Harnessing Twitter sentiment and historical price data to enhance Bitcoin market analysis with deep learning models

James Marcogliese and Yuri Zmytrakov

April 26, 2022

Abstract

Can the price of Bitcoin be predicted using deep learning techniques, and if so, can accuracy be enhanced with Tweet sentiments? We construct a variety of deep learning architectures to determine the best approach to predict Bitcoin price movements using historical price data. We then analyze sets of manually labeled stock-related Tweets, using them as training data and constructing a series of stock Tweet sentiment analyzers to predict positive or negative sentiment. The best preforming sentiment predictor is then used to label a set of Bitcoin related Tweets, which is then used as an additional feature in our Bitcoin price predictor. Results show that prediction on price is fairly successful against daily prices, but Tweet sentiment data did not increase prediction accuracy.

1 Introduction

1.1 Problem Definition

On Feb 4, Elon Musk Tweeted 'Dogecoin is the people's crypto'[1], causing a sudden jump of over 50% in one day. In general, cryptocurrency investor behaviour is heavily influenced by media proprietors, financial news sources and social media platforms, leading to unpredictable runups and selloffs in the Cryptocurrency Market market. Experts struggle to predict the future price swings based on historical asset prices only, so prediction models should be enhanced by sentiment indicators from financial news distribution platforms.

1.2 Research question

The global crypto market cap is $1.77T[2], yet it is extremely volatile and considered high risk for investors. Is crypto trading a gambling game or can market trends be understood? Recurrent Neural Networks are excellent technologies to be used to analyze and predict time series information, such as stock market asset price, crypto prices and currency exchange rates. But, do we know which RNN model and neural network structure will solve this problem with the highest accuracy? Can a model built on this approach be used to predict the future bitcoin asset values? Furthermore, to what extend can sentiment of social media and financial news impact bitcoin asset volatility, and how our RNN models can leverage from additional data?

2 Literature Review

2.1 Study group 1: Sentiment analysis

2.1.1 Deep Learning for Hate Speech Detection in Tweets[3]

- Background

  Content moderation staff are routinely subjected to disturbing content. One application of sentiment classification is detection of hate speech on online platforms so that human moderators are no longer necessary.

- Aim

  Compare effectiveness of modern deep learning techniques against current n-gram methods for the detection of hate speech.

- Methodology

  Baseline methods: 1. Char n-grams 2. TF-IDF 3. BoWV

  Proposed methods: 1. CNN 2. LSTM 3. FastText

  16k annotated Tweets labelled as 'sexist', 'racist', 'neither'

- Results

  Proposed methods are significantly better than the baseline methods. Among the baseline methods, the word TF-IDF method is better than the character n-gram method. Among proposed methods, CNN performed better than LSTM which was better than FastText.

- Conclusions

  Explored deep neural net architectures significantly outperformed existing methods.

2.1.2 Expressively Vulgar: The Socio-dynamics of Vulgarity[4]

- Background

  Most thoughts can easily be reworded to exclude vulgar language; the use of explicit words indicates a strong desire to send a specific message.

- Aim

  Answering: Is the expression of vulgarity and its function different across author demographic traits? Does vulgarity impact perception of sentiment? Does modeling vulgarity explicitly help sentiment prediction?

- Methodology

  6,800 Tweets labelled for sentiment, including data about user demographics.

  Analyze demographics & frequency of vulgarity (Pearson correlation), demographics & sentiment perception (Pearson correlation), vulgar vs. censored sentiment perception using crowdsourced sentiment analysis of censored Tweets (Kruskal-Wallis H-tests for significance), train LSTM network using new vulgarity feature using masking, token insertion and concatenation.

- Results

  Across sentiment labels, vulgarity features improves network performance at predicting negative Tweets, compared to baseline. Masking improved the prediction of positive Tweets. Token insertion and concatenation improved the prediction of negative Tweets.

- Conclusions

  Expressiveness of vulgarity interacts with a number of demographic features, including age, gender, education, income, religiosity, and political ideology. Vulgarity often used to intensify existing sentiment. Utilizing vulgarity-centric features yields increased sentiment analysis system performance.

2.1.3 Multilingual Twitter Sentiment Classification: The Role of Human Annotators[5]

- Background

  Most machine learning led sentiment classification is led by human annotated data.

- Aim

  What are the limits of automated Twitter sentiment classification?

- Methodology

  Quantify the quality of training data by applying various annotator agreement measures and identify the weakest points of different datasets. Use the data as training data in constructed automated classification models (6 classification models – 5 flavors of Support Vector Machines and Naïve bayes).

  Datasets containing 13 languages with other 1.6 million annotated Tweets.

- Results

  Generally, the top classifiers were not distinguishable.

  27% - 35% more difficult to distinguish between classes when utilized as unordered categories compared to ordered.

- Conclusions

  The quality of classification models depends much more on the quality and size of training data than on the type of the model trained. No statistically significant difference between the performance of the top classification models found.

  Humans tend to perceive sentiment classes (neg, neu, pos) as ordered rather than a set of unordered categories.

