International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V70I3P231

Volume 70 Issue 3, 275-283, March 2022

Original Article

Predictive Analytics to Determine the Bitcoin Price Rise using Machine Learning Techniques

Ricardo Leon-Ayala[1], Noe Vicente-Rosas[2], Nayely Quispe-Quispe[3], Luis Mesa-Salinas[4], Jean Pier Saldaña-Bartra[5], Alexi Delgado[6], Enrique Lee Huamaní[7]

[1,2,3,4,5]*Faculty of Science and Engineering, Universidad de Ciencias y Humanidades, Lima-Perú*
[6]*Faculty, Department of Engineering Mining Engineering Section, Pontificia Universidad Católica del Perú, Lima-Perú*
[7]*Faculty, Image Processing Research Laboratory, Universidad de Ciencias y Humanidades, Lima-Perú*

[1]ricleona@uch.pe, [5]kdelgadov@pucp.edu.pe, [6]ehuamaniu@uch.edu.pe

Abstract **-** *As we know, today there have been several variations in the price of Bitcoin as it is not stable, that is to say, the price of Bitcoin rises and falls exaggeratedly as there are several factors that focus on the opening and closing price values of Bitcoin, also the highest and lowest Bitcoin price values reached in a day. Bitcoin price is determined by supply and demand. If user demand for this cryptocurrency is high, the price will increase and if it is very low, will go to. The price of Bitcoin has fluctuated considerably in recent years. This means that it is constantly changing and is in a stable range without major changes. In this paper, a machine learning method, also called Machine Learning, is proposed, which will allow us to automatically search and interpret relevant information from a large amount of data. Machine learning is the branch of artificial intelligence science that creates automated learning systems. In the case study, the Recurrent Neural Network (LSTM) algorithm will be used for the predictive analysis of Bitcoin value. Recurrent Neural Networks are network layers for analyzing time series and time-series data. An LSTM recurrent neural network aims to learn long-term dependencies; that is to say, learn the dependencies of future values of a sequence on previous values. The results of this research work showed that, by applying the Machine Learning technique and the LSTM algorithm, it was possible to predict the Bitcoin price increase. These results could benefit different companies or financial institutions in the investment of money with the help of Bitcoin. Companies such as Microsoft, Destinia, WordPress, among many others, already allow purchases with bitcoins, or other cryptocurrencies, on their websites.*

Keywords - *Bitcoin, Machine Learning, Neural Networks, prediction, Intelligence Artificial, Price.*

I. INTRODUCTION

Currently, Bitcoin has recorded several declines, as the price is not stable, that is to say. it goes up and down exaggeratedly. Several factors are affecting the price of Bitcoin. Some of these factors center on Bitcoin's opening and closing price values, also the highest and lowest Bitcoin price values reached in a day, which worries investors and entrepreneurs in various parts of the world [1]. The price of Bitcoin is related to the transaction data of the digital currency, that is to say, an increase in trading volume influences the price of Bitcoin. So, Bitcoin is partially used as a medium of exchange. This implies, that the main driver of Bitcoin's price is speculative demand and, therefore, should be considered an investable asset [2]. Bitcoin is a commodity similar to gold or silver which has a value tied to their supply and demand. They are perceived as assets that retain value over the long term, regardless of economic developments. [3].

The research work focuses on predictive analytics that will help anyone who wants to know Bitcoin prices., as well as micro and/or macro enterprises. Since is considered the most famous and in-demand in the world. For this reason, there are different encryption exchanges, that enable traders to work in online markets. As well as, various entities in charge of the cryptocurrency exchange such as BitMex, Huobi, Okex, and Derbit, all of which work with Bitcoin-centric projects and contracts [4].

The application of predictive analysis with Machine Learning seeks to solve the need to provide updated prices of Bitcoin values. As well as, to know the exact data of its valuation in the last months and years. There are several cases on the rise of the Bitcoin price, which is in great demand among investors worldwide. Artificial Intelligence, as one of the methodologies within the discipline of Machine Learning, uses algorithms to analyze data and discover patterns, interactions, or rules that can help predict future trends that can provide any type of competitive advantage and aid in better decision making. There are several techniques within this branch of Artificial Intelligence, such as Neural Networks, Support Vectors, Genetic Algorithms, Rule Induction Systems, Decision Trees, or Rough Set Theory. However, the Generalized Linear Models, introduced in the early 1970s, have become important tools for statistical analysis in this area. For this reason, we will use Recurrent Neural Networks (LSTM), which will help us predict the value of Bitcoin to make the best decisions regarding its purchase.

