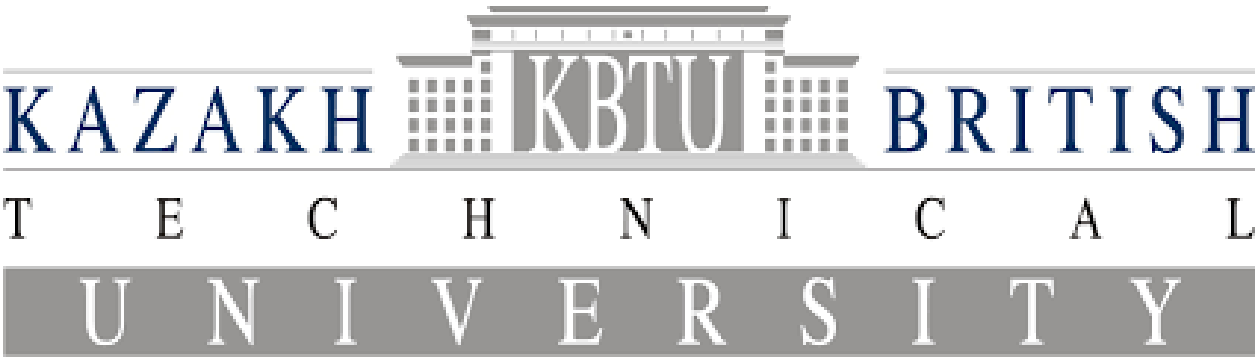


JSC “Kazakh-British Technical University”

Faculty of Information Technology

Machine Learning

Description of "A Study on Cryptocurrency Log-Return Price Prediction Using Multivariate Time-Series Model"

Done by: **Pirniyazov Miras**

School of Information Technology and Engineering,

Computer systems and Software, 3-year student, ID: 20B030498

Date of submission: **May 5, 2023**

Almaty, 2023

Abstract

The main conclusions of a study that used multiple attributes and time-series models to forecast the log-return prices of popular cryptocurrencies are summarized in the abstract. The volatility properties produced from the ARCH and GARCH models, along with the closing price, were examined by the authors in order to determine the most significant characteristics for each cryptocurrency. Through the use of two different time-series models, the traditional autoregressive integrated moving average (ARIMA) model and the neural network-based model, they were able to predict log-return prices using these features. The outcomes demonstrated that the neural network-based model outperformed the conventional model in terms of predictive ability. Overall, the work advances the field of bitcoin prediction by highlighting important characteristics and contrasting the effectiveness of various time-series models.

Introduction

The introduction emphasizes how popular blockchain technology is becoming as a new generation of data storage, as well as how popular cryptocurrencies are becoming. The study uses the volatility characteristics of cryptocurrencies to forecast the log-return price of cryptocurrency. In order to do this, the study forecasts the log-returns of the three most important cryptocurrencies, Bitcoin, Ethereum, and Binance Coin, using data from the top eleven most actively traded cryptocurrencies. This study broadens the field of study by incorporating volatility features into studies that use log-return price prediction to forecast cryptocurrency performance. The ARCH and GARCH volatility prediction models are utilized, and key features are chosen to assess the significance of attributes associated with the chosen cryptocurrency. The study employs conventional time-series approaches, including ARIMA and artificial neural network-based recurrent neural networks (RNN), LSTM, and GRU, to estimate the log-return price. The conclusion is offered in the last section of the paper, which is divided into sections that discuss the methods employed, data collecting and processing, experimental results, significance and limitations of the study, and experimental results.

Methods

Gini Impurity

The Gini impurity approach is a method for assessing the significance of various features. We must first comprehend the idea of impurity in order to

comprehend the significance of features. How mixed or diverse the values are within a classification is referred to as impurity. The impurity is close to zero if the values are significantly different from one another. However, if the values are extremely close, the impurity is nearer to one. In order to determine each feature's importance for classification, the Gini impurity technique calculates the level of impurity for each

$$G(S) = 1 - \sum_{k=1}^{m} p_k^2$$

feature.

