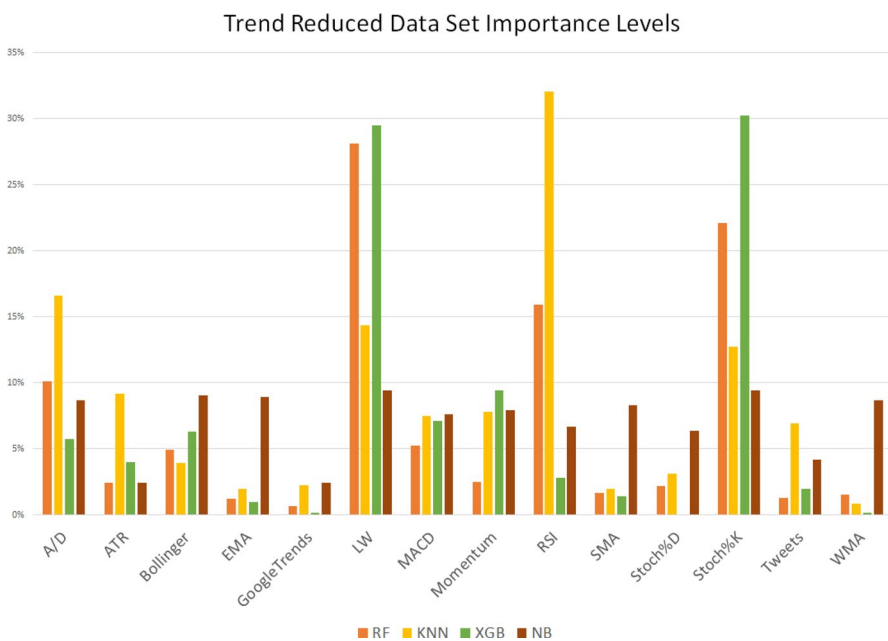


Fig. 11 Trend Reduced Data Set Importance Levels

Appendix C: Ablation Tests

See Figs. 12 and 13.

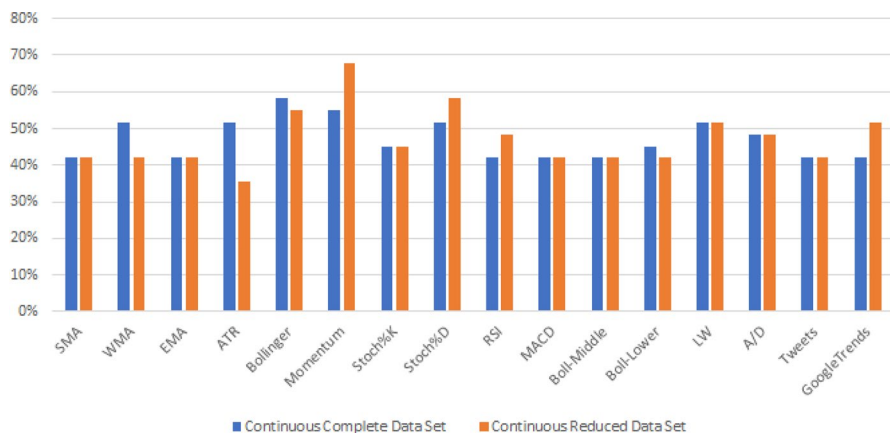


Fig. 12 Ablation Test Results for ANN Continuous Data Set

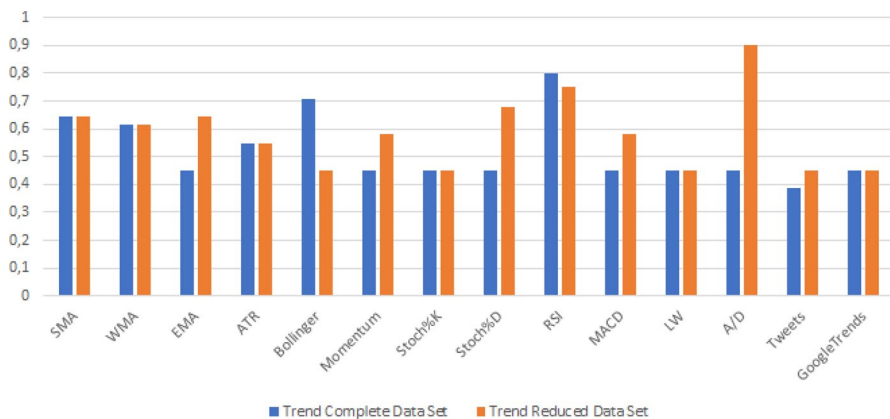


Fig. 13 Ablation Test Results for ANN Trend Data Set

Appendix D: Training Loss and Accuracy for ANN

See Figs. 14 and 15.

