BITCOIN PRICE PREDICTION USING LSTM

GUIDED
BY
Miss.VANISHREE A
AP/CSE

PRESENTED
BY
K.ANANDARAJ(712219104004)
M.HARISHKUMAR(712219104011)
K.JAYAPRAKASH(712219104012)
V.SIVABALAN(712219104025)

INTRODUCTION

- **Bitcoin** is a digital or virtual currency created in 2008 that uses peer-to-peer technology to facilitate instant payments.
- Bitcoin has recently gotten a lot of attention in the disciplines of economics, cryptography, and computer science.
- Bitcoin adoption has been increasing at an annual rate of 113%.
- **Machine learning (ML)** is a type of artificial intelligence that can predict the future based on past data.
- ML-based models have various advantages over other forecasting models as prior research has shown that it not only delivers a result that is nearly or exactly the same as the actual result, but it also improves the accuracy of the result.

ABSTRACT

Bitcoin is the first digital decentralized cryptocurrency that has shown a significant increase in market capitalization in recent years. The objective of this paper is to determine the predictable price direction of Bitcoin in USD by machine learning technique. We explored several algorithms of machine learning using supervised learning to develop a prediction model and provide informative analysis of future market prices. Due to the difficulty of evaluating the exact nature of a Time Series model, it is often very difficult to produce appropriate forecasts. Then we continue to implement long short-term memory cells (LSTM) algorithm. Thus, we analyzed the time series model prediction of bitcoin prices with greater efficiency using long short-term memory (LSTM) techniques.

PROBLEM STATEMENT

1. Bitcoin is the most complex cryptocurrency which value change in every second.
2. Investing money for bitcoin is more risk and less profit.

EXISTING SYSTEM

- Statistical methods including Logistic Regression for Bitcoin daily price prediction with an accuracy of 66%.
- In this paper,Compared with benchmark results for daily price prediction, Machine learning models including Random Forest, Support Vector Machine for predicted the bitcoin price and its accuracy of 66% and 65.3%, respectively.

DRAWBACKS

- Logistics Regression-It is used to the probability of event success and failure.So we can't get a good accuracy.
- Support Vector Machine-It is not suitable for large dataset.SVM does not perform very well when the dataset has more noise i.e..,target classes are overlapping.
- Random Forest- Random forest creates a lot of trees and combines their outputs.It requires much more time to train.