2.1.4 MELD: A Multimodal Multi-Party Dataset for Emotion Recognition[6]

- Background

  Although significant research work has been carried out on multimodal emotion recognition using audio, visual, and text modalities, significantly less work has been devoted to emotion recognition in conversation. One main reason for this is the lack of a large multimodal conversational dataset.

- Aim

  Create a Multimodal EmotionLines Dataset (MELD), an extension and enhancement of EmotionLines. Annotated with Ekman's six universal emotions (Joy, Sadness, Fear, Anger, Surprise, and Disgust)

- Methodology

  Analyze 13,000 utterances from 1,433 dialogues from the TV-series Friends. Each utterance is annotated with emotion and sentiment labels, and encompasses audio, visual, and textual

modalities. Completed by Amazon Mechanical Turk. Compare results with DialogueRNN – current state-of-the-art for conversational emotion detection, and bcLSTM as baseline (bidirectional RNN)

- Results

  DialogueRNN applied with MELD achieves higher performance than baseline bcLSTM by an F-score of 1%. The textual modality outperforms the audio modality by about 17%, which indicates the importance of spoken language in sentiment analysis. For positive sentiment, audio modality performs poorly. Overall, emotion classification performs poorer than sentiment classification. This observation is expected as emotion classification deals with classification with more fine-grained classes.

- Conclusions

  MELD dataset built.

2.2 Study group 2: Deep learning and machine learning applications in the finance and banking industry

2.2.1 A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module[7]

- Background

  Forecasting a financial asset's price is essential for investors, because it reduces the risk decision - making error, and ultimately loss of investment capital. Recently, the deep neural network is popularly applied in this area of research. However, it is prone to overfitting owing to limited availability of data points for training.

- Aim

  The main objective of this research paper is to evaluate the effectiveness of a novel data augmentation approach ModAugNet model for stock market index forecasting, which consists of two modules: overfitting prevention LSTM model(ModAugNet) and LSTM module(SingleNet).

- Methodology

  Researchers modified LSTM model, and incorporated a data augmentation (ModAugNet) approach to achieve robustness against overfitting in forecasting the stock market index. Then the model was trained and empirically tested on the validation data set, and compared against the LSTM module (SingleNet).

- Results

  Having investigated the proposed framework performances on different datasets, and conducted empirical tests on representative real world stock market SP 500 datasets, the researchers came to positive conclusion. The results confirm the excellent forecasting accuracy of the ModAugNet model, which lowered test error comparing to SingleNet, in which overfitting prevention LSTM module is absent. The test MSE, MAPE and MAE have decreased 54.1%, 35.5% and 32.7% respectively.

- Conclusions

  DNN methods have been adopted to solve financial problems, because of their ability to solve complex problems based on data. As data increases, in most cases, DNN performance improves, comparing to ML algorithms. But with large dataset, DNN tends to overfit. Based on this research paper, DNN overfitting can be prevented by implementing data augmentation approach, and ultimately increase the generalizability of DNN.

2.2.2 Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations[8]

- Background

  Stock market price data have non-linear, noisy and non-stationary structure, and therefore prediction of the price or its direction are both challenging tasks. As the result, investors fail to allocate the budget.

- Aim

  In this study, the researchers aim to predict the hourly stock price direction. While Artificial Neural Network, Support Vector Machine, Naive Bayes, Linear Regression and KNN are the leading algorithms in FinTech for classification problems due to its simplicity. This paper aims to evaluate Convolution Neural Networks performance by using 25 technical indicators against Linear Regression.

- Methodology

  Chi-square feature selection method were used for feature selection by using 25 technical indicators, and CNN model will built. The researchers aimed to extract spatial relations (similar to image classification task). L2 regularizer and dropout were used to prevent overfitting. The CNN was compared against Linear Regression.

- Results

  The experimental results were obtained based on 4 years training, 1 year test dataset. CNN - Corr was the leading algorithm with the best MA F-Measure scores in 54 out of 100 stocks.

- Conclusions

  The success of CNN - Corr model can be attributed to unique classification framework, this model combines feature selection and classification steps, and ultimately high MA F-Measure score.

2.2.3 Cross-domain Deep Learning Approach for Multiple Financial Market Prediction[9]

- Background

  The global financial system is structured by the worldwide framework of legal agreements, institutions, and economic actors, they facilitate international flows of financial capital for purposes of investment and trade financing. The financial analysis of global markets is extremely complex as they usually influence each other through various interactions. The cross-domain correlation, indicating the interactions between heterogeneous markets and the time series correlation, describing the transitional influence across different time points in a market.

- Aim

  In this paper, the researchers proposed a novel cross-domain deep learning approach (Cd-DLA) to learn real-world complex correlations for multiple financial market prediction. It analyzed the correlations between currency and stock markets in various countries.

- Methodology

  The method used in this paper utilizes the recurrent neural network (RNN) to analyze the time-series correlation among the temporal data inputs in all multiple financial markets (Cd-DLA), the inner-domain attention neural network (Id-ANN) to capture the inner-domain correlation in homogeneous markets, the cross-domain attention neural network (Cd-ANN) to model the cross-domain correlation in heterogeneous markets.

