Enhancing Stock Price Prediction with Bitcoin: A Comparative Analysis of Recurrent Neural Networks for Forecasting Microstrategy's Market Performance

Assignment 3: RNNs
Timo Brady
Trimester 3, 2023
Deep Learning Fundamentals

Abstract

This paper investigates the potential of Recurrent Neural Networks (RNNs) in predicting Microstrategy (MSTR) stock prices by incorporating Bitcoin (BTC) fiat-denominated price data. Recognizing the influence of BTC on MSTR, given the company's substantial Bitcoin holdings, various RNN models including a baseline RNN, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and a novel GRU with an attention layer were explored. The models were rigorously evaluated using Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-squared metrics through time series cross-validation and final testing. Results indicate a remarkable improvement in prediction accuracy when including BTC data. The GRU model enhanced with an attention layer and trained with a Huber loss function emerged as the most effective, achieving an MAE of 0.0107, MAPE of 1.997%, and an R-squared value of 0.9504 in the final test. This paper not only highlights the intricate relationship between BTC and stock market dynamics but also demonstrates the efficacy of advanced neural networks in financial forecasting.

1. Introduction

At the onset of the financial crisis of 2008, somewhere in the world, the pseudonymous Satoshi Nakamoto was binding the chaos and the order, the private cryptographic key with the validation of the public ledger. Bitcoin, the timechain, and "proof of work" were born to secure a decentralized network and its digital property, not through the brute force of the bullet or the bomb, but through the pacific, electromagnetic properties of bit-flipping, of converting electricity into cryptographic computation to verify, to validate—all without any trust whatsoever in the network's participants [2].

Years later, Microstrategy (MSTR) CEO Michael Saylor directed his company to begin buying vast sums of bitcoin (BTC) in August of 2020. Because there was and still is no bitcoin exchange-traded fund (ETF) available in the United States, Saylor sought to employ the MSTR stock as a *de facto* bitcoin ETF and to normalize and lead the way on bitcoin-backed financing on a grand scale [5].

Stock price prediction is inherently a tricky business given the innumerable macro and micro external variables that impose upward and downward price pressure. I thought it would be an interesting exercise to not only build a Recurrent Neural Network (RNN) and variations to predict the MSTR stock price, but to also incorporate the fiat-denominated USD price of BTC to see if doing so would enhance the model's learning process and thus show a correlation between the MSTR stock and BTC prices.

2. Data Acquisition, Preparation, and Pre-Processing

I selected the period from July 28, 2020 to November 15, 2023 as the data date range, coinciding with MSTR's announcement to initiate its Bitcoin investment, for both MSTR stock prices and the fiat-denominated USD prices of BTC.

Utilizing the yfinance library, historical data for MSTR and BTC was downloaded and saved in CSV format. The MSTR dataset comprised 832 entries, with stock features such as Open, High, Low, Close, Adjusted Close, and Volume. The BTC dataset was more extensive, with 1205 entries, and formatted similarly to the MSTR dataset. Stocks like MSTR are traded on exchanges that have specific trading days and hours, typically Monday through Friday, excluding holidays. In contrast, BTC is traded on cryptocurrency exchanges that operate 24 hours a day, 7 days a week, including weekends and holidays. This accounts for the gap in the number of data entries between MSTR and BTC. The datasets predominantly contained float64 and int64 data types, with dates represented as objects.

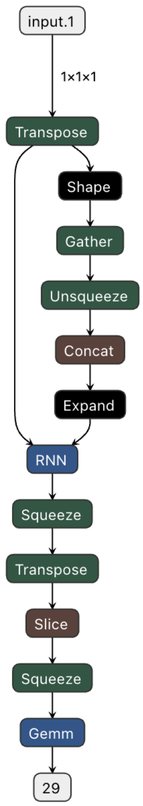

Figure 1. Baseline RNN Model Architecture

Following the Github example of Apple (AAPL) stock price prediction set forth by user rgkannan676 [3], a log transformation was applied to the MSTR and BTC closing prices to stabilize variance and normalize the data. The differencing of the log-transformed data helped in removing any trends or seasonality, making the series more stationary for both. These techniques were applied in order to avoid predicting the prices directly. In terms of normalization, I employed a MinMaxScaler to scale the differenced log values to a range of 0 to 1. The dataset was divided into training, validation, and testing sets (80% training, 10% validation, and 10% testing).

