Cryptocurrency Analysis and Prediction using Distributed Machine Learning

CSC-721 Distributed Systems

1st Padam Jung Thapa
Department of Computer Science
University of South Dakota
Vermillion, South Dakota
padamjung.thapa@coyotes.usd.edu

***Abstract*—Cryptocurrencies are digital currencies that have garnered significant investor attention in the financial markets. In this project, we have made an extensive analysis of various cryptocurrencies. First, we have taken the raw real-time data of the ten most popular cryptocurrencies, including Bitcoin, Ethereum, Ripple (XRP), Binance Coin (BNB), DogeCoin, Cardano (ADA), Polygon (MATIC), Polkadot (DOT), Solana (SOL), USDC, then we extract, transform and load the data using Spark(Pyspark), which is later saved in the Postgres Database and the Hadoop Distributed File System (HDFS). The aim of this project is to predict the hourly closing price of the cryptocurrencies mentioned above. This plays a vital role in making trading decisions. The machine learning model then uses the LSTM model to produce a prediction, which will be displayed in the dashboard using React. The models show excellent predictions depending on the root mean squared error (RMSE) and mean absolute percentage error (MAPE). Furthermore, the dashboard also displays analysis data obtained from PySpark, and at the end, the machine learning model is saved in the AWS S3 platform.**

***Index Terms*—Financial Market, Big Data, Cryptocurrency, Time Series Prediction (LSTM), PySpark.**

I. Introduction

The emergence of cryptocurrency as a new type of asset as a result of the development of financial technology has opened up several study opportunities. Forecasting cryptocurrency prices is challenging because of price volatility and dynamism. There are several different cryptocurrencies in use all around the world. Cryptocurrencies are a type of digital money that may be used for online transactions; unlike traditional money, they were created using encryption.

A prediction automation tool is required to address the aforementioned issue with fluctuations and assist investors in making decisions about whether to engage in the bitcoin or other cryptocurrency markets. Today, automated systems are frequently used to make general stock market forecasts, and we may apply the same processes and strategies to the cryptocurrency market.

When a workload is too heavy for a single computer or device to manage, distributed systems are utilized. Additionally, they come in handy when the workload is unpredictable, such as on Cyber Monday and Black Friday when there is a spike in online sales. Nowadays, nearly every online application connected to the internet is constructed using a distributed architecture. At its core, blockchain is a peer-to-peer distributed ledger that is append-only, immutable (very difficult to modify), cryptographically secure, and can only be updated by peer consensus. [1-3]

II. Why your approach is Important/Motivation

Most of the methodology mentioned in the literature only applies to prototypes and is mostly concerned with creating and fine-tuning models. It is uncommon to come across any implementation that has used the concept and presented it in an efficient manner. The project's goal is to offer a fully deployable and scalable solution in a breakeven manner, as different clusters can be fixed easily with no complications resulting in the avoidance of a single point of failure. Our method analyzes the distributed and cloud architecture that ensures the delivery of results while focusing on reusability and scalability along with the fast model transformation.

The necessity of such a strategy is made clear by the fact that many machine-learning initiatives fail even before they are ready for production. Given our system's versatility, we also want to offer a foundation for any form of the end-to-end crypto analysis system. Because they may be replaced with any other tools that benefit the consumers, every component was carefully considered. We have utilized PostgreSQL, HDFS, AWS S3, and LSTMs for time series sequential data and PySpark, for instance, as storage for computation, raw data, model training, and results. The user can easily swap this module if the Hadoop map-reduce feels more natural to them than PySpark. Even though we wrote our various engines in Python, one might choose a different language that offers either more complex or simpler algorithms, depending on the user's needs.

III. Related Work/Literature review

We had to conduct an extensive study and brainstorming to complete this project because it uses the PySpark library and applies the principles of distributed systems. We discovered

certain materials based on the PySpark library's implementation with respect to the ideas of distributed systems into practical applications after sifting through a large number of research papers, web publications, and periodicals. Here are a few connected pieces that are helpful to us in finishing this job and are similar to ours.

