Bitcoin Price Prediction using Sentiment and Historical Price

Shorya Sharma[1], Sarang Mehrotra[2]

[1-2]School of Electrical Sciences, Indian Institute of Technology Bhubaneswar, Bhubaneswar, Odisha, India

---***---

Abstract - *Bitcoin is one of the first digital currencies that were first introduced around 2008.The decentralized currency gained huge popularity in the past decade and it is now trading around USD 18000 per coin. Speculations around its future upside potential have attracted a large number of individual and institutional investors despite the price volatility. Therefore, price prediction for bitcoin has become a relevant but challenging task. In this project, we will be utilizing sentiment analysis and historical price data to maximize the accuracy of price prediction of bitcoin.*

Key Words: **Bitcoin, Cryptocurrency, Sentiment Analysis, Machine Learning, Historical Price, Deep Neural Networks**

1 INTRODUCTION

Bitcoin is a type of digital currency that can be sent between people without the need for intermediate administration. Transactions made using bitcoin can be more efficient as it is not tied to any country.Despite the fact that bitcoin is not officially recognized as a type of security, it still shares a lot of similarities with other securities like stocks. This means that the traditional techniques for trading stocks based on an analysis of historical price (MACD, EMA) can still be applied to it. On the other hand, bitcoin's institutional holding percentage is only around 36% compared with the 80% US equity institution holding. This means that bitcoin's price could be more easily impacted by market sentiment and news coverage because individual investors are more unstable. This paper intends to explore the potential of recent advancements in deep learning in improving the accuracy of making a price trend movement prediction for bitcoin (upward/downward) from the previous 50% achieved by xxx which is equivalent to random guessing.

2 RELATED LITERATURE

A wide range of Machine Learning models have been used for predicting the short-term/long-term price of bitcoin. The models are mostly either relying on sentiment analysis or an analysis of historical price data for future price prediction.

[1] Siddhi Velankar proposed the necessity of pre-processing the bitcoin price data through using different normalization techniques such as log normalization and Box-Cox normalization before feeding the data into the model. [6] Matthew Dixon1, Diego Klabjan2 and Jin Hoon Bang used DNN(basically MLP) to evaluate the accuracy of a model trained on the data collected from 43 CME listed commodity from March 1991 to September 2014 in 5-minute intervals and obtained around 40% accuracy in predicting whether the commodity price would increase, be neutral, or decrease over the next time interval, which is slightly better than random predictions.[7] Sean McNally, Jason Roche and Simon Caton made a comparison of the traditional ARIMA model for predicting time-series data and more recent deep learning model LSTM, RNN trained using Bitcoin Price Index data and concluded that the LSTM outperforms ARIMA and achieved 52% classification accuracy. They evaluated the impact of the window size of price data and concluded that the most effective length was 100 days for training LSTM.

[3] Giulia Serafini, Ping Yi analyzed the sentiment features from economic and crowd-sourced data and used ARIMAX and RNN for making price prediction to achieve an accuracy with a mean squared error lower than 0.14% with both models.[4] Rajua and Ali Mohammad Tarif used a combination of bitcoin historical price data and tweets for sentiment analysis. They extracted the polarity from the tweets and achieved a 197.515 RMSE with LSTM compared with 209.263 with ARIMA. [5] Ray Chen presented a special strategy to analyze the sentiment in tweets for stock market movement prediction. They pre-generated a dictionary of words and associate them with log probabilities of happiness and sadness associated with each word to compute an average sentiment for each tweet tomake stock market movement prediction and achieved around 60% accuracy.

3 Dataset

3.1 Source

We scrapped several major websites and decided to use comments from investing.com, a popular financial website that has quotes for most of the traded securities and their dedicated discussion pages, as they have higher desired qualities compared with social-media based Twitter dataset and forum-based Reddit dataset which are noisier.

From 2017-09-04 to 2020-11-01, there are an average 167 comments from Investing.com per day. The number of comments is not evenly distributed. The more volatile the price, the more the comments(figure 1a).

In terms of sentiment, the average sentiment of the comments calculated using TextBlob was not very volatile and tended to stay in a range. Part of the reason is that text sentiment related to securities tend to be more subtle than product reviews. An interesting finding is that, the higher the subjectivity score, the higher the sentiment score (figure 1b). It indicates that comments that express strong opinion tendto be the bullish ones. This pattern is shown in both datasets.

