FORECASTING PRICE FLUCTUATIONS OF MEME COINS USING LSTM-RNN AND FBPROPHET WITH SENTIMENTS OF TWEETS

DEAN NEWAR

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF SCIENCE IN COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE

DEPARTMENT OF
COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

student number

u181970

committee

dr. Giacomo Spigler
dr. Merel Jung

location

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

date

January 6, 2023

FORECASTING PRICE FLUCTUATIONS OF MEME COINS USING LSTM-RNN AND FBPROPHET WITH SENTIMENTS OF TWEETS

DEAN NEWAR

Abstract

Meme coins are a type of cryptocurrency that gained popularity on social media in the last few years. Social media might have an effect on the predictability of the price fluctuations of these coins, which is why this thesis tries to answer the following research questions: can the price fluctuations of meme coins be predicted using LSTM-RNNs and FBProphet models, and does the sentiment of tweets improve the predictive ability of LSTM–RNNs and FBProphet models on forecasting the price fluctuation of meme coins? Research has already been conducted where the prices of cryptocurrencies, such as Bitcoin and Ethereum, were predicted using both of these models. However, little research was found on the prediction of price fluctuations of meme coins, which is why this thesis tries to minimize the gap in this field of research. The historical data from four coins, namely DOGE, SHIBA and MONA, were extracted and their corresponding tweets were scraped from Twitter. This data was then used as input to two artificial intelligence models: an LSTM-RNN and an FBProphet model. From the results can be concluded that the predictability of the price fluctuation of meme coins did not improve for both of the used models with sentiment analysis.

1 INTRODUCTION

Crypto, a term that grew in popularity and largely influenced the world in the last thirteen years. During this period, the potential of cryptocurrencies to facilitate growth in the social and economic sectors has grown. By contributing easy and transparent access to capital and financial services, the world (including developing countries) is able to benefit from this facilitated growth (Oyelude, 2022). According to Giudici, Milne, and Vinogradov (2019), "cryptocurrencies are digital financial assets, for which ownership and transfers of ownership are guaranteed by a cryptographic decentralized technology". This technology, also called blockchain, makes the transactions protected, traceable, immutable, and transparent, which makes investing in cryptocurrencies attractive for investors (Patel, Tanwar, Gupta, & Kumar, 2020). Bitcoin, released in 2008, was the first decentralized cryptocurrency. After this release, the number of cryptocurrencies exploded exponentially. According to the data of CoinMarketCap (Cryptocurrency prices, charts and market capitalizations, n.d.), the number of cryptocurrencies that are in circulation is more than 20,200. These include altcoins, which are coins other than Bitcoin, such as payment tokens, stablecoins, governance tokens, and meme coins (Frankenfield, 2022). The latter type of altcoin is inspired by memes, which are comic images or pictures, distributed on the internet daily, especially on social media (C. Li & Yang, 2022). Most of these coins started as a "joke currency" before they were supported by these online communities, which eventually resulted in the growth of the price and the market cap (the amount of money that circulates in the market of the cryptocurrency) (Chohan, 2017). The prediction of a cryptocurrency's price has been researched extensively. By looking at the results of various related papers, it can be concluded that statistical and artificial intelligence methods LSTM-RNNs, and FBProphet can be used to predict the price of cryptocurrencies (discussed in Section 2). Even though the prediction of the price of cryptocurrencies has broadly been researched, the prediction of the price fluctuations of meme coins and the influence of social media on these coins has not substantially been researched yet. The influence of social media can be measured by calculating the sentiment (positive, neutral, negative) on tweets about the coins using sentiment analysis (opinion mining). The influence of social media on the financial world is discussed in Brans and Scholtens (2020), where the influence of negative-toned tweets by former US president Donald Trump about stocks was researched. The study found that tweets with negative sentiment from the ex-president do indeed impact the market value of a stock negatively.

In this thesis, the predictability of the price fluctuation of three meme coins: DOGE, SHIBA, and MONA will be researched. This will be done by the use of two artificial intelligence models, namely FBProphet and an LSTM-RNN. Furthermore, the addition of sentiments of tweets is researched to see if these help with the predictability of the models. The meme coins were chosen based on popularity where, according to the popularity list of CoinMarketCap (*Cryptocurrency prices, charts and market capitalizations*, n.d.), DOGE, SHIBA, and MONA are ranked at place one, two, and five, respectively.

1.1 *Research Question*

This thesis aims to answer the following research questions:

> *RQ 1.1: Can the price fluctuation of meme coins be forecasted using LSTM–RNNs and FBProphet?*

> *RQ 1.2: Does the sentiment of tweets improve the predictive ability of LSTM–RNNs and FBProphet models on forecasting the price fluctuation of meme coins?*

The answer to both questions is obtained by first discussing the related works about cryptocurrencies, forecasting models, sentiment analysis, and forecasting cryptocurrencies with sentiment analysis in section 2. Next, in section 3, the methods that will be used to answer the research question are explained and it will be described how they are used for the research. The results that the models generate will be discussed in section 4. Lastly, the conclusion of the results and the discussion of the limitations of the research are described in section 5.

2 RELATED WORK

2.1 *Brief introduction to the inner workings of cryptocurrencies*

As mentioned in section 1, Bitcoin was the first decentralized cryptocurrency that was released in the crypto world. Proposed by Nakamoto (2008), the Bitcoin was designed as a peer-to-peer version of digitized money and would allow online transactions to be transferred directly from one point to another without the verification and handling of a financial institution. The core technological mechanism for most cryptocurrencies, including Bitcoin, is blockchain. The mechanism was proposed in 2008 and implemented in 2009 by the same creator(s) as Bitcoin and has four main characteristics: 'decentralization, persistency, anonymity and auditability' (Zheng, Xie, Dai, Chen, & Wang, 2018). The authors of Zheng et al. (2018), describe blockchain as a public digital storage where all the information of every transaction is stored in a chain of blocks. The chain is growing new blocks when there is a new transaction of the cryptocurrency. Due to the integration of technologies, such as 'cryptographic hash, digital signature (based on asymmetric cryptography) and distributed consensus mechanism', a blockchain cannot be controlled by one authority (Ferdous, Chowdhury, & Hoque, 2021). The process of mining is described in the paper by Konoth et al. (2018)as a process which maintains the blockchain 'by a peer-to-peer network of miners'. These miners gather data from transactions from the network and validate it. The data is then added to the blockchain in the form of a new block. The miner has to solve a cryptographic mathematical problem which is fixed to the block. This technique stops mischievous users from adding incorrect blocks to the chain. A successfully mined block consists of the solution to the mathematical problem that consists of 'the hash of the previous block, the hash of the transactions in the current block, and a wallet address to credit with the reward' (Konoth et al., 2018). When a miner adds a valid block to the blockchain, the miner is awarded with a coin of the cryptocurrency that was mined for, which was not yet in circulation. This means that this coin of cryptocurrency will now be new in circulation.

2.2 *Introduction to stock forecasting models*

The forecasting of prices for the stock market is a complicated task due to the fact that stocks have "an intrinsically volatile and non-stationary nature" (Adam, Marcet, & Nicoli, 2016). Nowadays, micro and macroeconomic features, such as political events, social media, and company balance sheets, tend to influence price fluctuation (A. W. Li & Bastos, 2020). These features contribute to the complexity of the already non-linear and non-stationary environment of the stock market. A large number of different methods are proposed in finance literature for forecasting the stock market. In the paper by G. P. Zhang (2003), the author states that machine learning models can "be a promising alternative to the traditional linear methods". With the growth of interest in the field of AI, these machine learning models were developed to improve the results of traditional methods. A common model that is used to forecast the stock market is the Recurrent Neural Network with Long Short-Term Memory cells. In addition, a recent predictive model, created by Meta (previously Facebook), was proposed in 2017 and named FBProphet by Taylor and Letham (2017).

