Contents lists available online at TALENTA Publisher

DATA SCIENCE: JOURNAL OF COMPUTING AND APPLIED INFORMATICS (JoCAI)

Journal homepage: https://talenta.usu.ac.id/JoCAI

Time Series Prediction of Bitcoin Cryptocurrency Price Based on Machine Learning Approach

Eddie Ngai[1], Salwani Abdullah[2]*, Mohd Zakree Ahmad Nazri[3], Nor Samsiah Sani [4] and Zalinda Othman[5]

Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia

email: [1]a180095@siswa.ukm.edu.my, [2]salwani@ukm.edu.my, [3]zakree@ukm.edu.my, [4]norsamsiahsani@ukm.edu.my, [5]zalinda@ukm.edu.my,

ARTICLE INFO

Article history:
Received 15 June 2023
Revised 22 June 2023
Accepted 17 July 2023
Published online 31 July 2023

Keywords:
Bitcoin
Cryptocurrency
Machine Learning

Corresponding Author:
salwani@ukm.edu.my

ABSTRACT

Over the past few years, Bitcoin has attracted the attention of numerous parties, ranging from academic researchers to institutional investors. Bitcoin is the first and most widely used cryptocurrency to date. Due to the significant volatility of the Bitcoin price and the fact that its trading method does not require a third party, it has gained great popularity since its inception in 2009 among a wide range of individuals. Given the previous difficulties in predicting the price of cryptocurrencies, this project will be developing and implementing a time series approach-based solution prediction model using machine learning algorithms which include Support Vector Machine Regression (SVR), K-Nearest Neighbor Regression (KNN), Extreme Gradient Boosting (XGBoost), and Long Short-Term Memory (LSTM) to determine the trend of bitcoin price movement, and assessing the effectiveness of the machine learning models. The data that will be used is the close prices of Bitcoin from the year 2018 up to the year 2023. The performance of the machine learning models is evaluated by comparing the results of R-squared, mean absolute error (MAE), mean squared error (RMSE), and also through a visualization graph of the original close price and predicted close price of Bitcoin in a dashboard. Among the models compared, LSTM emerged as the most accurate, followed by SVR, while XGBoost and KNN exhibited comparatively lower performance.

IEEE style in citing this article:
E. Ngai, N.S. Sani, S. Abdullah, Z. Othman and M.Z.A. Nazri, "Time Series Prediction of Bitcoin Cryptocurrency Price Based on Machine Learning Approach," *DATA SCIENCE: JOURNAL OF COMPUTING AND APPLIED INFORMATICS (JoCAI)*, vol. 7, no. 2, pp. 81-95. 2023.

1. Introduction

In 2009, Bitcoin was introduced to the public. Since then, it has become the most famous cryptocurrency in the world. According to [1], about 18 million Bitcoins (BTC) are sold and exchanged. Satoshi Nakamoto, the pseudonym of Bitcoin's developer, declared that Bitcoin's purpose was to function as a decentralized electronic payment system based on cryptographic evidence rather than trust [2]. High price volatility implies that certain steps need to be taken to accurately predict bitcoin prices [3]. Investors are usually concerned about asset price volatility because price changes result in

e- ISSN: 2580-829X | p- ISSN: 2580-6769

immediate capital gains and losses. Given the volatility, it is always challenging to predict the bitcoin price. [4] found that accurate forecasting of bitcoin prices can provide decision support to investors and provide reference to the government to enact regulatory policies.

Accurately predicting Bitcoin price movements is challenging due to the volatile nature of the cryptocurrency market. Inaccurate forecasting will adversely affect investors, businesses, and organizations. Figure 1 shows a historical time series plot of Bitcoin prices from 2010 to 2022.

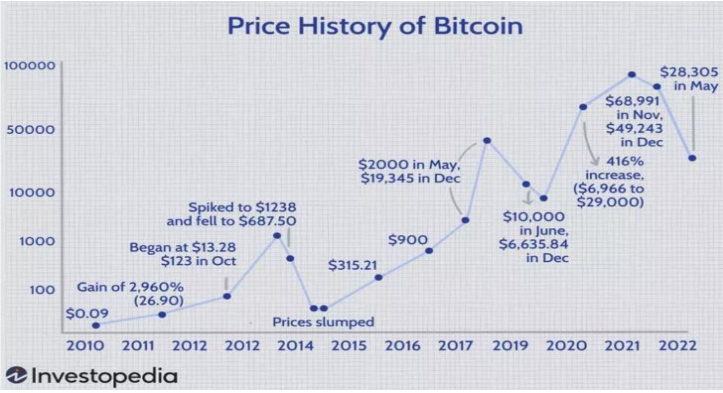

Figure 1. Time series plot for Bitcoin price history [5].

Based on this diagram, the increase and decrease in the price of Bitcoin are clearly shown by the time series plot method. There are certain years where the price of Bitcoin drops sharply but then rises again within a few years. However, the predictive potential of time series is a major concern, especially for Bitcoin price trends. An incorrect forecast will have a detrimental effect on investors, enterprises, and organizations.

Machine learning focuses on using data and algorithms to imitate how humans learn and gradually improve its accuracy. Selecting the most suitable machine learning model is vital in forecasting problems because the wrong algorithm selection can lead to suboptimal forecasting results. The same is true in the problem of predicting Bitcoin price movements. This will significantly impact investors' choice to buy and sell investment instruments at the micro level and have a macro effect on a country's economic policy [6]. With that in mind, designing machine learning models capable of producing high-accuracy results is essential. Many researchers have investigated the use of machine-learning approaches in currency prediction problems. Interested readers can find more [7 ; 8 ; 9 ; 10].