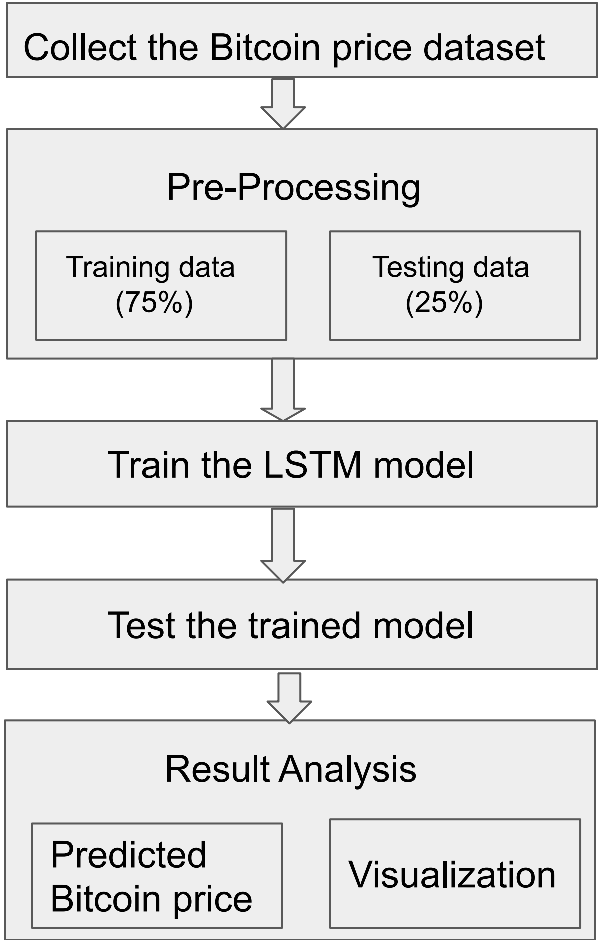
OBJECTIVES

- In our project, we used a time series model (LSTM) to leverage machine learning technology to predict the real-time price of Bitcoin.
- However, machine learning literature is lacking verification of whether or not the stock evaluation strategies are legitimate for the cryptocurrencies, and if so, how they may be modified.
- That is what features want to be eliminated or introduced as a foundation for price prediction, whether current machine learning algorithms work for cryptocurrencies, and which technique yields the excellent outcomes.

LITERATURE SURVEY

S.NO	TITLE	METHOD USED
1	Journal of computational and applied mathematics.	Logistic Regression, Random Forest, Support Vector Machine
2	Improving the Cryptocurrency Price Prediction Performance Based on Reinforcement Learning.	Reinforcement learning
3	Real-Time Prediction of BITCOIN Price using Machine Learning Techniques and Public Sentiment Analysis.	Recurrent Neural Network(RNN).
4	Stochastic neural networks for cryptocurrency price prediction.	Multilayer perceptron (MLP)

WORKFLOW


Collect the Bitcoin price dataset
Pre-Processing
Training data (75%)
Testing data (25%)
Train the LSTM model
Test the trained model
Result Analysis
Predicted Bitcoin price
Visualization

DATASET COLLECTION

- In our project,we are collecting dataset of 6 years (Oct2015-Oct2021) from site:https://www.kaggle.com/datasets/paulrohan2020/bitcoin-historic-prices-from-oct2015-to-oct2021.

FEATURES	DEFINITION
Time	Date and time with on that one minute time
Low	The lowest price of bitcoin on that one minute time
High	The highest price of bitcoin on that one minute time
Open	The open price of bitcoin on that one minute time
Close	The close price of bitcoin on that one minute time
Volume	Bitcoin volume on that one minute time

SNIPS OF DATASET

In [12]:
```
dataset.loc[:,:]
```

Out[12]:

	time	low	high	open	close	volume
0	2021-10-30 03:00:00	61868.81	61920.00	61888.18	61919.98	2.946079
1	2021-10-30 02:59:00	61882.74	61912.67	61903.16	61882.74	2.942357
2	2021-10-30 02:58:00	61854.82	61907.96	61854.82	61903.15	3.047848
3	2021-10-30 02:57:00	61848.04	61883.02	61883.01	61851.39	3.381070
4	2021-10-30 02:56:00	61879.33	61925.42	61907.47	61882.28	5.663128
...	...	...	...	...	...	...
3113271	2015-10-27 00:04:00	287.07	287.07	287.07	287.07	0.086100
3113272	2015-10-27 00:03:00	287.08	287.09	287.08	287.09	0.554400
3113273	2015-10-27 00:02:00	286.89	287.10	287.10	286.89	72.403200
3113274	2015-10-27 00:01:00	287.09	287.09	287.09	287.09	0.822760
3113275	2015-10-27 00:00:00	287.10	287.10	287.10	287.10	0.536200

3113276 rows × 6 columns

DATA PREPROCESSING

- In datasets,sometimes it have unnecessary records like null values,duplicate values and errors.
- It will affect our accuracy of result,so we use data preprocessing.
- And we employed some methods for clearing unnecessary records.

METHODS USED IN PREPROCESSING

1. Object dtype to datetime dtype - In our dataset,'time' column is being treated as an object rather than as dates. To fix this, I will use the (pd.to_datetime() function which converts the arguments to dates.