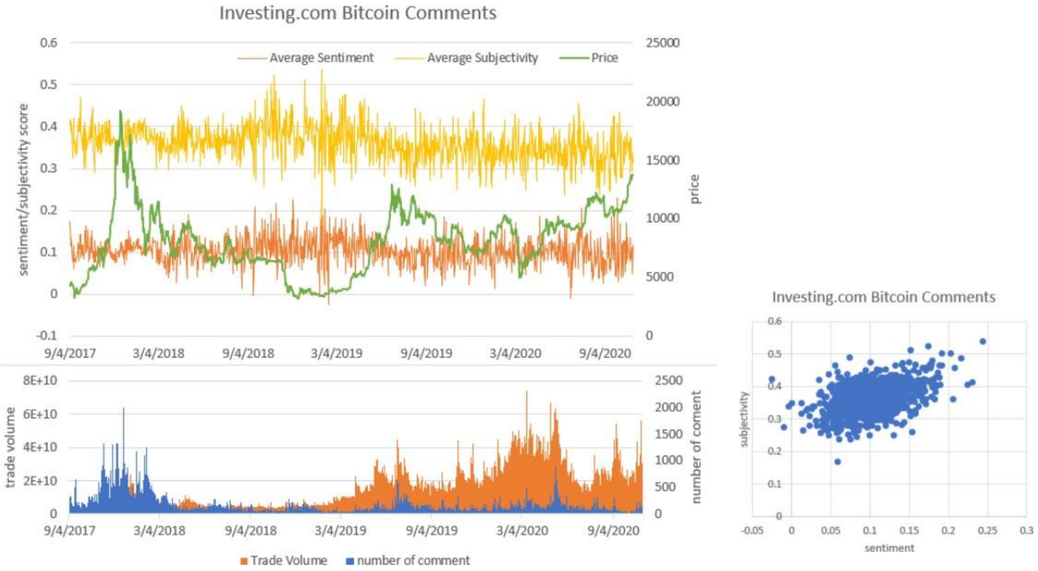


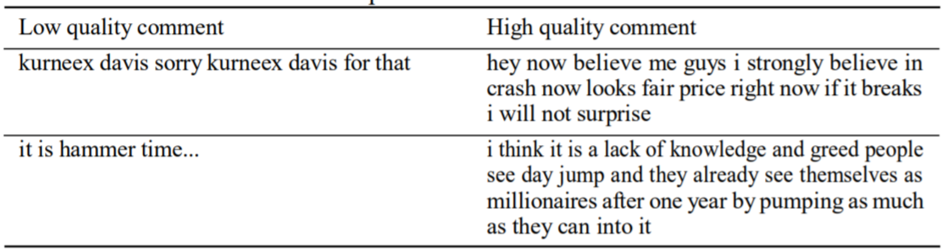
Figure 1: (a) Comment volume distribution (b) Comment overall subjectivity and sentiment

3.2 Data Processing

The bitcoin daily price data spans from Jan 2012 - Nov 2020. For the sentiment analysis part, the investing forum discussions include comments from 2017 till 2020, containing a total of 201,104 comments. Therefore, only the relevant price data between 2017-2020 is kept and is shuffled and split into train/validation by 70%/30% proportions.

The raw data scraped from online sources is noisy and contains a large number of toxic comments, which is the reason why data processing becomes particularly important for training. Below are some real examples taken from the sources.

Table 1: Sample Forum Discussion Comments



Low quality comment	High quality comment
kurneex davis sorry kurneex davis for that	hey now believe me guys i strongly believe in crash now looks fair price right now if it breaks i will not surprise
it is hammer time...	i think it is a lack of knowledge and greed people see day jump and they already see themselves as millionaires after one year by pumping as much as they can into it

In order to improve the overall quality of raw data, a pipeline is designed to clean the textual input.The pipeline aims to deal with several common problems of textual data in NLP tasks.

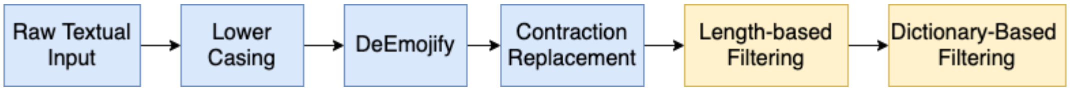
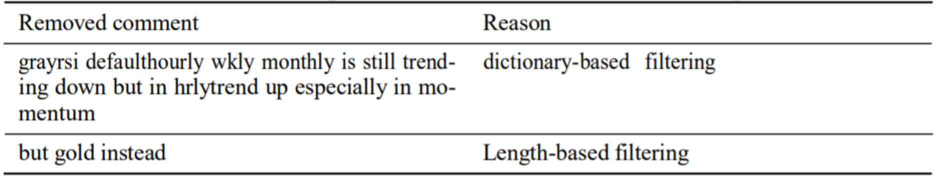