This work aims to develop and compare machine learning models and tests on the latest Bitcoin market data for time series forecasting. The contributions of this work are:

1) Evaluate the impact of different training-test data split ratios on the prediction results of machine learning models.
2) Evaluate the effect of hyperparameter tuning on the prediction results of machine learning models.
3) Comparing the performance and results of different machine learning models on Bitcoin time series forecasting.
4) Testing the applicability of Bitcoin time series forecasting models on different cryptocurrency data.

This work focuses on developing and optimizing machine learning algorithms for Bitcoin time series forecasting. The work in this project will involve several machine learning techniques such as Support Vector Machine Regression (SVR) [11], K-Nearest Neighbor Regression (KNN) (), Extreme Gradient Boosting (XGBoost) [12], and Long Short Term Memory (LSTM) [13] and tested on Bitcoin historical data sets, Ethereum, Ripple and Litecoin in the currency of the United States Dollar. The dataset was obtained from the Yahoo Finance website: https://finance.yahoo.com/

The remainder of the paper is organized as follows: Section 2 explains the Bitcoin prediction methodology and model development. Experimental results are discussed in Section 3 and brief concluding comments in Section 4.

2. Method

2.1 Methodology

This work is based on the CRISP-DM methodology [14 ; 15] that consists of 6 phases, i.e., business understanding, data Understanding, Data Preparation, Modeling, Evaluation), and Deployment. In this work, each phase aims to:

1) Business Understanding –Develop a machine learning model for a time series Bitcoin prediction problem
2) Data Understanding – Collect and analyze the data, and identify the features, form, and correlation between data. The data source used in this study is the historical data of Bitcoin in the currency of the United States Dollar obtained from the Yahoo Finance website, as many as 1827 records of the daily price of Bitcoin from May 7, 2018, to May 7, 2023.
3) Data preparation – Provide clean data for the machine learning model development that involves data discovery, data cleaning, attribute correlation, data scaling, and sliding window technique [16].
4) Modelling – four models are developed, i.e., Support Vector Regression (SVR), K Nearest Neighbor (K-NN), XGBoost, and Long Short Term Memory using Pythonation – RMSE, MAE and R2 Score are used as evaluation metrics [17].

2.2 Model Development

1) The preprocessed data set will be divided into three different ratios. The first ratio is 70% training data set and 30% test data set. This means that 70% of the data set will be used to train the model, while the other 30% will be used to test the performance of the trained model. The second ratio is 80% training data set, 20% test data set, and the third ratio is 90% training data set and 10% test data set.
2) Next, the dataset will be transformed using the sliding window technique for time series forecasting [18].
3) Four machine learning models suitable for time series forecasting, namely SVR, K-NN, XGBoost, and LSTM, will be implemented. Experiments will be performed by conducting a series of experiments using different training and test data division ratios, such as 70-30, 80-2,0, and 90-10. A hyperparameter tuning, i.e. grid search, will be implemented to optimize model performance. Appropriate hyperparameter values for each model will be determined based on negative mean squared error values. Descriptions for the type of hyperparameters for each model are found in Table he paragraph styles defined in the manuscript template can be seen in Table 1.

Table 1. Hyperparameter Description for Each Model

Model	Hyperparameter	Description
SVR	*C*	The regularization parameter controls the trade-off between parsimoniousness and matching of the training data
	Gamma	Kernel coefficients for the 'rbf' kernel
	Epsilon	The maximum allowable deviation or tolerance for regression predictions
KNN	*n_neighbors*	Number of neighbours to consider for classification/regression
	Weight	Weight function used in the prediction
XGBoost	*Colsample Bytree*	Column subsample ratio when building each tree
	Learning rate	Step size shrinkage used in the scaling process
	MaxDepth	Ma0ximum depth of the tree
	n_estimators	Number of upgrade rounds
LSTM	*Learning rate*	Step size for gradient descent optimization
	Units	Number of LSTM units in the layer

A comparative analysis will be conducted to compare the results obtained from experiments with and without hyperparameter tuning. The effect of different split ratios of test and training datasets on model performance will be examined.

3. Result and Discussion

This section presents the results of the metric evaluation for the prediction model, comparing different partition ratios as well as with and without hyperparameter tuning, presented in the form of tables and time series graphs.

3.1 Evaluation of Support Vector Regression (SVR) Modeling

Table 2 shows the metric evaluation results for the SVR prediction model. Figure 2 displays a time series graph of Bitcoin closing price prediction tests using the SVR model with different split ratios. Hyperparameter tuning is performed, and the results are displayed in Table 3. Next, the metric evaluation for the tuned SVR prediction model is shown in Table 4, and the timeline graph of the tuned SVR model is displayed in Figure 2.

Table 2. Metric Evaluation Results Of The Svr Prediction Model

Ratio (SVR)	Metrics Evaluation					
	RMSE (Training)	RMSE (Test)	MAE (Training)	MAE (Test)	R2 Score (Training)	R2 Score (Test)
70 - 30	1185.63	1914.77	830.08	1602.95	0.99	0.96
80 - 20	1266.41	2132.20	886.73	1847.03	0.99	0.70
90 - 10	1226.70	879.69	854.73	700.89	0.99	0.96

In Table 2, the performance of the Support Vector Regression (SVR) model in predicting the Bitcoin price was evaluated using different split ratios: 70-30, 80-20, and 90-10. The original hyperparameters for SVR are as follows: {C: 1, Gamma: 'scale', Epsilon: 0.1}. The 70-30 partition achieved a low RMSE of 1185.63 and 1914.77 for the training and test data, respectively, showing strong prediction performance. In addition, it has the lowest MAE values of 830.08 (train) and 1602.95 (test). However, the 90-10 split ratio consistently produced better results, with RMSE of 1226.70 (train) and 879.69 (test), MAE of 854.73 (train) and 700.89 (test), and R2 Score of 0.96 for test data. The 80-20 split ratio has a higher error and a lower R2 Score of 0.70 on the test data than the other ratios. Therefore, choosing a larger training portion, such as a 90-10 split, results in more accurate Bitcoin price predictions using the SVR model.

