Predicting the Fluctuations and Flow of Cryptocurrency Using ML Algorithms

Jeyashree kothai Alwan, Aryan Dwivedi,Raghav Malviya, Manojkumar Muthukumaran, Abhisek Mallick, Shafia Mahedi, Mohammed Ziyan Ahmed, Kamisetty Sai Smaran

INTRODUCTION

1.1 OBJECTIVE

The objectives of the proposed work is to identify the parameters such as high, open and close within the dataset for various types of cryptocurrencies and use it predict the cryptocurrency price through various deep learning algorithms of LSTM, Linear Regression, SGD Regression, Support Vector Regression, Random Forest Regression. Once the price prediction is achieved, comparative analysis can be performed between all the models to identify the most accurate model that best fits the practicality of using it within our application. This provides us the feasibility to understand the requirements of the best model and it‟s working for the of real-world cryptocurrency price prediction. We use MSE which helps to measure the differences between values predicted by a model and the values observed. The predicted output of the price can be evaluated within the bound of open and high values to get us a suitable prediction.

1.2 MOTIVATION

Lots of research has already been done for prediction analysis of markets like stock markets, gold market etc. So, the idea of time series prediction is a well- defined problem. Similarly, cryptocurrency can also be considered as a form of virtual currency with a purpose to serve as a medium of exchange. This allows it to come within the scope of other markets and thus be a time series prediction. Due to the current high volatility in the cryptocurrency market it provides opportunities of research on the prediction of cryptocurrency prices. Concurrently, cryptocurrencies like Bitcoin, Ethereum etc. that get increasingly getting adopted across the world. Cryptocurrencies tend to be of open nature where it is based on a peer to peer and decentralized system where all the transactions are transferred to a system called as blockchain. This is not known well in the traditional market which calls for means to develop research into the fields of cryptocurrency markets and use models of deep learning for their prediction of price and other parameters.

1.3 BACKGROUND

There has been a huge boost in popularity of cryptocurrencies due to several months of vast exponential growth of their market capitalization of around $300 Billion in 2020. This has led to cryptocurrency receiving a lot of attention from cryptocurrency enthusiasts and media. However, with the decentralization and open sourced nature of cryptocurrencies, it has reduced the level of central control over them which led to various other types of cryptocurrencies being created and introduced into the market. Correspondingly, this causes a wide range of differences in the cryptocurrency prices for between cryptocurrencies which indicate a important requirement for price prediction. This paper proposes the numerous ways to predict cryptocurrency price by inducing various factors such open price, high price and close price through deep learning techniques such as recurrent neural network (RNN) - long short-term memory (LSTM), Linear Regression, Stochastic Gradient Descent (SGD), Random Forest Regression, Support Vector Regression which are known effective machine learning models for training data, with LSTM(Long Short Term Memory) being recognized for longer term predictions. This proposed approach implements Python as the programming language and uses dataset that contains consolidated financial information for the top 10 cryptocurrencies sorted by Market Cap which also have various attributes for each type of cryptocurrency such as open, close, high, low, value and date. The cryptocurrencies used are; Tezos which is a technology that deploys a blockchain capable of modifying its set of rules with minimal disruption to its network through an on-chain governance model; Bitcoin which works without a central bank or administrator that can be sent from user to user on its peer-to-peer Bitcoin network; Ethereum works through on an operating system featuring smart contract functionality; XRP (Ripple)works as a fast, cost- efficient cryptocurrency develop for cross-border payments; Binance Coin (BNB) is the cryptocurrency of the Binance platform which is a global cryptocurrency exchange; EOS is designed to support large-scale applications; Tether is a cryptocurrency that has a value which matches the value of the U.S. dollar; Bitcoin Cash is cryptocurrency based on a fork of Bitcoin which is a spin-off that was created similar to Bitcoin SV which is a fork of Bitcoin Cash (BCH), it attempts to restore the original Bitcoin protocol as defined by version

0.1 of Bitcoin; Stellar cryptocurrency is used to fiat money transfers which allows for cross-border transactions; Litecoin cryptocurrency enables instant, near-zero cost payments; Cardano cryptocurrency is the first blockchain platform to evolve out of from a scientific philosophy and is based on a research driven approach.

The results that we accomplish from the project help us understand and examine the various types of model applicability of accurate prediction of cryptocurrency prices further facilitating the performances of the proposed deep learning prediction models.

2. PROJECT DESCRIPTION AND GOALS

2.1 LITERATURE SURVEY

Gorse and Phillips [1] worked on the prediction of cryptocurrency prices bubbles by using social media data along with epidemic modeling. Through using Hidden Markov model (HMM) to detect their bubble-like characteristics in the time series they concluded that the social media data plays an important role in the prediction of cryptocurrency prices.

Zheshi Chen, Wenjun Sun and Chunhong Li [2] proposed to predict Bitcoin prices by classifying by their respective everyday price and high frequency based prices. They use a set of high dimension features like that of property, network, attention, gold, trading and market spot price for Bitcoin price prediction, whilst they have used the trading features obtained from a cryptocurrency exchange based on a five minute interval-time price prediction. They have implemented statistical methods such as Logistic Regression and Linear Discriminant Analysis for Bitcoin daily price prediction along with the high-dimensional features which made them achieve an accuracy of 66% that has outperformed complicated machine learning algorithms on their testing. When they compared it with their benchmark results for daily price prediction, they seem to achieve a better performance of highest accuracies that are based on statistical methods and machine learning algorithms of 66% and 65.3% respectively. They have implemented machine learning models such as Random Forest, XGBoost, Quadratic Discriminant Analysis, and Support Vector Machine. They have used Long Short-term Memory model for Bitcoin five-minute interval price predictions that has proved to be superior to statistical methods with an accuracy reaching 67.2%.

Sneha Gullapalli [3] proposed the perform prediction based on the trained temporal neural networks such as time-delay neural networks and recurrent neural networks on prices of Bitcoin over few years. Parameters such as the opening price, highest price, lowest price, closing price and volume of its currency were taken into consideration so as to predict the highest and closing price of the next day. They have designed and implemented TDNNs and RNNs using NeuroSolutions‟s Artificial Neural Network development environment which was used to help build predictive models and was then evaluated them by computing various measures such as the Mean Square Error, Normalized Mean

Square Error and Pearson‟s Correlation Coefficient for the training data from each time series and was held out for validation.