The objective of the research work is to perform predictive analysis to determine the rise in the value of Bitcoin using Machine Learning techniques, Therefore, our predictive analytics ensure successful results, and consequently report on the current Bitcoin revaluation.

The present research work is structured as follows: in section II we will describe in detail the Methodology, in section III we will show the results and discussions of the data obtained and finally, in section IV, we will present the conclusions.

II. METHODOLOGY

Machine learning is capable of automatically searching and interpreting relevant information from a large amount of data. [5].

Machine Learning is a method of data science that provides computers with the ability to understand and learn without the need to be programmed with instructions. Also, allows you to build algorithms capable of analyzing, learning, and making predictions. Unlike other algorithms, this one makes better use of large amounts of data sets and can keep learning to generate more experience [6].

To perform predictive analysis of the Bitcoin price increase, the data science methodology was applied, because the methodology presents several stages that will be of great help for the development of the research work. In the following, each stage will be detailed and what is proposed in each one of them.

A. Analytica Approach

This first stage of the methodology involves a detailed analysis of the different parts of the system, as well as the identification of the strategies to be followed to achieve the objective, in addition, the problem must be identified to choose an appropriate model to provide a solution to an objective.

B. Data Requirements

This second stage consists of identifying the data required to achieve the objective, this requires a broad knowledge and mastery of the subject, In this way, the bases of the research are strengthened to obtain the necessary data for the investigation.

C. Data collection

This third stage of the methodology consists of collecting as much data as possible. In other words, to start collecting information, the origin of the data, its source, and its format must be documented to obtain information with a high level of quality [7]. It should be emphasized that the data collection process is a step that cannot be automated, which is why a thorough investigation is recommended until the data required for the research are found, to ensure the success of the project. Finally, the procedure for data collection is shown below.

D. Data Understanding

This fourth stage of the methodology indicates the following, after completing the collection process, the data collected must be understood and analyzed section by section, understanding the data collected in the previous stage is paramount to ensure a good result, it also helps to formulate future decisions and strengthen the quality of research.

E. Data preparation

This fifth stage of the methodology points out the following, after understanding each section of data, the next step is the preparation of the data to avoid future complications at the time of generating some kind of tables or graphs [8]. At this stage, it is necessary to identify those conflicting data that may generate possible errors in the analysis, for example, duplicated data, data that may be used in the analysis, etc, null values, empty data, excessively large numerals, confusing items, etc. The above examples must be normalized to obtain data ready for the application of the predictive model

F. Model Preparation

This sixth stage of the methodology indicates the following, after performing the data preparation, the next step is to choose the algorithm that best suits the needs of the target, also the algorithm in question must be able to manipulate the data previously identified in the stage of "Data Collection", also at this stage the training of the model is performed through a process known as training and learning, which generates the segmentation and output data. [9] For a network to be free of our intervention in its understanding and learning process, a large amount of test data must be entered, so that the training process can identify the most appropriate steps and begin to interconnect the data [9]. By applying data training with Machine Learning, a more assertive prediction is achieved.

G. Model Evaluation

This last stage of the methodology is fundamental because in this stage the effectiveness of the chosen model is tested and analyzed, if the result satisfactorily analyzes the data entered it is a good sign, however, the algorithm must be tested by entering new data to see the behavior of the algorithm, according to the results obtained it can be determined if the algorithm was correct or not.

III. RESULTS AND DISCUSSIONS

A. Analytical Approach

At this stage, the analytical approach to be used to achieve the objective is determined, which is why the following question is posed: Which analytical approach is the most appropriate to achieve the predictive analysis of the rise in the price of Bitcoin? To solve the problem, the objective must be analyzed in detail, in this case, the objective is to predict the rise in the price of Bitcoin using Machine Learning techniques, which is why the predictive model or also known as predictive analysis, is the most appropriate for the development of the project because the model is based on obtaining data to generate possible future results for a given event. This type of analysis uses a variety of statistical modeling techniques, which investigate the present and transcendental facts to make a forecast. [10].

B. Data requirement

This stage details the content of the data, its format, and the source from which the DataFrame was extracted to vouch for the veracity of the data. For this reason, the following is detailed in the Data Requirement stage: The DataFrame used for the research work was made by data scientist and neuroscience specialist Mark Zielinski, who was in charge of collecting the data concerning the price of bitcoin. [11]. The DataFrame has stored data on transactions from January 2012 to March 2021 which is perfect because it has the necessary data to make an assertive prediction [11].