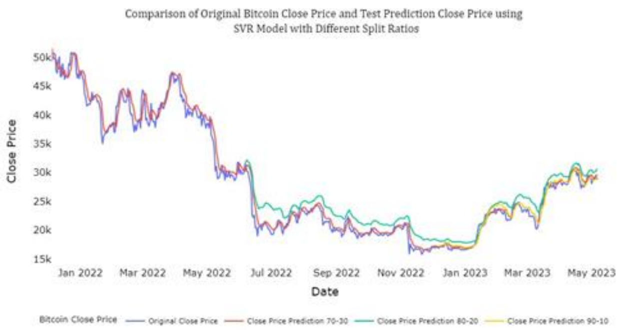

Figure 2. Time series graph of Bitcoin close price prediction test using SVR model with different split ratios

Based on the time series graph of Figure 2, it can be seen that the 70-30 split with low error and high R2 Score shows a strong alignment between the predicted price and the actual price line, resulting in the predicted price following the actual price trend. In the case of the 80-20 split ratio, there was a moderate alignment between predicted and actual prices with large deviations from the actual price trend compared to other ratios. A split ratio of 90-10 shows the best alignment between predicted and actual prices, exhibiting close similarity between predicted and actual prices, with minimal deviation. The evaluation results confirm the model's potential to accurately predict cryptocurrency prices, especially with a large portion of training data.

Table 3. Hyperparameter Tuning Results for The SVR Model

Hyperparameter Combination	*C*	*Gamma*	*Epsilon*	Ratio 70-30: Min MSE (Neg)	Ratio 80-20: Min MSE (Neg)	Ratio 90-10: Min MSE (Neg)
1	1	0.01	0.1	-0.05182	-0.00914	-0.02345
2	1	0.001	0.1	-0.02368	-0.01832	-0.02493
3	1	0.01	0.5	-0.12931	-0.10564	-0.16615
4	1	0.001	0.5	-0.13837	-0.12886	-0.15471
5	10	0.01	0.1	-0.01902	-0.00621	-0.01768
6	10	0.001	0.1	-0.01375	-0.00791	-0.01084
7	10	0.01	0.5	-0.11794	-0.09583	-0.14442
8	10	0.001	0.5	-0.10242	-0.08651	-0.11568

Based on Table 3, the 70-30 division shows that hyperparameter combination number 6 is the best where C=10, Gamma=0.001, and Epsilon=0.1, with a negative mean MSE of -0.01375. The 80-20 partition performed well with C=10, Gamma=0.01, and Epsilon=0.1, yielding a negative MSE of -0.00621. The 90-10 split achieves the same combination of hyperparameters as the 70-30 split, resulting in a negative MSE of -0.01084. Lower negative MSE values indicate better performance, indicating the effectiveness of GridSearchCV in selecting optimal parameters for improved cryptocurrency price forecasting.

Table 4. Metric Evaluation Result of the Tuned SVR Prediction Nodel

Ratio (SVR)	Hyperparameter	Evaluation Metric RMSE (Training)	RMSE (Test)	MAE (Training)	MAE (Test)	R2 Score (Training)	R2 Score (Test)
70 - 30	Untuned	1185.63	1914.77	830.08	1602.95	0.99	0.96
	Tuned	1256.18	1291.70	884.26	902.82	0.99	0.98
80 - 20	Untuned	1266.41	2132.20	886.73	1847.03	0.99	0.70
	Tuned	1120.96	1381.89	755.01	1167.09	0.99	0.87
90 - 10	Untuned	1226.70	879.69	854.73	700.89	0.99	0.96
	Tuned	1304.65	727.37	960.38	493.35	0.99	0.97

Table 4 shows the performance of the SVR model for bitcoin price forecasting with and without tuning. Model tuning generally leads to an increase in forecast accuracy. The tuned model achieved lower RMSE and MAE values, reducing errors in predicting bitcoin prices. Significantly, the 80-20 split ratio shows the most significant improvement after tuning, with RMSE decreasing from 1266.41 to 1120.96, MAE decreasing from 886.73 to 755.01 for test data, and R2 score increasing from 0.70 to 0.87. After tuning, the 90-10 division ratio showed better performance, achieving the lowest RMSE decrease from 879.69 to 727.37 and MAE decrease from 700.89 to 493.35 for the test data. The 70-30 split ratio also increased in RMSE and MAE, with the R2 score reaching the highest at 0.98. Therefore, tuning the SVR model can significantly improve its performance and make it more accurate in predicting the price of Bitcoin. In this case, the 90-10 split ratio seems to provide the best SVR model performance on the test data set based on the given evaluation metrics, as it exhibits the lowest prediction error (RMSE and MAE) and a relatively high R2 score.

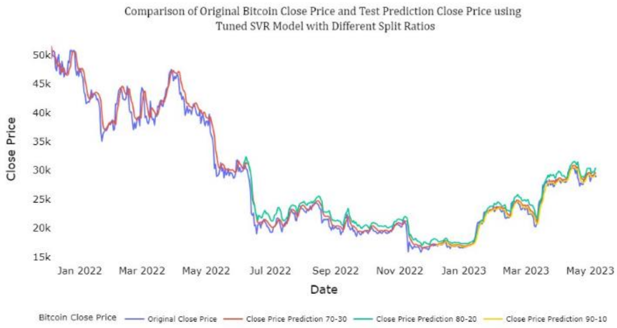

Figure 3. A graph of the results of the Bitcoin closing price forecast test using the SVR model tuned with different split ratios

Based on the time series graph of Figure 3, it can be seen that the 80-20 split shows an improvement in the alignment between predicted and actual prices compared to the time series graph of the untuned SVR model. As for the 70-30 and 90-10 splits, they still fit and show a close match between predicted and actual prices, with minimal deviations.