2.2.1 *LSTM*

Roondiwala, Patel, and Varma (2017), researched the use of an LSTM-RNN to predict the returns of the NIFTY 50 stock. Five years of historical data is collected from the stock and is used for the training and validation of the model. Six different LSTM-RNN models were trained and tested and the results were measured in an RMSE value. The models differed from each other in terms of parameters (Open/Close, High/Low/Close, High/Low/Open/ Close) and the number of epochs (250 and 500). The authors conclude that after running the six different simulations, the best results were obtained by taking the features High/Low/Open/Close and

500 epochs. The model with these features obtained an RMSE value of 0.00983 in training and an RMSE value of 0.00859 in testing. Looking at these results, it can be said that based on the RMSE score, the LSTM-RNN model can be a good predictor of stock prices.

2.2.2 *FBProphet*

The FBProphet model, proposed by Taylor and Letham (2017), is a time-series modelling analysis tool, which uses the logistic growth model and a linear model. The predictability of the FBProphet model was researched in the paper by Yenidogan, Cayir, Kozan, Dag, and Arslan (2018), where the authors compared the performance metrics of the FBProphet model with an ARIMA on the same dataset. The data consisted of historical data from Bitcoin ranging from May 2016 to March 2018 and the two models were trained and tested on this data. From the results that were obtained, the authors conclude that with a precision of 94.5% the FBProphet module outperforms the ARIMA model, which provided a precision score of only 68%.

2.2.3 *Comparison of the prediction models with cryptocurrency data*

A useful study was conducted by Mazed (2019), where the stock price forecasting performance of ARIMA, FBProphet, and LSTM-RNN was compared. The author used two years of data of the stocks Lockheed Martin (LMT) and Northrop Grumman (NOC) to be analyzed. The three models were trained and tested on the data of these stocks and the metric of performance that the author used was the RMSE. FBProphet performed the best on the task of predicting both stock prices. With an RSME score of 0.01% and 3.75% on the LMT and NOC stocks, respectively, the model outperformed ARIMA (RMSE (LTM) = 0.0%, RMSE (NOC) = 5%) and the LSTM-RNN (RMSE (LTM) = 5%, RMSE (NOC) = 5%). However, the author discussed that the LSTM-RNN model should have performed better, because the amount of input data should have been more. Another limitation that is discussed in the paper is the transparency of the FBProphet model, since the documentation states that it is an additive linear model we do not know the whole algorithm.

2.3 *Sentiment analysis on social media*

According to Drus and Khalid (2019), the environment of social media is changing since the rise of the participative social web (Web 2.0).The authors describe online social media as a way to “connect, share information and their personal opinion to others”. According to Dixon (2022), the number of social media users in 2021 was over 4.26 billion. This number is expected to be six billion in 2027. These users include strategic leaders, such as presidents, CEOs, or other influential individuals, who can use their knowledge and social network to expand their power or share their beliefs, as discussed in Ante (2023). With the rising number of social media users, the amount of data that is generated by these users is growing. According to Kumar and Sebastian (2012), this increase in data can be an opportunity to develop technologies that analyze the data and mine opinions shared on social media. Sentiment Analysis is a technology that uses textual data to identify and classify sentiments in order to express an opinion with a value (Neethu & Rajasree, 2013). VADER (Valence Aware Dictionary and sEntiment Reasoner), proposed by Hutto and Gilbert (2014), is a "lexicon and rule-based tool specifically for sentiments expressed in social media". A study by Kiani, Al Natour, and Turetken (2018), compared the results of the VADER algorithm with the Google Cloud Natural Language API. The task for the models was to analyze the sentiments of consumer product reviews. While one is a lexicon and rule-based method, the other considers a machine learning approach. The methods were used on the data that was collected from 625 consumer reviews, where each review was between 10 to 551 words. The models were used on the reviews and the star rating the reviewers provided. The authors deduced from the results of an ANOVA that the VADER algorithm outperformed the machine learning approach. Where the Google Cloud Natural Language API scored 80.2% accuracy, the VADER algorithm scored an accuracy of 83.4%.

2.4 *Forecasting cryptocurrencies with sentiment analysis*

The correlation between the price prediction of cryptocurrencies and sentiments of tweets has been shown in several studies, such as in the paper by Valencia, Gómez-Espinosa, and Valdés-Aguirre (2019). The authors compared three machine learning models to predict financial time series: neural networks (NNs), Support Vector Machines (SVMs) and Random Forests (RFs). These models were applied to four cryptocurrencies: Bitcoin, Ethereum, Ripple, and Litecoin. The social data is gathered with the sentiment analysis tool VADER. The paper uses three approaches for inputs of the models. In the first approach, the model trains exclusively on the social media data. In the second approach, the model trains only on the market data, and in the third approach, the model trains on both the social media data and the market data. The performance of each model was evaluated and from the results, the authors concluded that the best model was the NN with the Bitcoin data, which was trained and tested on both the social media data and the market data. This model had an accuracy of 0.72 and a precision score of 0.74. These results showed that there is indeed a correlation between cryptocurrencies and Twitter messages and that it is possible to predict cryptocurrencies using these messages. There are many variations of research that show that sentiment analysis of social media messages can help to predict the stock market movement (Ibrahim, 2021). However, in a paper proposed by Kraaijeveld and De Smedt (2020), there are limitations to current research on this topic. The authors describe in their research the influence of Twitter sentiment on the prediction of cryptocurrency prices. They compared the price returns and daily trading volumes of several cryptocurrencies with the influence of tweet volume and tweet sentiments. From the results, the authors concluded that there is a predictive relationship between tweet sentiments, tweet volume and the price of cryptocurrencies. The authors discussed three limitations that they found in other current literature. The first limitation they discussed was that there are only a few papers attempting to predict the price of an altcoin using tweet sentiments, which means that there is not a lot of background to research this topic. Furthermore, in most of the research that was conducted limited data was used due to the limitations of the Twitter API, a small set of search query terms, and/or only gathering tweets from a short time. Lastly, the papers published previously to the paper by Kraaijeveld and Smedt do not include specific slang or language that is used on social media for cryptocurrencies.

3 METHODS

In this section, the methods that are used and how they are used in this research are explained. The setup of the research will begin with a description of the extraction process of the financial data and the tweets. Secondly, the data preprocessing and the process of the calculation of sentiment analysis will be described. Furthermore, the models that are used (FBProphet and LSTM-RNN) will be extensively explained. Lastly, the experimental setup will be discussed.

3.1 *Data extraction*

This section is split up into two sections. The first section is the extraction of financial data of DOGE, SHIBA and MONA. The second section explains the process of the scraping of tweets using the snscrape module.

3.1.1 *Financial data*

The historical data of the coins mentioned above were extracted from Yahoo! Finance (*yahoo!*, n.d.) and are processed with the Python 3 (Van Rossum & Drake, 2009), using the Pandas library (McKinney, 2010). The features of the datasets are:

- Open: Opening price
- High: Highest price
- Low: Lowest price
- Close: Closing price
- Adj Close: Closing price that takes corporate actions into account, such as taxes and dividends.
- Volume: The number of shares that are traded

The specifics of the data are shown in Tables 4, 5 and 6 and the closing prices of the data are plotted in Figure 4 (See Appendix (A)). For DOGE, and MONA, not the complete historical data was used. This is because of the time complexity of the tweet scraping which will be elaborated in section 5.

