High-Frequency Prediction of Bitcoin Using Hybrid Convolutional and Long Short-Term Memory Networks

Jake Warby
School of Mathematics and Statistics
University of New South Wales
NSW, Australia
z5259166@ad.unsw.edu.au

***Abstract*—Cryptocurrencies have gained significant traction in recent years, with the major currencies, such as Bitcoin and Ethereum, consistently hitting all time highs. This growth has led to increased interest in the asset class, eliciting the question of whether the price movements can be predicted and subsequently exploited for profitable trading opportunities. This paper proposes a hybrid *convolutional neural network and long short-term memory* (CNN-LTSM) model to classify the next period in high-frequency Bitcoin time series data, leveraging correlations between the major currencies as a main proponent. This CNN-LSTM is also compared to CNN and LSTM models, providing prediction and simulated trading results for each type of model. Extensive empirical testing is performed using high frequency tick data of different tick lengths, collected from October 2017 to October 2021, for Bitcoin. The analysis present in this paper exhibits that signals can be extracted from the high frequency currency data, which can be subsequently used to accrue more profit over the test period than the simple buy and hold strategy.**

***Index Terms*—Bitcoin, Ethereum, Prediction, Trading, Convolutions, Long Short-Term Memory, Neural Network, Classification**

I. Introduction

Cryptocurrency is a relatively new asset class, with the first and most popular, Bitcoin, being first publicly traded in 2009 at $0.0008 [1]. These currencies are completely virtual and based on decentralised blockchain technology, with some believing it has the potential to rival traditional, fiat currency in the future as the favoured form of virtual transaction [2]. Recently, cryptocurrencies have been garnering interest from retail and institutional investors due to their history of extremely high volatility and positive returns, notably growing 112% in the past year [3].

With the popularisation of contemporary machine learning techniques, beating the market and generating alpha has been increasingly difficult [4], particularly as forecasting the future based on past time movements goes against the efficient market hypothesis [5]. Therefore, in order to be profitable emerging technologies must be used. Arguably, the most popular techniques for time series forecasting as of late have been neural networks, particularly due to their highly non-linear nature being able to learn extremely complex patterns in noisy data better than traditional machine learning techniques [6].

Initially *recurrent neural networks* (RNN) were used for forecasting [7], [8], due to their inherent ability to handle temporal data [9], [10]. However, RNNs have extremely short short-term memory, therefore *long short-term memory* networks [11] were later utilised with great success in time series prediction [12]–[15]. Exciting developments in *convolutional neural networks* (CNNs) in the image recognition space, such as GoogleNet and ResNet [16], [17], resulted in the technology being considered for time series feature extraction and forecasting with success [18], [19]. The natural development was to bring these two methodologies together, resulting in the CNN-LTSM hybrid model [20], [21].

For Bitcoin and other cryptocurrencies in particular, it has been established by Bouri et al. that there are potential trading opportunities within intraday returns [22], where functional time series analysis was considered and basic trading strategies formed. Following this Lahmiri and Bekiros were able to show that LTSM models could effectively learn the chaotic and non-linear patterns in cryptocurrency markets, across all major coins [23]. Li and Dai performed daily prediction using 2D CNN-LTSM for both classification and regression tasks with success [24]. Qiang and Shen also utilised CNN-LTSM for Bitcoin future movement classification [25], however this was done using 1D filters rather than 2D, as well as predicting for each minute rather than daily.

High-frequency analysis, such as within [6], [26], presents larger amounts of data points for a set period. This allows more predictions and therefore more opportunities to trade profitably. However, the accuracy of high-frequency predictions are often lower than low frequency predictions, such as daily, [13], likely due to the extra noisiness attributed to low tick lengths [22]. Once the tick length is chosen, the next decision is between regression and classification. While regression can lead to impressive looking plots and relatively low RMSE, it often becomes a lagging indicator which could result in ill-advised investment recommendations [19]. Therefore, when considering the goal of creating a trading strategy, classification is often used, with trading being done based on the directional prediction of the future time periods [6], [25].

This paper will explore the extreme market conditions that bitcoin continues to endure [3]. The majority of papers mentioned previously are written in early 2021 or earlier, and therefore do not consider the recent price surges and volatility. The question of how models trained on typical price data can handle contemporary, extreme price action will therefore be explored and the paper will hopefully contribute to further study of market predictability during perceived uncertainty. The paper will also contribute to study of deep learning based classification and translating predictions to profit, which is often disregarded when regression models are used [14], [19]. This paper will also add to the growing library of cryptocurrency analysis, which is lacklustre when compared to other traditional assets such as stocks and typical commodities due to the age of the asset class.

Within this paper, CNN and LSTM models are used separately and within a hybrid CNN-LSTM model to predict the next period Bitcoin price movement at a high frequency. The predictive ability of these models will be considered for different tick lengths ranging from 5 to 15 minutes and compared by their accuracy on unseen data. The possibility of converting predictions to profit will also be considered through trading simulations, where two simple trading strategies will be considered. The three models will be compared through their accuracy and their trading performance to determine whether each is suitable for use in the market, as well as whether the hybrid model tends to perform better than the singular models.