Scheuermann and Tschorsch [4] carried out a technical survey based on decentralized cryptocurrencies. Their research was based on the building blocks and protocols that correspond to the cryptocurrency known as Bitcoin; this highlights the characteristics of centralized cryptocurrencies, and various other findings. They have given in-depth insights of Bitcoin cryptocurrency. Their detailed work can be linked to understand various cryptocurrencies.

Mukhopadhyay [5] proposed a study based on cryptocurrency systems. The study consisted of several characteristics of cryptocurrencies, that comprises proof of stake, the proof of pork, and their combination that was used in data mining techniques. They elaborated on how the proof of stake is not self- dependent, while the proof of work is resource dependent. Hence, the combination of this can result in precise results.

Bruno Spilak [6] proposed the usage of Multilayer Perceptron, Recurrent Neural Network and Long Short-Term Memory models to predict the output of priced directions of cryptocurrencies that uses rolling window regression method. They constructed a classification problem that predicts if the price of the cryptocurrency will either increase or decrease based on a three-month trading strategy which will then compared to its performance with a passive index investment that relies on cryptocurrency market which follows CRIX.

2.2 GAPS IDENTIFIED IN THE SURVEY

The gaps that have been identified after reviewing the aforementioned research/work is that there was either one or two types of cryptocurrency being used for their price prediction. This limits the model‟s efficiency and accuracy for cryptocurrency price prediction to the unique parameters of that specific cryptocurrency type. So, in our project, we propose to find common parameters (open, high and close values) for ten different types of cryptocurrency train the models to these parameters and then predict their price. This facilitates to produce the efficiency of the deep learning and machine learning models by performing analysis such as using MSE on the price predictions by the various models implemented.

2.3 OVERVIEW OF PROPOSED SYSTEM / RELATED CONCEPTS

Python is a commonly used high level programming language. Python uses an open source object-oriented, interpreted, and interactive programming language. This language combines high performance with clear syntax which consists of modules, classes, exceptions, high level dynamic data types and dynamic typing. Python is used as an extension language for various applications that are written in several languages which require easy scripting and/or automation interfaces for the application.

Python is used for implementing Machine Learning algorithms. Compared to a lot of other high level languages the syntax is simple because of code libraries. Python is quickly changing into the go to language of Machine Learning. It is used to create models for Bayesian networks, decision trees etc. Prescriptive analytics otherwise known as decision science provides information about what anticipates what, when, and why specific outcomes will happen. Decision scientists frame their analysis of data around various business problems and implement many of the similar techniques and tools as data scientists. The goal for them is to make these insights usable, so that their models and visualization methods can be built to communicate on those insights. Therefore, Python is commonly used to create prescriptive analytics tools like deep learning, which uses artificial neural networks (ANN) to optimize outcomes.

Cryptocurrency is a virtual currency created to serve for online monetary needs. Crypto is prefix that originates from the method of using cryptography methodologies to verify and secure transactions which help produce new cryptocurrency units. The principle of cryptography is of that it can make information decipherable with a key than without one. Blockchain is composed of transactions that are made by cryptocurrencies on a peer to peer network. This consists of many different nodes which ensure that counterfeiting does not take place. Transactions between accounts and cryptocurrency wallets can be identified easily, thus ensuring security. There are a lot of advantages for cryptocurrencies due to its control over transactions and inflation. Investors can be seen using cryptocurrencies as assets for their portfolios. Cryptocurrencies are placed in the market of non-correlation that creates it as a potential hedge against risk for example, valuable metals such as platinum, diamond etc.

Django is used as a web based framework that is used for rapid deployment and development. It"s free and open source and Django was designed to assist developers to take applications from ideas to execution quickly. Django is also highly secure and helps developers to avoid common security risks. Popular websites around the world use Django for their platform.

2.4 FRAMEWORK AND MODULES OF PROPOSED SYSTEM

a. **Linear Regression:** The Linear Regression models are used to predict the relationship between two variables. The variable that is being predicted is called the as dependent variable. The variables that are used to predict the value of the dependent variable are called the independent variables. In Linear Regression, each observation consists of two values. One value is for the dependent variable and one value is for the independent variable. In this simple model, a straight line approximates the relationship between the dependent variable and the independent variable. We have used the sklearn.linear_model to import Linear Regression algorithm and implement it for our application. For our approach, we fit the Linear Regression algorithm with the scaled trained X and Y data and predict the value for the X test scaled data. Using the inverse transform function, we transform the predicted value and the Y test scaled values. We generate a loss function for Linear Regression using the Mean Squared Error for the predicted value and the Y test scaled value and save the generated model through pickle. Figure 3 shows the mathematical model of Linear Regression.

b. **Stochastic Gradient Descent (SGD) Regression:** Stochastic Gradient Descent (SGD) is an easy and methodical way of studying linear classifiers under convex loss functions like the Support Vector Machines and Logistic Regression. The SGDRegressor integrates a stochastic slope learning routine that supports different loss functions and setbacks in an appropriate linear regression model. SGDRegressor is accustomed in such a way that it has many training samples for regression problems. We use sklearn.linear_model to import SGDRegressor algorithm and implement it for our application. For our approach, we fit the SGD algorithm with the scaled trained X and Y data and predict the value for the X test scaled data. Using the inverse transform function, we transform the predicted value and the Y test scaled values. We generate a loss function for SGD Regression using the Mean Squared Error for the predicted value and the Y test scaled value and save the generated model through pickle. Figure 4 shows the mathematical model of SGD Regression

c. **Random Forest Regression:** A Random Forest is a method which has the capacity of executing both regression and classification problems by the use of several decision trees. It has an ability called Bootstrap Aggregation, which is also recognized as Bagging. Bagging incorporates numerous decision trees for deciding a definitive final output which relies on singular decision trees. We use sklearn.ensemble to import Random Forest Regressor algorithm for our application. For our approach, we fit the Random Forest Regression algorithm with the scaled trained X and Y data and predict the value for the X test scaled data. Using the inverse transform function, we transform the predicted value and the Y test scaled values. We generate a loss function for Random Forest Regression using the Mean Squared Error for the predicted value and the Y test scaled value and save the generated model through pickle. Figure 5 shows the mathematical model of Random Forest Regression.