3.1.2 *Tweet scraping*

In this research, messages from Twitter, tweets, are used for sentiment analysis. The snscrape module is used for the scraping of tweets from the Twitter API (JustAnotherAchivist, 2018-2021). The tool can extract various features, such as tweet date, tweet content, number of followers of a user, and tweet replies. For this research, the tweets on the dates corresponding to the dates of the financial data (section 3.1.1) are scraped. A limit of a hundred tweets per day is used, keywords corresponding to the cryptocurrencies are used, and the language is set to English. Table 1, shows the specifics of tweet scraping process. The snscrape module is used instead of other modules, because snscrape was easy to install and implement. Besides, the module has no limitations, which means that the user can scrape tweets unlimited.

3.2 *Data preprocessing and Sentiment Analysis*

The data preprocessing is mainly focused on sentiment analysis, as the financial data is already sufficient enough to use. The only preprocessing step of the financial data was the removal of the 'Volume' feature, since

Table 1: Specifics Tweet scraping

Coin	Number of tweets	Date range	Words
DOGE	84234	05/08/2020-16/11/2022	'doge', 'dogecoin', 'dogecrypto'
SHIBA	82530	17/02/2020-16/11/2022	'shiba', 'shibacoin', 'shibacrypto', 'shiba inu'
MONA	84234	05/08/2020-16/11/2022	'mona', 'monacoin', 'monacrypto'

it was not useful for this research. The sentiment analysis is done with the VADER algorithm on the scraped tweets (Hutto & Gilbert, 2014). The preprocessing of tweets can be divided into three parts: cleaning, VADER, and calculating sentiment per day.

3.2.1 *Cleaning*

In the first part, the tweets are cleaned to be applicable in the VADER algorithm. The tweets are stripped from 'RT' handles, usernames '@...', urls 'https://. . . ', and whitespaces. The date and time of the tweets extracted with snscrape have a format of 'YYYY-MM-DD hh:mm:ss sTZD', which has been adjusted to the format of 'YYYY-MM-DD'. Examples of cleaned tweets are presented in Table 2.

3.2.2 *VADER*

As mentioned before, VADER is used to perform the sentiment analysis on the cleaned tweets. The model is a Natural language Processing method that expresses sentiment polarity and intensity using sentiment lexicons in combination with grammatical rules and syntactical patterns. The authors listed existing word/phrase dictionaries, and added emoticons (":-("), acronyms("WTF"), and common slang("meh", "nah"). These instances were rated by humans with a score between -4 (Extremely Negative), 0 (Neural), and 4 (Extremely Positive). Next, the average of each lexicon was calculated and assigned to the corresponding lexicon. The list contains about 7500 lexicons with sentiment, if a word is not in this list a '0' (Neutral) is assigned to it as sentiment (Hutto & Gilbert, 2014). The grammatical rules in the VADER algorithm are formats of characters in a sentence that can change the polarity and/or intensity of the sentiment. Examples of such formats are: "but", "nope", "absolutely", and "flippin". The

Table 2: Example tweets and cleaned tweets

	Date + Tweet	Cleaned Date + Tweet
Tweet DOGE	2020-08-05 23:53:12+00:00,Strapped in n′ ready to go #bitcoin #doge #XRP https://t.co/6y3LfdJfLp	2020-08-05,Strapped in n′ ready to go #bitcoin #doge #XRP
Tweet SHIB	2021-09-02 23:33:52+00:00,@Shibtoken LFG SHIBA FAMILY!!!	2021-09-02,LFG SHIBA FAMILY!!!
Tweet MONA	2020-10-03 23:21:02+00:00,A big chance in a million! $MONA price: 127.3 MONA/JPY (2020/10/04 08:20)	2020-10-03,A big chance in a million! $MONA price: 127.3 MONA/JPY (2020/10/04 08:20)

designers of the algorithm BRON VADER implemented heuristic rules for managing adverbs, capitalization, punctuation and conjunctions with contrast. For example, the sentence "This algorithm is VERY GOOD :-)!!!" has a more positive sentiment than the sentence "This algorithm is good.". The sentiment score (compound) is calculated by using the lexicons and modifying the intensity and/or polarity of the text using the grammatical rules, to sum the scores found in the text. The formula:

$$\frac{x}{\sqrt{x^2 + \alpha}}$$

is used to normalize the compound between -1 and 1, where the x is the total sentiment score of a text and the is the maximum value that x can have. The VADER algorithm is used because this algorithm is specifically trained for social media messages, which contain, for example, slang and emoticons, which other models do not or barely pay attention to (Hutto & Gilbert, 2014).

3.2.3 *Calculate Sentiments per Day*

The 'compound' values of the tweets per day are calculated, which is done by summing the compound values of date X, dividing this by the number of days with the same date as date X and adding these to a list. When there is no tweet available on date X, a 'NaN' value is added to the list. These lists of compound values per day are transformed into NumPy arrays (Harris et al., 2020) and are added to the financial data. The 'NaN' values in the compound column of datasets with the cryptocurrencies, were replaced by the mean of that column in order to prevent missing values that can cause errors in the upcoming steps.

3.3 *Models*

The following section describes the models, FBProphet and LSTM-RNN, which are used to forecast the price fluctuations of the meme coins. These models are chosen, because both of the models have not been researched extensively on the price fluctuation prediction of meme coins. Furthermore, the process of how they were used, and the evaluation metric RMSE are described. Since the process consists of many tasks, Figure 5 presents this process in a diagram (See Appendix A). Firstly, an explanation of time series data is given, which is necessary for understanding the models.

3.3.1 *Time Series Data*

Both the LSTM-RNN and the FBProphet model rely on time series data, which can be found in a wide range of fields, such as in monitoring health data of patients, the price movements of stocks, or weather forecasting (Petelin, Cenikj, & Eftimov, 2023). A definition of time series data is given by Mallikarjuna and Rao (2019): "a time series is a series of observations x_t, observed over a period of time". These observations can vary in intervals, such as the whole interval, a fixed interval, or a random interval. To investigate and extract meaningful features of time series data, time series analysis is used. The understanding of temporal patterns of a set of data points is assisted by time series analysis, which is why the goal is to create a model that properly represents the pattern of the data and can be used for forecasting.

3.3.2 *FBProphet*

FBProphet Taylor and Letham (2017) is a forecasting method developed by Facebook that uses an additive model in which non-linear trends can be fitted with several basic effects (yearly, weekly, daily, seasonality, and holiday effects). Since these effects can be implemented, the model works optimally with strong seasonal effects and various seasons in the time series. In the research of this paper, these basic effects will not be included, since there are no strong seasonal effects in the used data. However, as mentioned in section 2.2.2, it is shown that cryptodata can be used as the input for an FBProphet model. The model is a decomposable time series model which consists of three parts, namely trend, seasonality, and holidays:

$$y(t) = g(t) + s(t) + h(t) + e(t)$$

where:

- t is the time,
- $g()$ is the trend which models non-periodic changes (linear or logistic),
- $s()$ is the seasonality, which models periodic changes (weekly, monthly, yearly),
- $h()$ ties in effects of holidays.
- $e()$ captures the distinctive changes that the model does not take into account.