ARCH

A statistical model called ARCH (Autoregressive Conditional Heteroskedasticity) is made to deal with highly volatile time-series data. This particular conditional heteroscedasticity model incorporates the lag in a time-series. Heteroscedasticity is a characteristic of financial time-series data, which means that the variance of the data changes over time. The assumption that most time-series models make—that the variance of the data is constant—is broken by this. Given that changing variance of the data over time is a crucial aspect of financial time-series data, the ARCH model is advantageous in this situation. The ARCH model can successfully capture the heteroscedasticity in the data and make more precise predictions by accounting for the lag in the time-series.

$$\sigma_t^2 = a_0 + \sum_{i=1}^{q} a_i \varepsilon_{t-i}^2$$

GIARCH

An addition to the ARCH model that solves some of its drawbacks is the GARCH model. In the ARCH model, the relevance of the variance estimate falls as the lag grows because it becomes more difficult to adequately reflect the structure. The GARCH model generalizes the ARCH model in a manner akin to the autoregressive moving average (ARMA) model in order to overcome this problem. This makes the GARCH model a more practical and valuable tool for studying financial time-series data because it demonstrates the same or higher explanatory power with significantly fewer parameters. As a result, in the majority of investigations, the

GARCH model is chosen over the ARCH model.

$$\sigma_t^2 = a_0 + \sum_{i=1}^{q} a_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2$$

ARIMA

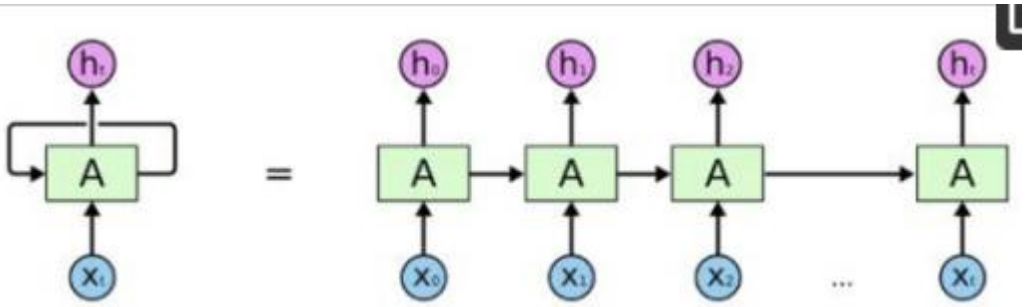
A potent tool for studying time-series data is the ARIMA model. It is a generalized variation of the ARMA model, which is employed to forecast future values in light of historical data. When analyzing unusual time-series data that might not adhere to the presumptions of conventional time-series analysis models, the ARIMA model is particularly helpful.

$$y'_t = c_0 + \emptyset_1 y'_{t-1} + \cdots \emptyset_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

Deep Neural Networks

RNN

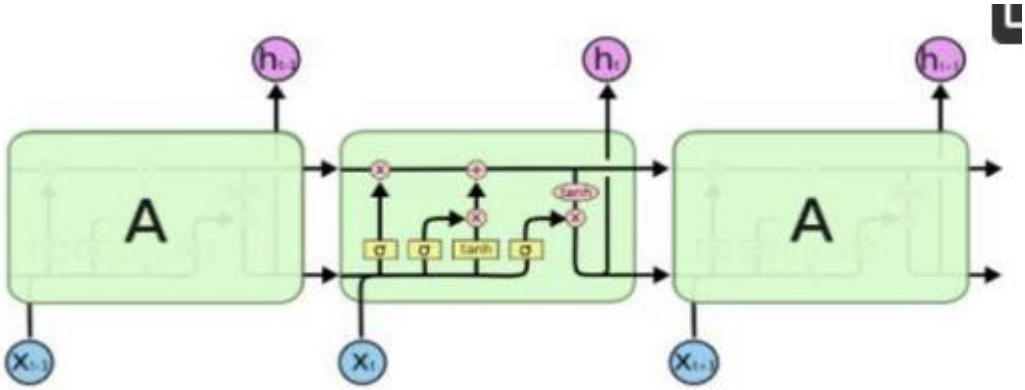
In order to handle time-series data, including audio, text, and sensor data, recurrent neural networks (RNNs) are frequently used. It is made to handle sequential data in situations where the order of the input is crucial, like when processing natural language or making long-term stock price predictions.