d. **Support Vector Regression:** This is a type of support vector machine which assists in linear and non-linear regression. SVR requires the training data"s Xand Y values where in it generates a correlation matrix using the X and Y values and then trains the model. We get a contraction coefficient which gives us the predicted value. For our approach, we fit the Support Vector Regression algorithm with the scaled trained X and Y data and predict the value for the X test scaled data. Using the inverse transform function, we transform the predicted value and the Y test scaled values. We generate a loss function for Support Vector Regression using the Mean Squared Error for the predicted value and the Y test scaled value and save the generated model through pickle. The figure no.6 shows the mathematical model of Support Vector Regression.

e. **Long Short-Term Memory:** Long Short-Term Memory is a division of a recurrent neural network (RNN). In RNN, the resultant product of the last step is used as the input in the present step. This is used to solve the problem of long-term dependencies of its usage. As time increases, RNN does not provide systematic performance however LSTM can by default preserve the desired information for a long time. It is used for processing, estimating and organizing by arranging it on the basis of time series data. We use tensorflow.keras.layers to import LSTM, Dense algorithms and tensorflow.keras.models to import Sequential, load_model, model_from_json from where our approach consists of two layers of LSTM input layers for representing an elliptical function for influencing the flow of information and memorization of patterns created inside a cryptocurrency data. The adam optimizer is used to repeatedly upgrade the network weights for the purpose of training it. The dense layer is used for making the model more accurate. This is done by using one LSTM layer and one dense layer. LSTM layers give their complete output sequences back, and the input sequence is converted into a single vector by the dense layer. We generate a loss function for LSTM using the Mean Squared Error for the predicted value and the Y test scaled value and save the generated model in .json and .h5 file formats. Figure 7 shows the mathematical model of LSTM.

f. **Dataset:** For our dataset, we use the consolidated financial information for the top ten cryptocurrencies that was pulled from the site www.CoinMarketCap.com.[7] Their attributes include: Currency name (e.g. Bitcoin), Date, Open, High, Low, Close, Volume, Market Cap. The descriptions of the attributes are:

g. **Parameter of the dataset:**

a. **Currency:** Name of cryptocurrency.

b. **Date:** Date refers to the calendar date for the particular row i.e. 24 hours from midnight to midnight.

c. **Open:** Open is what the price was at the beginning of the day.

d. **High:** Highest recorded trading price of the day.

e. **Low:** Lowest recorded trading price of the day.

f. **Close:** Close is what the price was at the end of the day.

g. **Volume:** Volume represents the monetary value of the currency traded in a 24 hours period, denoted in USD.

h. **Market Cap:** Market cap is circulating supply „x" price of the coin.

	Currency	Date	Open	High	Low	Close	Volume	Market Cap
count	28944	28944	28944	28944	28944	28944	28944	28944
unique	12	2412	12307	12057	12803	12294	16349	16058
top	stellar	Feb 26, 2015	1	1	1	1	0	4,51,600
freq	2412	14	1725	1511	1367	1729	2916	394

Figure 1: Attribute description for the cryptocurrency dataset

The attribute description gives the count of the various parameters used, unique gives the unique values of each parameter, top gives us the highest market cap and freq gives the frequency of the parameter values.

3. TECHNICAL SPECIFICATION

The proposed system model can be understood through the following phases:

a. **Data Analysis Phase:** This step examines the data and its parameters wherein it checks for any redundancy in the values of the data which might affect the predicted results. In case the dataset has any insignificant parameters, then the values of the data are erased. We improve the model predictability by merging the data.

b. **Data Filtration Phase:** The data is filtered so that it can get rid of all unimportant values.

c. **Train Test Split Phase:** In this phase the data is divided into subsets of training and testing. The division of these training and testing subsets are in the ratio of 70% and 30% respectively.

d. **Data Scaling Phase:** The data is scaled accordingly based on the parameters needed in the model. The data scaling makes it compatible for its use in the model.

e. **Model Building Phase:** We have implemented this paper using Python and its libraries such as sklearn, keras that contain the required models in Python that can be imported directly such as the sklearn.linear_model package that gives Linear and SGD regression, sklearn.ensemble gives the Random Forest Regression, sklearn.svm gives Support Vector Regression and keras gives LSTM. We cannot we use these models directly to build a LSTM model. Hence we have used Keras which implements tensorflow as a backend library that makes the model accurate. The model of Keras is composed of LSTM layer and dense layer. The data is processed in these layers to identify and form patterns present in the dataset so that the model becomes more accurate. The model then takes in this data for training.

f. **Model Learning and Evaluation Phase:** In this phase, the algorithm is fitted with the X and Y trained scaled data that is open, high and close values and it is formatted to get a better output.

g. **Prediction Phase:** In this phase we implement the prediction based on the model that has been generated in which the input values are fed into the model to give the cryptocurrency price prediction. The accuracy and losses are calculated by comparing the test data to the resultant output and losses through the MSE function.

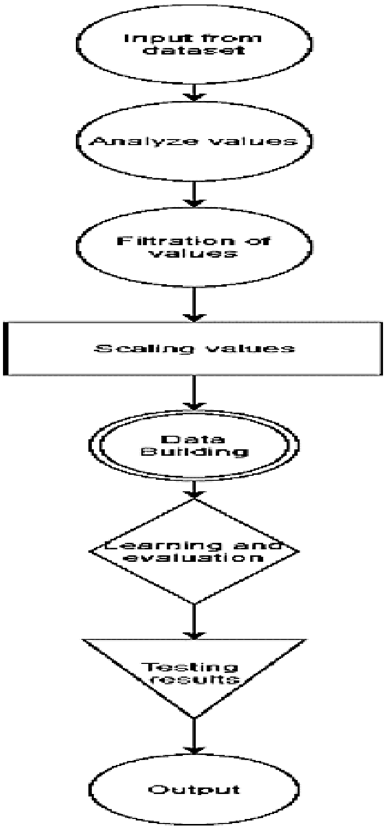


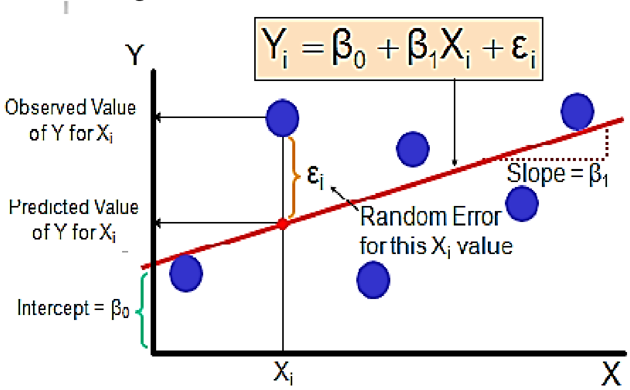
Figure 2: Workflow of proposed approach

3.1 MATHEMATICAL MODELS

This is used to elaborate a system"s mathematical model by its formulation and theories. This method of creating a mathematical model is called as mathematical modeling. The following are the mathematical models of the algorithms that have been discussed in the earlier section.