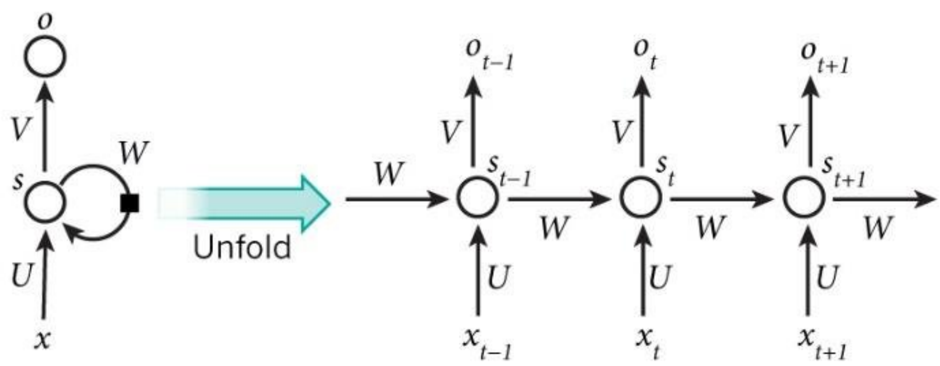
Instead of examining each observation's explicit time-based dependency, the model frames the forecasting issue as a curve-fitting problem. The formula describes the basic formula that is used in a basic model for univariate prediction, with time as a regressor. However, adding multiple regressors results in multivariate prediction.

3.3.3 *LSTM-RNN*

Artificial Neural Networks (ANNs) are briefly described to get a better understanding of Long Short Term Memory (LSTM) units in Recurrent Neural Networks (RNNs). ANNs consist of artificial neurons, which are connected in a network to mimic the biological neurons in the brain (Zupan, 1994). These neural networks are used in a wide range of tasks, such as "classification, clustering, pattern recognition and prediction in many disciplines" (Abiodun et al., 2018). ANNs can be used as a basis for more complex networks that can do different tasks. Typically, ANNs consist of three layers: the input layer, (one or more) hidden layer(s), and the output layer. Each layer consists of connected individual neurons with their own weights and activation function. The input layer is presented with the training data and the weights are adjusted using backpropagation. Error backpropagation is the most popular network learning technique, where the weights are learned by the use of gradient descent. The technique uses adjustable iterative steps to calculate the error of the network and adjust the weights from the output layer back to the input layer (Staudemeyer & Morris, 2019). In this training phase patterns that are in the data can be learned. To use ANNs for more complex data such as time series, a variation of ANNs is created, namely Recurrent Neural Networks (RNNs).

RNNs are based on sequence modelling, which is the method of predicting the next value on time $t+1$. In the paper by Yu, Si, Hu, and Zhang (2019), the authors describe a basic RNN and its possible variations. They mention that the recurrent layers (hidden layers) in RNNs consist of cells that depend on the past output state and current input state with feedback connections. These connections make a cycle, which lets data move in multiple directions, contrasting the ANN that was previously described.

Figure 1: Basic RNN by W. Zhang et al. (2019)



In Figure 1 (W. Zhang et al., 2019), a basic RNN is shown, which can be described in the following equations:

$$s_t = \phi(Ux_t + Ws_{t-1})$$
$$y_t = softmax(Vs_t)$$

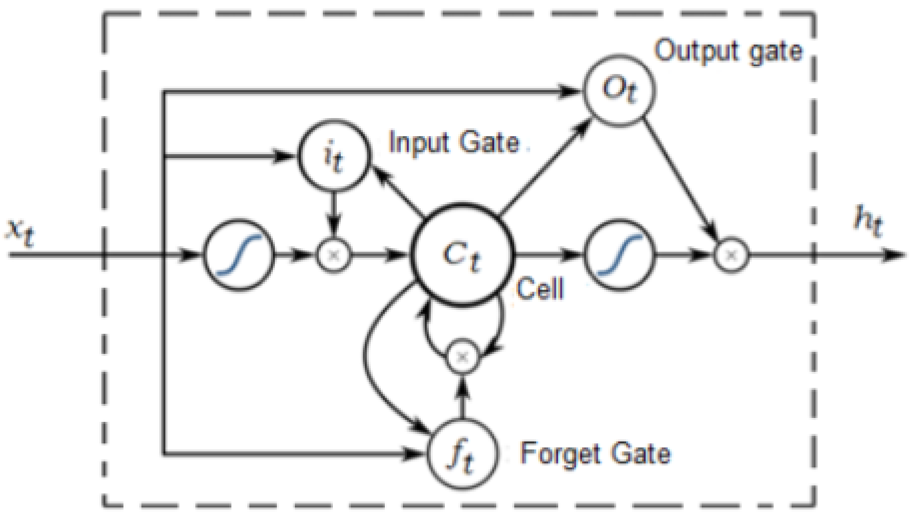
where:

- x_t is the input at time t
- W, U, V are the weight parameter matrices
- s_t is the t step state of the hidden layer (memory unit of the network)
- y_t is the output of the step on time t
- ϕ is the nonlinear activation function which can be, for example, sigmoid or tanh.

Yu et al. (2019), described that the layers are connected with matrices with weight parameters assigned to W, U, V, which update the learned parameters that can be trained by the network with backpropagation. Backpropagation is used to update the weights of the networks for which the model can recognize patterns in the data. When new data is input into the model, the information from $t-1$ is calculated to become the output at that current moment.

However, according to Staudemeyer and Morris (2019), a standard RNN cannot learn from more than 5-10 time steps, because the error that is back-propagated increases or decreases with each time step. When there are many time steps, the error increases (blows up) or decreases (vanishes) too much. The increased error can result in "oscillating weights", where the learning takes too long or does not learn at all with the decreased error. One of the solutions to this "vanishing gradient" problem is by implementing an LSTM unit into the RNN. By implementing a "gate" cell, Hochreiter and Schmidhuber (1997) proposed an LSTM unit which enhanced the memory capacity of the standard recurrent cell in the standard RNN. There are three types of these 'gate' cells developed, namely the 'LSTM without a forget gate', 'LSTM with a peephole connection', and 'LSTM with a forget gate'. However, since this research is using the latter cell, this is briefly described. The cell is a modified LSTM unit with an added forget gate, proposed by Gers, Schmidhuber, and Cummins (2000). The LSTM cell is described by Bulut (2021) and can be seen in Figure 2.

Figure 2: LSTM cell with forget gate by Bulut (2021)



The author describes three different gates that control the flow of information, namely the input gate (i), the forget gate (f), and the output gate (o). By combining the output of t-1 and the current input, these gates can be regulated. To discuss the LSTM more extensively, Figure 2 is explained. The input gate can inspect if the current cell state allows it to add new external information to it, which can be described in the formula:

$$i_t = \sigma_k(W_i x_t + U_i h_{t-1} + b_i,$$

where:

- I_t is the entrance gate vector,
- σ_k is the sigmoid function,
- x_t is input vector,
- W_i and U_i are parameter matrices,
- b_i is the bias vector.

The output vector tells us whether or not to keep the current cell state, which influences other cells. This process can be described in the following formula:

$$o_t = \sigma_k(W_o x_t + U_o h_{t-1} + b_o$$

The forget gate can reset the currency state of the LSTM and its formula is:

$$f_t = \sigma_k(W_f x_t + U_f h_{t-1} + b_f$$

The cell state and the output are the results from combining the formulas above:

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_f x_f + U_c h_{t-1} + b_c)$$
$$(h_t = o_t \odot \sigma_h(c_t) \;\; or \;\; (h_t = o_t \odot tanh(c_t)$$

The $\odot$ is the Hadamart product which is a type of vector multiplication, σ_h represents hyperbolic tangent functions.

3.4 *Experimental Setup*

As can be seen in Figure 5 (See Appendix A), the experiment runs over the two models described in section 3.3. The experimental setup for FBProphet is almost equivalent to the setup for the LSTM-RNN. However, since they do differ in certain parts, they are described apart from each other. Both the experimental setups start with an initial preprocessed dataset acquired from sections 3.1 and 3.2. Based on the research question we want to predict the percentage change per n_days of the Close price with and without the use of sentimental analysis.