- Results

  For each data set, Cd-DLA outperformed all baselines in both F-measure and AUC measurement. It was observed that Id-ANN and the Cd-ANN based on the attention mechanism showed better classification performance on most of the data sets than the baseline RNN networks for all homogeneous and heterogeneous markets. Cd-DLA showed obvious improvements in terms of F-measure and AUC in the data set for the financial crisis period.

- Conclusions

  This paper presented a novel cross-domain deep learning approach with parallel multi-tasking architecture and complex correlations. The result of the experiments with real-world financial data sets demonstrated the superior performance over baseline neural networks.

2.2.4 Sequence classification for credit-card fraud detection[10]

- Background

  Due to the growing volume of electronic payments, the monetary strain of credit-card fraud is turning into a substantial challenge for financial institutions and service providers, thus forcing them to continuously improve their fraud detection systems. Modern data-driven and learning-based methods, despite their popularity in other domains, only slowly find their way into business applications.

- Aim

  The main objective of this paper was to compare sequential learners LSTM with static leaner Random Forest on the real-world fraud detection dataset. In addition to that, the study evaluated the benefit of state of the art manual feature engineering.

- Methodology

  Fraudulent transactions can be understood as anomalies in the purchase behavior of customers. Firstly, feature engineering was completed to summarize purchase activities. The LSTM model had two recurrent layers and logistic regression classifier stacked on top of the last layer. For both models, optimal hyper parameters were searched and configuration via grid search.

- Results

  Having trained models for each combinations of feature set, the researchers tested classification performance on the held out test set. Both approaches were consistent in frauds detection. But, it was determined that LSTM and RF detect different frauds. On face to face (F2F), RF model agree on 50.8% of their true positives versus 37.8% in LSTM model. However, in ecomm transactions, LSTM classified 47.5%, whereas RF only 35

- Conclusions

  This paper employed LSTM as a means to aggregate the historical purchase behaviour of credit card holders with the goal to improve fraud detection accuracy on new incoming transactions. The study demonstrated that online and offline transactions exhibit different qualities. For online transactions, LSTM improves the detection accuracy, whereas RF outperformed LSTM with F2F data set. Finally, a subsequent analysis of true positives revealed that both approaches tend to detect different frauds, which suggests a combination of the two.

3 Methods

To properly answer the research questions posed in this project, both sub-solutions of time-series price prediction and stock Tweet sentiment analysis require development before they can be combined in the last step to produce a multi-input deep learning architecture.

Firstly, gathering of data sources and preliminary analysis on datasets. A series of neural net architectures will be trained on Bitcoin time-series price data to determine the architecture that yields the highest accuracy. Secondly, a series of machine learning classifiers will be trained on the stock Tweets to determine the algorithm that yields the highest accuracy. The Bitcoin Tweets will then be labeled using the successful classifier. Once the previous steps are complete, time-correlation of the sentiments with asset prices will be performed. Daily price and sentiment of Bitcoin Tweets will lastly be fed into the chosen deep learning technique from the price prediction portion of this project to determine if the model preforms better or worse than baseline.

3.1 Time-series forecasting

Long short-term memory (LSTM) is a variant of a recurrent neural network (RNN) architecture used in the field of deep learning. LSTM networks are well-suited to making predictions based on time series data, which is a good starting point for this project. Gated Recurrent Unit (GRU) is the newer generation of RNN similar to LSTM, it utilizes a different way fusing previous timestep information with gates to prevent vanishing gradients. GRU performance surpasses the LSTM in the scenario of long text, such as news articles. RNN has short term memory problems. Having reviewed multiple research papers, we have determined that there is no superior RNN model to solve this problem. In this study, we will compare the performance of these models to determine the winner using accuracy, recall, and F1 ratio. As a baseline, we will use the Bitcoin value from the same day a week prior as a naive predictor.