The remainder of this paper will be organised as follows. Section 2 will focus on the models and methodology. Section 3 will include the experimental set-up and empirical results from the experiments. Following from this will be Section 4, where trading simulations will be run and discussed. Section 5 will discuss the results and explain areas of improvement and future research, with Section 6 concluding the main findings of report and ideal future directions.

II. Methodology

Bitcoin and Ethereum time series data will be reconstructed using Takens' Theorem. This is useful as it allows creation of many training batches that can be utilised for learning, particularly when using temporal deep learning machines, such as LSTM, where time dependencies are lost as sequence size increases. Increasing the sampling size through subsampling, such as through Takens' theorem, also works to reduce forecast error through a decrease in variance. This data will then be used by a range of models for a one step ahead classification.

A. Takens' Theorem

Takens' theorem is a delay embedding system that can reconstruct a time series while maintaining information about underlying dynamics [27].

by temporal deep learning machines, such as LSTM.

For a given time series $x(t)$, through Takens' theorem we can embed a phase space as

$$Y(t) = [x(t), x(t-T), ..., x(t-(D-1)T)]$$

where D is the embedding dimension, T is the delay time, $t = 0, 1, 2, ..., N-DT-1$, and N is the length of the original time series.

The values for D and T are problem specific and typically chosen through experimentation [28].

B. LSTM

LSTM networks were developed to address training instabilities (vanishing gradients) as well as short term memory problems existing in RNNs [10].

Due to these improvements, LSTM networks are able to detect longer term dependencies within data, as well as converge faster and more consistently.

Within the LSTM network, there are four gates and two states (long and short-term) that are passed through to the next cell. The LSTM network is examined in Figure 1 by Géron [29].

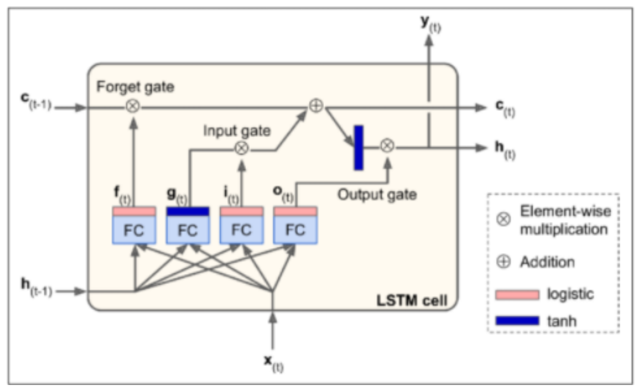

Fig. 1. LSTM Network, Géron [29]

The main gate does the majority of the analysis of the inputs $\mathbf{x}_t$ and the previous short-term state.

$$g_t = \tanh(x_t U^g + h_{t-1} W^g)$$

The input gate decides what information from g_t should be added to the long term state.

$$i_t = \sigma(x_t U^i + h_{t-1} W^i)$$

The forget gate decides what information in the long term state should be forgotten

$$f_t = \sigma(x_t U^f + h_{t-1} W^f)$$

The output gate decides what information should be output at the current time as well as what information is maintained in the long-term state.

$$o_t = \sigma(x_t U^o + h_{t-1} W^o)$$

The short-term state is passed through the cell through time and contains short-term information.

$$c_t = \sigma(f_t * c_{t-1} + i_t * g_t)$$

The long-term state is passed through the cell through time and contains long-term information.

$$h_t = \tanh(C_t) * o_t$$

Note that W and U are the weight matrices connected to the long-term state and input features respectively. c_t and h_t are denoted as the short-term and long-term states respectively.

C. CNN

CNN is a neural network architecture inspired by the brain's visual cortex. CNNs are built through convolutional layers, where the network learns useful filters for feature information extraction. These filters are created by analysing the input image through receptive fields, where filters are learned by finding the most important information found within the receptive field. The feature information is then passed through the filter, and the filtered information is passed through to the next layer. Stacking these filters allows the network to create complex filter patterns, which are used to extract the most important information present in the features. This is visualised in Figure 2 by Géron [29].

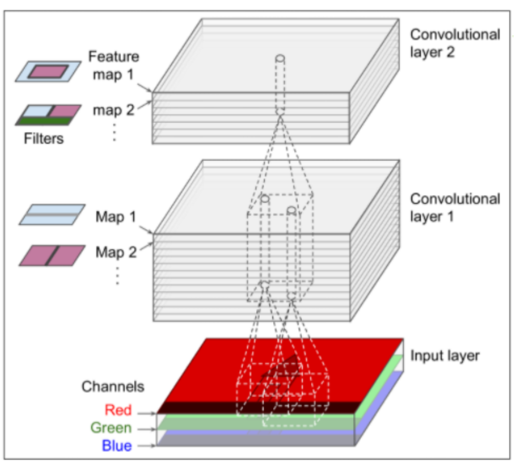

Fig. 2. Stacked Feature Maps, Géron [29]

These filters can also be used to decrease the dimensionality of an image, through the padding 'valid', or can keep the feature space the same size through the padding 'same'. The filters can be further simplified through pooling layers which work similarly to convolutional layers, however it aggregates the values in the kernel through a function such as max or mean.