C. Data Collection

At this stage, the tools and language to be used are determined, as well as the DataFrame data, the tool to be used is Colab, which helped to make the predictive model and to program in the Python programming language. Before starting with the programming, it is recommended to install all the libraries that were required, that is why the code starts importing the library "pandas" because it helped to perform the reading of the DataFrame, the library "matplotlib. pyplot" library which facilitated the creation of graphs contained in a list of arrays, the "DateTime" library was used to manage the date variable, the "nunpy" library was used to analyze a large amount of data and finally "seaborn" was used to generate graphs with different styles, the import code is shown in Fig. 1.

```
# we import the libraries for the forecast
import pandas as pd
import matplotlib.pyplot as plt
import datetime
import numpy as np
import seaborn as sns
```

Fig. 1 Import of libraries

In addition, those libraries that were used in the Model Processing stage were imported as shown below. The library "skelearn" was used for the predictive analysis, the library "Keras. models" helped us to apply the model, and finally the library "Keras. layers" was imported and helped us to develop neural networks, as shown in Fig. 2.

```
# we import the libraries for the graphs
from sklearn.metrics import mean_absolute_error
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout,Flatten
```

Fig. 2 Import of libraries for forecasting

For reading data we created the variable "data" in it we saved the DataFrame, then we used "pd. read_csv" to load our DataFrame as shown in Fig. 3.

```
#we save the data set in the data variable
databt=pd.read_csv('BitCoin.csv')
```

Fig. 3 Data reading

To visualize the DataFrame items, "columns" were used, as shown in Fig. 4.

```
#shows the name of the items
databt.columns

Index(['Timestamp', 'Open', 'High', 'Low', 'Close', 'Volume_(BTC)',
       'Volume_(Currency)', 'Weighted_Price', 'date'],
      dtype='object')
```

Fig. 4 Ítems del dataframe

To quickly visualize the content of the DataFrame and to generate a table, the function head () was used, which shows the first 5 rows, as shown in Fig. 5.

```
#first 5 rows of the dataset
databt.head()
```

	Timestamp	Open	High	Low	Close	Volume_(BTC)	Volume_(Currency)	Weighted_Price	date
0	1325317920	4.39	4.39	4.39	4.39	0.455581	2.0	4.39	2011-12-31
1	1325317980	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2011-12-31
2	1325318040	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2011-12-31
3	1325318100	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2011-12-31
4	1325318160	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2011-12-31

Fig. 5 Dataframe rows and columns

To find out how many rows and columns the DataFrame has, we used "shape" which shows the number of rows followed by the number of columns, as shown in Fig. 6.

```
#number of rows and columns
databt.shape

(2656467, 9)
```

Fig. 6 Number of rows and columns

Finally, we use the function info () which shows us the type of variable for each item, as shown in Fig. 7.

```
#information about the variables of each item
databt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2656467 entries, 0 to 2656466
Data columns (total 9 columns):
 #   Column              Dtype
---  ------              -----
 0   Timestamp           int64
 1   Open                float64
 2   High                float64
 3   Low                 float64
 4   Close               float64
 5   Volume_(BTC)        float64
 6   Volume_(Currency)   float64
 7   Weighted_Price      float64
 8   date                object
dtypes: float64(7), int64(1), object(1)
memory usage: 182.4+ MB
```

Fig. 7 Type of variable for each item

D. Type of the variable for each item

In this stage, each item of the DataFrame is detailed and the information contained in each of them. There are a total of 9 items, the first one is "Timestamp" which shows a numerical amount indicating the day of transaction made by the network, "Open" indicates the entry price, "High" indicates the maximum price reached by the coin, "Low" indicates the minimum value reached by the coin, "Close" indicates the exit price, "Volume_(BTC)" shows the value of buying and selling in a certain period, in other words, the volume increases or decreases depending on the transactions made, in this way it is possible to measure the interest in the use of the coin, "Volume_(Currency)" indicates the value acquired or sold and finally "Weighted_Price" is the value acquired by averaging "Open" and "Close".