a. Linear Regression Mathematical Model

Figure 3: Mathematical model for Linear Regression [12]



X and Y are the variables that are used in Linear Regression. Linear Regression"s equation elaborates about the relation between X and Y and is called as the regression model. It is represented by the equation y = β0 +β1x+ε, where, the y-intercept is known by β0, the slope is known as β1 and the mean is known as E(y). The Linear Regression models represented „ε". β0 and β1 are used to represent the parameters for the population studies.

b. SGD Regression Mathematical Model

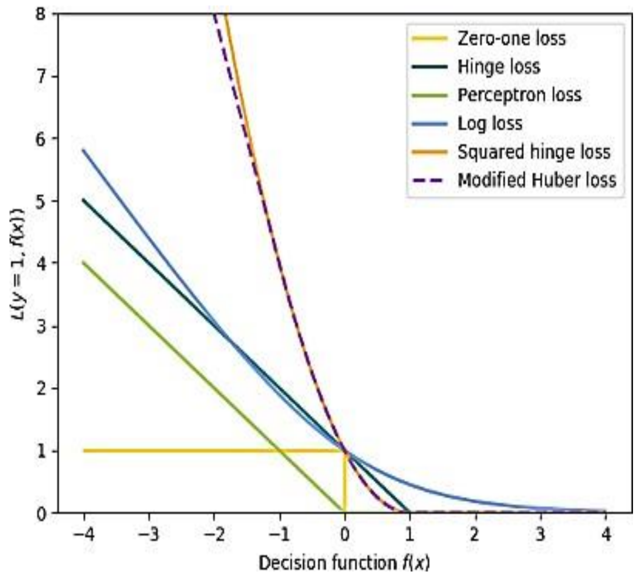


Figure 4: Mathematical model for SGD Regression [13]

The SGD Regression is based on various gradient descent functions as shown in the figure above.

c. Random Forest Regression Mathematical Model

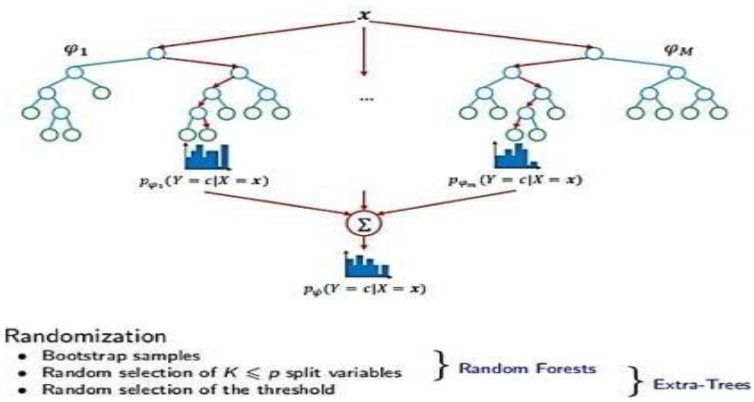


Figure 5: Mathematical model for Random Forest Regression[14]

Random Forest Regression works by selecting random K data points(K) from the training data subset which is then used to build the decision tree based on these data points, we then choose „N" number of trees that we require to build and the previous steps are repeated. When there is a newly created data point we create each one of the N data trees which are used to predict the value of Y for the given data point. The new data point is then assigned to represent as an average for all the Y predicted values.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 08 | August - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

d. Support Vector Regression Mathematical Model

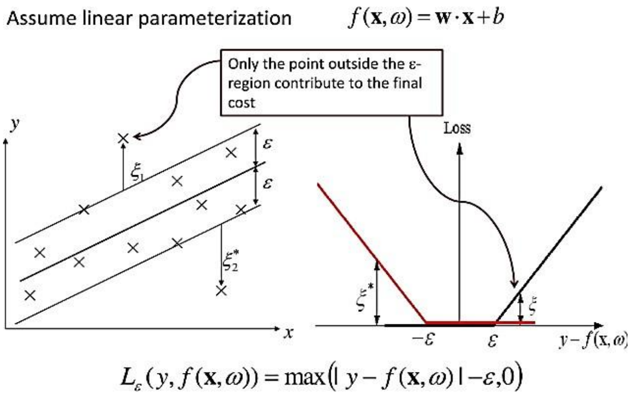


Figure 6: Mathematical model for Support Vector Regression [15]

The process is to collect a training c set of X and Y. We then choose a kernel, parameter and the required regularization if needed. We form the correlation matrix and train the machine to obtain a contraction coefficient by using the algorithm. We use this coefficient to create an estimator. The aim of SVR is to ensure that the errors obtained do not cross the threshold value.

e. LSTM Mathematical Model

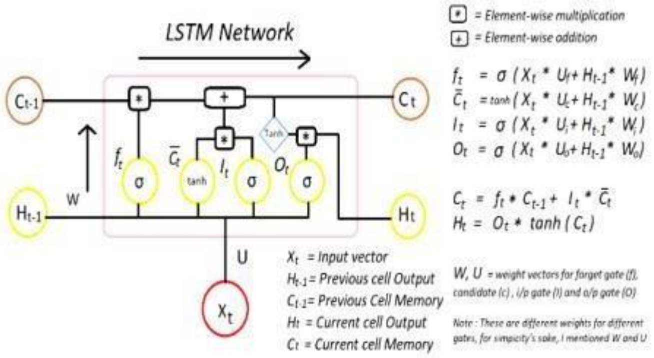


Figure 7: Mathematical model for LSTM [16]

LSTM is composed of a chain like structure which comprises of four neural networks and memory blocks known as cells. Gates perform memory manipulations. In these gates, there is a Forget Gate in which the information that is not required is removed using the forget gate. In the input gate addition of required information is performed. In the output gate where extraction of useful information from the current cell state is passed as the output for the output gate.

f. Mean Squared Error (MSE) Mathematical Model

MSE is a mathematical model which is the average squared error that occurs in a loss function in minimal squares regression. It is a combination of all data points such that it is also the square of the difference between the variables of predicted and target values. The data points are split and used. It is through the MSE that we can determine the loss of the model.

$$\sum_{i=1}^{n} \frac{\left(w^{T}x(i) - y(i)\right)^{2}}{n}$$

Figure 8: MSE Formula [17]

g. Pearson's Correlation Coefficient Mathematical Model

The linear correlation between the two variables X and Y is given by the Pearson‶s Correlation Coefficient.