A. Cryptocurrency Price Prediction Model using variations of RNNs including LSTMs

The motive of this study was to predict the three types of cryptocurrencies including bitcoin, ethereum, and litecoin with three different types of Recurrent Neural Network Algorithms (RNN) including the Gated recurrent networks (GRU), Long Short Term Memory (LSTM), and bi-LSTM models. Out of all three, the GRU outperformed all the other two models. The author has concluded with the validation that the AI algorithm is reliable and acceptable for cryptocurrency prediction and GRU can predict cryptocurrency prices better than LSTM and bi-LSTM but overall all algorithms represent excellent predictive results. [4]

B. Bitcoin Price Prediction with PySpark using Random Forest

The following body of literature focuses on utilizing Random Forest to forecast the price of cryptocurrencies. PySpark was used to parallelize tree-creation during the training of the Random Forest to handle large amounts of data. The root mean square error (RMSE) and Pearson's correlation coefficient (r) were computed on test data to evaluate the random forest model, which was developed, built, and tested using the pyspark and scikit learn frameworks. The past values of Bitcoin over a number of years were used to train the random forest model, which makes predictions. To anticipate the closing price of the following day, factors like the starting price, highest price, lowest price, closing price, volume of Bitcoin, volume of other currencies, and weighted price were taken into account. (Yakup Görür [5])

C. Time Series Analysis of Cryptocurrency Prices Using LSTM

In order to examine volatility and comprehend this behavior, this work investigates time series analysis using deep learning. A long short-term memory model is used to identify patterns in cryptocurrency closing prices and forecast future prices. The suggested model picks up knowledge from nearby values. The root-mean-squared error and a comparison to an ARIMA model are used to assess the performance of this model. [6]

D. Deep Learning Algorithm to Predict Cryptocurrency Fluctuation Prices

LSTM algorithm is presented in this work and is used to predict the prices of four different types of cryptocurrency. The LSTM model was assessed using mean square error (MSE), root mean square error (RMSE), and normalize root mean square error (NRMSE) assessments. The results of these models demonstrated that the LSTM algorithm performed better than other algorithms in forecasting all varieties of cryptocurrencies. As a result, it may be said to be the most efficient algorithm. The LSTM model delivered hopeful and precise predictions for all coins. The model was used to project future cryptocurrency closing values over a 180-day timeframe. The correlation between the prediction and the observed data was evaluated using the Pearson correlation measure. In the training and testing stages, the Pearson correlation measure was used to evaluate the correlation between the prediction and target values. In training, the LSTM algorithm predicted XRP currency prices with the strongest correlation values. [7]

IV. Materials and Methods

A. Datasets

The data used in this study included daily historical data from the website Binance. The data used in this study included daily historical data from the website Binance-com (accessed on 10 November 2022). In this study, four cryptocurrencies, namely AMP, ketCap.com. In this study, four cryptocurrencies, namely, Bitcoin, Ethereum, Ripple (XRP), Binance Coin (BNB), DogeCoin, Cardano (ADA), Polygon (MATIC), Polkadot (DOT), Solana (SOL), were investigated. When working with cryptocurrency data, it is helpful and vital to understand the distribution and behavior of the data by using a chart of steady and understandable fluctuation prices of the cryptocurrencies. All data sets were collected from May 2015 through April 2022 at 1 hr intervals. The Hyperparameters used in dataset creation were forecast variables and loopback variables. Investors have been engaged in active trading in 2022 with cryptocurrencies. Table 1 shows the features of cryptocurrencies used in this dataset, such as open, high, low, close, volume ETH, volume USDT, and trade count.

The dataset includes the following features as an input to the datasets: OPEN PRICE: The open represents the first price traded during the candlestick.

HIGH PRICE: The high is the highest price traded during the candlestick.

LOW: The low shows the lowest price traded during the candlestick.

CLOSE: The close is the last price traded during the candlestick.

Volume (ETH): Volume, in ETH traded in the stock market during a given measurement interval

Volume (Currency): Volume, in USD, traded on stock market during a given measurement interval. Weighted Price: Measure of the average price.

	unix	date	symbol	open	high	low	close	Volume ETH	Volume USDT	tradecount
0	1.669080e+12	11/22/2022 0:00	ETH/USDT	1107.34	1114.87	1100.73	1111.61	26840.0886	29705403.99	22373.0
1	1.669070e+12	11/21/2022 23:00	ETH/USDT	1107.98	1116.46	1106.82	1107.34	18732.0253	20795361.07	24670.0
2	1.669070e+12	11/21/2022 22:00	ETH/USDT	1094.97	1123.27	1093.70	1107.97	41393.4403	45901590.56	56059.0
3	1.669060e+12	11/21/2022 21:00	ETH/USDT	1104.94	1108.68	1082.36	1094.97	36953.8499	40469838.86	44548.0
4	1.669060e+12	11/21/2022 20:00	ETH/USDT	1091.22	1114.29	1089.51	1104.94	43447.0118	47929625.55	48732.0

Table 1: Sample Data

B. Long Short Term Memory (LSTM)

LSTM is another type of module provided for RNNs. It is an updated version of RNN, the difference is the connection between the hidden layers of RNN. We develop a long short-term memory network (LSTM) based analyzer for cryptocurrency using distributed machine learning. LSTM enables the network to learn long-term relations by utilizing forget and remember gates that allow the cell to decide which information to block or transmit based on its strength and importance. We take the data from Postgres every 1 hr and predict the closing price for the next hour. The LSTM-based recurrent network is trained by taking the sequence of the embedding feature vector. A fully connected layer is used to transform the output of the LSTM later and activated with TanH to output the prediction.