3. Baseline Models

3.1. Architecture and Training Procedure

The baseline model for training with both the MSTR dataset alone (Training A) and the MSTR and BTC combined datasets (Training B) was constructed with a single-layer RNN with 50 hidden units and a fully connected layer (Figure 1). The model was trained over 50 epochs using the Adam optimizer with a Mean Squared Error (MSE) loss function and a Cosine Annealing Learning Rate Scheduler.

Training and validation losses were monitored across epochs, showing a consistent decrease in both Training A and B. The losses were visualized to verify the model's convergence and performance stability (Figures 2 and 3). For both Training A and B, the model's predictions were compared against actual MSTR stock prices in the validation set. An inverse transformation was applied to the scaled predictions and actual prices to bring them back to the original scale.

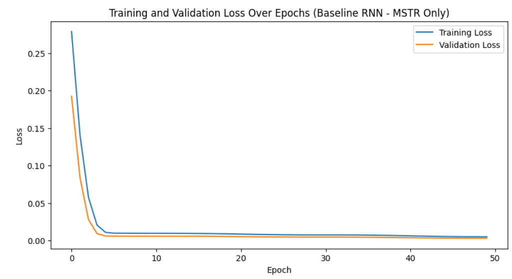

Figure 2. Training and Validation Loss Over Epochs of Training A (Baseline RNN - MSTR Only)

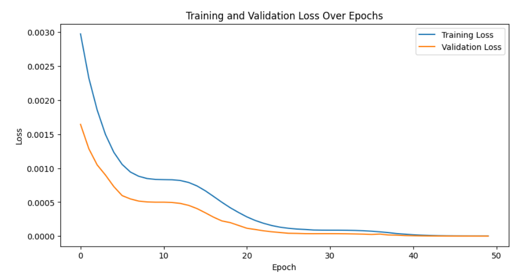

Figure 3. Training and Validation Loss Over Epochs of Training B (Baseline RNN - MSTR with BTC)

3.2. Baseline Model Results

The inclusion of Bitcoin (BTC) prices in Training B dramatically improved the model's predictions (Table 1). This is evident from the significantly lower Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) and the higher R-squared value (see equations 1, 2, and 3). It suggests that BTC prices have a strong correlation with MSTR stock prices. With just the MSTR data in Training A, the model's performance is moderate, indicated by an R-squared of 0.62. However, when BTC data is included in Training B, the R-squared jumps to an almost perfect score of 0.9999, indicating that nearly all the variance in MSTR stock prices is explained by the model. MAE and MAPE decrease drastically when including BTC data, indicating much more accurate predictions. The MAPE value of about 0.154% with BTC data is particularly promising, showing very high precision.

Metric	Training A	Training B
MAE	0.042	0.0008
MAPE	8.12%	0.15%
R-squared	0.62	0.999

Table 1. Baseline Model Results for Training A (MSTR only) and B (MSTR with BTC)

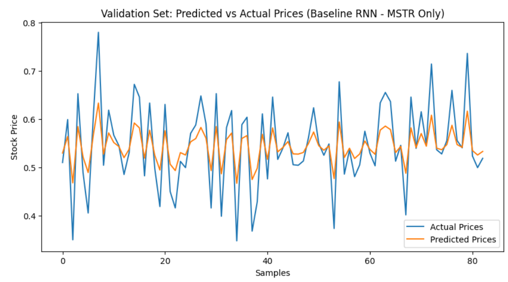

Figure 4. Validation Set: Predicted vs Actual Prices of Training A (Baseline RNN - MSTR Only)

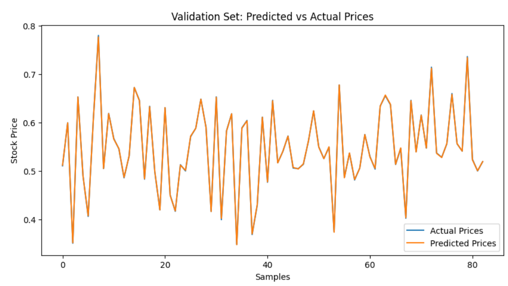

Figure 5. Validation Set: Predicted vs Actual Prices of Training B (Baseline RNN - MSTR with BTC)