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$

Figure 9: Pearson‶s Correlation Coefficient Formula [18]

4. DESIGN APPROACH

In this section, we present the proposed approach in predicting cryptocurrency price by the means of deep learning and machine learning models that were discussed above. We use the programming language of Python as the backend and Django for the front end which uses HTML, CSS and JS for the user‶s interaction. The workflow of the proposed approach is shown in figure 3. The algorithms that are used for our system are Linear Regression, Stochastic Gradient Descent (SGD)

Regression, Random Forest Regression, Support Vector Regression and Long Short-Term Memory (LSTM) where the regression algorithms are imported from the sklearn package. We also find correlation between the deciding parameters of the price prediction i.e. open, high and close in determining the relationship of the numerous characteristic features in the data set. This gives us an insight to the required parameters that help us obtain the values for the required price prediction.

In the below Use Case Diagram, the User launches the Django Server or command prompt window to run the interface of the application. The user is it to then input the cryptocurrency name, date, open

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 08 | August - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

value, high value and should then chose the preferred algorithm. Taking in all these input parameters the system computes the given data to the model and presents the predicted value for the given cryptocurrency.

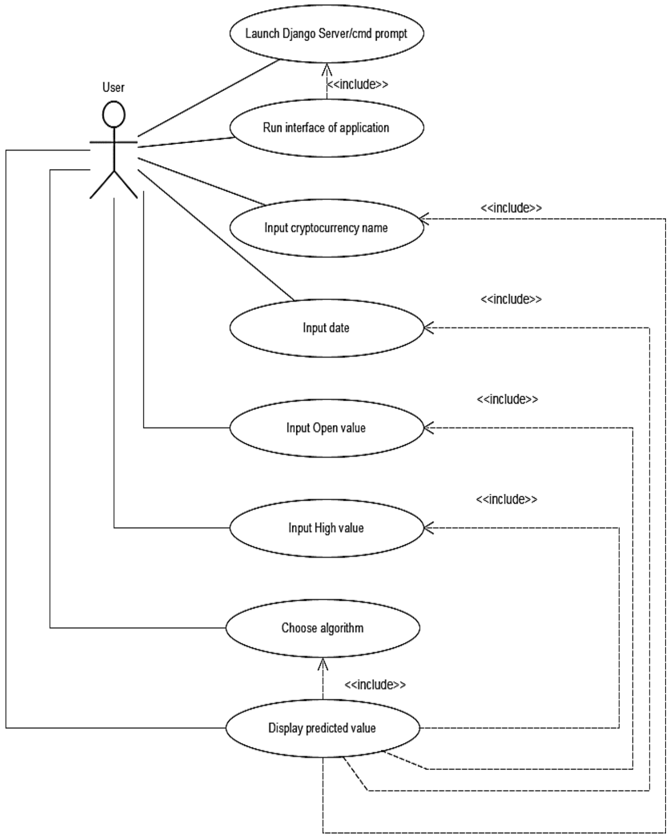


Figure 10: Use Case Diagram for proposed system

4.1 System Requirements

a. H/W Requirements: The proposed approach is tested on the following hardware specifications.

- Processor and processor speed: Intel Core i5 8300H processor, up to 4.0 GHz.
- RAM: 16 GB, 2667 Mhz.
- Graphics Card: NVIDIA GeForce 1060M, 6GB.
- Hard Disk: 1TB

b. **S/W Requirements:** The proposed approach is tested on the following software specifications.

- IDE platform: Visual Studio 2019, Jupyter Notebook.
- Programming language: Python (with Django), HTML, CSS, JS.
- Operating System: Windows 10 Single Language 64 Bit.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 08 | August - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

5. PROJECT DEMONSTRATION

We demonstrate our proposed work using a web browser which is Google Chrome (in this case) to implement the Django framework. We launch a localhost server from Microsoft Visual Studio from which we enter the interface and use our application. The cryptocurrency dataset is loaded from the code.

Figure 11: User interface through Django

In figure 11, we see the user‟s interface that has been developed through Django from which they can enter the input values of currency name, date, open and high price and the preferred algorithm.

Figure 12: Predicted cryptocurrency price

In figure 12, the output from the workings of figure 12 is displayed for a selected algorithm.

Figure 13: Clipping for out of range predicted cryptocurrency price

In figure 13, the output of the predicted cryptocurrency price is clipped using a range of high value+10 and open value-11 from the user‟s input. This ensures uniformity in price prediction. We derive these ranges from the mean difference of all the open prices and mean difference of all the high prices in the dataset.

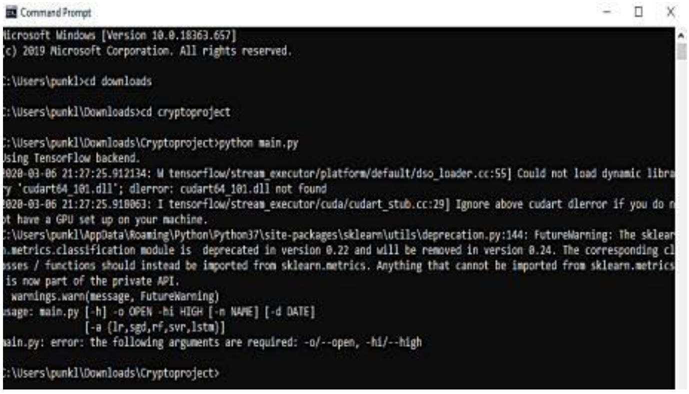


Figure 14: Execution through command line prompt

In figure 14, we have additionally added the feature of predicting the cryptocurrency price directly through the command line prompt which doesn‟t need the use of a web browser. Furthermore, since we can‟t implement .sav files for LSTM which makes it compatible for Django, we should use the command line prompt for predicting using LSTM as show in figure 15.