3. 2 Evaluation of K-Nearest Neighbor (KNN) Modeling

Table 5 shows the metric evaluation results for the KNN prediction model. Figure 4 displays the time series graph of the Bitcoin closing price prediction test using the KNN model with different split ratios. Hyperparameter tuning is performed, and the results are displayed in Table 6. Next, the metric evaluation for the tuned KNN prediction model is shown in Table 7. The timeline graph of the tuned KNN model is displayed in Figure 5.

Table 5. Metric Evaluation Result of the KNN Prediction Model

Ratio (KNN)	Evaluation Metric					
	RMSE (Training)	RMSE (Test)	MAE (Training)	MAE (Test)	R2 Score (Training)	R2 Score (Test)
70 - 30	688.38	4937.51	369.65	4223.59	0.99	0.76
80 - 20	754.85	4758.47	434.53	4061.35	0.99	-0.50
90 - 10	743.42	2272.56	432.66	1912.36	0.99	0.74

Based on Table 5, when comparing different partition ratios in the K-neighbor regressor (KNN) algorithm. The original hyperparameters for KNN are as follows: {n-neighbors: 5, Weight: 'Uniform'}. The 90-10 ratio produced the best results, with an RMSE of 2272.56, an MAE of 1912.36, and an R2 score of 0.74 on the test dataset. This shows the lowest prediction error and a relatively high level of explained variance. A ratio of 80-20 performed the worst with a negative R2 score, while a ratio of 70-30 performed moderately. The KNN algorithm used in this analysis may be more sensitive to data division in the 80-20 division ratio. If the test set contains events significantly different from the training set, it may affect the model's performance and lead to a negative R2 score.

Based on the time series graph in Figure 4, the KNN model shows overlapping lines between all three different ratios. This behaviour can occur due to the nature of the K-neighbor algorithm. The 70-30 split forecast price shows that it has a similar trend line to the actual price line, but the error is large. Likewise, with the 80-20 split, the forecast line does follow the actual price trend, but the deviation is high. Finally, the 90-10 split forecast line shows moderate alignment with the actual price line, which is the best for the KNN model.

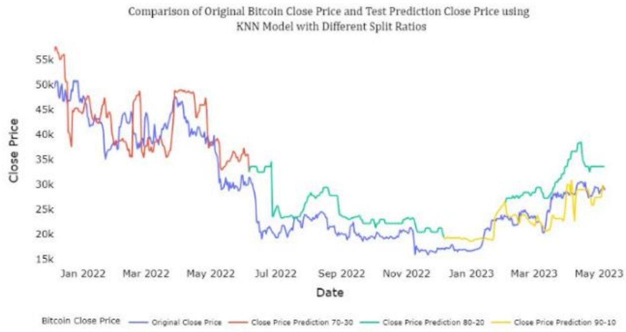

Figure 4. Graph of the results of the Bitcoin closing price prediction test using the KNN model with different partition ratios

Table 6. Hyperparameter Tuning Result for The KNN Model

Hyperparameter Combination	*n-neighbours*	*Weight*	70/30 Ratio: Mean MSE (Neg)	80/20 Ratio: Mean MSE (Neg)	90/10 Ratio: Mean MSE (Neg)
1	5	*Uniform*	-0.06411	-0.06231	-0.07983
2	5	*Distance*	-0.06442	-0.06228	-0.07999
3	10	*Uniform*	-0.04874	-0.05611	-0.07334
4	10	*Distance*	-0.04962	-0.05611	-0.07362
5	20	*Uniform*	-0.05240	-0.04796	-0.07678
6	20	*Distance*	-0.05200	-0.04772	-0.07628
7	25	*Uniform*	-0.06389	-0.04782	-0.08126
8	25	*Distance*	-0.06068	-0.04708	-0.07982

Table 6 shows the results of the KNN model with different hyperparameter configurations and training-test partitioning ratios. For the 70-30 split, hyperparameter combination number 3 is the best where n-neighbors = 10, and Weight = 'Uniform', with a negative mean MSE of -0.04874. The 80-20 split achieves a lower negative MSE (-0.00621) with n-neighbors = 25, and Weight = 'Distance', resulting in 8 hyperparameter combinations to choose from. The 90-10 split achieves the same combination of hyperparameters as the 70-30 split, resulting in a negative MSE of -0.07334.

Table 7. Hyperparameter Tuning Result for The KNN Model

Ratio (KNN)	Hyperparameter	Penilaian Metrics					
		RMSE (Training)	RMSE (Test)	MAE (Training)	MAE (Test)	R2 Score (Training)	R2 Score (Test)
70 - 30	Untuned	688.38	4937.51	369.65	4223.59	0.99	0.76
	Tuned	1093.16	4631.46	577.90	3938.92	0.99	0.79
80 - 20	Untuned	754.85	4758.47	434.53	4061.35	0.99	-0.50
	Tuned	0.0038	4175.52	0.0010	3347.56	0.99	-0.16
90 - 10	Untuned	743.42	2272.56	432.66	1912.36	0.99	0.74
	Tuned	1176.24	2228.91	679.87	1886.91	0.99	0.75

Based on Table 7, the tuned KNN model achieved lower RMSE and MAE values, reducing errors in predicting bitcoin prices. After tuning, the 90-10 split ratio performed better, achieving the lowest

RMSE (2228.91) and MAE (1886.91) for the test data. Although the RMSE on the 90-10 and 70-30 ratio training data is increased and may seem counterintuitive, it can be a sign of better generalization. It shows that the model becomes less overfit with the training set and captures more meaningful patterns that can be applied to new data. Therefore, parameter tuning is proven to improve the performance of the KNN model. In this case, the 90-10 split ratio seems to provide the best model performance on the test dataset based on the given evaluation metrics, as it exhibits the lowest RMSE and MAE and a relatively good R2 score (0.75).