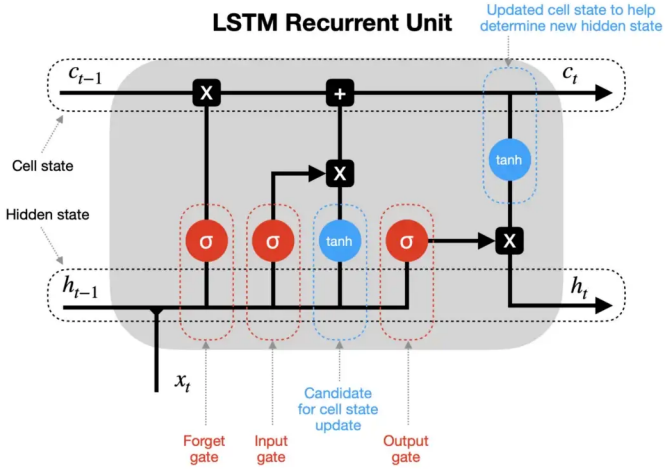
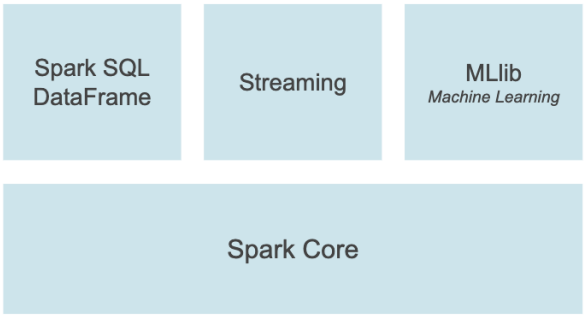


Fig: Long Short Term (LSTM) Neural Networks

where,
h_{t-1} = hidden state at previous timestep t-1 (short-term)
c_{t-1} = cell state at previous timestep t-1 (long term memory)
x_t = input vector at current timestep t
h_t = hidden state at current timestep t
c_{t-1} = cell state at current timestep t
X , + = vector pointwise multiplication and addition

C. PySpark

PySpark is nothing more than a Python-based interface for Apache Spark. It enables the development of Spark applications using Python APIs and offers the PySpark shell, enabling interactive data analysis in a distributed setting. The majority of Apache Spark's functionality, including Spark SQL, Mlib (Machine Learning), DataFrame, Streaming, and Spark Core, is supported by PySpark.



A key tool in the Hadoop Ecosystem is Spark. Hadoop's MapReduce can only be used for batch processing; it cannot be utilized to process real-time data. Spark may operate independently or atop the Hadoop framework to use huge data and perform real-time data analytics in a distributed computing environment.

It can enable several types of complicated analysis, including batch, stream, and machine learning, as well as business intelligence. Because Spark conducts in-memory computations, it is 100 times quicker than the Hadoop MapReduce framework for processing massive amounts of data. The big data era has led us to develop platforms for machine learning (ML) techniques, which have applications across many areas and frameworks for quick data storage and processing. With so many ML tools accessible, selecting one that can effectively analyze and deploy ML algorithms has proven difficult. Thankfully, Spark offers a versatile framework for carrying out various machine-learning tasks, such as classification, regression, optimization, clustering, dimensionality reduction, etc.

V. Proposed Methodology/Framework

With its PySpark framework and distributed system ideas, the cryptocurrency analysis system strongly emphasizes the execution of distributed systems. Everything in this project is developed on numerous servers, which gives it its singularity. Smaller modules or services are separated from the complicated framework. Our continuous data is obtained through WebSockets, and we have a scheduler set up that generates data in CSV format every ten minutes. This is the first factor that comes into play. The CSV data is converted using PySpark and Pandas in the second component, which is PySpark Data Processing. This stage completes all data processing and ETL activities, including data filtering, data cleaning, data, data transformation, and data extraction. The PostgreSQL database is subsequently filled with the cleansed data. The WebAPI, which is the third component to follow, uses dotnet 6 (.Net6) as an entity framework and object-relational mapper (ORM). It is a method that enables object-oriented data manipulation and querying from databases. ORM makes data manipulation simpler since DotNet6 is OS-independent and works everywhere.