E. Data preparation

After having understood each item of the DataFrame comes the stage of data preparation, as indicated by the methodology, the first thing to do is to find those data that may generate some conflict or prevent the data from being altered, that is why the first thing that was done was to replace the NaN values by zeros because they can generate problems when making the prediction, likewise, first, the DataFrame was called with the variable "data", then the items where NaN values exist were entered, then "filling" was used to replace the NaN values, as shown in Fig. 8.

```
# replacing null values
databt['Volume_(BTC)'].fillna(value=0, inplace=True)
databt['Volume_(Currency)'].fillna(value=0, inplace=True)
databt['Weighted_Price'].fillna(value=0, inplace=True)
```

Fig. 8 Replacing NaN values

The values "Open", "High", "Low" and "Close" were also normalized because they are values of continuous series, i.e. they are quite small time values and this was not reflected in the table presented in Fig. 5, therefore the normalization is applied as shown in Fig. 9.

```
# we correct the values "Open", "High"
#"Low" and "Close" which is a continuous time series
databt['Open'].fillna(method='ffill', inplace=True)
databt['High'].fillna(method='ffill', inplace=True)
databt['Low'].fillna(method='ffill', inplace=True)
databt['Close'].fillna(method='ffill', inplace=True)
```

Fig. 9 Replacing NaN values

After applying the normalization, the function "head ()" was used to observe the replaced data, as shown in Fig. 10.

```
#Printing the replaced and modified values
databt.head()
```

	Timestamp	Open	High	Low	Close	Volume_(BTC)	Volume_(Currency)	Weighted_Price	date
0	1325317920	4.39	4.39	4.39	4.39	0.455581	2.0	4.39	2011-12-31
1	1325317980	4.39	4.39	4.39	4.39	0.000000	0.0	0.00	2011-12-31
2	1325318040	4.39	4.39	4.39	4.39	0.000000	0.0	0.00	2011-12-31
3	1325318100	4.39	4.39	4.39	4.39	0.000000	0.0	0.00	2011-12-31
4	1325318160	4.39	4.39	4.39	4.39	0.000000	0.0	0.00	2011-12-31

Fig. 10 NaN values replaced

The item "Timestamp" must also be changed for a better understanding because it shows numeric values and to make the prediction we are interested in knowing the transaction date, that is why it is transformed by a DateTime variable, for this we will create an additional column named "Date", there the date is shown, to achieve this we used the function "to_datetime ()". Then the whole process was saved in an array called "group", also the array was stored in a variable called "data", finally the results were displayed using the function "head ()" as shown in Fig. 11.

```
# we add column "date" to the DataFramer and convert the data in the Timestamp column to date format.
databt["date"]=pd.to_datetime(databt["Timestamp"],unit="s").dt.date
# group the data from the columns "date" and "close" to a variable named "datac".
group=databt.groupby("date")
datac=group["Close"].mean()
#show the data of the variable "group"
group.head()
```

Date	Timestamp	Open	High	Low	Close	Volume_(BTC)	Volume_(Currency)	Weighted_Price
2011-12-31	1325346930	4.472552	4.472624	4.472552	4.472624	0.098469	0.439381	0.018478
2012-01-01	1325419170	4.680778	4.680778	4.680778	4.680778	0.015001	0.073458	0.010014
2012-01-02	1325505570	5.000000	5.000000	5.000000	5.000000	0.013228	0.066139	0.003472
2012-01-03	1325591970	5.145917	5.145917	5.145917	5.145917	0.061137	0.322781	0.029181
2012-01-04	1325678370	5.176708	5.228729	5.176708	5.228729	0.074468	0.394497	0.032551

Fig. 11 Standard date

Finally, a graph was generated using the "matplotlip" library, on the X-axis was found "Date" (item previously added) and on the Y-axis I used "Close" (exit price of the currency), the whole procedure is shown in Fig. 12.

```
# Graph generated with x = Date and y =Close
price_by_date=databt["Close"]
price_by_date.plot()
plt.xlabel('Date')
```

```
Text(0.5, 0, 'Date')
```

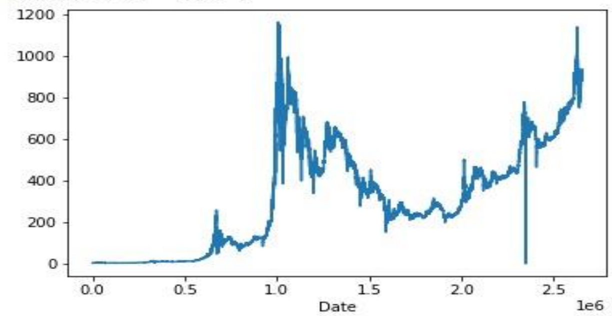

Fig. 12 Chart using Date as X-axis

As can be seen in Fig.12 the graph shows how the price has been varying over the years, in the "Date" axis the range goes from (0.0 - 2.5) which refers to the years (2012 - 2021), this shows an idea of how the price of Bitcoin has been evolving over the years.