While these networks were created and popularised through image processing [15], [16], their powerful feature extraction mechanism has inspired uses outside of this field, such as within time series feature extraction [17], [18].

D. CNN-LSTM

The hybrid CNN-LSTM architecture combines the feature extraction capabilities of CNN with the time series processing abilities of LSTM. This has been recently applied to many time series investigation tasks [18], [19], [23]. A simple CNN-LSTM architecture can be seen in Figure 3.

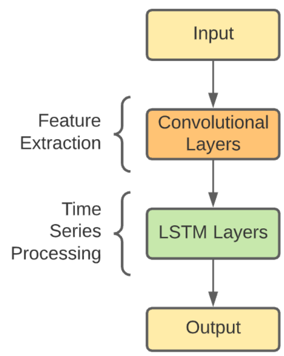

Fig. 3. CNN-LSTM Architecture

III. Results

This section is dedicated to the predictive results and detailed architecture and parameters of each model.

A. Experimental Design

CNN, LSTM and hybrid CNN-LSTM models will be used for classification of Bitcoin's next tick movement over a range of high frequency tick lengths. The experimental steps are summarised:

- Pre-process the data by creating relevant features from the pricing data.
- Perform trial experiments with validation data to determine appropriate topology and related parameters for each model.
- Evaluate the accuracy of each model (LSTM, CNN and CNN-LSTM).
- Run trading simulations using these predictions and report the profit.
- Determine the most accurate and most profitable strategies for each tick length, as well as the best overall time series set up (model and tick length).

The data is one minute resolution cryptocurrency price data (open, close, high, low, volume) for various coins collected and compiled from the Bitfinex exchange by Klein [29]. Many features were created from this data, some inspired by Wu et al. [20], with the following highlighted features being the best performing in trial runs and therefore were chosen for experimentation. Ethereum price and volume changes were chosen as features as they are highly correlated to Bitcoin price change, with the price correlations of three large coins seen in Figure 4.

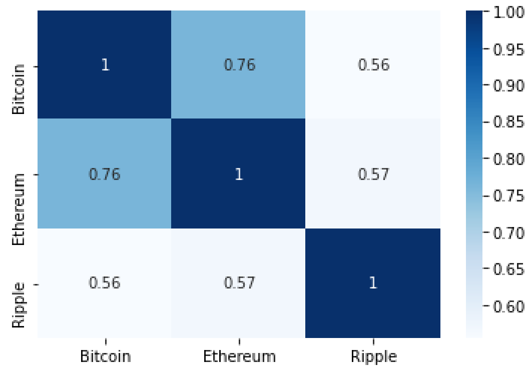

Fig. 4. Coin Price Correlation

Fig. 5. Data Splits

Dataset	Up	Down
Train	0.488	0.512
Valid	0.489	0.511
Test	0.497	0.503

TABLE I
T = 5 PROPORTIONS

- Percent change in price - Bitcoin (variable to be predicted)

$$c_t = \frac{\text{close}_t - \text{close}_{t-1}}{\text{close}_{t-1}}$$

- Percent change in volume - Bitcoin

$$v_t = \frac{\text{vol}_t - \text{vol}_{t-1}}{\text{vol}_{t-1}}$$

- Percent change in price - Ethereum
- Percent change in volume - Ethereum
- High Spread - Bitcoin

$$h_t = \frac{\text{high}_t - \text{open}_t}{\text{open}_t}$$

- Low Spread - Bitcoin

$$l_t = \frac{\text{low}_t - \text{open}_t}{\text{open}_t}$$

- Volatility of last 10 periods - Bitcoin

$$\sqrt{\frac{\sum_{t=1}^{10}(c_t - \bar{c_t})^2}{9}}$$

The transformed time series data was then reconstructed through the use of Takens' Theorem [26], with the embedding dimension $D = 8$ and delay time $T = 1$, as chosen through trial runs.
The time series data is split into a train, validation and test set to determine how the models would perform in the contemporary, extremely volatile market. The validation set was used for trial runs to determine the optimal topology of the neural networks and other various parameters, whereas the test set was used for predictive and trading performance evaluation. The splits can be seen in Figure 5, where the blue data is for training, orange for validation and green for test. The relative proportion of price movements being exhibited in Table 1 (5 min), Table 2 (10 min), Table 3 (15 min). Note that proportion of movements will vary based on tick length.

All of the proceeding models have had their respective hyper-parameters and topologies chosen from trial runs on the validation set. All models were optimized with the Adam optimizer [30] with a learning rate of 0.001, using the binary cross entropy loss measure, seen in Equation 1. Note that y_{ij} is the true result and $\hat{y}_{ij}$ is the model's predicted probability, where this is summed over each class for each observation. Softmax is used to create class probability outputs, which is equivalent to a sigmoid activation in this 2 class setting.

$$L = -\frac{1}{N}\sum_{i}^{N}\sum_{j}^{M} y_{ij}\log(\hat{y}_{ij}) \tag{1}$$

B. Model Set-Up

1) CNN: 6 2D convolution layers are connected to a dense layer with a 0.5 dropout. The respective hyper-parameters specified in Figure 6. For the convolutional layers, the hyper-parameters are, in order, amount of filters, convolutional kernel, padding. Note that no max-pooling is used as it had no material impact on the performance. This is likely due to the fact that the network isn't very deep [23].