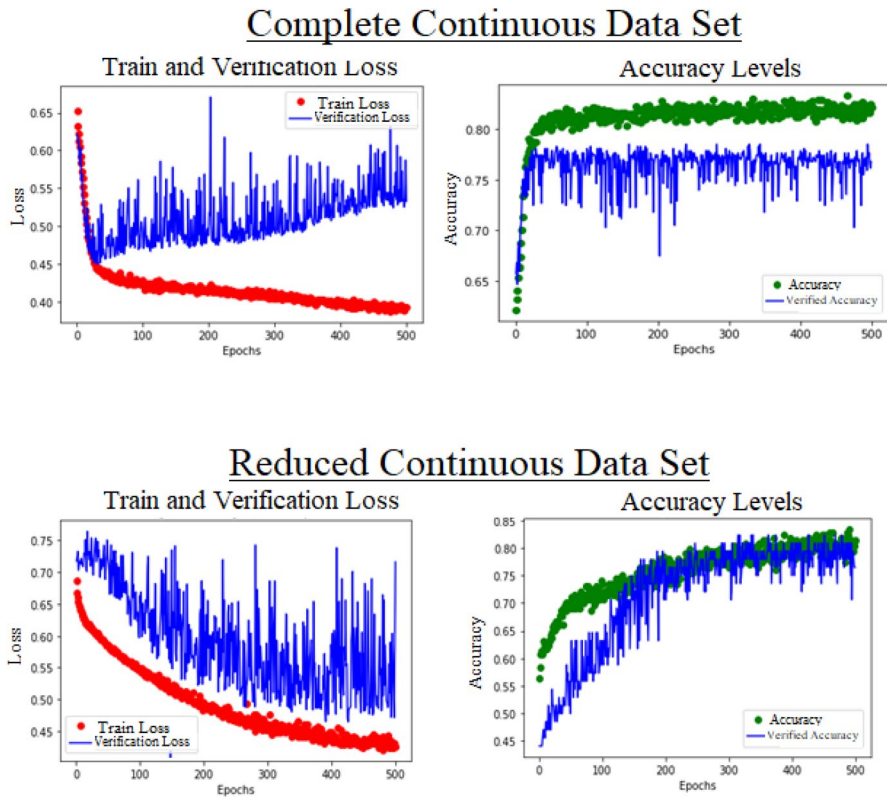
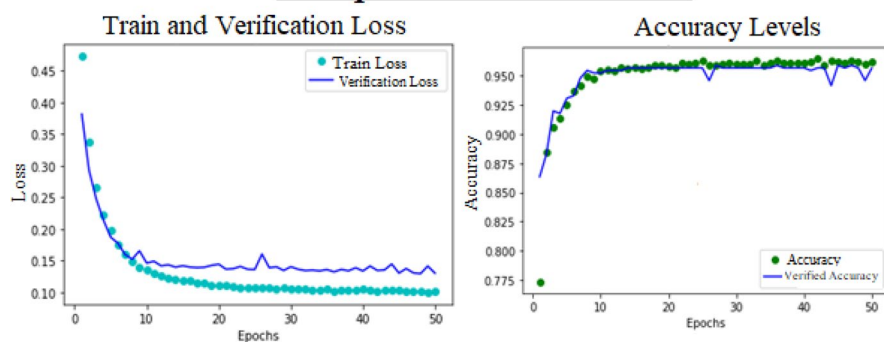