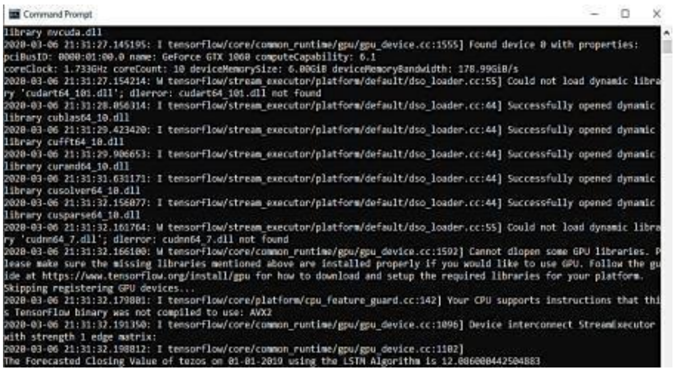


Figure 15: Predicting cryptocurrency price using LSTM

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 08 | August - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

6. RESULTS AND DISCUSSION

In our results, the model that we have implemented using our datasets has given us our predicted cryptocurrency prices. The Pearson correlation coefficient was used to find the features from the dataset where correlations were calculated between Open - Low, Open - Low, Open - High values that show the impact as the deciding factors of our input and output values. We also find the correlation between all the parameters in the dataset as seen in figure 16.

	Open	High	Low	Close	Volume	Market Cap
Open	1.000000	0.999268	0.998868	0.998551	0.560012	0.953655
High	0.999268	1.000000	0.998588	0.999403	0.561073	0.954372
Low	0.998868	0.998588	1.000000	0.999205	0.559649	0.954388
Close	0.998551	0.999403	0.999205	1.000000	0.560458	0.955007
Volume	0.560012	0.561073	0.559649	0.560458	1.000000	0.591818
Market Cap	0.953655	0.954372	0.954388	0.955007	0.591818	1.000000

Figure 16: Attribute correlation for the cryptocurrency dataset

In figure 16, the attribute correlation gives us the amount of linear correlation between two types of parameters, X and Y. Here, the higher the correlation coefficient, the greater is their linear correlation.

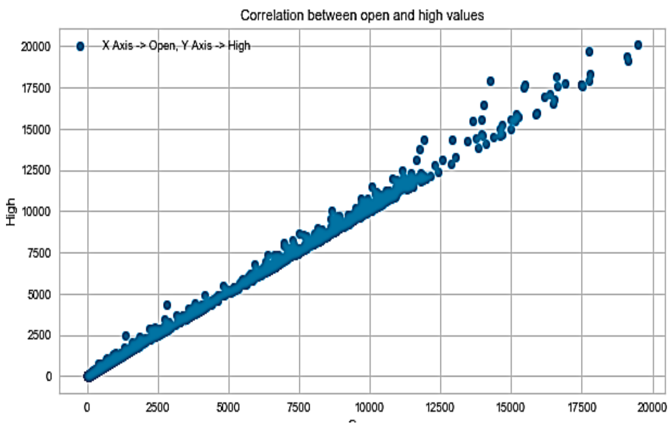


Figure 17: Correlation between open and high values

In figure 17, the correlations between Open and High values are observed to be highly correlated with noticeable dispersion after the 12500 and 12500 of Open and High values respectively. The Pearson correlation coefficient was found to be 0.999268.

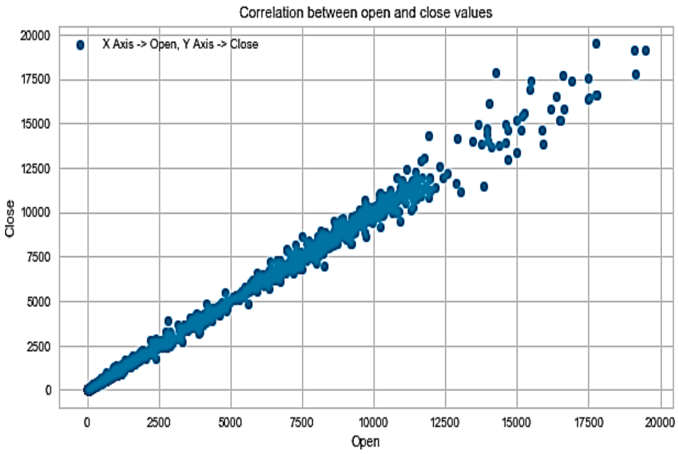


Figure 18: Correlation between open and close values

In figure 18, the correlations between Open and Close values are observed to be highly correlated with high dispersion after the 15000 and 15000 of Open and Close values respectively. The Pearson correlation coefficient was found to be 0.998551.

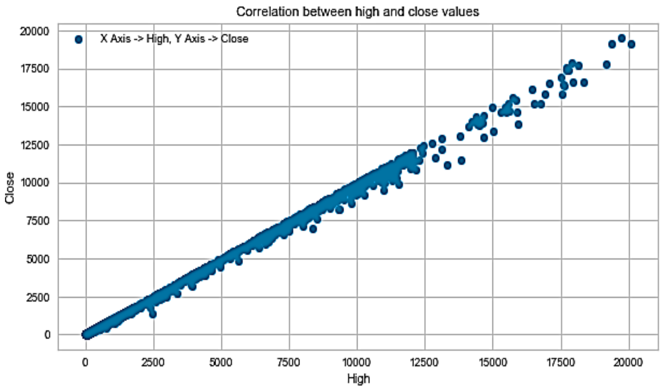


Figure 19: Correlation between high and close values

In figure 19, the correlations between High and Close values are observed to be highly correlated with high dispersion after the 12500 and 12500 of High and Close values respectively. The Pearson correlation coefficient was found to be 0.999403.

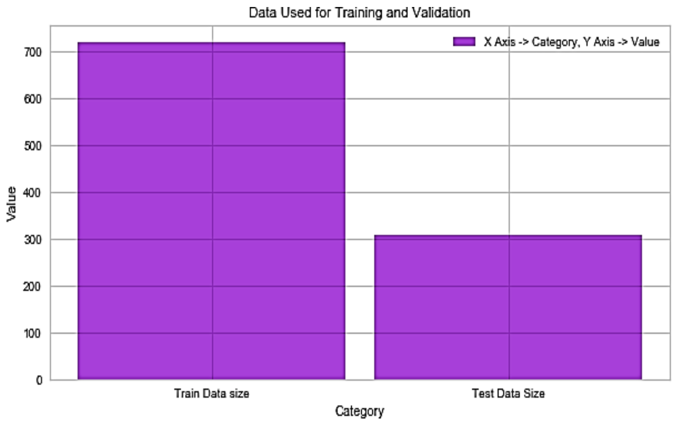


Figure 20: Train and test data sizes

In figure 20, for our implementation, we have split the data for our training and testing to the ratio of 70% and 30% respectively. We"ve used the random state of 1 in which an internal number has been randomly generated using the attribute of random state which decides how the training data and test data gets split.