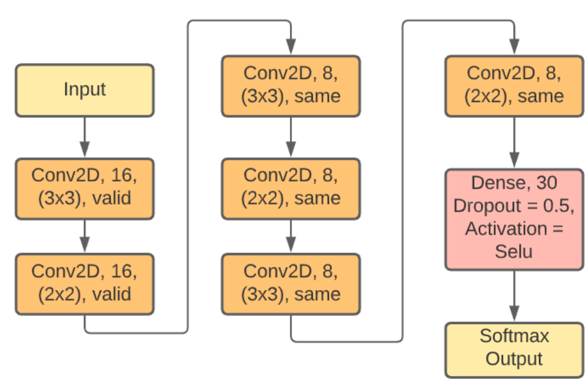

Fig. 6. CNN Topology

Dataset	Up	Down
Train	0.498	0.502
Valid	0.491	0.509
Test	0.498	0.502

TABLE II
T = 10 PROPORTIONS

Dataset	Up	Down
Train	0.502	0.498
Valid	0.498	0.502
Test	0.500	0.500

TABLE III
T = 15 PROPORTIONS

2) LSTM: 2 subsequent LSTM layers with 30 neurons are connected to a dense layer with 30 neurons, 0.5 dropout and selu activation function. This can be seen in Figure 7.

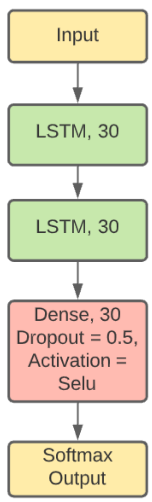

Fig. 7. LSTM Topology

3) CNN-LSTM: The CNN-LSTM model takes inspiration from Qiang and Shen [23], where 2D convolutional layers are used to create horizontal 1D filters. This allows convolution only on the feature axis, leaving all temporal activity to the LSTM layer connected at the end. The total architecture can be seen in Figure 8, with parameters specified in the same format as previously.

C. Predictive Performance

For these experiments, the models were trained for 5 epochs for 5 experiments with different random weight initialization each experiment. This was done for each model and each tick length. Accuracy was assessed for each model on the test data, defined in section 3, and are recorded numerically in Table 4 and displayed visually in Figure 9.

As seen in the results, the LSTM model consistently performs the best. The CNN-LSTM performed better than the CNN model for tick lengths of 5 and 15 minutes convincingly, however for 10 minute ticks the CNN model performed

T	Model	Av. Accuracy
5	CNN	0.522 (0.0021)
	LSTM	**0.529 (0.0009)**
	CNN-LSTM	0.524 (0.0020)
10	CNN	0.519 (0.0015)
	LSTM	**0.523 (0.0015)**
	CNN-LSTM	0.518 (0.0012)
15	CNN	0.520 (0.0025)
	LSTM	**0.525 (0.0012)**
	CNN-LSTM	0.521 (0.0013)

TABLE IV
ACCURACY

marginally better. Overall, the tick length of 5 minutes was the most accurate for all models, followed by 15 minute and 10 minute. Therefore, if choosing the most accurate model, LSTM would be used on a 5 minute time frame.

IV. TRADING SIMULATION

As per the motivation of this paper, the potential trading profitability of each model is assessed. There are 2 strategies employed based on the predicted next period movement. The first strategy is to go long or short 1 unit worth of the coin each period, selling at the end of the period. The results of this strategy are seen numerically in Table 5 and visually in Figure 10. This strategy will be compared to the benchmark where 1 unit is bought at the start of each period and sold at the end. The second strategy invests the entire accumulated amount into the next position, selling at the end of each period, effectively compounding the returns. The results of this strategy are seen numerically in Table 6 and visually in Figure 11. The trading performance of this strategy is compared to the benchmark strategy of buying and holding the coin.

T	Model	Av. No Compound Return
5	CNN	1.005 (0.14)
	LSTM	**1.453 (0.20)**
	CNN-LSTM	1.083 (0.20)
	Long Only	0.691 (0)
10	CNN	0.708 (0.11)
	LSTM	**0.732 (0.07)**
	CNN-LSTM	0.501 (0.18)
	Long Only	0.691 (0)
15	CNN	0.504 (0.17)
	LSTM	0.591 (0.15)
	CNN-LSTM	0.327 (0.21)
	Long Only	**0.691 (0)**

TABLE V
NO COMPOUND RETURN

From the results, it is obvious that compounding returns is the superior strategy. When considering the 3 models, the lowest compound return for 5 minute ticks was 2.568 while the highest no compound return was 1.453. This trend is true for all time periods.