Fig. 14 ANN Continuous Data Loss and ACCs

Complete Trend Data Set



Reduced Trend Data Set

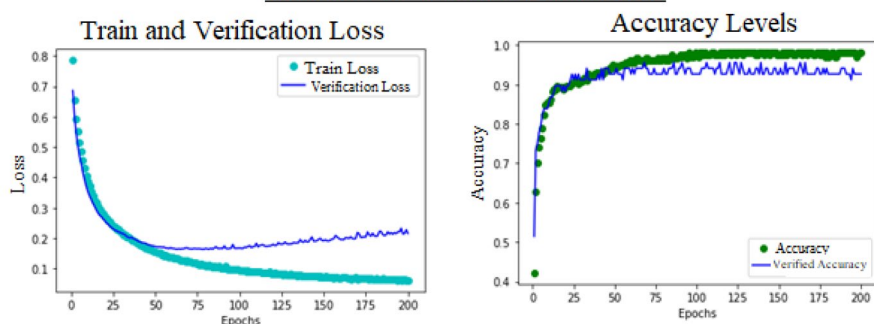


Fig. 15 ANN Trend Data Loss and ACCs

Appendix E: Training Loss and Accuracy for LSTM

See Figs. 16 and 17

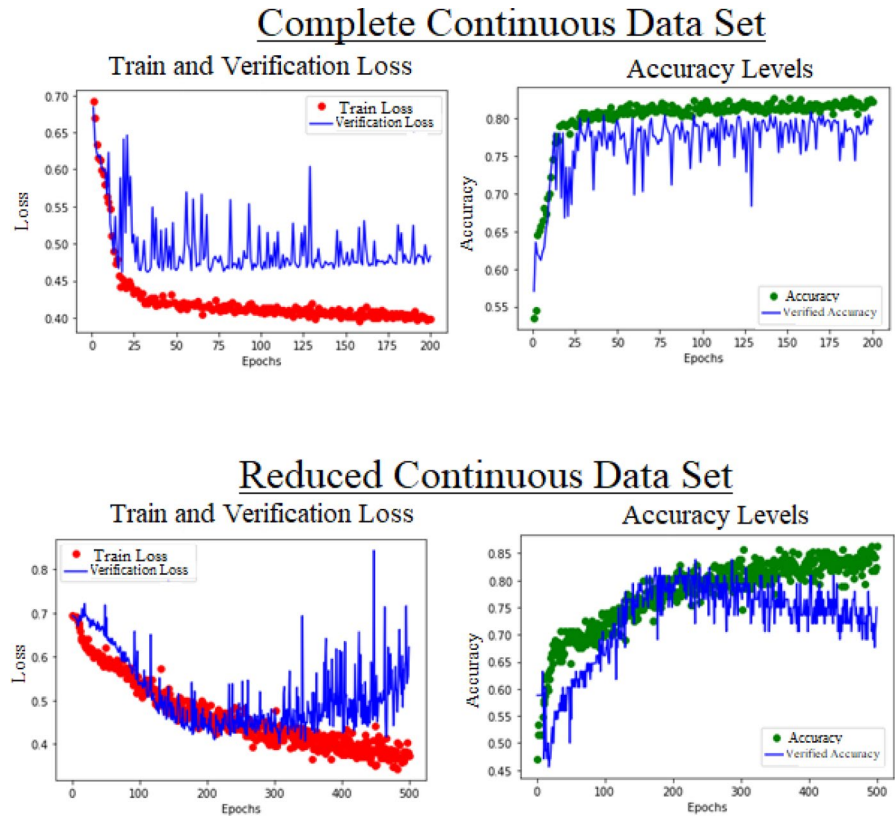
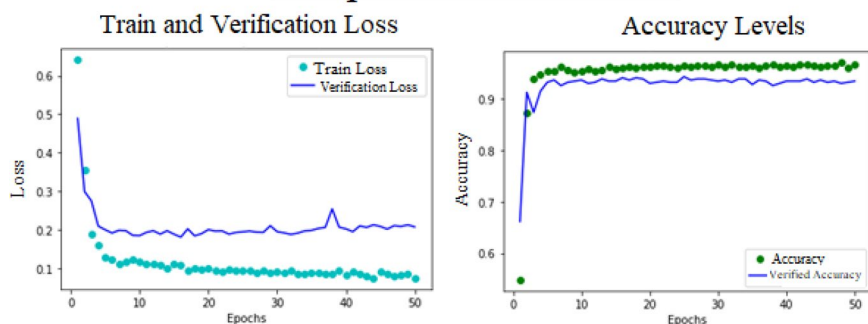