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

$$\text{MAPE} = \left(\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|\right) \times 100\% \tag{2}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3}$$

This experiment highlights the importance of external factors in stock price models. BTC's influence on MSTR stock prices is a specific example, but it broadly suggests the significance of considering relevant external economic or financial indicators in machine learning. When comparing the graphs of predicted vs. actual MSTR stock prices on the validation set, I was concerned with potential overfitting of Training B given its overtly high precision as evidenced in the graph (Figures 4 and 5).

Metric	AAPL Only	with BTC
MAE	0.002	0.03
MAPE	4.25%	5.54%
R-squared	0.79	0.68

Table 2. Baseline Model Results for AAPL and AAPL with BTC

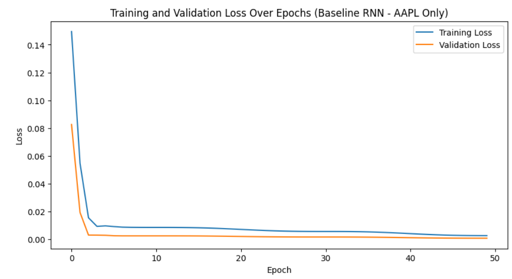

Figure 6. Training and Validation Loss Over Epochs of Baseline RNN - AAPL Only

3.3. Baseline Verification with Apple Stock

In order to conduct a sort of sanity check on the baseline Training B model with the BTC dataset included, I conducted the same experiments with Apple (AAPL) stock alone and combined with the BTC dataset for the same time period as the earlier baseline models (July 28, 2020 to November 15, 2023). My intention was to ensure that the model improvements with including the BTC dataset in the baseline was not a fluke, and that the BTC price action really did have bearing on the predictability of the MSTR stock price.

In summary, for the AAPL-only training, the model performs better without the BTC dataset. This is evident from the lower MAE and MAPE and higher R-squared for the AAPL-only model. This suggests that the inclusion of BTC data does not add predictive value for AAPL stock prices, which aligns with the expectation as Apple is not known to have BTC in its treasury, according to the publicly available financial documents [1]. See Table 2 for metrics and Figures 6, 7, 8, and 9 for the loss curves and predicted stock graphs. The inclusion of the irrelevant BTC dataset with the AAPL stock dataset has ultimately degraded the model's performance. This further confirms the relevance of the combining of the MSTR and BTC datasets.

4. LSTM and GRU Experiments

4.1. Architecture and Training Procedure

Two more advanced RNNs architectures, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), were implemented as experiments to compare with the baseline models. Both models were constructed with a sim-

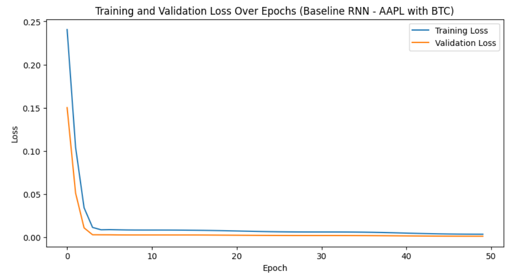

Figure 7. Training and Validation Loss Over Epochs of Baseline RNN - AAPL with BTC

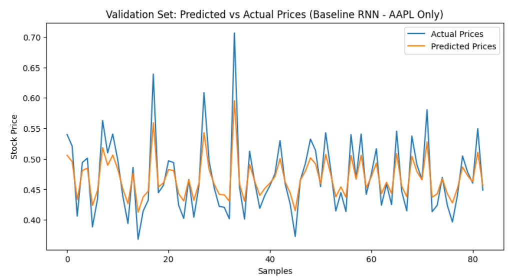

Figure 8. Validation Set: Predicted vs Actual Prices of Baseline RNN - AAPL Only

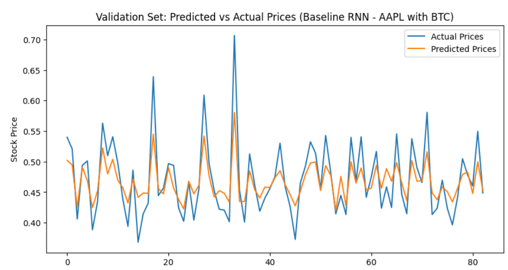

Figure 9. Validation Set: Predicted vs Actual Prices of Baseline RNN - AAPL with BTC