In [18]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3113276 entries, 0 to 3113275
Data columns (total 6 columns):
 #   Column  Dtype  
---  ------  -----  
 0   time    object 
 1   low     float64
 2   high    float64
 3   open    float64
 4   close   float64
 5   volume  float64
dtypes: float64(5), object(1)
memory usage: 142.5+ MB
```

In [19]:
```
dataset = dataset.astype({'time': 'datetime64'})
```

In [20]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3113276 entries, 0 to 3113275
Data columns (total 6 columns):
 #   Column  Dtype         
---  ------  -----         
 0   time    datetime64[ns]
 1   low     float64       
 2   high    float64       
 3   open    float64       
 4   close   float64       
 5   volume  float64       
dtypes: datetime64[ns](1), float64(5)
memory usage: 142.5 MB
```

2.groupby('date') -The unit of the arg (D,s,ms,us,ns) denote the unit, which is an integer or float number. This will be based off the origin. Example, with unit='ms' and origin='unix' (the default), this would calculate the number of milliseconds to the unix epoch start.

Say you pass an int as your arg (like 20203939), with unit, you'll be able specify what unit your int is away from the origin. In the example here, if we set unit='s', this means pandas will interpret 20203939 as 20,203,939 seconds away from the origin. Available units are [D,s,ms,us,ns]

In [30]:
```python
dataset['date'] = pd.to_datetime(dataset['time'],unit='s').dt.date

display(dataset.head())

group = dataset.groupby('date')

btc_closing_price_groupby_date = group['close'].mean()
```

	time	low	high	open	close	volume	date
0	2021-10-30 03:00:00	61868.81	61920.00	61888.18	61919.98	2.946079	2021-10-30
1	2021-10-30 02:59:00	61882.74	61912.67	61903.16	61882.74	2.942357	2021-10-30
2	2021-10-30 02:58:00	61854.82	61907.96	61854.82	61903.15	3.047848	2021-10-30
3	2021-10-30 02:57:00	61848.04	61883.02	61883.01	61851.39	3.381070	2021-10-30
4	2021-10-30 02:56:00	61879.33	61925.42	61907.47	61882.28	5.663128	2021-10-30

In [31]:
```python
display(btc_closing_price_groupby_date.head(10))

print("Length of btc_closing_price_groupby_date :", len(btc_closing_price_groupby_date))
```

```
date
2015-10-27    292.887484
2015-10-28    301.991834
2015-10-29    310.537530
2015-10-30    326.665808
2015-10-31    323.877132
2015-11-01    319.527295
2015-11-02    339.226860
2015-11-03    386.773759
2015-11-04    450.035105
2015-11-05    400.249521
Name: close, dtype: float64

Length of btc_closing_price_groupby_date : 2196
```

3.Mix Max Scaling of Data post Train-Test Split

- Transform features by scaling each feature to a given range. This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g. between zero and one.
- Scaling must be done after the data has been split into training and test sets — with each being scaled separately.
- A common mistake when first using the LSTM is to first normalize the data before splitting the data.
- The reason this is erroneous is that the normalization technique will use data from the test sets as a reference point when scaling the data as a whole. This will inadvertently influence the values of the training data, essentially resulting in data leakage from the test sets.

TRAIN TEST SPLIT

- In our dataset,we split it for training and testing.
- Training have 75% of dataset called train dataset because of getting good accuracy.
- Testing have 25% of dataset called test dataset.

REFERENCES

1. Chen, Z., Li, C. and Sun, W., 2020. Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics*, *365*, p.112395.
2. Shahbazi, Z. and Byun, Y.C., 2021. Improving the Cryptocurrency Price Prediction Performance Based on Reinforcement Learning. *IEEE Access*, *9*, pp.162651-162659.
3. Raju, S.M. and Tarif, A.M., 2020. Real-Time Prediction of BITCOIN Price using Machine Learning Techniques and Public Sentiment Analysis. *arXiv preprint arXiv:2006.14473*.
4. Jay, P., Kalariya, V., Parmar, P., Tanwar, S., Kumar, N. and Alazab, M., 2020. Stochastic neural networks for cryptocurrency price prediction. *Ieee access*, *8*, pp.82804-82818.
5. Hamayel, M.J. and Owda, A.Y., 2021. A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms. AI, 2(4), pp.477-496.