Between the 3 models examined in the report, the LSTM has the highest returns, for all time periods and strategies. This could be guessed due to the superior performance in accuracy, however this is not always true. While the CNN accuracy was lower than the CNN-LSTM for 15 minute tick lengths, it

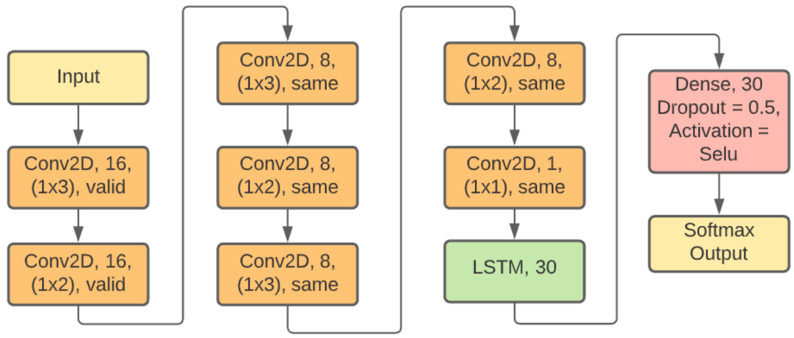

Fig. 8. CNN-LSTM Topology

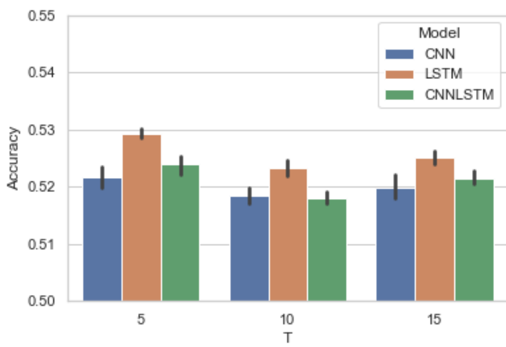

Fig. 9. Accuracy

T	Model	Av. Compound Return
5	CNN	2.568 (0.37)
	LSTM	**4.058 (0.80)**
	CNN-LSTM	2.806 (0.56)
	Buy and Hold	1.860 (0)
10	CNN	1.901 (0.19)
	LSTM	**1.940 (0.13)**
	CNN-LSTM	1.562 (0.28)
	Buy and Hold	1.860 (0)
15	CNN	1.563 (0.30)
	LSTM	1.700 (0.25)
	CNN-LSTM	1.317 (0.26)
	Buy and Hold	**1.860 (0)**

TABLE VI
COMPOUND RETURN

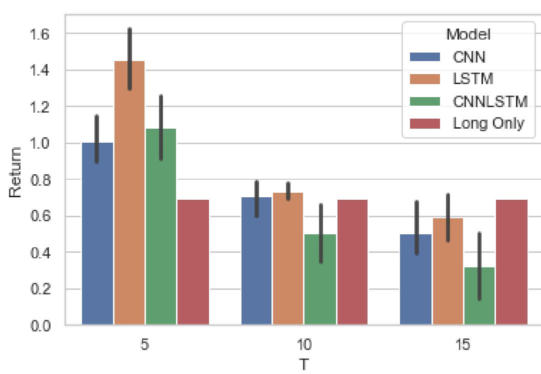

Fig. 10. No Compound Return

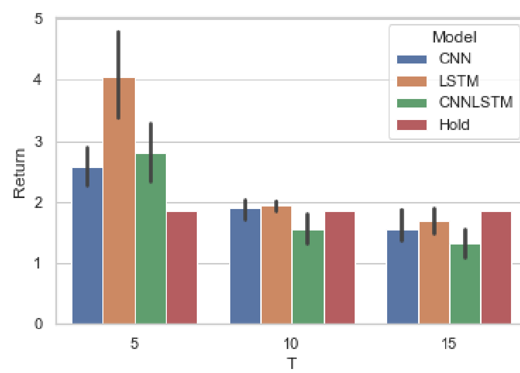

Fig. 11. Compounded Return Results

performed better for this tick length using both strategies. The CNN performed better than the CNN-LSTM for both strategies as well for 10 minute tick length, however the opposite is true for the 5 minute tick length.

Note that the benchmark strategy performed particularly well due to the market trending upwards throughout the training set. However, for the tick length of 5 all the models beat its performance, and for a tick length of 10 both the LSTM and CNN beat its performance on average. For a tick length of 15 minutes however the benchmark strategy beat all models. Due to the ability of the models to execute long and short trades.

Example profit paths are presented for each of the tick lengths and for each strategy. Figure 12, 13 and 14 show paths for the no compound strategy for 5, 10 and 15 minute length ticks respectively. Figure 15, 16 and 17 show paths for the compound strategy for 5, 10 and 15 minute length ticks respectively.

V. Discussion

Three different deep learning models, CNN, LSTM and CNN-LSTM, were created and tested on varying tick lengths.