ilar configuration: an input layer, a hidden layer with 50 units, and a fully connected output layer (Figures 10 and 11). As with the baseline models, and to keep the experiments consistent, a training loop was employed for both models at 50 epochs. The Adam optimizer with a learning rate of 0.001 and a Cosine Annealing Learning Rate Scheduler were used. The Mean Squared Error (MSE) loss function was the criterion.

The combined MSTR and BTC datasets were used. My thinking was to see how well these models would perform in terms of the potential overfitting observed in the baseline Training B model. Both experimental models exhibited a decrease in training and validation losses over epochs as with the baseline Training A and B models. Loss plots provided visual confirmation of the models' learning stability

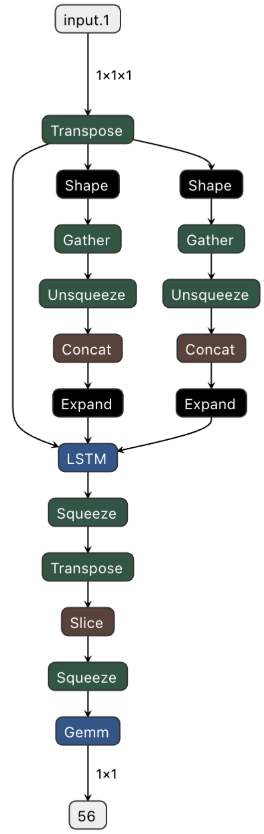

Figure 10. LSTM Model Architecture

and convergence (Figures 12 and 13).

To maintain consistency in the experiments, predictions were generated on the validation set, and an inverse transformation was applied to scale back to the original MSTR stock price range, as with the baseline models.

4.2. LSTM and GRU Results

The Baseline RNN model, when enhanced with BTC data, demonstrates significantly superior performance over the LSTM and GRU models in all metrics (Table 3). The high precision (low MAE and MAPE) and nearly perfect R-squared value suggest an extremely accurate fit to the data. However, as mentioned previously, this level of precision raises concerns about overfitting. Both the LSTM and GRU models show higher error rates and lower R-squared values compared to the Baseline RNN with BTC. The LSTM and GRU models, though less precise, might offer a more balanced and realistic performance, particularly on unseen data. The graphs of the predicted vs. actual MSTR stock prices for both models are in Figures 14 and 15.

This analysis indicates a need for further experiments, involving adjusting the loss function and subjecting the models to cross-validation in order to assess the true gen-

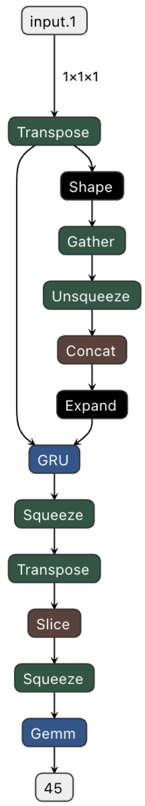

Figure 11. GRU Model Architecture

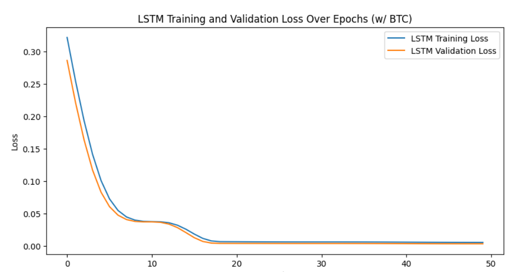

Figure 12. LSTM Training and Validation Loss Over Epochs (w/ BTC)

Metric	LSTM	GRU	Training B
MAE	0.046	0.00	0.0008
MAPE	8.91%	7.5%	0.15%
R-squared	0.54	0.67	0.999

Table 3. Experimental Model Results for LSTM and GRU with BTC)

eralizability of these models.