3.1.1 Approaches

- Naive baseline
- LSTM
- LSTM-stacked
- LSTM-bidirectional
- GRU
- GRU-stacked
- GRU-bidirectional

```
def build_LSTM(lookback_window, neurons, opt):
    model = Sequential()
    model.add(LSTM(neurons, activation='relu', input_shape=(lookback_window, 1)))
    model.add(Dense(1))
    model.compile(optimizer=opt, loss='mse', metrics=['mse'])
    return model

def build_LSTM_stacked(lookback_window, neurons, opt):
    model = Sequential()
    model.add(LSTM(neurons, activation='relu', return_sequences = True,
                                              input_shape=(lookback_window, 1)))
    model.add(LSTM(math.floor(neurons/2)))
```

```
    model.add(Dense(1))
    model.compile(optimizer=opt, loss='mse', metrics=['mse'])
    return model

def build_LSTM_bidirectional(lookback_window, neurons, opt):
    model = Sequential()
    model.add(Bidirectional(LSTM(neurons, activation='relu', return_sequences =
                                            True, input_shape=(lookback_window,
                                            1))))
    model.add(Bidirectional(LSTM(math.floor(neurons/2))))
    model.add(Dense(1))
    model.compile(optimizer=opt, loss='mse', metrics=['mse'])
    return model

def build_GRU(lookback_window, neurons, opt):
    model = Sequential()
    model.add(GRU(neurons, activation='relu', input_shape=(lookback_window, 1)))
    model.add(Dense(1))
    model.compile(optimizer=opt, loss='mse', metrics=['mse'])
    return model

def build_GRU_stacked(lookback_window, neurons, opt):
    model = Sequential()
    model.add(GRU(neurons, activation='relu', return_sequences = True,
                                            input_shape=(lookback_window, 1)))
    model.add(GRU(math.floor(neurons/2)))
    model.add(Dense(1))
    model.compile(optimizer=opt, loss='mse', metrics=['mse'])
    return model

def build_GRU_bidirectional(lookback_window, neurons, opt):
    model = Sequential()
    model.add(Bidirectional(GRU(neurons, activation='relu', return_sequences =
                                            True, input_shape=(lookback_window,
                                            1))))
    model.add(Bidirectional(GRU(math.floor(neurons/2))))
    model.add(Dense(1))
    model.compile(optimizer=opt, loss='mse', metrics=['mse'])
    return model
```

3.1.2 Data sources

Daily Bitcoin historical prices data set was provided by Yahoo Finance[14].

3.2 Tweet sentiment analysis

The following machine learning models were considered for the classification task of stock Tweet data:

3.2.1 Approaches

- C-Support vector machine classifier
- Multinomial naive bayes classifier
- Logistic regression classifier
- Random forest classifier

- Linear classifier with SGD (stochastic gradient descent)
- Mutli-layer perceptron classifier

3.2.2 Data sources

Labelled stock Tweets were provided by publicly available data sets on Kaggle.com[12][13]. Bitcoin Tweets were provided by publicly available data set on Kaggle.com[11].

4 Dataset and experimental setup

4.1 Tweet sentiment analysis

To train our sentiment analyzer, we combined two partially labelled stock Tweet datasets made available on Kaggle.com[12][13], which when combined yielded 6667 labelled stock Tweets. Of the labelled Tweets, 4213 were labelled as positive and 2454 as negative. Tweets were pre-processed by lowering, removing special characters and stop words, and applying a word stemmer based on the Porter stemming algorithm. To prepare for machine learning model training, cleaned text was further passed through a count vectorizer and then a term frequency–inverse document frequency (tf-idf) transformer. To balance the uneven dataset, over-sampling of the negative sentiment class was preformed using synthetic minority oversampling technique (SMOTE) which ensured that both positive and negative sentiments both contained 4213 instances, for a final total of 8,426 training instances. Sentiment labels were standardized as 0 and 1 to correspond to negative and positive sentiment for consistency.

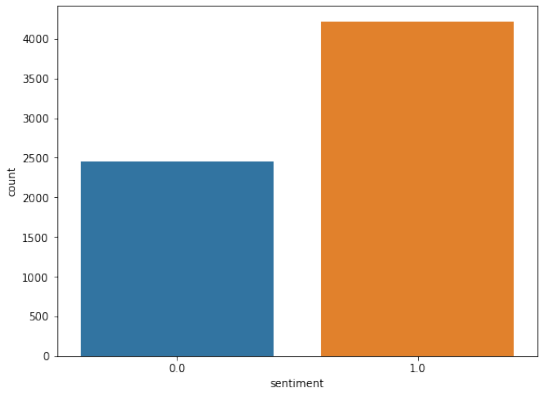


Figure 1: Combined stock Tweet dataset sentiment counts before balancing

During training and testing of the models listed in 3.2.1 of this paper, 5-fold cross-validation was employed to evaluate estimator performance, with average fold accuracy and f1-score gathered from the runs, shown in Table 1. Of the six models, the support vector classifier proved to produce the highest accuracy and f1-score.

Once the sentiment analyzer was trained, a dataset of Bitcoin Tweets were prepared in an identical fashion to the pre-processing steps outlined above and feed through the sentiment analyzer to be labelled. The time span of the Tweets ranged from 2021-02-10 to 2022-03-02, the same coverage of Bitcoin price data gathered. Daily counts of positive and negative sentiment-labelled Tweets were scaled as a single value between 0 and 1 to denote a summary of that day's sentiment, as shown in Table 3.