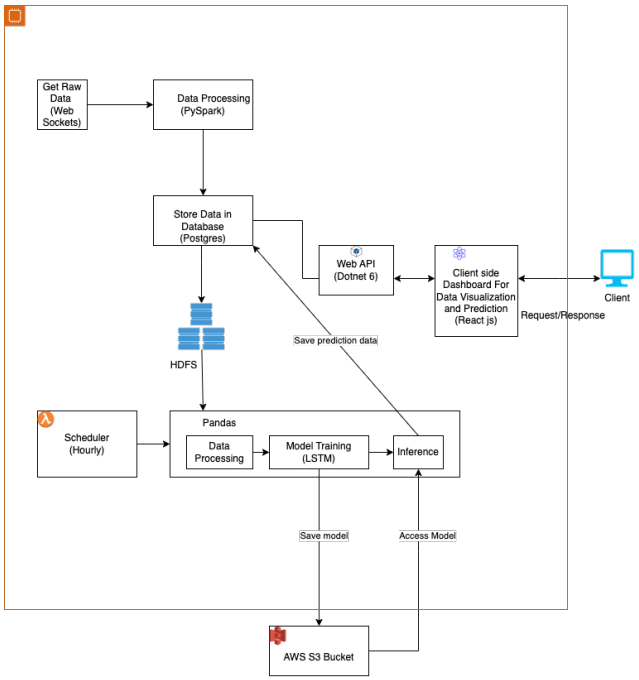


Fig: Model Architecture

The machine learning component, which makes up the fourth segment, uses data from the Postgres database every hour to forecast the closing price for the following hour. LSTM has been widely used in time series modeling since it can overcome the problem of vanishing gradients and better capture long-term dependencies of time series. It is frequently used, for example, to forecast stock prices, Bitcoin prices, etc. The LSTM model, which offers the greatest fit and most accurate results, receives data that is practically real-time. The predicted data is then saved into Postgres, which is then shown to the Dashboard. And finally, the model is saved in an Amazon S3 bucket.

The final part that comes into execution is the Front-end part, where the data is visualized in the dashboard using React. React is an open-source front-end JavaScript library for building user interfaces based on UI components, which works on the basis of component-wise rendering enabling dynamic web pages. We developed the cryptoanalysis dashboard where the Analysis chart and Prediction chart are portrayed using our LSTM model. For the chart section, we have used the APEX chart, and for the UI part, we have used the react component called Material UI (MUI).

VI. Results

A. Evaluation Metrics:

1. RMSE (Root Mean Square Error) - The difference between values (sample or population values) predicted by a model or an estimator and the values observed is typically measured using the root mean square error (RMSE). A measure of how widely distributed prediction mistakes are is the RMSE. It can be stated this way:

$$\text{RMSE} = \sqrt{(\tfrac{1}{n})\sum_{i=1}^{n}(y_i - x_i)^2}$$

In the above formula, n is the number of exemplars in the data set.

2. MAPE (Mean Absolute Percentage Error) - The average or mean of forecasts' absolute percentage errors is known as the mean absolute percentage error (MAPE). Actual or observed value less predicted value is the definition of error. To calculate MAPE, percentage errors are added without respect to sign.

B. Experimental Results

In this paper, we investigated and attempted to forecast the varying prices of several cryptocurrencies. In addition, we give a clear description of the issue and go through four different sorts of features below. This research can potentially expand the cryptocurrency research field and give investors more resources for evaluating investments. We suggest the novel LSTM model solve the issue of price fluctuation prediction. An embedding network is shown to capture the hidden representations from connected cryptocurrencies, and the LSTM model is employed to capture the time-dependent elements of cryptocurrency price dynamics. Both networks are used in tandem with one another. As a method of long-term forecasting, the created approach was utilized to demonstrate the future changes in the values of the various types of cryptocurrencies over a 180-day period. We experimentally showed the utility of our LSTM model in the real-world bitcoin market. Additionally, LSTM showed state-of-the-art performance that was superior to those of all other existing models.