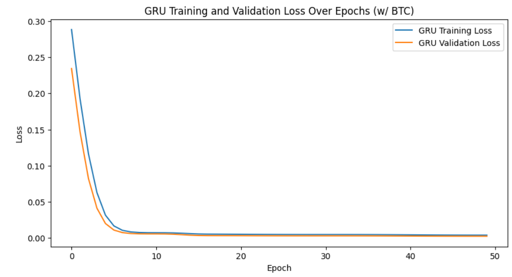

Figure 13. GRU Training and Validation Loss Over Epochs (w/ BTC)

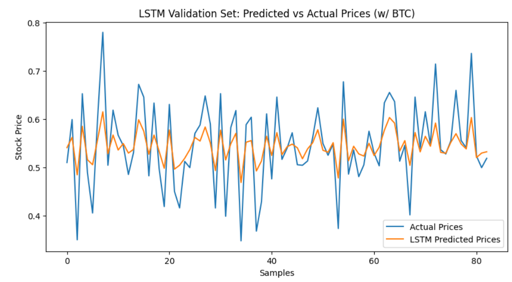

Figure 14. Validation Set: Predicted vs Actual Prices of LSTM Experiment (with BTC)

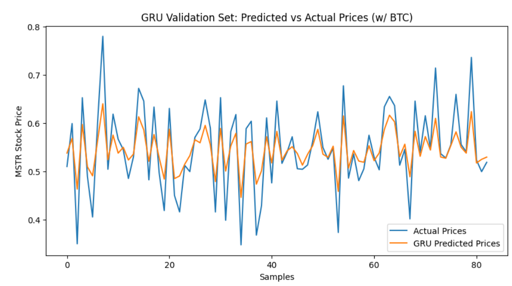

Figure 15. Validation Set: Predicted vs Actual Prices of GRU Experiment (with BTC)

4.3. Further Experiments: Huber Loss Function

Now, given that MSE is more sensitive to outliers (in our case this correlates to volatility, especially in the fiat-denominated USD price of BTC), I experimented with running both the Baseline RNN and GRU models with the loss function set to Huber. The Huber loss function is less sensitive to outliers [4] (Equation 4).

$$\text{Huber}(a, \delta) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \delta, \\ \delta(|a| - \frac{1}{2}\delta) & \text{otherwise.} \end{cases} \quad (4)$$

The results are telling (Figure 4). The RNN and GRU models with the Huber loss function show higher MAE and MAPE for both models compared to the MSE loss function. This suggests that the extreme precision indicated by MSE

Metric	RNN	GRU
MAE	0.0140	0.0099
MAPE	2.67%	1.89%
R-squared	0.9617	0.9814

Table 4. RNN and GRU Model Results with Huber Loss Function (MSTR and BTC)

Metric	MSE	Huber
MAE	0.062	0.064
R-squared	0.464	0.379

Table 5. Time Series Cross-Validation for Baseline RNN (MSTR and BTC)

might be overly optimistic, likely due to its high sensitivity to outliers, and indicative of overfitting.

With the Huber loss function, the GRU model outperforms the Baseline RNN across all metrics, indicating it might be more robust to outliers and noise in the data. In contrast with the MSE loss function, the GRU model's performance drops significantly. The Baseline RNN shows exceptionally high performance with MSE, as discussed in the previous section, but when using Huber loss, the performance metrics become more conservative. This indicates that, while the Baseline RNN may perform well on average, it might not handle outliers as effectively as the GRU model. Because handling outliers and more extreme market conditions is crucial, particularly in the case of BTC, the GRU model with Huber loss merits further exploration in the sections below.

5. Time Series Cross-Validation

I subjected the Baseline RNN and GRU models with both MSE and Huber loss functions to time series cross-validation (with n splits = 10), and the results are telling (Figures 5 and 6). I should note the extremely high MAPE values across all models and loss functions. My assumption is that because the time series cross-validation is utilizing smaller slices of time, significant volatility and fluctuations in BTC in one fold (i.e. in that particular slice of time) can skew the average MAPE across all folds, which would account for the large MAPE errors. Therefore, I have deprioritized MAPE as a metric for evaluating model performance of the time series cross-validations.