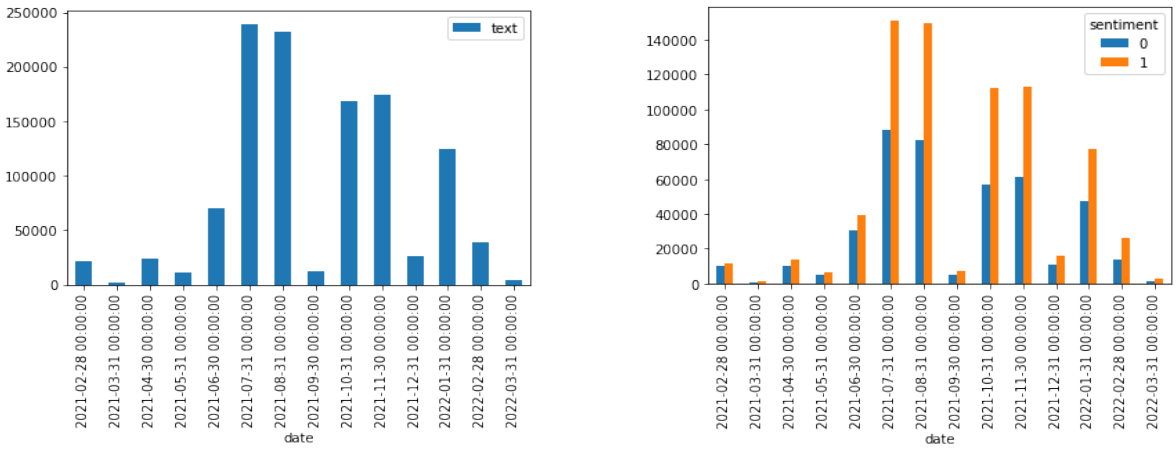
Several correlation measures were performed against the daily Bitcoin price and Tweet sentiment features. Typical correlation measurements like Pearson and Spearman may not be good measurements against combined binary and continuous targets so point-biserial correlation was also

Model	ACC	ACC SD	F1-Score	F1-Score SD
LogR	0.837	0.081	0.839	0.08
MLP	0.841	0.079	0.843	0.079
MNB	0.712	0.076	0.747	0.017
RF	0.823	0.057	0.83	0.069
SGD	0.695	0.137	0.73	0.069
SVC	**0.888**	**0.056**	**0.889**	**0.056**

Table 1: Performance comparison of trained stock Tweet sentiment classifiers

Date	Processed Tweet Text	Sentiment
2021-02-10	blue ridge bank shares halted nyse #bitcoin at...	0
2021-02-10	network secured nodes today soon biggest bea...	0
2021-02-10	trade #crypto #binance enjoy #cashback tradin...	1
2021-02-10	lt fire amp mangt #bitcoin #crypto #btc https ...	1
2021-02-10	teslas #bitcoin investment revolutionary #cryp...	1

Table 2: Sample of labelled Bitcoin Tweets



(a) Before sentiment labelling

(b) After sentiment labelling

Figure 2: Distribution of Bitcoin Tweets dataset

Date	Sentiment
2021-02-05	0.181395
2021-02-06	0.164090
2021-02-07	0.083223
2021-02-08	0.463641
2021-02-09	0.358913

Table 3: Sample of resampled scaled sentiment

considered. Very low (˜0) correlation coefficients were observed among the correlation metrics (Table 4) each accompanied with high p-values, indicating evidence is not strong enough to suggest that correlations exist in the data. See Table 4 for results. Tweet sentiment feature was then used during hyperparameter tuning and testing of the RNN's in section 4.2.

Measurement	Corr Coefficient	P-value
Pearson	0.029	0.563
Spearman	0.028	0.583
Kendall	0.026	0.582
Point Biserial	0.029	0.563

Table 4: Correlation measurements between daily price direction and Tweet sentiments

4.2 Hyperparameter tuning

Historical time-series data of daily Bitcoin prices was provided by Yahoo Finance[14] over the time period of 2021-02-10 to 2022-03-02. Open and close prices were included in the dataset.

Figure 3: Time-series of Bitcoin price data

4.2.1 Hyperparameter options

- Neuron count $\in \{10, 20, 30, 40, 50\}$
- Optimizer $\in \{adam, sgd, rmsprop, ftrl\}$
- Epoch count $\in \{10, 20, 30, 40, 50\}$
- Prediction horizon $\in \{5, 10, 30, 60, 90\}$
- Look-back window $\in \{25, 50, 75, 100\}$

4.2.2 Hyperparameter tuning on price data only

Data was split into training and testing data depending on the chosen prediction horizon and look-back window. Hyperparameter tuning was undertaken by applying a grid search of parameters while training an array of single-layer LSTM networks, then recording the resulting root mean squared error (RMSE) when making predictions on the withheld test set. Cartesian product of the hyperparameters

yielded 2000 distinct combinations. Batches of temporal data were fed into the model by way of a time-series generator included in the Keras preprocessing sequence API. Top 3 results of testing are displayed in Figure 4. Naive prediction was determined by rolling forward the price of the period proceeding the test period.