3.4.1 *Experimental Setup FBProphet*

The first step of the setup for FBProphet is to create and add percentage change columns to our dataset, since we want to predict the price fluctuation of the Closing price. This is done by a function that looks n_days back in time and calculates the percentage change between $t - n_days$ and t on the Closing price. The number of n_days are: 1, 7, 14 and 30 days, which means that the data from $t - n_days$ till t is used to calculate the price percentage change on day t. An example of the dataset can be seen in Figure 3.

Figure 3: Example of dataframe for FBProphet

	Date	Open	High	Close	Low	Adj Close	1_day	7_day	14_day	30_day
829	2022-11-12	0.400014	0.402853	0.375867	0.371395	0.375867	-0.065107	-0.173130	-0.143116	-0.113279
830	2022-11-13	0.375490	0.380881	0.368894	0.366283	0.368894	-0.018552	-0.176592	-0.153954	-0.121756
831	2022-11-14	0.369111	0.369990	0.359320	0.353833	0.359320	-0.025953	-0.180964	-0.213754	-0.168360
832	2022-11-15	0.361521	0.379844	0.375734	0.358986	0.375734	0.045681	-0.113714	-0.145562	-0.120782
833	2022-11-16	0.375686	0.379038	0.371419	0.369329	0.371419	-0.011484	-0.038962	-0.155609	-0.122213

Due to the fact that there are no days before $t = 0$, the columns of percentages are missing the first n_days. This is solved by making four different datasets for 1, 7, 14, and 30 days lookback, where the first n_days rows are deleted. The next step is to rename the 'Date' feature to 'ds' and the column we want our model to train on to 'y', in this case, the n_days price fluctuation column. This is done because FBProphet uses these names to detect what it needs to forecast. The following step consists of splitting the dataset into a train and test set, with a size of 85% and 15% of the total data, respectively. The fourth step is the fitting of the FBProphet model on the training set, where the model only detects the 'ds' and the 'y' column. To use FBProphet on multivariate data, it is required to add regressors to the model. Since all the features of the dataset will be used in the forecasting, the remaining columns are added to the model as regressors. For the forecasting with sentiment analysis, a regressor for the compound value is added. Using the fitted model, the test process can start. As mentioned in section 1, this thesis will research if the models can predict the percentage change over the following 1, 7, 14, and 30 days. To clarify, each model (trained on 1, 7, 14 and 30 days lookback) will predict the price fluctuation for 1, 7, 14, and 30 days. This is done by creating a rolling window of the test set, where the test set is split up into smaller sets. These sets have the length of n_days and are each step increased with 1, e.g. 0-n_days, 1-n_days+1, 2-n_days+2. The predicted values and the true values were compared and the Root Mean Squared Error (RMSE) was calculated which will be explained later on. The baseline, which the models will be compared to, is the RMSE value of the prediction of one day with a lookback period of one day without the addition of sentiment. This baseline will be used throughout the paper.

3.4.2 *Experimental Setup LSTM-RNN*

The initial step for the setup of the LSTM-RNN is equal to that of the FBProphet model. The only difference is that the calculated fluctuation of the Closing prices over n_days is added to a separate list called the y set. The X set is a copy of the financial dataset with the columns: Close, Open, High, Low, and Adj Close. The next step is the scaling of the values in the X and y sets. There were two types of MinMaxScalers (module by sci-kit learn (Pedregosa et al., 2011))) used, which normalize the data between the given range. The values of X are scaled between 0 and 1 and the values of y are scaled between -1 and 1. The scaling of values is necessary because the model can recognize patterns better in scaled data.

When the model has to be trained with the sentiment analysis, the compound column is also added to the X set. LSTM-RNNs for multivariate data allow a certain shape of input from the training set, [N, n_days, num_features]. This shape of data is created by implementing a rolling window on the X set, where the lookback period of n_days is used to make subsets of $t - n_days$ till day t and add them into one array. When the sentiment analysis is used, the number of features is six, because there is one more column in the dataset, namely 'compound'. To create the y set with the true values that will be used for calculating the RMSE value, another rolling window is implemented. The window takes the y set and splits it into separate arrays with a length of n_forecast.

In the fourth step, the X and y set are split into training and test sets, with a size of 85% and 15%, respectively. The model is made with the use of the tensorflow.keras module, which was proposed by Chollet (n.d.). The model uses four LSTM units with 128,64,64,128 neurons, respectively, in each hidden layer and a dropout rate of 0.1. Lastly, a Dense layer of n_days is added to output the data in the desired shape, namely the amount of days that the model wants to predict. The dropout layer randomly sets weights to zero with each time step during the training of the model, which prevents the model from overfitting. Lastly, the model is compiled using the 'adam' optimizer and the MSE loss function. The optimizer 'Adam' (Kingma & Ba, 2014) is used to reduce the error of the model by changing the weights and learning rate through back-propagation. The MSE loss function is used to calculate the error of the model. The next step in the setup is to define the parameters that are used for fitting the model, which are:

- X_train, y_train: The model is fitted on the training data for each of the lookback periods,
- Epochs = 30: Define how many cycles the data goes through the network,
- Batchsize = 16: The number of samples that are passed through the models at once,
- Validation data: Used for scoring the ability of the model to predict based on an unseen data point. The X_test and y_test of the corresponding lookback periods were used.

The sixth step is to make predictions on the X_test and y_test sets using the fitted model. Since the Dense layer in the model is equal to n_days that the model wants to predict, the outcome of the predictions are also of size n_days. For example, when the model wants to predict 14 days into the future, the test set will predict the following: 0:14 days of the test set, 1:15 days of the test set, etc. Since the y test set also has this shape, it can be easily compared to the predictions. This is done by first unscaling the y test and predictions with the MinMaxScaler, which then can be used to calculate the RMSE score. There are in total four models trained and tested for each n_days.

3.4.3 *RMSE*

The evaluation metric Root Mean Squared Error (Chai & Draxler, 2014) is used to be a comparative value between models and is calculated by taking the square root of the mean squared error. The formulas as described below:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2$$

$$RMSE = \sqrt{(MSE)}$$

where:

- n is the number of samples
- y is the the true value
- $\hat{y}$ is the predicted value

The lower the RMSE value the better the model can predict the price fluctuation.

4 RESULTS

In this section, the results from the models are explained, which are the basis for answering the research questions proposed in section 1. The results are divided into several tables, each with the RMSE score as a comparative component between the models. As mentioned in section 3.4, the data of each coin is used to predict the price fluctuation using four variations of LSTM-RNN and FBProphet models. This results section will present three different findings:

- Best result overall
- Best method
- Comparison of the models with and without sentiment analysis

Since the total number of RMSE scores is 256 for all predictions using four models for LSTM-RNN and FBProphet, all the results are presented in tables 7 and 8, which can be found in the Appendix A. These tables are divided into the four coins that are used, which are divided into four rows (number of predicted days) and four columns ('model 1', 'model 7', 'model 14', and 'model 30'). The columns are subdivided into two cells: 'with' and 'without'. The features of the rows and columns are explained as follows:

- Model 1: Method trained with 1-day lookback
- Model 7: Method trained with 7-day lookback
- Model 14: Method trained with 14-day lookback
- Model 30: Method trained with 30-day lookback
- With/without: with or without the added sentiment of tweets
- Number days: number of predicted days

The best results per coin, method, and with/without sentiment are presented in table 3

4.1 *Best Results Overall*

The best results per coin, method, and with/without sentiment are presented in table 3. The best result by the LSTM-RNN model is obtained on the MONA dataset. It is a model trained on 7-days lookback and predicting 14 days without added sentiment (RMSE = 0.0003). The best result by the FBProphet model is obtained on the MONA dataset. It is a model trained