Fig. 16 LSTM Continuous Data Loss and ACCs

Complete Trend Data Set



Reduced Trend Data Set

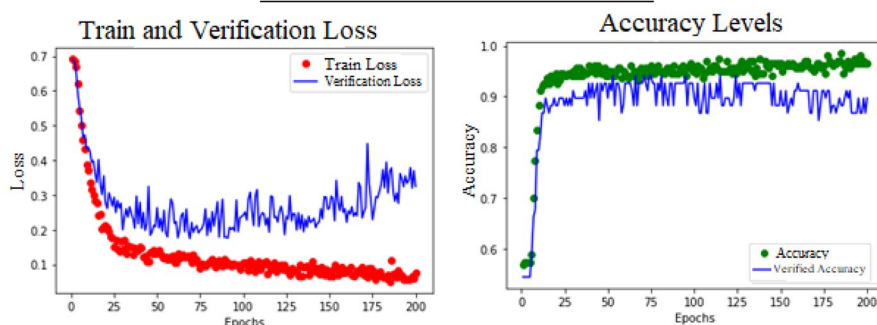


Fig. 17 LSTM Trend Data Loss and ACCs

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). We declare that we have no relevant or material financial interests that relate to the research described in this paper. The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Declarations

Conflict of interest The authors have not disclosed any Conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abraham, J., Higdon, D., Nelson, J., & Ibarra, J. (2018). Cryptocurrency price prediction using tweet volumes and sentiment analysis. *SMU Data Science Review*, 1(3), 1.
- Ahsan, M., Saha, K., & Zahed (2021). Effect of data scaling methods on machine learning algorithms and model performance. *Technologies*, 9(3).
- Breiman, L. (2001). *Random forests*. *Machine learning*, 45, 5–32.
- Cahyani, N. D. W., Nuha, & H. H., et al. (2021). Ransomware detection on bitcoin transactions using artificial neural network methods. In: *2021 9th International conference on information and communication technology (ICOICT)*, pp. 1–5. IEEE.
- Chen, W., Xu, H., Jia, L., & Gao, Y. (2021). Machine learning model for bitcoin exchange rate prediction using economic and technology determinants. *International Journal of Forecasting*, 37(1), 28–43.
- Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics*, 365, 112395.
- Colianni, S., Rosales, S., & Signorotti, M. (2015). Algorithmic trading of cryptocurrency based on twitter sentiment analysis. *CS229 Project*, 1(5):1–4.
- Cortes, C., & Vapnik, V. (1995). *Support vector machine*. *Machine learning*, 20(3), 273–297.
- Dash, R., & Dash, P. K. (2016). A hybrid stock trading framework integrating technical analysis with machine learning techniques. *The Journal of Finance and Data Science*, 2(1), 42–57.
- Delfabbro, P., King, D., & Williams, J. (2021). The psychology of cryptocurrency trading: Risk and protective factors. *Journal of Behavioral Addictions*, 10.
- Demir, A., Akılotu, B. N., Kadiroğlu, Z., & Şengür, A. (2019). Bitcoin price prediction using machine learning methods. In: *2019 1st International informatics and software engineering conference (UBMYK)*, pp. 1–4. IEEE.
- Dong, Y., Yan, D., Almudaifer, A. I., Yan, S., Jiang, Z., & Zhou, Y. (2020). Belt: A pipeline for stock price prediction using news. In: *2020 IEEE international conference on big data (Big Data)*, pp. 1137–1146. IEEE.
- Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, 27(8), 861–874. ROC Analysis in Pattern Recognition.
- Goutte, S., Le, H. V., Liu, F., & Von Mettenheim, H. J. (2023). Deep learning and technical analysis in cryptocurrency market. *Finance Research Letters*, 54, 103809.
- Huang, J.-Z., Huang, W., & Ni, J. (2019). Predicting bitcoin returns using high-dimensional technical indicators. *The Journal of Finance and Data Science*, 5(3), 140–155.
- Julian, T., Devrison, T., Anora, V., & Suryaningrum, K. M. (2023). Stock price prediction model using deep learning optimization based on technical analysis indicators. *Procedia Computer Science*, 227, 939–947.
- Khosravi, G. (2023). Novel insights into the modeling financial time-series through machine learning methods: Evidence from the cryptocurrency market. *Expert Systems with Applications*, 234, 121012.
- Kim, H. J., Hong, J. S., Hwang, H. C., Kim, S. M., & Han, D. H. (2020). Comparison of psychological status and investment style between bitcoin investors and share investors. *Frontiers in Psychology*, 11.
- Kim, H.-M., Bock, G.-W., & Lee, G. (2021). Predicting ethereum prices with machine learning based on blockchain information. *Expert Systems with Applications*, 184, 115480.
- Lahmiri, S., & Bekiros, S. (2020). Intelligent forecasting with machine learning trading systems in chaotic intraday bitcoin market. *Chaos, Solitons and Fractals*, 133, 109641.
- Linardatos, Kotsiantis, S. (2020). *Bitcoin price prediction combining data and text mining*, pp. 49–63. Springer Singapore, Singapore.
- Luo, J., Zhang, Z., Fu, Y., & Rao, F. (2021). Time series prediction of covid-19 transmission in america using lstm and xgboost algorithms. *Results in Physics*, 27, 104462.
- Madan, I., Saluja, S., & Zhao, A. (2015). Automated bitcoin trading via machine learning algorithms. URL: <http://cs229.stanford.edu/proj2014/Isaac%20Madan>, 20.
- Mangla, N., Bhat, A., Avabratha, G., & Bhat, N. (2019). Bitcoin price prediction using machine learning. *International Journal of Information And Computing Science*, 6(5), 318–320.
- Matta, M., Lunesu, I., & Marchesi, M., et al. (2015). Bitcoin spread prediction using social and web search media. In: *UMAP workshops*, pp. 1–10.