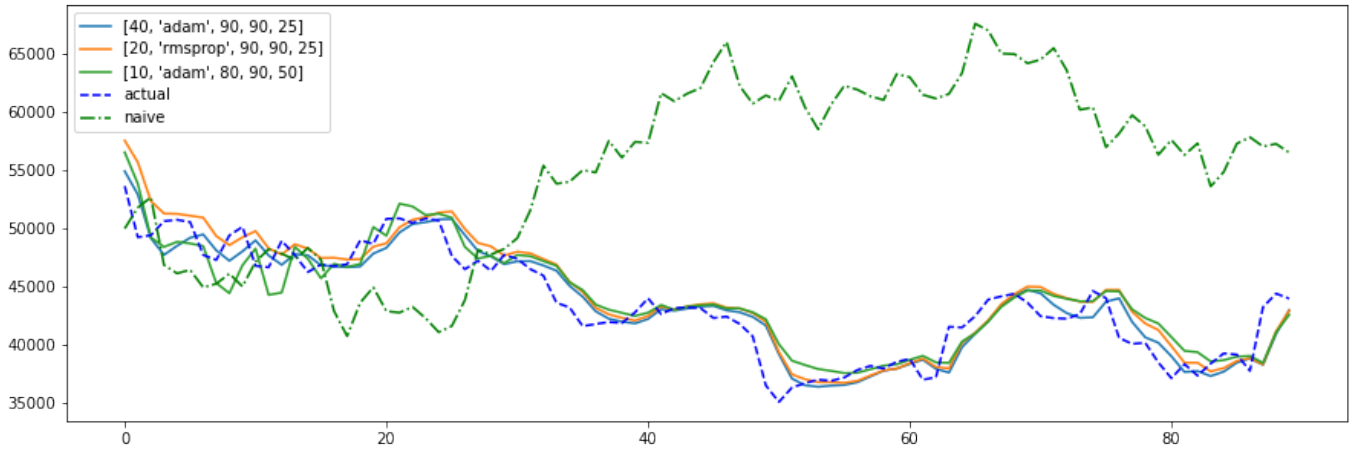


Figure 4: Top 3 hyperparameter combinations for basic LSTM network over 90 days with price data only. Parameters are listed in format: [neurons, optimizer, epochs, prediction horizon, look-back window]

4.2.3 Hyperparameter tuning on price and sentiment data

Hyperparameter tuning was undertaken again with an identical approach as 4.2.2 against an array of single-layer LSTM networks, in this experiment using both the price and sentiment feature produced from section 4.1. Top 3 results of testing are displayed in Figure 5. Again, naive prediction was determined by rolling forward the price of the period proceeding the test period.

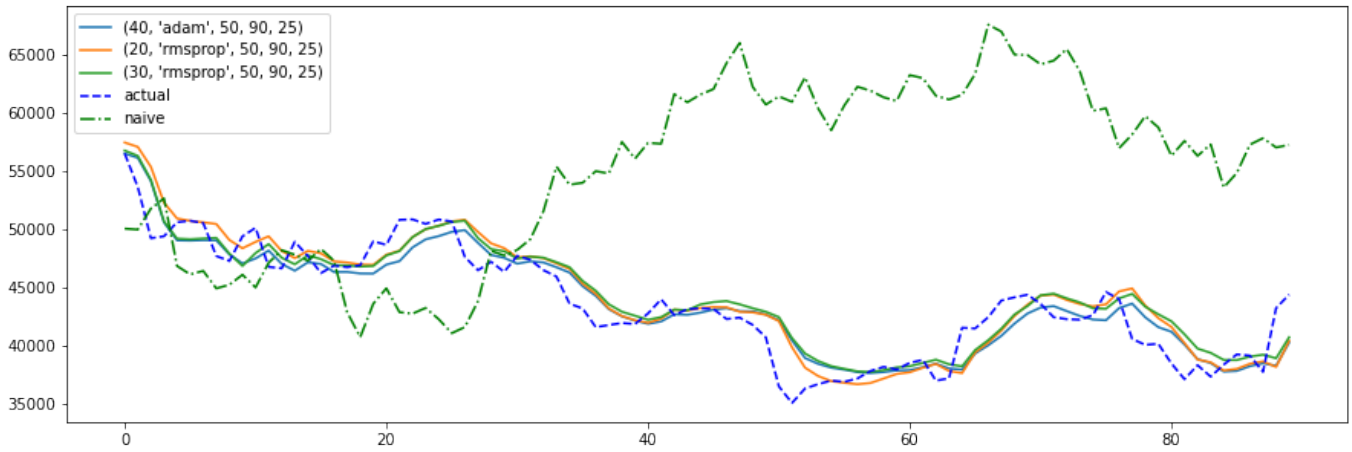


Figure 5: Top 3 hyperparameter combinations for basic LSTM network over 90 days with price and sentiment data. Parameters are listed in format: [neurons, optimizer, epochs, prediction horizon, look-back window]

5 Results

5.0.1 Model selection for price data only

The top combination of hyperparameters from 4.2.2 were used to train a set of RNN architectures listed in 3.1.1 and measurements for MSE, RMSE and MAPE were taken against withheld test data (Table 5). The single-layer GRU model produced the most accurate results, with an RMSE value of 1564 over a forecast of 90 days. The final GRU model forecast is displayed in Figure 5.