F. Data preparation Preparation of the Model

The algorithm used in the project for the predictive analysis is the Recurrent Neural Network (LSTM) because it can make predictions with a large amount of data, the preparation of the model was started using the function "is null ()" which is used to ignore the zero values and the function "sum ()" to add the numerals contained in the item "Close", as shown in Fig. 13.

```
#we identify null values
datac.isnull().sum()
```

```
0
```

Fig. 13 Funciones isnull () y sum ()

Then the variable "data c" created previously was used, together with the function "head ()", thus obtaining the following result Fig. 14.

```
#data sampling and item to be used
datac.head()
```

```
date
2011-12-31     4.482500
2012-01-01     4.806667
2012-01-02     5.000000
2012-01-03     5.252500
2012-01-04     5.223333
Name: Close, dtype: float64
```

Fig. 14 Function head ()

To train the predictive model, the last 50 rows of the item "Close", which is represented by the variable "data c", were collected, then the operation was stored in the variable "close_train", as shown in Fig. 15.

```
#we use the last 50 rows of the DataFramer
close_train=datac.iloc[:len(datac)-50]
close_test=datac.iloc[len(close_train):]
```

Fig. 15 DataFrame, last 50 records

The previously created variable was used to store an array, in which the position of the "Close" values was assigned in zeros and ones. This is shown in Fig. 16

```
#we set the values between 0 and 1 by scaling them
close_train=np.array(close_train)
close_train=close_train.reshape(close_train.shape[0],1)
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))
close_scaled=scaler.fit_transform(close_train)
```

Fig. 16 Scaling values between 0 and 1

Then, the X and Y values of the last 50 days were established by creating the arrays "x_Day" and "y_Close". For this purpose, the "for" command was implemented and the "print()" function was used to display the structure of the arrays, as shown in Fig. 17.

```
# we set the values x = day and y = close
days=50
x_Day=[]
y_Close=[]

for i in range(days,close_scaled.shape[0]):
    x_Day.append(close_scaled[i-days:i,0])
    y_Close.append(close_scaled[i,0])

# we set the x and y values in arrays

x_Day,y_Close=np.array(x_Day),np.array(y_Close)
x_Day=x_Day.reshape(x_Day.shape[0],x_Day.shape[1],1)
print("x_Day =",x_Day.shape)
print("y_Close =",y_Close.shape)
```

```
x_Day = (1747, 50, 1)
y_Close = (1747,)
```

Fig. 17 Establishing values X y Y

Now, for the elaboration of the prediction, the LSTM model was used as mentioned above. For this purpose, the data of the sequence model was loaded with the "Sequential ()" function, as shown in Fig. 18.

```
# we load the data using the LSTM model
model=Sequential()
model.add(LSTM(10,input_shape=(None,1),activation="relu"))
model.add(Dense(1))
model.compile(loss="mean_squared_error",optimizer="adam")
model.fit(x_Day,y_Close,epochs=100,batch_size=32)
```

Fig. 18 Implementation of the sequence model

After having loaded the data, a variable named "inputs" was created to deliver the data to be trained. As shown in Fig.19.

```
#data input
inputs=datac[len(datac)-len(close_test)-days:]
inputs=inputs.values.reshape(-1,1)
inputs=scaler.transform(inputs)
```

Fig. 19 Data Entry

Now, an array called "x_testing" was created, in it the data stored in the variable "inputs" was entered, then using the "for" command, to add all the data to the array. As shown in Fig. 20.

```
# testing the input data
x_testing=[]
for i in range(days,inputs.shape[0]):
    x_testing.append(inputs[i-days:i,0])
x_testing=np.array(x_testing)
x_testing=x_testing.reshape(x_testing.shape[0],x_testing.shape[1],1)
```

Fig. 20 Input data testing

Then, we proceeded to prepare the data to perform the prediction, for this purpose we created the variable "predicted_datac" in it we used "model" to indicate the model, then with the function "predict ()" we entered as parameter the variable "x_testing" previously created, then the variable was equalized again to use the function "inverse_transform ()" the function helped us to prepare the data to perform the test, as shown in Fig. 21.

```
#preparing data for prediction
predicted_datac=model.predict(x_testing)
predicted_datac=scaler.inverse_transform(predicted_datac)
```

Fig. 21 Data prepared for prediction

Now we proceed to test the last 50 rows of data of the Close item, for this, we again saved in an array the data of the Close item, finally, we used "reshape ()" which helped to convert and simplify the array in 2 values (X, Y), it should be noted that the function does not alter the values, they are still the same initial value. As shown in Fig. 22.

```
# testing close_test data
data_testing=np.array(close_test)
data_testing=data_testing.reshape(len(data_testing),1)
```

Fig. 22 Test al ítem Close

Finally, for the creation of the graph, we used the library "matplotlib. pyplot" for this we called the variable "plt" in this way we used the tools to indicate the size of the figure, the color, to mention the name of the X-axis and the Y-axis respectively and also the function "show ()" to show the graph. As shown in Fig. 23.