Table 1: Actual vs predicted values of Linear Regression

Actual value	Predicted Value
1272.830	1231.021
2464.580	2446.526
7466.860	7233.420
5337.890	5168.912
703.700	681.736
3911.480	3799.789
517.790	510.210
222.880	221.410
77.530	76.062
12.750	14.027

In table 1, we have compared the actual values to the predicted values based on the obtained predictions by the Linear Regression method.

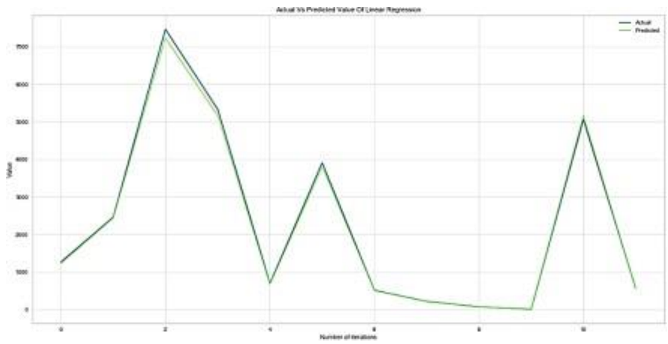


Figure 21: Actual vs predicted values of Linear Regression

In the figure 21, using the help of table 1 we have plotted a graph where we use the blue line to depict the actual value and green line to depict the predicted values where it can be seen that the graph is uniform this giving us nearly approximate predicted closing values and can be far off from the actual value if the values given are higher. Mean Squared Error loss of Linear Regression was found to be 0.00125.

Table 2: Actual vs predicted values of SGD Regression

Actual value	Predicted Value
1272.830	1245.713
2464.580	2465.031
279.560	281.801
7466.860	7280.149
124.340	124.163
737.330	733.633
5337.890	5218.530
703.700	677.676
3911.480	3838.615
222.880	225.163

In table 2, we have compared the actual values to the predicted values based on the obtained predictions by the SGD Regression method.

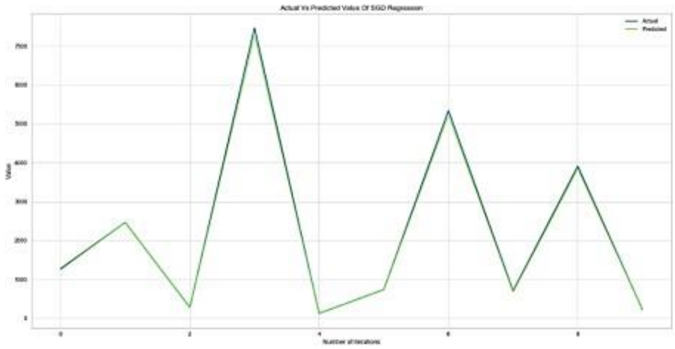


Figure 22: Actual vs predicted values of SGD Regression

In the figure 22, using the help of table 2 we have plotted a graph where we use the blue line to depict the actual value and green line to depict the predicted values where it can be seen that the graph is uniform this giving us nearly approximate predicted closing. Mean Squared Error loss of SGD Regression was found to be 0.00148.

Table 3: Actual vs predicted values of Random Forest Regression

Actual value	Predicted Value
1272.830	1240.018
2464.580	2465.188
279.560	280.378
7466.860	7337.790
92.570	81.804
124.340	121.482
737.330	736.832
5337.890	5349.382
703.700	686.999
3911.480	3828.997

In table 3, we have compared the actual values to the predicted values based on the obtained predictions by the Random Forest Regression method.

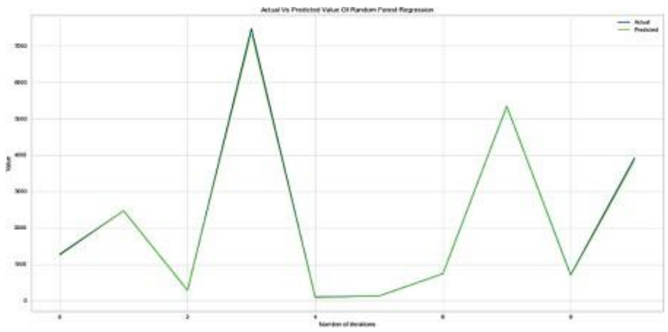


Figure 23: Actual vs predicted values of Random Forest Regression

In the figure 23 using the help of table 3 we have plotted a graph where we use the blue line to depict the actual value and green line to depict the predicted values where it can be seen that the graph is uniform this giving us nearly approximate predicted closing. Mean Squared Error loss of Random Forest Regression was found to be 0.00139.

Table 4: Actual vs predicted values of Support Vector Regression

Actual value	Predicted Value
1272.830	1218.887
0.005	-24.238
8.710	-14.487
2464.580	2446.319
279.560	255.301
2.730	-21.529
7466.860	7281.328
0.002	-24.241
5337.890	5197.377
222.880	198.695

In table 4, we have compared the actual values to the predicted values based on the obtained predictions by the Support Vector Regression method.

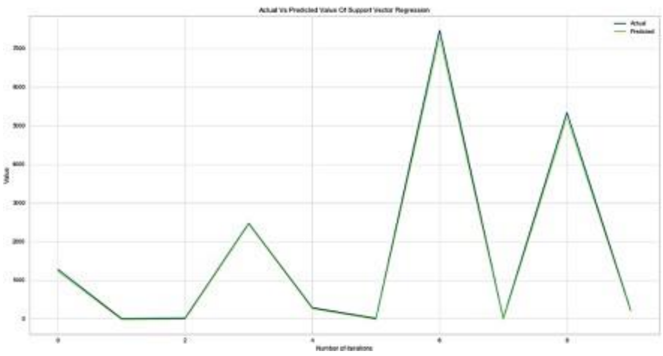


Figure 24: Actual vs predicted values of Support Vector Regression

In the figure 24, using the help of table 4 we have plotted a graph where we use the blue line to depict the actual value and green line to depict the predicted values where it can be seen that the graph is uniform this giving us nearly approximate predicted closing. Mean Squared Error loss of Support Vector Regression was found to be 0.00160.