LSTM

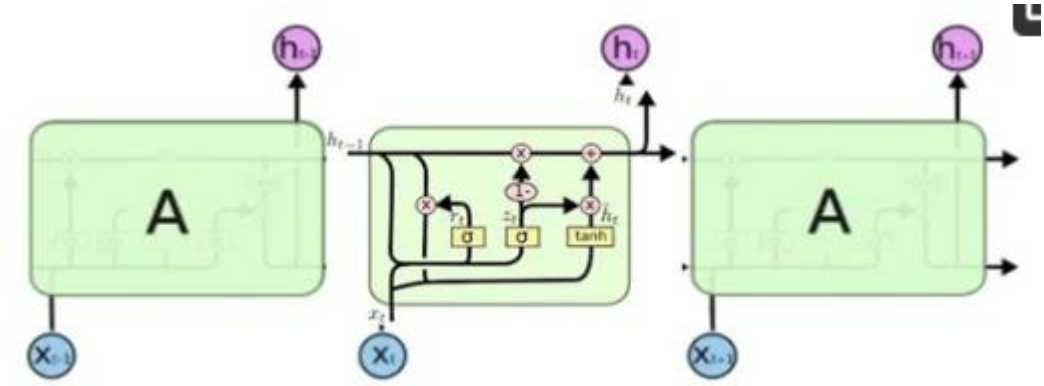
Recurrent neural networks (RNNs) are frequently employed for time-series data processing. The gradients utilized in the backpropagation process during training, however, suffer from the issue of long-term reliance, which makes it challenging to learn from lengthy data sequences. Due to the information from previous time-steps being lost as a result of this problem, the network struggles to remember and use the long-term dependencies in the input sequence. Long Short-Term Memory (LSTM) networks were developed to handle this problem. RNNs of the

LSTM type have a memory cell that enables selective data storage, reading, and writing. By managing the information passing through their gates, which control how much information is stored, forgotten, or output at each time-step, they are able to handle long-term dependencies in the input sequence.



GRU

The neural network technique known as the GRU, or gated recurrent unit, is made to deal with the issue of long-term dependencies in time-series data. It was first introduced in 2014 and resembles the LSTM in terms of structure, however it contains less parameters. Reset and update gates are used by GRU to regulate information flow and stop gradients from vanishing. The hidden state of GRU is created by combining the cell state and hidden state of the LSTM. This makes it possible to process time-series data effectively and has been demonstrated to be effective at a number of speech recognition and natural language processing



applications.

Data Collection

To forecast their volatility, the study gathered data from 11 cryptocurrencies with large market caps. Using their API, the cryptocurrency exchange Binance provided the data. The closing price was recorded together with the data that was

collected at various time intervals (minutes, hours, days, and months). The analysis concentrated on the chosen cryptocurrency' daily closing prices. To account for the recent volatility in the cryptocurrency market, data was gathered for nearly 4 years, from May 31, 2018, to May 31, 2022.

	BTC	ETH	BNB	XLM	ADA	XRP	IOT	QTU	EOS	LTC	NEO
Count	1462	1462	1462	1462	1462	1462	1462	1462	1462	1462	1462
Mean	21,416.9	1148.6	147.1	0.2	0.5	0.5	0.6	5.2	4.0	98.1	22.7
Std.	18,576.8	1327.1	194.6	0.1	0.7	0.3	0.5	4.3	1.9	61.3	18.3
Min	3211.7	83.8	4.5	0.0	0.0	0.1	0.1	1.0	1.2	23.1	5.4
25%	7197.5	187.1	15.4	0.1	0.1	0.3	0.3	2.1	2.6	51.7	10.2
50%	10,253.8	347.6	25.3	0.1	0.1	0.4	0.4	3.2	3.5	75.8	16.8
75%	38,670.0	2159.4	321.9	0.3	1.1	0.6	0.9	7.0	4.9	133.8	28.8
Max	67,525.8	4808.0	676.2	0.7	3.0	1.8	2.5	27.4	14.7	387.8	122.8
Skewness.	0.82	1.07	1.07	1.16	1.38	1.38	1.27	1.61	1.96	1.31	2.17
Kurtosis.	−0.87	−0.31	−0.41	1.13	0.88	1.42	0.88	2.53	5.69	1.69	6.15