Fig. 23 Generating predictive graphics

Next, the prediction on the price of Bitcoin is displayed on a graph. This is shown in Fig. 24.

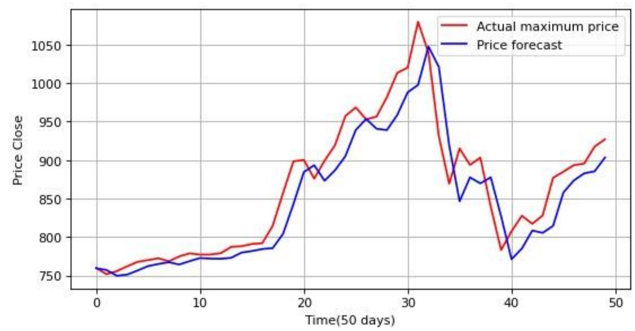

Fig. 24 Bitcoin Price Forecast Chart

As shown in Fig. 24. The X-axis shows how the Bitcoin price has been evolving in the last 50 days, the Y-axis shows the Bitcoin exit price. As can be seen, the red line is the current price and the blue line is the prediction generated by applying the LSTM algorithm.

G. Model Evaluation

In this stage of the methodology, the LSTM model applied for prediction was tested and analyzed using the same procedure applied in the Model Preparation stage, but in this case, the item "Weighted_Price" was used in the Y-axis, in this way the model evaluation was performed, in Fig. 25 the change that was made is shown, now the variable "data c" stores "Weighted_Price" and "date".

```
# we add column "date" to the DataFramer and convert the data in the Timestamp column to date format.
databt["date"]=pd.to_datetime(databt["Timestamp"],unit="s").dt.date
# group the data from the columns "date" and "close" to a variable named "datac".
group=databt.groupby("date")
datac=group["Weighted_Price"].mean()
```

Fig. 25 We use the "Weighted_Price" item for evaluation

After performing the same procedure that was applied in the Model Preparation, the predictive graph was generated, as shown in Fig. 26.

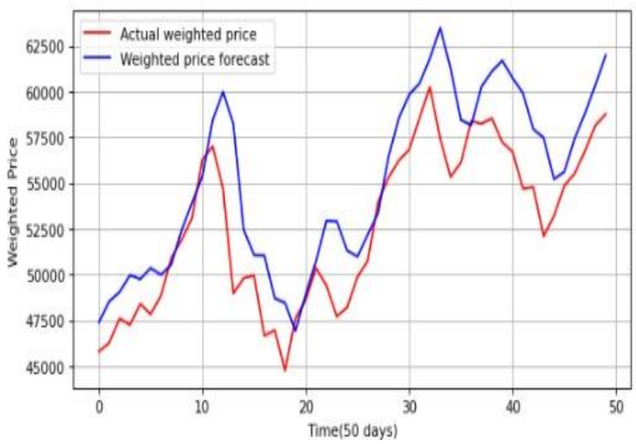

Fig. 26 Predictive Graph using the item "Weighted_Price"

As shown in Fig. 23, the tracking prediction is quite similar to the original result, therefore, the model used for the prediction is adequate

H. About the Methodology

The Data Science methodology, allowed for a more orderly deployment of the project, in addition to adding precision to the data processing, in combination with Machine Learning techniques and the LSTM algorithm.

Throughout the development of the project, the following advantages and disadvantages were found.

Table 1. Comparative table on the use of Machine Learning

MACHINE LEARNING	
ADVANTAGES	**DISADVANTAGES**
It can handle large amounts of data without the need for area knowledge or parameter settings. [13].	Guiding the program through all phases of the system so that it knows how to identify each category automatically, therefore, this modality requires supervised learning. [15].
Machine learning and adaptation to changing environments [13].	The training time is longer because the program needs to be guided [13].
They have good information accuracy and support incremental learning. [13].	Machine Learning applications are limited and focus on specific processes, rather than the entire manufacturing program or manufacturing system. [15].
It helps solve problems when existing methods require manual changes or long lists of rules. In addition, machine learning algorithms always simplify code and improve performance [14].	When you select a Machine Learning algorithm suitable for the requirements of the research problem. Since it implies a careful analysis of the algorithms to be used [15].
A comprehensive solution that traditional methods fail to provide [14].	Description of hidden patterns in the data without making a prior hypothesis [15].
Obtain information on complex problems and large amounts of data [14].	Unconstrained inquiry of patterns and relationships [15].