Table 5: Actual vs predicted values of LSTM

Actual value	Predicted Value
1272.83	1265.203
7466.860	7327.019
124.340	121.496
737.330	733.369
5337.890	5183.345
3911.480	3823.547
517.790	512.080
222.880	220.139
37.750	45.973
77.530	75.984

In table 5, we have compared the actual values to the predicted values based on the obtained predictions by the LSTM method.

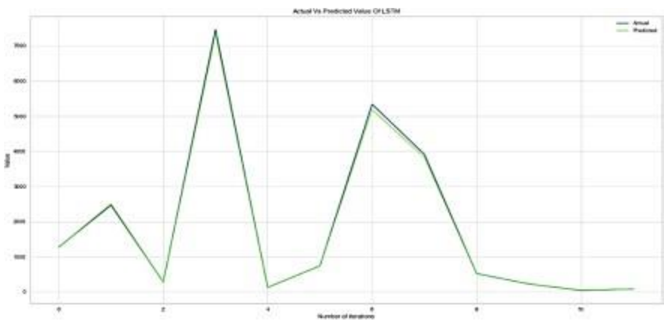


Figure 25: Actual vs predicted values of LSTM

In the figure 25, using the help of table 5 we have plotted a graph where we use the blue line to depict the actual value and green line to depict the predicted values where it can be seen that the graph is uniform this giving us nearly approximate predicted closing values and can be far slightly off from the actual value due to the sudden change in the values which can happen randomly based on how LSTM works. Mean Squared Error loss of LSTM was found to be 0.00111, which makes it the lowest compared to all the other models.

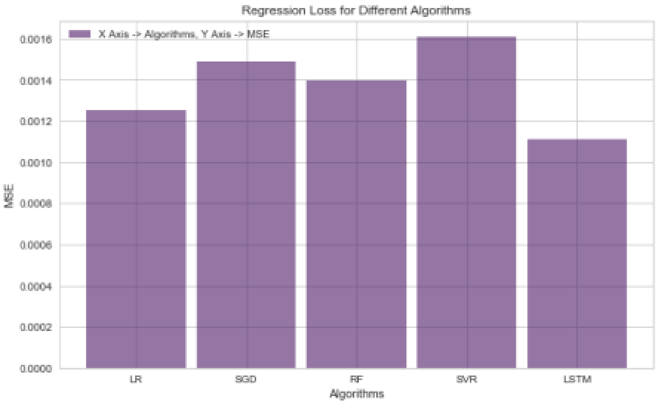


Figure 26: Regression loss for all the implemented algorithms with respect to MSE

Therefore, in comparison to all the models the figure 26 shows the difference between all the implemented models. As shown, LSTM proves to have the lowest MSE making it the most accurate.

7. SUMMARY

We can summarise our project by concluding that cryptocurrencies has weakened the central control over them by the years. Alongside with this, there have been variations and drastic changes in their prices which require an important need to predict cryptocurrency prices. This paper proposes several methods which are used to predict cryptocurrency prices by utilizing several parameters such as open, high and low values from the benchmark dataset. We have implemented several algorithms such as Linear Regression, SGD Regression, Support Vector Regression, Random Forest Regression and LSTM in Python. The results show the optimality and practicality of the proposed approach to reach the expected cryptocurrency price prediction as we train the models for several types of cryptocurrencies thus proving its applicability. Future research can extend to implementing a specific date from which the cryptocurrency price gets predicted.

8. REFERENCES

[1].Phillips R C , Gorse D, "Predicting cryptocurrency price bubbles using social media data and epidemic modeling", IEEE Symposium Series on Computational Intelligence (SSCI), (2017).

[2].Zheshi C, Chunhong L, Wenjun S, "Bitcoin price prediction using machine learning: An approach to sample dimension engineering", Journal of Computational and Applied Mathematics, (2019).

[3].Sneha Gullapalli, "Learning to predict cryptocurrency price using artificial neural network models of time series ",(2018), Retrieved from https://krex.k-state.edu/dspace/bitstream/handle/2097/38867/SnehaGullapalli2018.pdf.

[4].Tschorsch F , Scheuermann B, "Bitcoin and beyond: A technical survey on decentralized digital currencies", IEEE Communications Surveys & Tutorials, Vol.18, No.3, (2016).

[5].Mukhopadhyay U, Skjellum A, Hambolu O, Oakley J, Yu L & Brooks R, "A brief survey of cryptocurrency systems", IEEE 14th Annual Conference on Privacy, Security and Trust (PST), (2016).

[6].Bruno Spilak, "Deep neural networks for cryptocurrencies price prediction"(2018), Retrieved from https://d-nb.info/1185667245/34.

[7].Yecheng Y, Jungho Y, Shengjun Z, Yuwen L, Taekseung K, Guihongxuan Z, Leonard Y "Predictive Analysis of Cryptocurrency Price Using Deep Learning", International Journal of Engineering & Technology, 7 (3.27) (2018).

[8].https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term- memory/

[9].www.kaggle.com [https://www.kaggle.com/philmohun/cryptocurrency-financial- data].

[10]. Abhinav Sagar, "Cryptocurrency Price Prediction Using Deep Learning" [https://towardsdatascience.com/cryptocurrency-price-prediction-using-deep- learning-70cfca50dd3a].

[11]. Marco Santos" Predicting Bitcoin Prices with Deep Learning" [https://towardsdatascience.com/predicting-bitcoin-prices-with-deep-learning- 438bc3cf9a6f].

[12]. https://madhureshkumar.wordpress.com/2015/07/21/regression-analysis- linear-regression-sse-assumption-of-linear-regression-error-term-and-best-fit-line.

[13]. https://scikit-learn.org/stable/modules/sgd.html.

[14]. https://www.kdnuggets.com/2017/10/random-forests-explained.html.

[15]. https://stats.stackexchange.com/questions/13194/support-vector-machines- and-regression.

[16]. https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory.

[17]. https://www.oreilly.com/library/view/machine-learning-with/9781785889936/669125cc-ce5c-4507-a28e-065ebfda8f86.xhtml.

[18]. https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/correlation-coefficient-formula.