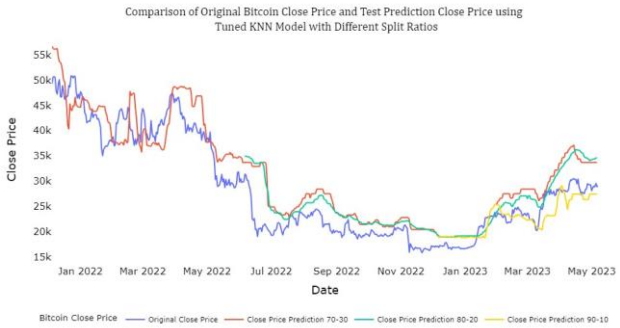

Figure 5. A graph of the results of the Bitcoin closing price prediction of the test using the KNN model tuned with different division ratios

Based on the time series graph in Figure 5, it can be seen that the forecast lines stop overlapping each other after the model is tuned. The 80-20 split shows an increase where the forecast line is closer to the actual price line, unlike the 70-30 split, where the forecast line is not much different than the untuned KNN time series graph. However, the 90-10 split also shows improvements where there is little error compared to the untuned model, especially in early May 2023.

3.3 Evaluation of Extreme Gradient Boosting (XGBoost) Modeling

Table 8 shows the metric evaluation results of the XGBoost prediction model. Figure 6 displays the time series graph of the Bitcoin closing price prediction test using the XGBoost model with different split ratios. Hyperparameter tuning is performed, and the results are displayed in Table 9. Next, the metric evaluation for the tuned XGBoost prediction model is shown in Table 10, and the timeline graph of the tuned XGBoost model is displayed in Figure 7.

Table 8. Metric Evaluation Result of the XGBoost Prediction Model

Ratio (XGBoost)	Evaluation Metric					
	RMSE (Training)	RMSE (Test)	MAE (Training)	MAE (Test)	R2 Score (Training)	R2 Score (Test)
70 - 30	41.43	5851.85	28.87	4646.62	0.99	0.66
80 - 20	58.42	7438.66	39.97	6497.72	0.99	-2.67
90 - 10	80.39	1547.61	54.38	1267.09	0.99	0.88

The original hyperparameters for XGBoost are as follows: {Colsample Bytree: 0, Learning rate: 0, MaxDepth: 0, n-estimators: 100}. Based on Table 4.9, the XGBoost model for the 70-30 partition achieves an RMSE of 41.43 on the training data and 5851.85 on the test data, indicating that it will perform poorly on new and unseen data. The MAE values follow the same pattern, with 28.87 on the training data and 4646.62 on the test data. The 90-10 partition ratio shows the best overall results with low RMSE (1547.61) and MAE (1267.09) and high R2 Score (0.88) for the test data. However, the R2 score of the 80-20 split ratio for the test data was -2.67, indicating that the model performed poorly and failed to capture the underlying patterns in the test data. This negative R2 score reflects overfitting. Further analysis and potential model adjustments are needed to improve performance and avoid overfitting in the XGBoost model for Bitcoin price prediction.

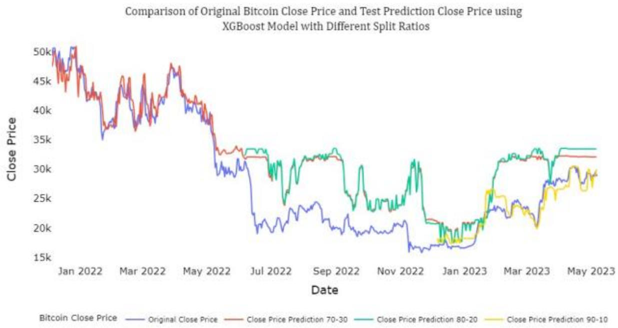

Figure 6. Graph of Bitcoin closing price results of test predictions using X the GBoost model with different split ratios

Based on the time series graph of Figure 6, the 70-30 ratio split price forecast line closely aligns with the actual price until mid-May 2022, then it deviates within a large margin of error, indicating not a good sign. For the 80-20 split, the RMSE for the test data is large compared to the others, so it makes sense that the prediction line deviates by a large margin. However, the 90-10 split manages to get a good fit where it shows a close similarity between the actual price line and the predicted price.

Table 9. Hyperparameter Tuning Result for the XGBoost Model

Hyperparameter Combination	***Colsample Bytree***	***Learning Rate***	***Max_ Depth***	***n-estimators***	**70/30 Ratio: Mean MSE (Neg)**	**80/20 Ratio: Mean MSE (Neg)**	**90/10 Ratio: Mean MSE (Neg)**
1	0.5	0.01	3	300	-0.05003	-0.03673	-0.06044
2	0.5	0.01	3	1000	-0.04493	-0.03296	-0.04853
3	0.5	0.01	5	300	-0.06644	-0.04226	-0.07031
4	0.5	0.01	5	1000	-0.05677	-0.03775	-0.05905
5	0.5	0.1	3	300	-0.05251	-0.03469	-0.04954
6	0.5	0.1	3	1000	-0.05256	-0.03471	-0.04984
7	0.5	0.1	5	300	-0.06340	-0.03880	-0.05779
8	0.5	0.1	5	1000	-0.06346	-0.03873	-0.05781
9	0.8	0.01	3	300	-0.05107	-0.03498	-0.06343
10	0.8	0.01	3	1000	-0.04454	-0.03030	-0.05105
11	0.8	0.01	5	300	-0.06358	-0.04179	-0.07105
12	0.8	0.01	5	1000	-0.05172	-0.03735	-0.06040
13	0.8	0.1	3	300	-0.04586	-0.03094	-0.05251
14	0.8	0.1	3	1000	-0.04632	-0.03086	-0.05307
15	0.8	0.1	5	300	-0.05139	-0.03771	-0.05813
16	0.8	0.1	5	1000	-0.05149	-0.03757	-0.05825

