**Comparative Analysis of Deep Learning Models for Bitcoin Prediction**

**Abstract**

In this project, we explore the efficacy of various machine learning and deep learning models in predicting the bitcoin price in the market, including long short-term memory (LSTM), Gated Recurrent Network model (GRU), and XGBoost to compare them and see which one works better.

**1. Introduction**

The ability to accurately predict stock prices and digital currencies is of paramount importance in financial markets. Various approaches, including mathematical models like the Black-Scholes Merton equation and machine learning techniques such as RL and DL, have been employed for this purpose. In this project, we aim to compare the predictive capabilities of deep learning models using real-world bitcoin data.

**2. Literature Review**

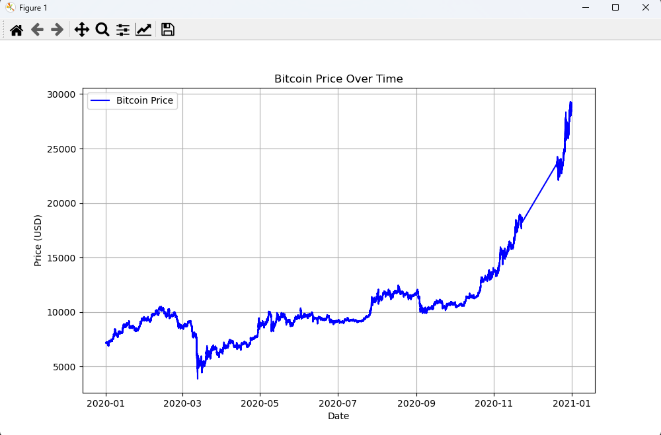
Previous studies on Bitcoin price prediction using deep learning have explored models such as RNNs, LSTMs, and Bayesian neural networks, often outperforming traditional methods like ARIMA. These studies have also incorporated blockchain and macroeconomic data, with varying degrees of success. However, RNN models and diverse combinations of deep learning techniques remain largely unexplored.

**3. Methodology**

**3.1 Data Collection**

We collected historical stock price data from Yahoo Finance using the python library yfinance and from the website Kaggle.com for the previous year of bitcoin price in the market.

Our dataset contains 600,000 instances of bitcoin prices starting from January 2020 to December 2020 year. Our features are date, open, high, low, volume BTC and volume USD. Our target is “close” which is the price at which the bitcoin closes the day.



**3.2 Model Implementation**

3.2.1 long short-term memory (LSTM)

LSTM is the most used for bitcoin prediction practices in finance. LSTM models offer a powerful framework for Bitcoin price prediction, leveraging their ability to capture complex temporal dependencies and nonlinear relationships inherent in cryptocurrency markets.

3.2.2 Gated Recurrent Network model (GRU)

GRU models offer a compelling framework for Bitcoin price prediction, combining efficiency in training, effective handling of short-term dependencies, and scalability to large datasets.

3.2.2 XGBoost

XGBoost excels in capturing complex patterns and relationships within cryptocurrency markets, offering a robust framework for forecasting Bitcoin prices. It effectively handles both linear and nonlinear dependencies, making it suitable for capturing the intricate dynamics of the cryptocurrency market.

**4. Results**

**4.1 Preliminary results**

We started our project by conducting a statistical analysis on the behavior of our target column. We derived the Augmented Dickey-Fuller (ADF) test to see if the ‘close’ column had stationary behavior or non-stationary.

|  |  |
| --- | --- |
| Test | Value |
| ADF Statistic | -6.829 |
| p-value | 1.9144e-09 |

Based on these values we can see that our p-value is extremely small compared to some important significance levels (0.05). These results indicate that our target column follows stationary behavior. This indicates that some algorithms like ARIMA can work good under these conditions.

Besides this, we decided to follow our path to conduct the GRU and LSTM.

The choice of use GRU and LSTM is predicated on each observation in the time series being dependent on the previous observation. Based on this, the ordering of the observations matters, and we are supposing that our target column is not independent.

**4.2 GRU**

For the GRU preprocessing stage we split the train and test on 80% and 20% respectively. Since we are dealing with a time series model, we delete all the columns, and we just leave the time column and our target column. We scale our target column using standardization since we are dealing with univariate time series.