Model	MSE	RMSE	MAPE
build_LSTM	4.322378e+06	2079.032983	12.124374
build_LSTM_stacked	7.417760e+06	2723.556534	12.822548
build_LSTM_bidirectional	6.447592e+06	2539.210996	12.021906
build_GRU	**2.448763e+06**	**1564.852434**	**12.136807**
build_GRU_stacked	2.591019e+06	1609.664222	12.346874
build_GRU_bidirectional	2.988129e+06	1728.620579	12.400062
naive_baseline	2.601689e+08	16129.753089	12.400062

Table 5: Test performance against RNN architectures using best hyperparameter combination from Figure 4 when using only price data

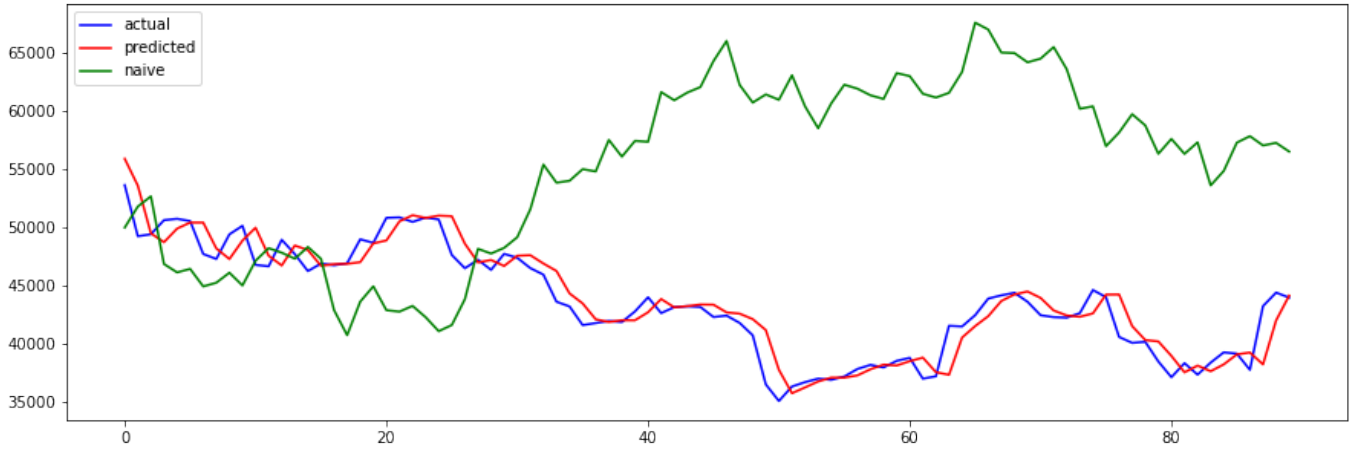


Figure 6: Forecast of selected GRU model with tuned hyperparameters, price data only

5.0.2 Model selection for price and sentiment data

The top combination of hyperparameters from 4.2.3 were used to train a set of RNN architectures listed in 3.1.1 and measurements for MSE, RMSE and MAPE were taken against test data (Table 6). As with model performance comparisons in section 5.0.1, the single-layer GRU model produced the most accurate results, with an RMSE value of 1727 and a MAPE value of 2.9 over a forecast of 90 days. The final GRU model forecast is displayed in Figure 7.

Models	MSE	RMSE	MAPE
build_LSTM_sentiment	5.618086e+06	2370.250161	4.332274
build_LSTM_stacked_sentiment	1.006564e+07	3172.639348	5.881247
build_LSTM_bidirectional_sentiment	9.076195e+06	3012.672383	5.544754
build_GRU_sentiment	**2.985088e+06**	**1727.740726**	**2.903595**
build_GRU_stacked_sentiment	3.340000e+06	1827.566726	3.057440
build_GRU_bidirectional_sentiment	4.259321e+06	2063.812252	3.555111
naive_baseline	2.588805e+08	16089.763053	3.555111

Table 6: Test performance against GRU architectures using best hyperparameter combination when using price and sentiment data