Table 9 shows the results of the XGBoost model with different hyperparameter configurations and test and train partition ratios. For a 70-30 split ratio, hyperparameter combination number 10 is the best where Colsample Bytree = 0.8, Learning Rate = 0.01, Max_Depth,= 3 and n-estimators = 1000, with a negative mean MSE of -0.04454. The 80-20 split achieves the same combination of hyperparameters but with a lower mean negative MSE (-0.03030). The 90-10 split scores a negative -0.04853 mean MSE, resulting in hyperparameter combination number 2 where Colsample Bytree = 0.5, Learning Rate = 0.01, Max_Depth = 3 and n-estimators= 1000.

Table 10. Metric Evaluation Results of the Tuned XGBoost Prediction Model

Ratio (XGBoost)	Hyperparameter	Evaluation Metric RMSE (Training)	RMSE (Test)	MAE (Training)	MAE (Test)	R2 Score (Training)	R2 Score (Test)
70 - 30	Untuned	41.43	5851.85	28.87	4646.62	0.99	0.66
	Tun	633.912	4860.66	343.38	3860.80	0.99	0.77
80 - 20	Untuned	58.42	7438.66	39.97	6497.72	0.99	-2.67
	Tuned	734.95	5943.47	412.72	5214.26	0.99	-1.34
90 - 10	Untuned	80.39	1547.61	54.38	1267.09	0.99	0.88
	Tuned	770.32	1222.05	457.17	1026.77	0.99	0.93

Table 10 presents the performance of the tuned XGBoost model for predicting the price of Bitcoin. Before tuning, the model showed moderate results, with RMSE values ranging from 41.43 to 80.39 on the training data and higher values between 1222.05 and 7438.66 on the test data. After tuning, the model's performance improved significantly, achieving lower RMSE values on training and test data. For example, in the 70-30 split, the tuned model achieved an RMSE of 633.912 on the training data and 4860.66 on the test data, showing a significant reduction in the prediction error. MAE values follow the same pattern, with a significant decrease after tuning. Moreover, the R2 scores for the tuned model remained consistently high, especially the 90-10 split (0.93), indicating an excellent fit to the data and better prediction accuracy. These results show that tuning improves the model's performance, leading to more accurate Bitcoin price predictions.

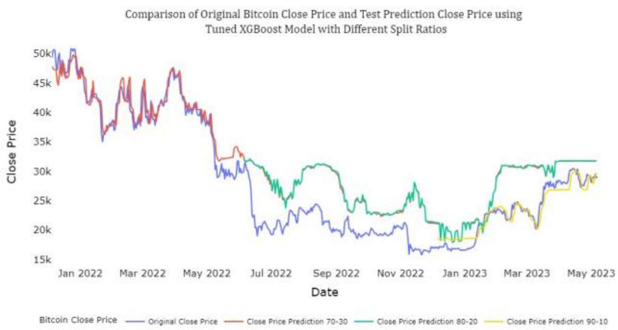

Figure 7. A graph of the results of the Bitcoin closing price prediction test using the XGBoost model tuned with different split ratios

Based on the time series graph in Figure 7, it can be seen that the gap between the actual price and the forecast line for the 70-30 split and the 80-20 split has only a minimal increase after tuning. On the other hand, the 90-10 split forecast improves and shows the best fit compared to the others, where it aligns more closely towards the actual price line.

3.4 Evaluation of Long Short-Team Memory Modeling (LSTM)

Table 11 presents the metric evaluation results for the LSTM prediction model. Meanwhile, Figure 8 displays the time series graph of the Bitcoin closing price forecast of the test set using the LSTM model with different division ratios. Hyperparameter tuning was performed, and the resulting data is presented in Table 12. Following this, Table 13 displays the metric evaluation for the XGBoost prediction model, while Figure 8 shows the timeline graph of the LSTM model.

Table 11. Metric Evaluation Results of the Tuned XGBoost Prediction Model

Ratio (LSTM)	Evaluation Metric					
	RMSE (Training)	RMSE (Test)	MAE (Training)	MAE (Test)	R2 Score (Training)	R2 Score (Test)
70 - 30	3391.44	2122.01	2414.32	1525.41	0.96	0.95
80 - 20	2732.21	1950.05	1828.37	1373.63	0.97	0.74
90 - 10	2856.42	1908.52	1910.56	1485.41	0.97	0.81

Table 12 shows the results of the effectiveness of the LSTM model in capturing basic patterns in the data. The original hyperparameter for LSTM is as follows: {Learning rate: 0.001, Units: 64}. The model generally shows good results at all partition ratios. In the 70-30 split, the model showed the best overall result, achieving an RMSE of 3391.44 on the training data and 2122.01 on the test data, demonstrating its ability to generalize well to unseen data. The MAE values also show promising results and the highest R2 score (0.95) for the test data, illustrating a strong fit with the data and a reliable prediction accuracy. These results show that the LSTM can effectively model complex and dynamic Bitcoin price data and make accurate predictions.