Data Preprocessing

To determine the volatility of daily bitcoin closing price data, the researchers in this study employed log transformation.

$$\text{Log return} = \log\left(\frac{p_t}{p_{t-1}}\right) * 100$$

This transformation aids in calculating the return and comprehending the data's volatility. The issue of variations in the unit of volatility and transaction amount for each cryptocurrency was resolved using min-max normalization. The KPSS test was used to determine whether the preprocessed time series data were stationary. Volatility was predicted using the ARCH (1) and GARCH (1, 1) models, and the outcomes were verified for the chosen cryptocurrencies. The old dataset was supplemented with features obtained using these models to produce a new dataset with a total of 33 features. Major features for predicting log-return prices were extracted from these features.

Results

In this study, the ARIMA, a conventional time-series prediction approach, and several artificial neural network-based time-series methods were used to assess the

log-return price prediction of three cryptocurrencies: Bitcoin, Ethereum, and Binance Coin. MAE, MSE, and RMSE values—which stand for various components of error calculation—were used in the evaluation. Based on auto-correlative function analysis, the hyperparameters for ARIMA were set to p=2, d=1, and q=0. Six architectures were constructed for the artificial neural network-based models, and the GRU of Architecture 6 displayed the best performance in predicting the log-return price for Ethereum. The model used the data from the previous seven days to forecast the value for the following day. The prediction interval was set to seven days. The outcomes demonstrated that approaches based on artificial neural networks performed better in terms of prediction accuracy than the conventional ARIMA method.

Model	Composition of Layers
Architecture 1	RNN (32)/LSTM (32)/GRU (32) + dense (64-32-16-8-1)
Architecture 2	RNN (32)/LSTM (32)/GRU (32) + dense (32-16-8-1)
Architecture 3	RNN (32)/LSTM (32)/GRU (32) + dense (16-8-4-1)
Architecture 4	RNN (32)/LSTM (32)/GRU (32) + dense (16-8-1)
Architecture 5	RNN (32)/LSTM (32)/GRU (32) + dense (64-1)
Architecture 6	RNN (32)/LSTM (32)/GRU (32) + dense (16-1)
Activation	Linear
Loss	Mean squared error
Optimizer	Adam

Methods		MAE	MSE	RMSE
ARIMA (2, 1, 0)		0.0422	0.0028	0.0532
	RNN	0.0377	0.0024	0.0492
Architecture 1	LSTM	0.0383	0.0025	0.0502
	GRU	0.0378	0.0025	0.0504
	RNN	0.0376	0.0024	0.0491
Architecture 2	LSTM	0.0381	0.0024	0.0497
	GRU	0.0391	0.0026	0.0509
	RNN	0.0382	0.0025	0.0497
Architecture 3	LSTM	0.0383	0.0025	0.0501
	GRU	0.0382	0.0025	0.0500
	RNN	0.0376	0.0024	0.0491
Architecture 4	LSTM	0.0382	0.0025	0.0496
	GRU	0.0377	0.0025	0.0497
	RNN	0.0374	0.0024	0.0491
Architecture 5	LSTM	0.0381	0.0024	0.0494
	GRU	0.0381	0.0025	0.0496
	RNN	0.0377	0.0025	0.0495
Architecture 6	LSTM	0.0379	0.0025	0.0498
	GRU	0.0384	0.0024	0.0492

Conclusion

The study uses both conventional time-series prediction techniques and methods based on artificial neural networks to forecast the volatility of popular cryptocurrencies. It selects significant factors affecting the log-return price of sample cryptocurrencies using a variety of cryptocurrency data, such as log-return price and volatility statistics based on ARCH and GARCH. The study builds conventional time-series prediction models and models of artificial neural networks like RNN, LSTM, and GRU using these chosen features. According to the study, neural network models perform better at predicting the log-return price of cryptocurrencies than conventional time-series models when they have a reasonably simple design. It does admit that there may be other macroeconomic factors that influence bitcoin volatility, and that taking these factors into account may result in more precise and all-encompassing prediction models. Future studies may look into using new macroeconomic factors and more bitcoin data to boost the models' predictive power.