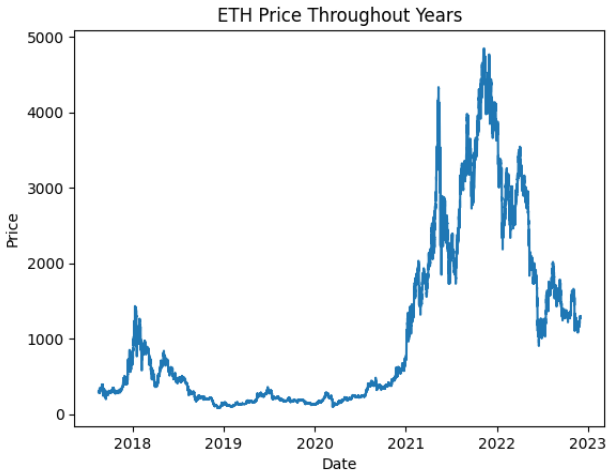


Fig: ETH Price Throughout Years

	Actual	Predicted
0	3729.92	3618.018555
1	3729.12	3609.883301
2	3747.72	3602.238281
3	3733.96	3597.714844
4	3745.54	3593.635254
...	...	...
9074	1563.69	1543.020264
9075	1564.79	1540.242798
9076	1557.71	1538.135010
9077	1565.83	1535.784424
9078	1572.69	1534.450439

9079 rows × 2 columns

Fig: Line Graph of Prediction on Testing Dataset

```
from sklearn.metrics import mean_absolute_error

rmse = mean_squared_error(y_test,predictions_test,squared=False)
mape = mean_absolute_error(y_test,predictions_test)

print(f"rmse: {rmse}")
print(f"mape: {mape}")
```

```
rmse: 0.049191636035688845
mape: 0.03465241977875644
```

Fig: Evaluation Metrics

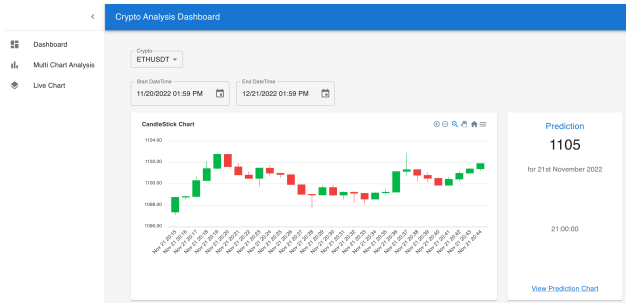


Fig: Candle Stick Chart of Cryptocurrency Data

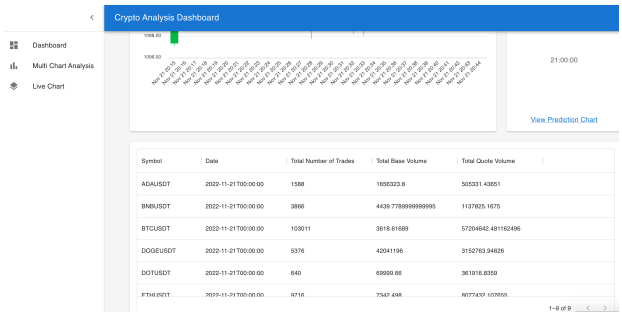


Fig: Crypto Analysis Dashboard (React JS, Material UI)

VII. Conclusion

Our model provides an excellent base model to add further input parameters of the cryptocurrencies, as well as being able to add other hyperparameters and changes to the model layers. Our LSTM approach produces a more accurate RMSE than the ARIMA model at the cost of a longer runtime. The prediction lines had a minimal deviation from the actual recorded close values. Additionally, the model has very high accuracy as the predictions had a minimal root-mean-squared error, meaning the predicted value was close to the actual price. The model must be hyper-parametrized to account for variables such as investor sentiment. The limitations of our project are that we have not explored the architecture of introducing such hyperparameters in a detailed manner. For future research, the model can accept a hyperparameter such as sentiment analysis of tweets from Twitter pages or even simply the measurement of tweet volume.

VIII. References

[1] Bedell, Michael. "Client-Server Architecture." vol. 1, no. 1, 2003, p. 549. Science Direct, https://www.sciencedirect.com/topics/computer-science/client-server-architecture

[2] "Distributed Systems - The Complete Guide." Confluent, 2014, https://www.confluent.io/learn/distributed-systems/. Accessed 22 November 2022.

[3] Kindberg, Tim, et al. Distributed Systems: Concepts and Design. Addison-Wesley, 2012. Accessed 22 November 2022.

[4] Hamayel, Mohammad J., and Amani Yousef Owda. 2021. "A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms" AI 2, no. 4: 477-496. https://doi.org/10.3390/ai2040030

[5] https://www.researchgate.net/publication/330669028

[6] https://www.researchgate.net/publication/361728742

[7] Ammer, Mohammed Abdullah, and Theyazn H. H. Aldhyani. 2022. "Deep Learning Algorithm to Predict Cryptocurrency Fluctuation Prices: Increasing Investment Awareness" Electronics 11, no. 15: 2349. https://doi.org/10.3390/electronics11152349