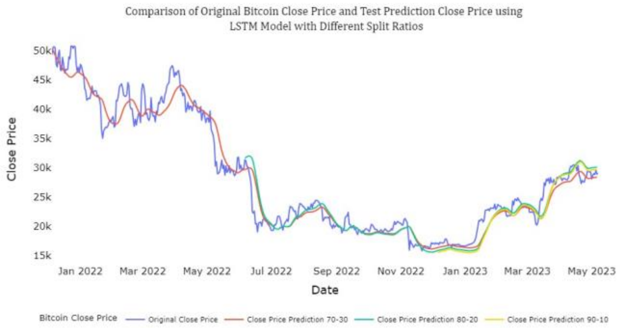

Figure 8. A graph of the results of the Bitcoin closing price prediction of the test using the LSTM model with different division ratios

Based on the time series graph of Figure 8, it is clear that all the forecast lines of the fractional ratio closely follow the trend of the actual price line, which indicates a good sign. No specific division ratio shows a bad linear alignment between predicted and actual prices with large deviations. Although there are slight variations and deviations between the predicted price and the actual price, the overall trend of the predicted line shows the model's ability to provide meaningful insight into Bitcoin market dynamics.

Table 12. Hyperparameter Tuning Results for the LSTM Model

Hyperparameter Combination	*Learning rate*	*Units*	70/30 Ratio: Mean MSE (Neg)	80/20 Ratio: Mean MSE (Neg)	90/10 Ratio: Mean MSE (Neg)
1	0.001	32	-0.05849	-0.00715	-0.01014
2	0.001	64	-0.04496	-0.00604	-0.00634
3	0.01	32	-0.01112	-0.00814	-0.01855
4	0.01	64	-0.03393	-0.00989	-0.01140

Table 12 shows the results of the LSTM model with different hyperparameter configurations and training-test data partition ratios. For the 70-30 split, hyperparameter combination number 3 is the best where Learning Rate = 0.01 and Units = 32 obtain a negative mean MSE of -0.01112. The 80-20 partition scores -0.00604 negative MSE mean, indicating the best hyperparameter combination is number 2 where Learning Rate = 0.001, and Units = 64. The 90-10 partition achieves the same hyperparameter combination with a negative MSE mean of -0.00634.

Table 13. Metric Evaluation Results of the Tuned LSTM Prediction Model

Ratio (LSTM)	Hyper-parameter	Evaluation Metric RMSE (Training)	RMSE (Test)	MAE (Training)	MAE (Test)	R2 Score (Training)	R2 Score (Test)
70 - 30	Untuned	3391.44	2122.01	2414.32	1525.41	0.96	0.95
	Tuned	1130.55	1215.43	590.09	889.51	0.99	0.98
80 - 20	Untuned	2732.21	1950.05	1828.37	1373.63	0.97	0.74
	Tuned	1314.18	821.58	799.92	555.46	0.99	0.95
90 - 10	Untuned	2856.42	1908.52	1910.56	1485.41	0.97	0.81
	Tuned	1162.97	661.66	745.57	462.63	0.99	0.98

Based on Table 13, which shows the tuned results of the LSTM model, the untuned model initially produced good results, with RMSE values ranging from 2732.21 to 3391.44 on the training data and 1908.52 to 2122.01 on the test data. After tuning, the performance of the model improved significantly. For example, in the 70-30 split, the tuned LSTM model achieved an RMSE of 1130.55 on the training data and 1215.43 on the test data, showing a significant reduction in the prediction error. Similar improvements were observed across the other split ratios, especially the 90-10 split ratio, which had the lowest RMSE (661.66) and MAE (462.63) on the test data. MAE values and R2 scores also significantly improve after tuning, confirming better model accuracy and fit to the data. The 90-10 split seems most promising in this case because it has a low prediction error and a high R2 score (0.98).

Figure 9. Graph of Bitcoin closing price results of test prediction using LSTM model tuned with different division ratios

Based on the time series graph of Figure 9, it can be seen that all the split ratio forecasts show excellent performance with a tight fit and close similarity to the actual price, tracking its fluctuations with remarkable accuracy. A large margin has reduced the deviation by comparing it with the time series graph of the untuned LSTM model. This good accuracy reflects the effectiveness of the LSTM architecture in understanding the underlying patterns and dependencies in time series data.

3.5 Overall Model Comparison

This section compares the performance of four different models for Bitcoin currency price forecasting: SVR, KNN, XGBoost, and LSTM. To accurately evaluate the model, the researchers considered three different training-test data split ratios: 70-30, 80-20, and 90-10, then investigated the effect of hyperparameter tuning on model performance, comparing their results before and after tuning.

After analyzing the results, it became apparent that the 90-10 training-test data split ratio consistently outperformed the other ratios across all models. This finding shows that allocating a large portion of the data for training (90%) and a smaller portion for testing (10%) results in better prediction accuracy. Moreover, all the models show a significant improvement in performance after undergoing hyperparameter tuning. This indicates the importance of carefully selecting optimal hyperparameters to improve the model's predictive ability. Given these findings, the researchers made an informed decision to continue the experiment with the ratio using a 90-10 split ratio and model tuned for the researcher's next experiment. Table 14 below shows a comparison between the 90-10 division ratio of the tuned model.

Table 14. Comparison of Tuned Prediction Models (90-10 split ratio)

Model	**Tuned Hyper-parameter**	**Evaluation Metric**						**Ranking**
		RMSE (Training)	RMSE (Test)	MAE (Training)	MAE (Test)	R2 Score (Training)	R2 Score (Test)	
SVR	*{C: 10, Gamma: 0.001, Epsilon: 0.1}*	1304.65	727.37	960.38	493.35	0.99	0.97	2
KNN	*{n-neighbors: 10, Weight: Uniform}*	1176.24	2228.91	679.87	1886.91	0.99	0.75	4
XGBOOST	*{Colsample Bytree: 0.5, Learning rate: 0.01, MaxDepth: 3, n-estimators: 1000}*	770.32	1222.05	457.17	1026.77	0.99	0.93	3
LSTM	*{Learning rate: 0.001. Units: 64}*	1162.97	661.66	745.57	462.63	0.99	0.98	1