- Nita, S. (2016). Machine learning techniques used in big data. *Scientific Bulletin of Naval Academy*, 19.
- Pabuçcu, H., Ongan, S., & Ongan, A. (2020). Forecasting the movements of bitcoin prices: an application of machine learning algorithms. *Quantitative Finance and Economics*, 4(4), 679–692.
- Rahaman, A., Bitto, A. K., Biplob, K., Bijoy, M. H. I., Jahan, N., & Mahmud, I. (2022). Bitcoin trading indicator: a machine learning driven real time bitcoin trading indicator for the crypto market. *Bulletin of Electrical Engineering and Informatics*, 12, 1762–1772.
- Sagi, O., & Rokach, L. (2021). Approximating xgboost with an interpretable decision tree. *Information Sciences*, 572, 522–542.
- Shah, D., Isah, H., & Zulkernine, F. (2019). Stock market analysis: A review and taxonomy of prediction techniques. *International Journal of Financial Studies*, 7(2), 26.
- Shah, D. & Zhang, K. (2014). *Bayesian regression and bitcoin*. In *2014 52nd annual Allerton conference on communication, control, and computing (Allerton)*, pp. 409–414. IEEE.
- Sismanoglu, G., Onde, M. A., Kocer, F., & Sahingoz, O. K. (2019). Deep learning based forecasting in stock market with big data analytics. In: *2019 scientific meeting on electrical electronics bio-medical engineering and computer science (EBBT)*, pp. 1–4.
- Sossi-Rojas, S., Velarde, G., & Zieba, D. (2023). A machine learning approach for bitcoin forecasting. *Engineering Proceedings*, 39(1).
- Tarekegn, A. N. (2020). Cross-validation approach to evaluate clustering algorithms: An experimental study using multi-label datasets. *SN Computer Science*, 1(5), 263.
- Vargas, M. R., dos Anjos, C. E. M., Bichara, G. L. G., & Evsukoff, A. G. (2018). Deep learning for stock market prediction using technical indicators and financial news articles. In: *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8.
- Wang, C., Du, W., Zhu, Z., & Yue, Z. (2020). The real-time big data processing method based on lstm or gru for the smart job shop production process. *Journal of Algorithms and Computational Technology*, 14, 1748302620962390.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Hulusi Mehmet Tanrikulu¹  · Hakan Pabuccu¹ 

✉ Hulusi Mehmet Tanrikulu
hmtanrikulu@gmail.com

Hakan Pabuccu
hakanpabuccu86@gmail.com

¹ Faculty of Economics and Administrative Sciences, Bayburt University, Bayburt, Turkey