Figure 2: Textual input processing pipeline

The pipeline contains two major parts. The first part is related to textual cleaning and it basically ensures that when the text data is passed into the embedding layer, it has a consistent and clean format. The goal of the second part is to maximize the quality of comments by filtering out gibberish comments that contain a large number of words that never showed up in the dictionary of common English words and short comments that are toxic and contain minimal sentiment-related information.

Table 2: Sample Comments removed after Processing



Removed comment	Reason
grayrsi defaulthourly wkly monthly is still trending down but in hrlytrend up especially in momentum	dictionary-based filtering
but gold instead	Length-based filtering

4 Methodology

The price movement trend prediction is approached from two different methods. Sentiment-based and historical price-index based.

4.1 Historical price-index

The predictive power of the price index was investigated using stand-alone models to investigate andisolate the auto-regressive performance of the price index upon itself. The dependent variable beingthe 1-day directional price movement in the Bitcoin price index which is either 1 in the case that theclose price 1-day in the future is higher than or equal to the current close price or 0 when it is lower.

In keeping with the objective, two simple architectures were selected. The first was a bidirectional LSTM network with 2 layers of 64 neurons each followed by a linear layer to narrow the 128-dimensional output to a single neuron and a sigmoid activation which captures the binary response.The second, a multi-layer perceptron (MLP) network model, is a simple 3-layer model with 540, 256and 128 dimensions in the layers respectively which then are combined into a single output layer and sigmoid activation. While several architectures which combined an LSTM model with prior convolutions were also experimented with, none of these yielded results better than chance. As such,these final two models were chosen for simplicity as an initial baseline.

The historical data was summarized into daily segments and for each day the following features wereextracted: firstly, the daily close price and secondly the historical movement indicator data (indicating an up movement or down movement in the price). In the case of both variables, the prior 270 trading days worth of data was used to predict the following day's movements. The rationale being that thistime period would capture medium term seasonality which may occur in a year e.g., annual bonuses which people receive or seasonal investment flows.

In this case, one-fifth of the data was kept as a hold-out validation set and the rest was used to train the models. The models were trained over 20 epochs optimizing for a BCE loss function (Binary cross-entropy loss function) and using an Adam search optimization algorithm. The BCE loss function is the most appropriate in this scenario as we are predicting binary categorical variables.The model performance was measured using by calculating an accuracy percentage. The following table summarizes the results of the model performance:

Table 3: Best performance of two baseline models on Historical Price Models

	Architecture	
Model	LSTM based model	Simple MLP model
Training Accuracy	55.2%	64.05%
Training Loss	0.6639	0.7454
Validation Accuracy	54.81%	53.80%
Validation Loss	0.7159	0.6800

While the models may be only just better than chance, this is typical prediction performance for financial stocks and indices such as Bitcoin. The best performing model of the two is the LSTM Model with 54.8% overall accuracy. It is noted that the cost of error is asymmetrical for this context.Incorrectly predicting a down movement results in a missed opportunity for a trader with a long-onlystrategy while incorrectly predicting an up movement would result in portfolio losses.

4.2 Sentiment

4.2.1 Input/Embeddings

The improvements in the quality of the pretrained embeddings in the last few years have significantly boosted the accuracy of models related to NLP tasks such as machine translation, sentiment analysis,question answering. Earlier NLP models that are based on pretrained word-level embeddings have been gradually replaced by novel sentence-level embeddings. Word-level embeddings such as Glove (left) are able to capture the semantic similarity between words, on the other hand, sentence embeddings such as UHS or Infersent's major advantage is that it also captures syntactical information on top of word by word based semantic information.