Based on Table 14 above, among the four models evaluated for Bitcoin currency time series forecasting, SVR, XGBoost, and LSTM show strong performance, while KNN exhibits relatively weak results. SVR exhibits accurate forecasts with low RMSE and MAE values, capturing both short-term fluctuations and long-term trends. XGBoost captures patterns effectively with low RMSE and MAE values, showing accurate predictions and a robust linear relationship with actual prices. LSTM, as a recurrent neural network, excels at modelling temporal dependence, producing low RMSE and MAE values and high R2 scores. In contrast, KNN displays higher errors and lower R2 scores, illustrating limitations in capturing the complexities of Bitcoin currency price dynamics. Overall, the models were ranked based on the R2 Score, which resulted in LSTM being first, SVR second, XGBoost third, and KNN last. The model has been saved and will be adapted to evaluate the performance of the model using other cryptocurrency data.

In conclusion, research on Bitcoin price time series forecasting using various machine learning models and different split ratios of training-test data reveals deep insights and understanding. The choice of split ratio impacts forecast performance, with a 90-10 split emerging as the best fit for all models. Moreover, the prediction results clearly show the importance of hyperparameter tuning, as the tuned model consistently outperforms the untuned one. Among the compared models, LSTM exhibits the highest prediction accuracy, followed by SVR and XGBoost, while KNN Regressor is slightly behind. These findings emphasize the importance of proper data segmentation and hyperparameter optimization in time series forecasting tasks and highlight LSTM as a robust choice for Bitcoin price forecasting.

4. Conclusion

In this work, we work on Bitcoin time series forecasting using machine learning models, namely Support Vector Regression (SVR), Long Short Term Memory (LSTM), Extreme Gradient Boosting (XGBoost) and K-Nearest Neighbor Regression (KNN) models. Researchers explore the effect of different training-test data split ratios on model performance and conduct hyperparameter tuning to improve prediction accuracy. Findings reveal that the choice of split ratio significantly affects model performance, with a 90-10 split being the most appropriate for all models. Additionally, hyperparameter tuning proved to be important, as tuned models consistently outperformed untuned ones. Among the compared models, LSTM showed the highest prediction accuracy, followed by SVR, while XGBoost and KNN showed relatively poor results. The research also extended its investigation to a dataset of alternative cryptocurrencies (ETH, XRP, and LTC), and the results showed that LSTM and SVR remained robust and effective in predicting price trends across these assets.

For future work, it is suggested to include additional features such as sentiment analysis of social media data related to Bitcoin, combining data from financial markets, and incorporating external factors such as regulatory changes, macroeconomic events, or cryptocurrency news can improve the accuracy of Bitcoin price prediction models.

References

[1] Abdallah, Z., Du, L., & Webb, G. (2017). Data Preparation. Encyclopedia of Machine Learning and Data Mining, pp 318-327, 2017.

[2] Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2(2), 121-167.

[3] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T. P., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step data mining guide.

[4] Edwards, J. (2022, July 2, 2022). Bitcoin's Price History. Investopedia. Retrieved November 30, 2,022 from https://www.investopedia.com/articles/forex/121815/bitcoins-price-history.asp

[5] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-term Memory. Neural Computation, 9, 1735-1780.

[6] Hotz, N. (2022). What CRISP-DM DM? https://www.datascience-pm.com/crisp-dm-2/

[7] Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. International Journal of Forecasting, 22(4), 679-688.

[8] Matkovskyy, R., & Jalan, A. (2019). From financial markets to Bitcoin markets: A fresh look at the contagion effect. Finance Research Letters, 31, 93-97.

[9] McNally, S., Roche, J., & Caton, S. (2018). Predicting the Price of Bitcoin Using Machine Learning. PDP), Cambridge, UK, 2018, pp. 339-343, doi: 10.1109/PDP2018.2018.00060.

[10] Morde, V. (2019). XGBoost Algorithm: Long May She Reign! https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

[11] Mudassir, M., Bennbaia, S., Unal, D., & Hammoudeh, M. (2020). Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach. Neural Computing and Applications. doi: 10.1007/s00521-020-05129-6

[12] Muniye, T., Rout, M., Mohanty, L., & Satapathy, S. (2020). Bitcoin Price Prediction and Analysis Using Deep Learning Models. In (pp. 631-640). doi: 10.1007/978-981-15-5397-4_63

[13] Murray, K., Rossi, A., Carraro, D., & Visentin, A. (2023). On Forecasting Cryptocurrency Prices: A Comparison of Machine Learning, Deep Learning, and Ensembles. Forecasting, 5(1), 196–209.

[14] Norwawi, N. M. (2021). Sliding window time series forecasting with multilayer perceptron a multi regression of COVID-19 outbreak in Malaysia. Data Science for COVID-19, 547–564. doi: 10.1016/b978-0-12-824536-1.00025-3

[15] Rahouti, M., Xiong, K., & Ghani, N. (2018). Bitcoin Concepts, Threats, and Machine-Learning Security Solutions. IEEE Access, 6, 67189-67205.

[16] Sabry, F., Labda, W., Erbad, A., & Malluhi, Q. (2020). Cryptocurrencies and Artificial Intelligence: Challenges and Opportunities. IEEE Access, 8, 175840-175858.

[17] Sӧze, K. (2017). Bitcoin guidebook for beginners: two books in one: Bitcoin Blueprint & Invest in digital gold wallets, Bitcoin ATM-s, Bitcoin mining, investment, and trading. Createspace Independent Publishing.

[18] Zhang, S., Li, M., & Yan, C. (2022). The Empirical Analysis of Bitcoin Price Prediction Based on Deep Learning Integration Method. Computational Intelligence and Neuroscience, 2022, 1265837.