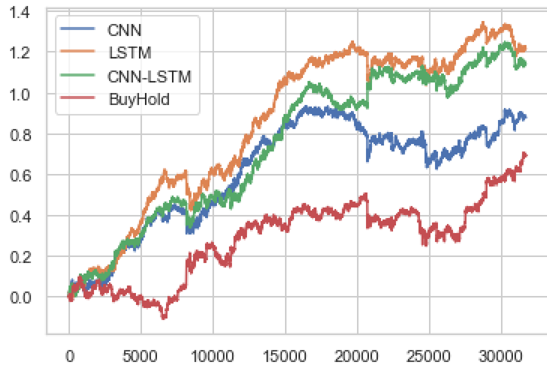

Fig. 12. No Compounded Return T=5

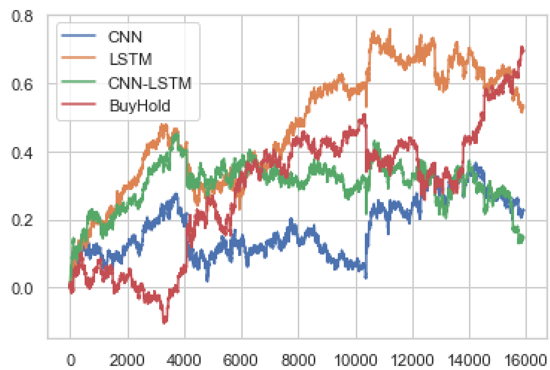

Fig. 13. No Compounded Return T=10

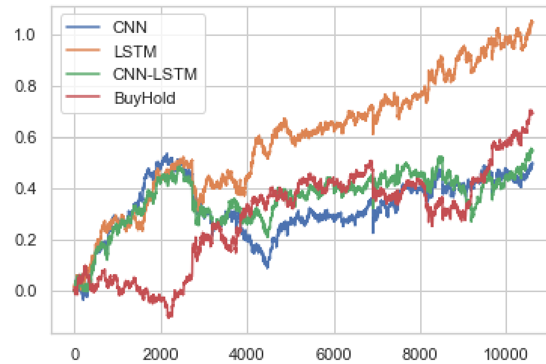

Fig. 14. No Compounded Return T=15

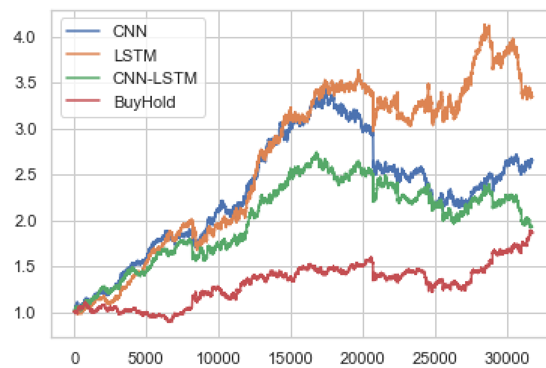

Fig. 15. Compounded Return T=5

Both accuracy and profitability were tested, with the LSTM dominating the models in all conditions. CNN tended to perform better than CNN-LSTM in profit based measures while the CNN-LSTM tended to perform better in accuracy measures. This disparity is interesting, indicating that if the end goal is maximising profit, choosing a model based on accuracy only may not be the best idea. While a model may be able to correctly predict more movements, the movements of greater magnitude have more impact on the profit. Basically, all predictions are not worth the same. One way to improve this is to create profit focused model with a custom loss function that takes into account the magnitude of the movements, having a greater cost if big movements are misclassified compared to smaller. The tick lengths of 5,10 and 15 minutes were chosen to test the potential of the chosen models with large amounts of noisy data, and they were successful in these ranges. 5 minute tick length was the best performing for all models, which may be due to more data being avaliable for lower tick lengths and more potential trading periods, however this does not explain the 15 minute tick dominating the 10 minute tick experiments. Further testing can be done with different tick lengths, both greater and smaller than those presented in this paper, to see if there is a more well defined link between the performance of the model and tick size. All the models were compared to simple benchmark strategies which comprised of going long in the market. This is the stance of a typical investor, and the test period had an uptrend which greatly benefits this type of investing. However, despite the above average performance of the benchmark strategy, the models created were able to consistently beat the benchmark in for 5 minute ticks, and the CNN and LSTM beat the benchmark for 10 minute ticks as well.Due to the ability of the models to execute long and short trades they should have similar performance in a downtrending market while the benchmark would falter, however further testing needs to be done to prove this. The CNN and CNN-LSTM had worse than expected performance in this predictive task when compared to other similar methodologies [18], [19], [21], [22], [24], [25], however this may be due to the low amount of features, 7, while other models utilising CNN architectures in their comparison use up to 40 features [25]. Possible price related features could be in the form of popular technical indicators, such as those used in [25], [32]. Sentiment analysis on the internet related to the asset in question has also been successfully used as a feature for prediction of stock and cryptocurrency movements [33], [34], and would likely improve the results of these models if implemented. Computational cost had a large impact on the selection of the hyper-parameters and empirical testing. Due to the large cost of computing LSTM and convolutional layers, only a small range of hyper-parameters were considered in the creation of the models with respect to the validation data.