The GRU model consists of a single GRU layer followed by a fully connected (linear) layer. The GRU layer processes the input sequence and outputs hidden states, which are then passed to the fully connected layer to produce the final output. During training, the model was trained for 20 epochs using the Adam optimizer with default parameters. The learning rate was adjusted using a ReduceLROnPlateau scheduler with a patience of 10 epochs and a factor of 0.1.

Once the training was finished, we calculated the mean squared error for the training set and testing set obtaining the following results:

|  |  |
| --- | --- |
| Training Mean Squared Error (MSE) | |
| GRU | 0.0116243959180 |

|  |  |
| --- | --- |
| Test Mean Squared Error (MSE) | |
| GRU | 1.92956876e-05 |

This indicates that the model performed well in both training and testing, with low MSE values indicating good accuracy in predicting the target values as we can appreciate in the following plot.

A green line graph with red and green lines

Description automatically generated

**4.2 LSTM**

For the LSTM we split the data the same way as with GRU, 80% training and 20% test. We created a sequential layer, and we passed 3 LSTM layers with 50 units each followed with a dropout layer with a value of 0.2. During the training, the model was trained for 10 epochs using the Adam optimizer with a batch size of 32.

Once the training was finished, we calculated the MSE as we did in the GRU obtaining the following results:

|  |  |
| --- | --- |
| Test Mean Squared Error (MSE) | |
| LSTM | 2.34e-05 |

We obtained good values as we did in GRU model as we can see in the following image.

**A graph showing the price of bitcoin

Description automatically generated**

**4.3 XGBoost**

For the XGBoost since it is a supervised ML model, we created a target. For the data processing we dropped the symbol and unix columns which are the non-useful columns. We also created a tomorrow column shifting the closing prices to one day before. With the tomorrow column and closing columns we created the target column in which if tomorrow’s value is bigger than the closing value the target value is 1, and 0 otherwise. We split the data in training and test with divisions of 70% and 30% respectively. After the split we initialized the model with parameters n\_estimators = 200, max\_depth = 8, learning\_rate = 0.2, random\_state = 1, stratify = y, and trained the model. We obtained the following results.

|  |  |
| --- | --- |
| Accuracy Score | |
| \XGBoost | 64.5% |

|  |  |
| --- | --- |
| F1 Score | |
| \XGBoost | 56.2% |

This indicates the model didn’t perform as well as the DL models but performed fairly well achieving more than a 50%.

**5. Discussion**

Throughout the course of this project, we encountered various challenges inherent in delving into novel technologies, unfamiliar libraries, and innovative deployment strategies for deep learning models. The integration of GRU, LSTM, and XGBoost models presented a unique set of obstacles, each demanding meticulous attention to detail and innovative problem-solving approaches.

One of the primary challenges we confronted was navigating the intricacies of new technologies essential for implementing our predictive models. From understanding the intricacies of recurrent neural networks (RNNs) for GRU and LSTM models to mastering the nuances of gradient boosting algorithms for XGBoost, our team grappled with the learning curve associated with these cutting-edge technologies. However, through collaborative efforts and a commitment to continuous learning, we successfully overcame these hurdles, leveraging our collective expertise to navigate through uncharted territory.

**6. Conclusion**

In this study, we evaluated the performance of GRU, LSTM, and XGBoost models for predicting Bitcoin prices. Our results indicate that the GRU model outperforms LSTM and XGBoost, showcasing its ability to capture short-term dependencies in cryptocurrency data effectively.

The recurrent nature of GRU networks proves advantageous in discerning subtle fluctuations in Bitcoin prices, highlighting the potential of RNN-based models for cryptocurrency price prediction tasks. While LSTM and XGBoost demonstrate competitive performance, they fall slightly short compared to GRU.

Moving forward, further exploration using techniques like GridSearchCV to optimize hyperparameters could enhance model performance. By leveraging advanced machine learning techniques, we can continue to refine our understanding of cryptocurrency markets and develop robust forecasting models for informed decision-making.

**7. References**

Ji, S., Kim, J., & Im, H. (2019). A comparative study of bitcoin price prediction using deep learning. Mathematics, 7(10), 898.

Awoke, T., Rout, M., Mohanty, L., & Satapathy, S. C. (2020). Bitcoin price prediction and analysis using deep learning models. In Communication Software and Networks: Proceedings of INDIA 2019 (pp. 631-640). Singapore: Springer Singapore.

Jiang, X. (2019). Bitcoin price prediction based on deep learning methods. Journal of Mathematical Finance, 10(1), 132-1