Table 3: Best results of the LSTM-RNN and FBProphet models

LSTM	DOGE	SHIB	MONA
Without sentiment	Model 1 7 days 0.0026	Model 7 30 days 0.0058	Model 7 14 days 0.0003
With sentiment	Model 1 7 days 0.0005	Model 1 30 days 0.0006	Model 1 30 days 0.0018
FBProphet	**DOGE**	**SHIB**	**MONA**
Without sentiment	Model 1 1 day 0.0732	Model 1 1 day 0.0608	Model 1 1 day 0.0218
With sentiment	Model 1 1 days 0.0732	Model 11 1 day 0.0652	Model 1 1 day 0.0218

on 1-day lookback and predicting 1 day with and without added sentiment (RMSE = 0.0218). The best FBProphet model is the baseline (model trained on one day, predicting one day without added sentiment) and there is no other model that performed better. When looking at the overall results in table 9, it can be observed that for the LSTM-RNN model trained on all the coins (DOGE, SHIB, and MONA), the difference between the best (RMSE = 0.0005, RMSE = 0.0006 and RMSE = 0.0014, respectively) and second best result (RMSE = 0.0003, RMSE = 0.0058 and RMSE = 0.0005, respectively) is not significant. For the FBProphet model, the difference of DOGE and MONA between the best (RMSE = 0.0732 and RMSE = 0.0218, respectively) and second best model (RMSE = 0.1542 and RMSE = 0.0424, respectively) is significant.

4.2 *LSTM-RNN or FBProphet*

From the results that are presented in tables 7 and 8, the best performing method can be derived. When comparing the RMSE values of both the methods, it can be seen in table 7 and 8 that there are more RMSE values that are lower in the results of the LSTM-RNN method in comparison to the FBProphet method. Furthermore, the FBProphet method shows that it does not perform better than the baseline RMSE scores of all the coins (DOGE (RMSE = 0.0732), SHIB (RMSE = 0.0608), and MONA (RMSE = 0.0218)). However, the LSTM-RNN method does show that it performs better with most models than the baseline RMSE scores.

4.3 *With or Without Sentiment*

In this section, the comparison of the RMSE scores with and without added sentiment is described. In table 7 there are several cases where the added sentiment results in a lower RMSE value. From the RMSE scores obtained by the LSTM-RNN method it can be seen that model 7, model 14, and model 30, trained on SHIB, do indeed show a significant difference. However, for the rest of the table, the results show that this comparison has no significant results. In several cases, the RMSE scores with sentiment are higher than the RMSE scores without sentiment added. For example, for model 30 trained on DOGE it can be seen that the results for the model with added sentiment are significantly higher than the scores without added sentiment. For the FBProphet method, table 8 can be used to determine if added sentiment improves the RMSE value of the method. As can be observed in the table, there is no significant difference in all models with or without added sentiment. There are models where the RMSE scores are higher with sentiment than without sentiment, such as for models 7 and 14, on the DOGE and SHIB datasets, where the values of the column with sentiment are substantially higher than the column without sentiment.

In this section, the comparison of the RMSE scores with and without added sentiment is described. In table 7 there are several cases where the added sentiment results in a lower RMSE value. From the RMSE scores obtained by the LSTM-RNN method, it can be seen that the following models showed a significant difference between the RMSE values with and without added sentiment:

1. DOGE
   (a) model 14 with lookback days 7 (with sentiment RMSE = 0.0121 and without sentiment RMSE = 0.9042)
2. SHIB
   (a) Model 1 with lookback days 30 (with sentiment RMSE = 0.0006 and without sentiment RMSE = 0.0122)
   (b) Model 7 with lookback days 7 (with sentiment RMSE = 0.0411 and without sentiment RMSE = 0.2559)
   (c) Model 14 with lookback days 30 (with sentiment RMSE = 0.0698 and without sentiment RMSE = 0.3491)
3. MONA
   (a) Model 7 with lookback days 30 (with sentiment RMSE = 0.0049 and without sentiment RMSE = 0.0314)

(b) Model 14 with lookback days 7 (with sentiment RMSE = 0.0083 and without sentiment RMSE = 0.0135)

In table 8 can be seen that there is no case where the RMSE score of the model with added sentiment is significantly lower than the RMSE score of the model without added sentiment.

5 DISCUSSION & CONCLUSION

To come back to section 1, the goal of this thesis was to see if the price percentage change of meme coins could be predicted with an LSTM-RNN and the FBProphet method. In addition, a second goal was to see if the sentiment of tweets would improve the predictive ability of the methods in predicting the price percentage change of meme coins. By looking at the results, it can be concluded that it is indeed possible for both models to predict the price percentage change of meme coins, since both methods score relatively low RMSE scores. This indicates that the models predict closely to the actual values. Overall, the LSTM-RNN method does predict the price percentage change better than the FBProphet method, as can be seen in section 3. The LSTM-RNN method has an improved predictive ability with the addition of sentiments of tweets of the prediction of the price percentage change. However, the predictive ability of the FBProphet method does not improve with the addition of sentiments of tweets.

Nonetheless, the research has limitations that might give a misrepresentation of the results. The first limitation is the amount of data that was used in this research. This amount for all the coins was too limited to properly train the models, which was caused by the period of time that the coins were released. This limitation caused that there was no convergence of the training and validation loss while training the LSTM-RNN method. This resulted in a higher training loss than the validation loss, which can conclude that the LSTM-RNN method was underfitting. The amount of data was partially caused by another limitation, the time of scraping the tweets. Due to this time limitation, not every date in the historical datasets could be scraped. For one data set of about eight hundred historical data points, the function for scraping a hundred tweets per day took around 130 minutes. This had to be done on three different datasets. Furthermore, the scraping tool only took the last hundred tweets of a day, which means, for example, that the first tweet of a day was scraped at around 23:45 on that day. This is a limitation on its own because the tone of these tweets could be different from the tone of tweets at the beginning or in the middle of the day.

Another limitation of the tweets is the fact that there are bots on social media that post messages on a minute basis. This means that most of the tweets are from bots, which can be misleading, because, in this thesis, it is more helpful to get tweets from real people. The last limitation concerns the hyperparameter tuning of the LSTM-RNN model. Due to time limitations, this hyperparameter tuning was not implemented in the code. This would have been helpful to get a model that was adjusted based on the data that was presented instead of manually adjusting the parameters.

REFERENCES

(n.d.).

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018, November). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, *4*(11), e00938.

Adam, K., Marcet, A., & Nicoli, J. P. (2016). Stock market volatility and learning. *The Journal of Finance*, *71*(1), 33–82.

Ante, L. (2023, January). How elon musk's twitter activity moves cryptocurrency markets. *Technol. Forecast. Soc. Change*, *186*(122112), 122112.

Brans, H., & Scholtens, B. (2020, Mar). *Under his thumb the effect of president donald trump's twitter messages on the us stock market.* Public Library of Science. Retrieved from https://journals.plos.org/plosone/article?id=10.13712/journal.pone.0229931

Bulut, M. (2021). Hydroelectric generation forecasting with long short term memory (LSTM) based deep learning model for turkey.

Chai, T., & Draxler, R. R. (2014, June). Root mean square error (RMSE) or mean absolute error (MAE)? – arguments against avoiding RMSE in the literature. *Geosci. Model Dev.*, *7*(3), 1247–1250.

Chohan, U. W. (2017). A history of dogecoin. *SSRN Electron. J.*.