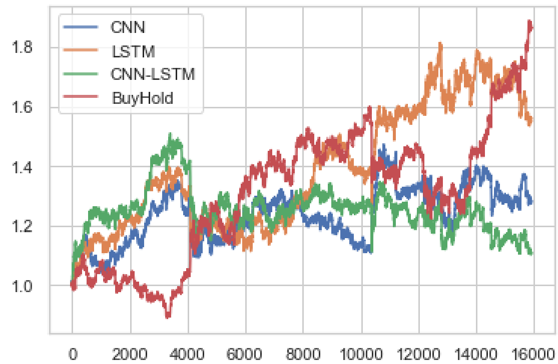

Fig. 16. No Compounded Return T=10

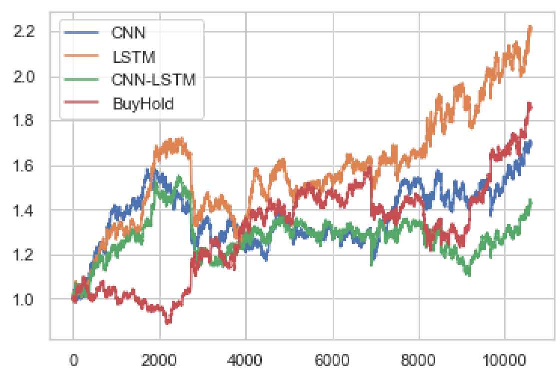

Fig. 17. Compounded Return T=15

The depth and complexity of the model itself was also limited by this. More optimal models could possibly be created if a large hyper-parameter space was searched. For testing, only 5 experimental runs were averaged for each situation due to the time constraints on the results of this paper. Improved validity of results could be attained through running more experiments for the average results. With respect to the profitability tests, the strategies were very simple and the no transaction cost assumption is not realistic. Further research could be done into better trading strategies, such as using the probability, rather than just the classification, of a future movement dictating how much capital is invested in a certain position. More classes could also be created, such as bins of return magnitude, to guide investment size. Tests should also be done with varying commission sizes. Each model in this paper were considered separately and are rather simple, further investigation into more complex non-linear architectures, for example the CNN architecture used in ResNet [17]. Considerations of ensemble learning using these deep learning models could result in a higher performing model, as it is obvious that different models are better at detecting different movements through the discrepancy between the accuracy and profitability scores. Bagging [35], boosting [36] and stacking [37] are methods that could be utilised.

VI. Conclusion

Three model alternatives were presented and assessed for the prediction of bitcoin time series data for tick lengths varying from 5-15 minutes long. These predictions were then used to test the potential profitability of the models. LSTM was determined to be the best model in both predictive and profitability tests, with the highest accuracy and profit being found when considering 5 minute tick data. In this tick range, all models were able to outperform the long only strategy despite the market being in a heavy uptrend, which is a positive result and justifies the work in the paper. Despite this positive result, many possible improvements to the model and experiment environment have been presented within the discussion which could further push the limits of the models and likely result in better performing models in the future. With the goal of profitability analysis in mind, the next logical step would be enhancing profitability testing through improved strategies and consideration of transaction costs. These considerations will help decide optimal models in the trading space when designing higher accuracy models in the future.

References

[1] Williams, G. (2021, October 23). *Bitcoin: A Brief Price History of the First Cryptocurrency*. Investing News Network. https://investingnews.com/daily/tech-investing/blockchain-investing/bitcoin-price-history/

[2] *More than half of experts in a recent crypto survey said bitcoin will replace fiat currency by 2050.* (2021, July 18). Markets.Businessinsider.Com. https://markets.businessinsider.com/news/currencies/bitcoin-vs-fiat-global-finance-2050-crypto-experts-finder-2021-7

[3] Statista. (2021, November 15). *Bitcoin (BTC) price history up until November 15, 2021*. https://www.statista.com/statistics/326707/bitcoin-price-index/

[4] Huck, N. (2019). Large data sets and machine learning: Applications to statistical arbitrage. *European Journal of Operational Research*, 278(1), 330–342. https://doi.org/10.1016/j.ejor.2019.04.013

[5] Hsu, M. W., Lessmann, S., Sung, M. C., Ma, T., Johnson, J. E. (2016). Bridging the divide in financial market forecasting: machine learners vs. financial economists. *Expert Systems with Applications*, 61, 215–234. https://doi.org/10.1016/j.eswa.2016.05.033

[6] Long, W., Lu, Z., Cui, L. (2019). Deep learning-based feature engineering for stock price movement prediction. *Knowledge-Based Systems*, 164, 163–173. https://doi.org/10.1016/j.knosys.2018.10.034

[7] Zhu, Y. (2020). Stock price prediction using the RNN model. *Journal of Physics: Conference Series*, 1650, 032103. https://doi.org/10.1088/1742-6596/1650/3/032103

[8] Saad, E., Prokhorov, D., Wunsch, D. (1998). Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on Neural Networks*, 9(6), 1456–1470. https://doi.org/10.1109/72.728395

[9] Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404, 132306. https://doi.org/10.1016/j.physd.2019.132306

[10] Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986). *Learning representations by back-propagating errors*. nature, 323(6088), 533-536.