The machine learning methodology stands out for being agile and for its excellent work with the algorithms that are programmed, unlike other methodologies, such as Deep Learning, which is very similar, but which is different in certain points, for example, it usually works automatically and agile by recognizing similar files or documents according to their search, an example of this are for example those facial recognition applications, the features learned from the first layer can be the edges, addresses, and some local information. The second layer usually detects some parts of the object that are a combination of the edges and directions. The upper layers can further abstract the image of the face by combining the features of the previous layers (eye contours, noses, lips) [16].

On the other hand, Machine Learning is very parallel to one of the most popular methodologies called evolutionary algorithms (EA), which is part of a category of stochastic optimization algorithms that are inspired by the Darwinian principle and natural selection, which have been used mainly in applications. Also, EA is simple and efficient to use, since theoretical knowledge of the principles of operation and complexity is much more complicated and lags far behind practical use due to the difficulty of the mathematical analysis with which it usually works [17].

I. About the case study

For the case study, research was carried out to rescue some contributions from articles related to our topic. In one paper they applied 4 methods, two of which (method 1 and method 2) were based on decision trees and method 3 applied short- and long-term recurrent neural networks. As a conclusion, they state that LSTM recurrent neural networks performed better when predictions were based on 50 days of data as they can capture also long-term dependencies and are very stable with price volatility, the result allowed to obtain gains also when considering transaction fees of up to 1%. This is shown in Fig. 24 y 25 [18].

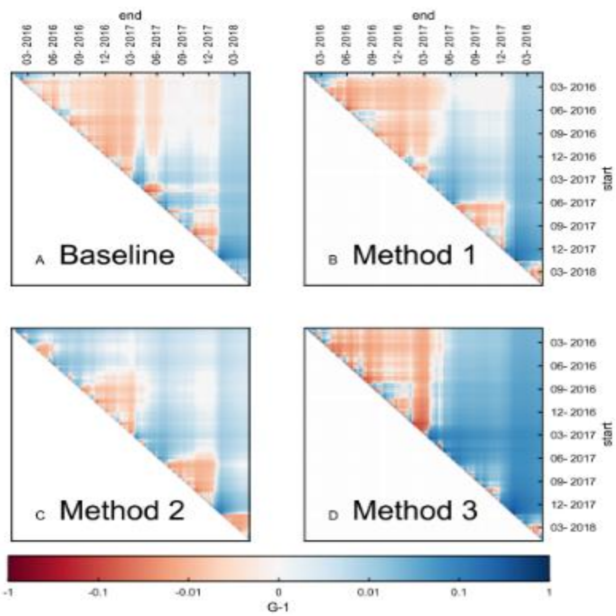

Fig. 27 The 4 methods applied in the research

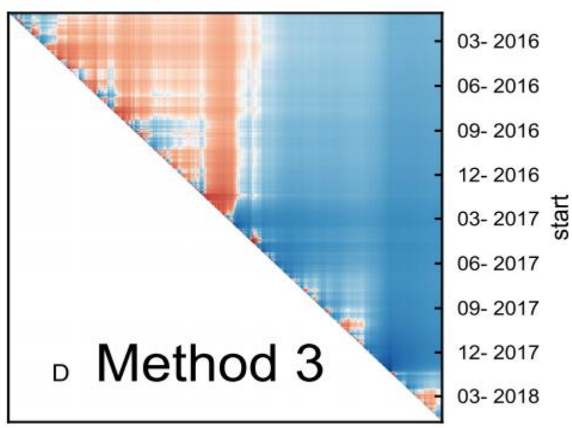

Fig. 28 Method 3: LSTM Model

The figure shows the translated geometric mean return G-1. Shades of red refer to negative returns and shades of blue to positive returns (see color bar) [18].

V. CONCLUSION

After having presented the prediction of the bitcoin price increase by applying Machine Learning techniques, it is concluded that the LSTM model is quite accurate when it comes to making predictions. As observed in the present research work, the results were exceptional because the predictive graph of the last 50 rows of the item "Close" was generated considering as time the item "Day" of the last 50 days. Based on the results obtained, it could be observed how the predicted price considerably resembles the rise and fall of the real bitcoin price during those days. Also, according to the chart, the predicted price of bitcoin has had a high and low value during that period. Therefore, we can say that it was possible to know the price of Bitcoin on the rise for the next few months and this trend will probably be with us for a long time to come.