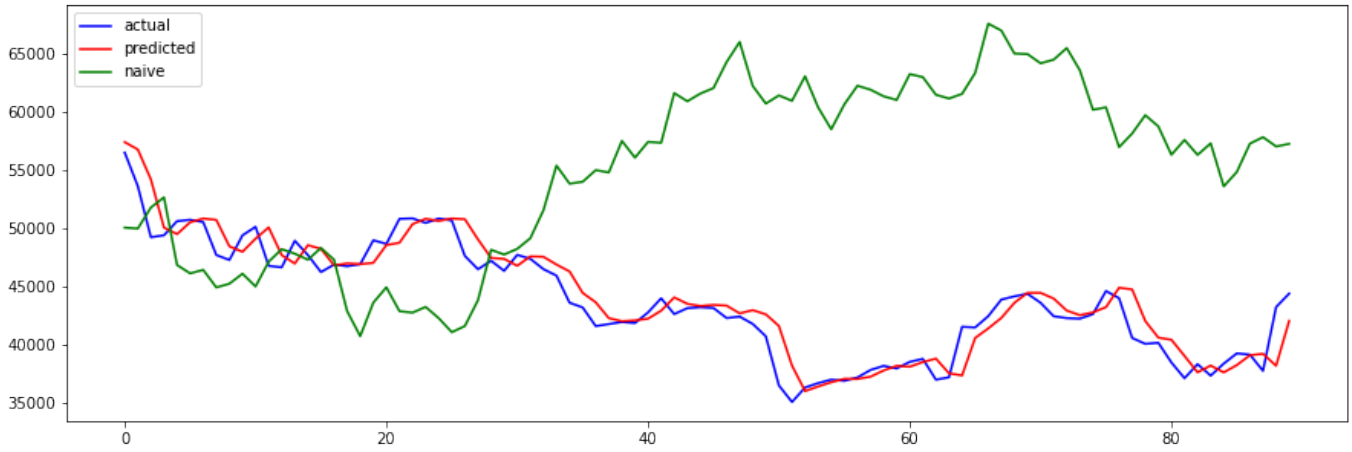


Figure 7: Forecast of selected RNN architecture with tuned hyperparameters, price and sentiment data

6 Conclusions

To summarize the results reported in section 5, a single tuned GRU layer architecture worked best for Bitcoin price prediction in both scenarios where only price data was input, and when price with sentiment data was input. This model could reasonably predict Bitcoin price direction, with the best model producing an RMSE measurement of 1564 over a 90 day period. We found that sentiments in Bitcoin Tweets did not improve model prediction accuracy and in actuality served to decrease the accuracy of the model when utilized, even after hyperparameter tuning. This could be explained by the findings in Table 4 which uncovered a distinct lack of correlation between price and gathered sentiment, which would indicate that the sentiment data only served as noise when used with price data.

6.1 Considerations

Given more time, several more approaches and/or weaknesses could be considered:

- A transformer architecture model could have been used as an additional approach, given its natural effectiveness when processing sequential data.
- Although large, we were unsure if the Bitcoin Tweet dataset used was exhaustive enough to produce a true, representative sentiment value for each day in the time span considered.
- Given availability lack of datasets containing labelled Bitcoin Tweets, we were forced to train a classifier on generic stock Tweets which we then used to label the Bitcoin Tweets by way of domain transfer learning. The true accuracy of the classifier against a Bitcoin Tweet specific dataset is therefore unknown.
- A more intelligent hyperparameter selection method should be used as the grid search approach used in this project yielded an unreasonably high number of parameter combinations which proved extremely time consuming to test.

7 References

1 Elon Musk Tweeted. Dogecoin surged more than 50%
`https://www.cnn.com/2021/02/04/investing/elon-musk-dogecoin/index.html`

2 Today's Cryptocurrency Prices by Market Cap
`https://coinmarketcap.com/`

3 Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep Learning for Hate Speech Detection in Tweets. In Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 759–760. DOI:https://doi.org/10.1145/3041021.3054223

4 Isabel Cachola, Eric Holgate, Daniel Preoţiuc-Pietro, and Junyi Jessy Li. 2018. Expressively vulgar: The socio-dynamics of vulgarity and its effects on sentiment analysis in social media. In Proceedings of the 27th International Conference on Computational Linguistics, pages 2927–2938, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

5 Mozetic, Igor & Grcar, Miha & Smailovic, Jasmina. (2016). Multilingual Twitter Sentiment Classification: The Role of Human Annotators. PLOS ONE. 11. 10.1371/journal.pone.0155036.

6 S. Poria, D. Hazarika, N. Majumder, G. Naik, R. Mihalcea, E. Cambria. MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversation. (2018)

7 Yujin Baeka, Ha Young Kim: "A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module." ModAugNet, 2018.

8 Hakan Gunduz, Yusuf Yaslan, Zehra Cataltepe: "Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations"

9 Xinxin Jiang, Shirui Pan, Jing Jiang and Guodong Long: "Cross-domain Deep Learning Approach for Multiple Financial Market Prediction", University of Technology Sydney, Australia 2018.

10 J. Jurgovskya, M. Granitzer, K. Ziegler, S. Calabretto, P. Portier, L. He-Gueltonc, O. Caelen: "Sequence classification for credit-card fraud detection", 2018

11 Kash. (2021-10-16). Bitcoin Tweets, 1. Retrieved 2022-03-06 from https://www.kaggle.com/kaushiksuresh147/bitcoin-Tweets.

12 Zeus. (2020-07-04) Stock Market Tweet — Sentiment Analysis lexicon, 1. Retrieved 2022-03-06 from https://www.kaggle.com/datasets/utkarshxy/stock-markettweets-lexicon-data

13 Yash Chaudhary. (2020) Stock-Market Sentiment Dataset, 1. Retrieved 2022-03-06 from https://www.kaggle.com/datasets/yash612/stockmarket-sentiment-dataset

14 Bitcoin CAD. (BTC-CAD) (2013). Historical Data. Yahoo!Finance. https://ca.finance.yahoo.com/quote/BTC-CAD/history?p=BTC-CAD.

15 Steven Loria. (2013-07-28) TextBlob (Version 0.16.0) https://github.com/sloria/TextBlob

16 Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.