Chollet, F. (n.d.). *keras: Deep learning for humans.*

Cryptocurrency prices, charts and market capitalizations. (n.d.). Retrieved from https://coinmarketcap.com/

Dixon, S. (2022, Sep). *Number of worldwide social network users 2027.* Retrieved from https://www.statista.com/statistics/278414/number-ofworldwide-social-network-users.

Drus, Z., & Khalid, H. (2019). Sentiment analysis in social media and its application: Systematic literature review. *Procedia Comput. Sci.*, *161*, 707–714.

Ferdous, M. S., Chowdhury, M. J. M., & Hoque, M. A. (2021, May). A survey of consensus algorithms in public blockchain systems for crypto-currencies. *J. Netw. Comput. Appl.*, *182*(103035), 103035.

Frankenfield, J. (2022, Aug). *Altcoin explained: Pros and cons, types, and future.* Investopedia. Retrieved from https://www.investopedia.com/terms/a/altcoin.asp

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000, 10). Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, *12*(10), 2451-2471. Retrieved from https://doi.org/10.1162/089976600300015015 doi: 10.1162/089976600300015015

Giudici, G., Milne, A., & Vinogradov, D. (2019, September). Cryptocurrencies: market analysis and perspectives. *Journal of Industrial and Business Economics*, *47*(1), 1–18. Retrieved from https://doi.org/

10.1007/s40812-019-00138-6 doi: 10.1007/s40812-019-00138-6

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T. E. (2020, September). Array programming with NumPy. *Nature*, *585*(7825), 357–362. Retrieved from https://doi.org/10.1038/s41586-020-2649-2 doi: 10.1038/s41586-020-2649-2

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hutto, C. J., & Gilbert, E. E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international conference on weblogs and social media (ICWSM-14).* Ann Arbor, MI.

Ibrahim, A. (2021). Forecasting the early market movement in bitcoin using twitter's sentiment analysis: An ensemble-based prediction model. In *2021 IEEE international IOT, electronics and mechatronics conference (IEMTRONICS).*

JustAnotherAchivist. (2018-2021). *snscrape: A social networking service scraper in python.* https://github.com/JustAnotherArchivist/snscrape. GitHub.

Kiani, A., Al Natour, S., & Turetken, O. (2018). A comparison of sentiment analysis tools. In *AMCIS 2018 proceedings.*

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization.

Konoth, R. K., Vineti, E., Moonsamy, V., Lindorfer, M., Kruegel, C., Bos, H., & Vigna, G. (2018, January). MineSweeper. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security.* ACM. Retrieved from https://doi.org/10.1145/3243734.3243858 doi: 10.1145/3243734.3243858

Kraaijeveld, O., & De Smedt, J. (2020, March). The predictive power of public twitter sentiment for forecasting cryptocurrency prices. *J. Int. Financ. Mark. Inst. Money*, *65*(101188), 101188.

Kumar, A., & Sebastian, T. (2012, 07). Sentiment analysis on twitter. *International Journal of Computer Science Issues*, *9*, 372-378.

Li, A. W., & Bastos, G. S. (2020). Stock market forecasting using deep learning and technical analysis: A systematic review. *IEEE Access*, *8*, 185232–185242.

Li, C., & Yang, H. (2022, December). Will memecoins' surge trigger a crypto crash? evidence from the connectedness between leading cryptocurrencies and memecoins. *Finance Research Letters*, *50*, 103191. Retrieved from https://doi.org/10.1016/j.frl.2022.103191 doi: 10.1016/j.frl.2022.103191

Mallikarjuna, M., & Rao, R. P. (2019, December). Evaluation of forecasting methods from selected stock market returns. *Financial Innovation*, *5*(1).

Retrieved from https://doi.org/10.1186/s40854-019-0157-x doi: 10.1186/s40854-019-0157-x

Mazed, M. (2019). *Stock price prediction using time series data* (Unpublished doctoral dissertation). Brac University.

McKinney, W. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th python in science conference.* SciPy.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*.

Neethu, M., & Rajasree, R. (2013). Sentiment analysis in twitter using machine learning techniques. *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, 1-5.

Oyelude, A. A. (2022, August). Cryptocurrency: the future money here now. . . . *Library Hi Tech News*, *39*(8), 1–4. Retrieved from https://doi.org/10.1108/lhtn-07-2022-0089 doi: 10.1108/lhtn-07-2022-0089

Patel, M. M., Tanwar, S., Gupta, R., & Kumar, N. (2020, December). A deep learning-based cryptocurrency price prediction scheme for financial institutions. *Journal of Information Security and Applications*, *55*, 102583. Retrieved from https://doi.org/10.1016/j.jisa.2020.102583 doi: 10.1016/j.jisa.2020.102583

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Petelin, G., Cenikj, G., & Eftimov, T. (2023, March). Towards understanding the importance of time-series features in automated algorithm performance prediction. *Expert Syst. Appl.*, *213*(119023), 119023.

Roondiwala, M., Patel, H., & Varma, S. (2017). Predicting stock prices using LSTM. *International Journal of Science and Research (IJSR)*, *6*(4), 1754–1756.

Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM – a tutorial into long Short-Term memory recurrent neural networks.

Taylor, S. J., & Letham, B. (2017, September). *Forecasting at scale.*

Valencia, F., Gómez-Espinosa, A., & Valdés-Aguirre, B. (2019, June). Price movement prediction of cryptocurrencies using sentiment analysis and machine learning. *Entropy (Basel)*, *21*(6), 589.

Van Rossum, G., & Drake, F. L., Jr. (2009). *Python 3 reference manual*. Createspace.

Yenidogan, I., Cayir, A., Kozan, O., Dag, T., & Arslan, C. (2018, September). Bitcoin forecasting using ARIMA and PROPHET. In *2018 3rd international conference on computer science and engineering (UBMK).* IEEE.

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019, July). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.*, *31*(7), 1235–1270.

Zhang, G. P. (2003, January). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, *50*, 159–175.

Zhang, W., Huang, G., Wang, G., & Wang, Y. (2019, October). Prediction high frequency parameters based on neural network. *IOP Conf. Ser. Mater. Sci. Eng.*, *631*(5), 052035.

Zheng, Z., Xie, S., Dai, H. N., Chen, X., & Wang, H. (2018). Blockchain challenges and opportunities: a survey. *Int. j. web grid serv.*, *14*(4), 352.

Zupan, J. (1994). Introduction to artificial neural network (ANN) methods: what they are and how to use them. *Acta Chimica Slovenica*, *41*, 327–327.

[titletoc]appendix

float

A APPENDIX A

Table 4: Exploratory Data Analysis of DOGE [1]

	Open	High	Low	Close	Adj Close	Volume
Count	834	834	834	834	834	834
Mean	0.1283	0.1359	0.1209	0.1284	0.1284	2.1772e+09
Std	0.1149	0.1251	0.1049	0.1149	0.1149	4.9740e+09
Min	0.0025	0.0025	0.0025	0.002	0.0025	2.2755e+07
Max	0.6878	0.7376	0.6082	0.6848	0.6848	6.9411e+10

[1] The data was gathered between 05-08-2020 and 16-11-2022

Table 5: Exploratory Data Analysis of SHIBA [1]

	Open	High	Low	Close	Adj Close	Volume
Count	638	638	638	638	638	638
Mean	0.16e-04	0.17e-04	0.15e-04	0.16e-04	0.16e-04	1.28e+09
Std	0.14e-04	0.15e-04	0.12e-04	0.14e-04	0.14e-04	2.90e+09
Min	0.00e-04	0.00e-04	0.00e-04	0.00e-04	0.00e-04	3.17e+04
Max	0.79e-04	0.88e-04	0.67e-04	0.80e-04	0.80e-04	3.91e+10