The methodology used has made it possible to cover various aspects ranging from the analytical, going through the requirement, the collection, understanding, and preparation of data, as well as the preparation and evaluation of the chosen model; all this, to apply all the necessary knowledge to obtain a more assertive prediction, all this was achieved thanks to the methodology and the different stages it presents, since it was essential for good research development.

As future work, we recognize that these predictive analytics should be a reality in the coming months, given that bitcoin is currently considered the best cryptocurrency, used for business purposes. In addition to this, this predictive analysis will benefit different companies or financial institutions in the investment of money with the help of bitcoin. It is suggested that topics related to the process of creation and inner workings of bitcoin be addressed, as this will help to have more knowledge about its important development in the economic and business world.

REFERENCES

[1] Aggarwal, I. Gupta, N. Garg, and A. Goel, Deep Learning Approach to Determine the Impact of Socio-Economic Factors on Bitcoin Price Prediction, 2019 Twelfth International Conference on Contemporary Computing (IC3). (2019) 1-5. doi: 10.1109/IC3.2019.8844928.

[2] Yhlas. Sovbetov, Factors Influencing Cryptocurrency Prices: Evidence from Bitcoin, Ethereum, Dash, Litcoin, and Monero, Journal of Economics and Financial Analysis. 2(2) (2018) 1-27.

[3] I. Makarov y A. Scholar Price Discovery in Crypto Currency Markets. 109 (2019) 97-99.

[4] Alexander C, & Heck D. F, Price Discovery in Bitcoin: The Impact of Unregulated Markets. Journal of Financial Stability. 50 (2020) 100776.

[5] H. Liu and B. Lang, Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey, Applied Sciences. 9(20) (2019) 4396. www.doi.org/10.3390/app9204396

[6] G. Choy, O. Khalilzadeh, M. Michalski, S. Do, A. E. Samir, O. S. Pianykh, J. R. Geis, P. V. Pandharipande, J. A. Brink, and K. J. Dreyer, Current Applications and Future Impact of Machine Learning in Radiology. 288(2) (2018) 318–328. https://doi.org/10.1148/radiol.2018171820

[7] J. D. Pineda-Jaramillo, A Review Of Machine Learning (Ml) Algorithms Used for Modeling Travel Mode Choice Dyna. 862(11) (2019) 32. www.doi.org/10.15446/dyna.v86n211.79743

[8] O. Gutierrez and A. Moreno, Bitcoin Financial and Legal Considerations about its Nature and Proposed Approach for its Regulation. 69 (2017) 133-180.

[9] D. Hinostrosa, The Machine Learning through the Times, and the Contributions to Humanity, Thesis, Universidad Libre Seccional Pereira. (2018).

[10] C. Herraiz, Blockchain: An Instrument of Transformation for the Sector, Bulletin Of Academic Studies. 75 (2020) 27-52.

[11] (2021). M. Zielinski, Mark Zielinski Personal Profile, CV. [Online]. Available: www.mcz.fyi/cv/

[12] L. Alessandretti, A. ElBahrawy, L. Aiello, and A. Baronchelli, Anticipating cryptocurrency prices using machine Learning, SSRN, 9–11(2019).

[13] H. Bhavsar and A. Ganatra, A Comparative Study of Training Algorithms for Supervised Machine Learning, International Journal of Soft Computing and Engineering (IJSCE). 2(4) (2012) 2231–2307.

[14] K. C. A. Khanzode and R. D. Sarode, Advantages and Disadvantages of Artificial Intelligence and Machine Learning: A Literature Review, Int. J. Libr. Inf. Sci. 9(1) (2020) 30–36.

[15] T. Wuest, D. Weimer, C. Irgens, and K. D. Thoben, Machine Learning in Manufacturing: Advantages, Challenges, and Applications, Prod. Manuf. Res. 4(1) (2016)23–45. doi: 10.1080/21693277.2016.1192517.

[16] J. S. Bhatt and M. V. Joshi, Deep Learning in Hyperspectral Unmixing: A Review, IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium. (2020) 2189-2192. doi: 10.1109/IGARSS39084.2020.9324546.

[17] W. Zheng, H. Chen and X. Yao, Analysis of Evolutionary Algorithms on Fitness Function with Time-linkage Property, in IEEE Transactions on Evolutionary Computation. doi: 10.1109/TEVC.2021.3061442.

[18] (2019). L. Alessandretti, A. ElBahrawy, L. Aiello, and A. Baronchelli, Anticipating cryptocurrency prices using machine Learning. [Online]. Available: www. papers.ssrn.com/sol3/papers.cfm?abstract_id=3183792.