[11] Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[12] Ghosh, P., Neufeld, A., Sahoo, J. K. (2021). Forecasting directional movements of stock prices for intraday trading using LSTM and random forests. *Finance Research Letters*, 102280. https://doi.org/10.1016/j.frl.2021.102280

[13] Sirignano, J., Cont, R. (2018). Universal Features of Price Formation in Financial Markets: Perspectives From Deep Learning. *SSRN Electronic Journal*. Published. https://doi.org/10.2139/ssrn.3141294

[14] Ma, Q. (2020). Comparison of ARIMA, ANN and LSTM for Stock Price Prediction. *E3S Web of Conferences*, 218, 01026. https://doi.org/10.1051/e3sconf/202021801026

[15] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning For Image Recognition, *Computing Research Repository*. https://arxiv.org/abs/1512.03385

[16] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2014). Going Deeper with Convolutions. *Computing Research Repository*. https://arxiv.org/abs/1409.4842

[17] Sezer, O.B., Ozbayoglu, A.M. (2018) Algorithmic financial trading with deep convolutional neural networsL Time series to image conversion approach. *Appl. Soft Comput.*. 70, 525-538. https://doi.org/10.1016/j.asoc.2018.04.024.

[18] Borovykh, A., Bohte, S., Oosterlee, C. (2017). Dilated Convolutional Neural Networks for Time Series Forecasting. *Journal of Computational Finance*. 22 (4), 73-101. https://doi.org/10.21314/JCF.2019.358.

[19] Lu, W., Li, L., Li, Y., Sun, A., Wang, J. (2020). A CNN-LSTM-based model to forecast stock prices. *Complexity*. 2020. 1-10. https://doi.org/10.1155/2020/6622927.

[20] Wu, J.MT., Li, Z., Herencsar, N. Vo, B., Lin. J.CW. (2021). A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. *Multimedia Systems* https://doi.org/10.1007/s00530-021-00758-w

[21] Bouri, E., Lau, C. K. M., Saeed, T., Wang, S., Zhao, Y. (2021). On the intraday return curves of Bitcoin: Predictability and trading opportunities. *International Review of Financial Analysis*, 76, 101784. https://doi.org/10.1016/j.irfa.2021.101784

[22] Lahmiri, S., Bekiros, S. (2019). Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, Solitons Fractals*, 118, 35–40. https://doi.org/10.1016/j.chaos.2018.11.014

[23] Li, Y., Dai, W. (2020). Bitcoin price forecasting method based on CNN-LSTM hybrid neural network model. *The Journal of Engineering, 2020*(13), 344–347. https://doi.org/10.1049/joe.2019.1203

[24] Qiang. Z., Shen. J. (2021) Bitcoin High-Frequency Trend Prediction with Convolutional and Recurrent Neural Networks. http://cs230.stanford.edu/projects_winter_2021/reports/70308950.pdf

[25] Nakano, M., Takahashi, A., Takahashi, S. (2018b). Bitcoin Technical Trading With Artificial Neural Network. *SSRN Electronic Journal*. Published. https://doi.org/10.2139/ssrn.3128726

[26] Livieris, I. E., Kiriakidou, N., Stavroyiannis, S., Pintelas, P. (2021). An Advanced CNN-LSTM Model for Cryptocurrency Forecasting. Electronics, 10(3), 287. https://doi.org/10.3390/electronics10030287

[27] Takens, F. (1981). *Detecting strange attractors in tubulence*. *Dynamical Systems and Turbulence* pp. 366–381. http://crcv.ucf.edu/gauss/info/Takens.pdf

[28] Frazier, C., Kockelman, K. (2004) Chaos theory and transportation systems: Instructive example. *Transportation Research Record: Journal of the Transportation Research Board*, vol. 20, pp. 9–17, 2004

[29] Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd ed.). O'Reilly Media.

[30] Klein, C. (2021, November 13). *400+ crypto currency pairs at 1-minute resolution*. Kaggle. https://www.kaggle.com/tencars/392-crypto-currency-pairs-at-minute-resolution

[31] Kingma, D.P., Ba, J. (2015). Adam: A Method for Stochastic Optimization. *Computing Research Repository*, abs/1412.6980.

[32] Taylor, M. P., Allen, H. (1992). The use of technical analysis in the foreign exchange market. *Journal of International Money and Finance*, 11(3), 304–314. https://doi.org/10.1016/0261-5606(92)90048-3

[33] Derakhshan, A., Beigy, H. (2019). Sentiment analysis on stock social media for stock price movement prediction. *Engineering Applications of Artificial Intelligence*, 85, 569–578. https://doi.org/10.1016/j.engappai.2019.07.002

[34] Kim, Y. B., Lee, J., Park, N., Choo, J., Kim, J. H., Kim, C. H. (2017). When Bitcoin encounters information in an online forum: Using text mining to analyse user opinions and predict value fluctuation. *PLOS ONE*, 12(5), e0177630. https://doi.org/10.1371/journal.pone.0177630

[35] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123–140. https://doi.org/10.1007/bf00058655

[36] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5). https://doi.org/10.1214/aos/1013203451

[37] Džeroski, S., ŽEnko, B. (2004). Is Combining Classifiers with Stacking Better than Selecting the Best One? *Machine Learning*, 54(3), 255–273. https://doi.org/10.1023/b:mach.0000015881.36452.6e