Both models show a noticeable difference in MAE when using Huber loss compared to MSE due to Huber loss's robustness to outliers. The GRU model, in particular, shows a higher MAE with Huber loss than MSE, suggesting it may be more sensitive to the Huber loss function's handling of outliers. The R-squared values are significantly lower when

Metric	MSE	Huber
MAE	0.086	0.082
R-squared	-0.067	0.05

Table 6. Time Series Cross-Validation for GRU (MSTR and BTC)

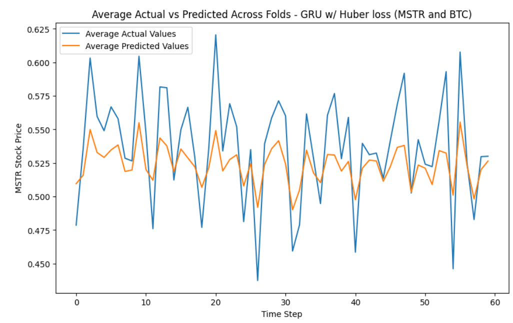

Figure 16. Average Actual vs Predicted Across Folds - GRU w/ Huber loss (MSTR and BTC)

using Huber loss compared to MSE, suggesting that while Huber loss provides a more robust error handling, it may capture less variance in the data compared to MSE.

Based on the above analysis, the GRU model with Huber loss appears to be the most promising candidate for further fine-tuning and final testing. The GRU model with Huber loss has shown a good balance between MAE and robustness to outliers. Although the GRU model with Huber loss has a slightly higher MAE compared to its performance with MSE, this could be indicative of better generalization and mitigation against overfitting.

Now, although the R-squared values are lower with Huber loss, this is expected as Huber loss focuses more on robustness rather than capturing the maximum variance in the data. In my view, a slightly lower R-squared value is a reasonable trade-off for a model that generalizes better and is less sensitive to outliers. This is all the more evident in a graph of the average actual vs. predicted MSTR stock prices across all 10 folds (Figure 16).

Based on the graph, the predicted values appear to be generally close to the actual values, suggesting that the model is capturing the overall trend of the data reasonably. However, the model appears not to capture the peaks and troughs with very much precision. The model does show consistency across different time steps, an indicator of its stable performance.

6. Final Experiment: Attention Layer

I endeavored to build upon the incremental success of the GRU with Huber loss, targeting an improvement in the model's capability to capture critical points in the MSTR stock price volatility, observed in the peaks and troughs ear-

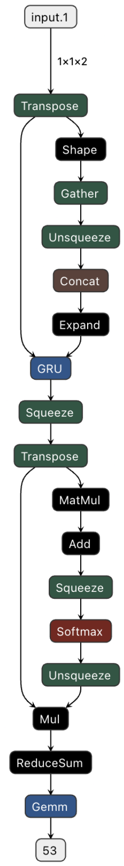

Figure 17. GRU with Attention Layer Model Architecture

lier. Adding an attention layer allows the model to focus on the sorts of features across important time steps to reduce the earlier smoothing effect and therefore increase the model's accuracy and predictive power. The architecture remains largely the same except for the addition of the attention layer (Figure 17).

Observable in the graph (Figure 18), the predicted values are quite close to the actual values as in the GRU model without the attention layer. This closeness is a good indicator of the model's overall accuracy. While the model still appears to smooth out some of the extremes, those peaks and troughs, the addition of the attention mechanism appears to help it to better capture significant changes in the data compared to the previous GRU model without the attention layer.

The GRU model with attention has a slightly lower MAE compared to the one without the attention layer, indicating the the model with the attention layer has smaller predictive errors (Table 7). There is a significant difference in R-squared values, with the model with the attention layer showing a substantially higher value, indicating that this new model is much better at capturing the volatility in the MSTR stock prices compared to the model without the attention layer.