[1] The data was gathered between 17-02-2020 and 16-11-2022

Table 6: Exploratory Data Analysis of MONA [1]

	Open	High	Low	Close	Adj Close	Volume
Count	834	834	834	834	834	834
Mean	1.3043	1.3484	1.2561	1.3024	1.3024	4.8779e+06
Std	0.6114	0.6539	0.5680	0.6119	0.6119	1.1133e+07
Min	0.3615	0.3700	0.3538	0.3593	0.3593	2.3089e+04
Max	4.0293	4.3577	3.1491	4.0293	4.0293	1.5883e+08

[1] The data was gathered between 05-08-2020 and 16-11-2022

graphicx stfloats [caption=false,font=footnotesize]subfig

Figure 5: Diagram of the Experiment Process

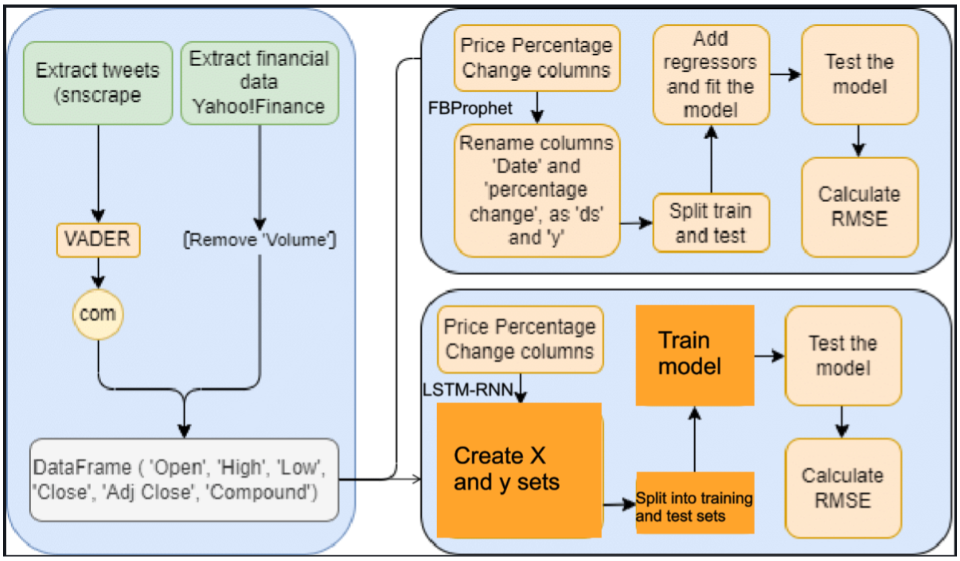


Figure 4: Closing prices of DOGE, SHIB, and MONA

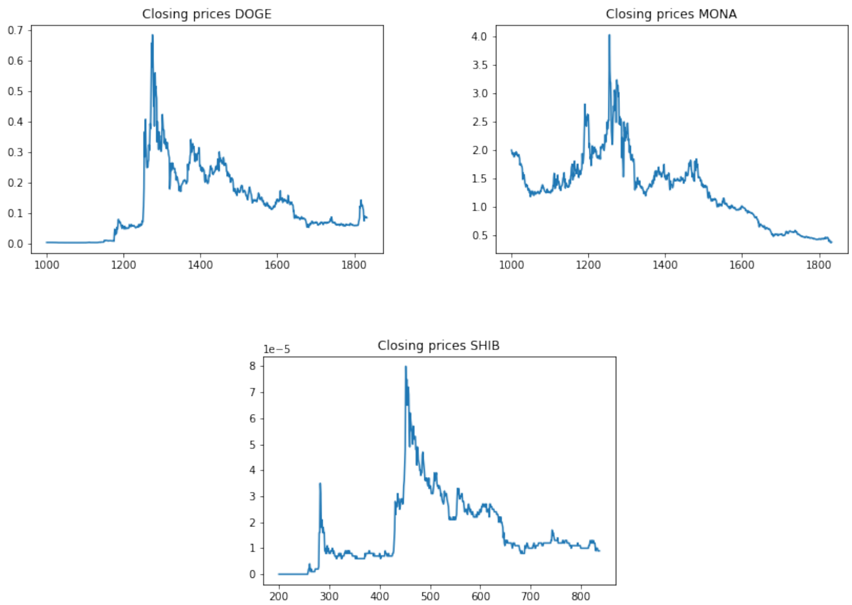


Table 7: Results of the LSTM-RNN models

DOGE	model 1		model 7		model 14		model 30	
	with	without	with	without	with	without	with	without
1 day	0.0335	0.0194	0.0355	0.0103	0.1336	0.1748	0.5535	0.5789
7 day	**0.0005**	0.0026	0.0102	0.0387	0.0121	0.9042	0.5269	0.6324
14 day	0.0014	0.0063	0.1874	0.0588	0.3031	0.0292	0.2003	0.0830
30 day	0.0162	0.0088	0.2757	0.1030	0.2579	0.1472	0.5828	0.9553
SHIB	model 1		model 7		model 14		model 30	
	with	without	with	without	with	without	with	without
1 day	0.0131	0.0101	0.4177	0.1248	0.5621	0.1325	0.0325	0.0534
7 day	0.0177	0.0100	0.0411	0.2559	0.3724	0.1547	0.1123	0.1873
14 day	0.0270	0.0196	0.1023	0.0777	0.6681	0.2958	0.1656	0.1146
30 day	**0.0006**	0.0122	0.1781	0.0559	0.0698	0.3491	0.2747	0.0058
MONA	model 1		model 7		model 14		model 30	
	with	without	with	without	with	without	with	without
1 day	0.0049	0.0055	0.0179	0.0005	0.0210	0.0537	0.0092	0.0926
7 day	0.0038	0.0030	0.0187	0.0047	0.0083	0.0135	0.0257	0.0616
14 day	0.0270	0.0196	0.0163	**0.0003**	0.0145	0.0210	0.0427	0.0489
30 day	0.0018	0.0013	0.0049	0.0314	0.0125	0.0044	0.0838	0.0952

Table 8: Results of the FBProphet models

DOGE	model 1		model 7		model 14		model 30	
	with	without	with	without	with	without	with	without
1 day	**0.0732**	**0.0732**	0.2584	0.2579	0.3904	0.3897	0.2559	0.2514
7 day	0.1542	0.1542	0.3015	0.2808	0.4517	0.4374	0.3288	0.2674
14 day	0.2391	0.2391	0.2127	0.1907	0.4720	0.4005	0.4825	0.4153
30 day	0.3365	0.3365	0.2893	0.3249	0.1852	0.2542	0.1270	0.2382
SHIB	model 1		model 7		model 14		model 30	
	with	without	with	without	with	without	with	without
1 day	0.0652	**0.0608**	0.1031	0.0985	0.1548	0.1482	0.3057	0.3023
7 day	0.2917	0.2246	0.3033	0.2633	0.2922	0.2853	0.3212	0.3051
14 day	0.6800	0.5444	0.7542	0.5889	0.8439	0.6360	0.6230	0.5854
30 day	0.7788	0.8561	0.8242	0.8342	0.8093	0.8257	0.6459	0.7671
MONA	model 1		model 7		model 14		model 30	
	with	without	with	without	with	without	with	without
1 day	**0.0218**	**0.0218**	0.0630	0.0625	0.0837	0.0827	0.1267	0.1273
7 day	0.0424	0.0433	0.0680	0.0685	0.0887	0.0887	0.1163	0.1165
14 day	0.0650	0.0602	0.0747	0.0718	0.0935	0.0906	0.1178	0.1180
30 day	0.1297	0.1308	0.1240	0.1250	0.1070	0.1092	0.0487	0.0513