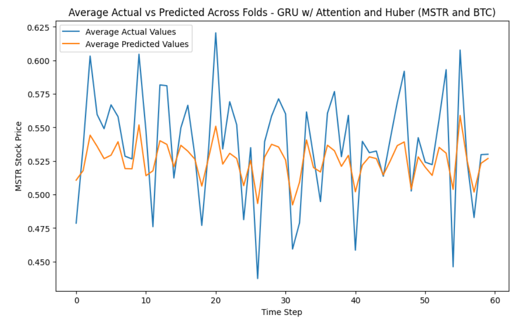

Figure 18. Average Actual vs Predicted Across Folds - GRU w/ Attention Layer (MSTR and BTC)

Metric	Attention	No Attention
MAE	0.0672	0.082
R-squared	0.4054	0.05

Table 7. Time Series Cross-Validation for GRU w/ Attention Layer and Huber Loss (MSTR and BTC)

tention layer. Therefore, the GRU model with the attention layer and Huber loss function will be subjected to final testing.

7. Final Model Testing

7.1. Results

The GRU model with the added attention layer and the Huber loss function was subjected to the test set, and the results are impressive (Table 8 and Figure 19). The final test MAE is significantly lower than the average MAE from cross-validation. The test MAPE shows an improvement from the previous model trainings. The R-squared value during testing is substantially higher than the cross-validation average, indicating that the model does indeed explain a large proportion of the variance in the test data.

It appears that the process of cross-validation and subsequent adjustments have led to a better-tuned model for the test set. It is clear that the GRU model with the attention layer, and when trained with the Huber loss function, shows exceptional performance in predicting MSTR stock prices on the test set, significantly outperforming the cross-validation phase.

7.2. Conclusion

This paper has demonstrated the significant impact of incorporating Bitcoin prices in predicting the stock prices of a company heavily invested in Bitcoin, like Microstrategy. The baseline RNN model, when supplemented with BTC data, showed an almost perfect R-squared value, suggesting a strong correlation between MSTR stock prices and BTC.

Metric	Test Set	Cross-Validation
MAE	0.010706	0.0672
MAPE	1.997%	NA
R-squared	0.9504	0.4054

Table 8. GRU w/ Attention Layer and Huber Loss on Test Set (MSTR and BTC)

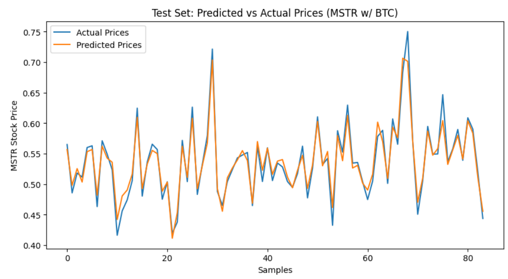

Figure 19. GRU w/ Attention Layer and Huber Loss on Test Set (MSTR and BTC)

The introduction of the Huber loss function provided a more balanced performance in the GRU model compared to the baseline RNN, making it a superior choice for handling the inherent volatility of BTC. The addition of an attention layer to the GRU model further refined its ability to capture critical shifts in stock price movements, achieving impressive results in the final testing phase.

In conclusion, this paper highlights the importance of selecting appropriate data, neural network architecture, and loss functions in the domain of financial predictions. The findings emphasize the interconnectedness of cryptocurrency markets with traditional financial assets and underscore the potential of more nuanced machine learning techniques in financial modeling.

8. Code

The supporting code for this paper can be accessed at the following private respository: https://github.com/bluebindu/DeepLearningAssignments/blob/main/DeepLearningAssignment3.ipynb. Access has been granted to the following individuals: jinan.zou@adelaide.edu.au, haiyao.cao@adelaide.edu.au, and jiaxin.wang01@student.adelaide.edu.au.

References

[1] Inc. Apple. Sec filings. https://investor.apple.com/sec-filings/default.aspx. Accessed: November 2023. 3

[2] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf, 2008. Accessed: November 2023. 1

[3] rgkannan676. Fit-and-predict-apple-stock-prices. https://github.com/rgkannan676/Fit-and-predict-Apple-stock-prices/, 2021. Accessed: November 2023. 2

[4] scikit-learn Development Team. sklearn.linear_model.HuberRegressor Documentation. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.HuberRegressor.html. Accessed: November 2023. 5

[5] MacKenzie Sigalos. Michael saylor says investing in microstrategy is the closest you'll get to a bitcoin spot etf. https://www.cnbc.com/2022/04/21/bitcoin-spot-etf-michael-saylor-says-investing-in-microstrategy-is-closest-youll-get.html, 2022. Accessed: November 2023. 1