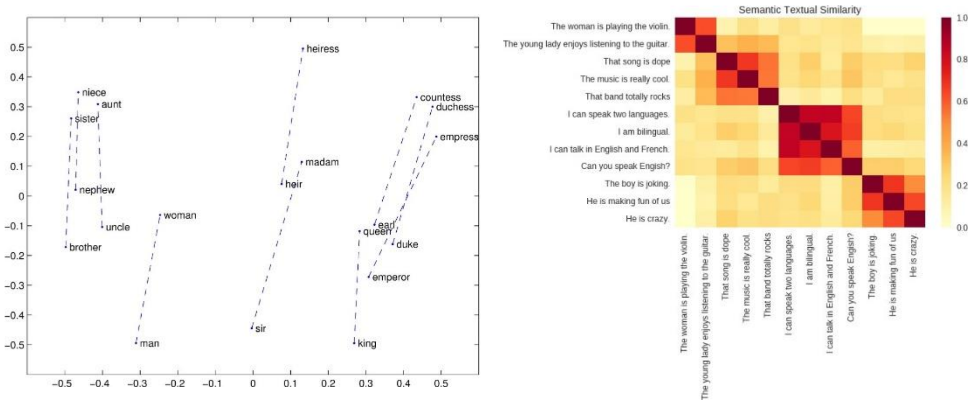


Figure 3: (a) Glove word-embedding (b) USE sentence-embedding

A simple visualization created for our raw data indicates how sentence-level embedding couldpotentially recognize the importance of each word and help improve the model training.

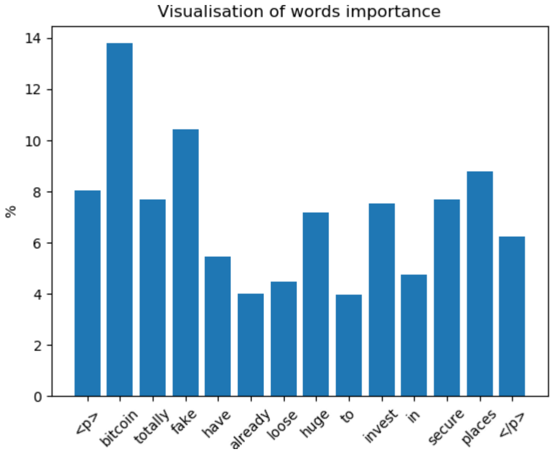


Figure 4: Infersent-based sentence embedding visualization

4.2.2 Architecture

The overall architecture contains three major components: the preliminary encoder, the secondary encoder for a window and the final decoder for prediction generation. The overall architecture is based on a CNN-LSTM baseline except that we introduce the concept of window size as a new hyper-parameter that can be experimented and tuned.

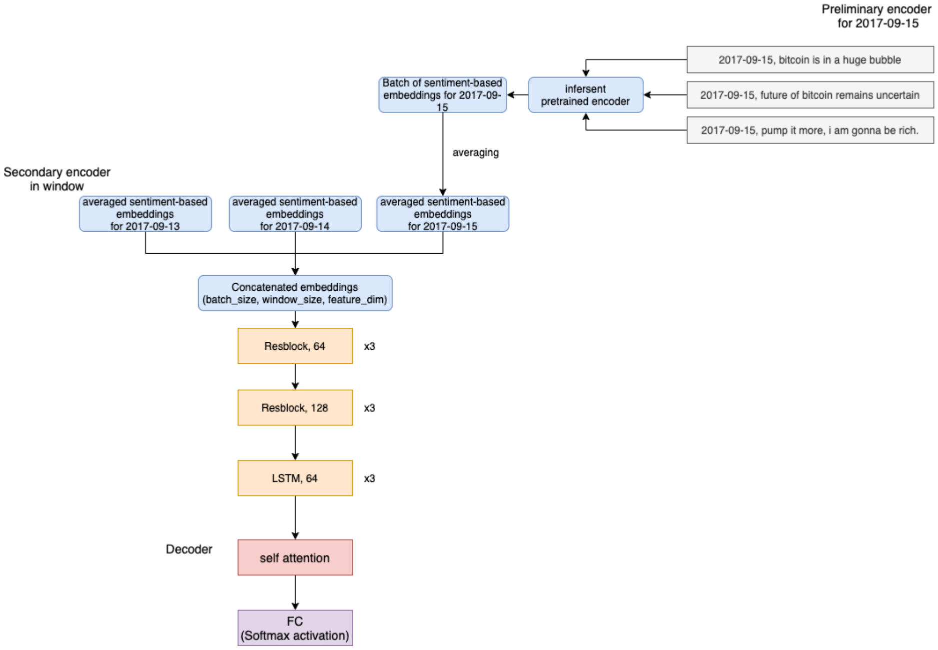


Figure 5: Overall architecture

In the first part, the preliminary encoder, we take the previously processed and filtered investing forum comments for each day and use the pre-trained infersent model to generate a high dimensional(4096), sentiment/semantic-based representation for each comment. The reason why the infersent model is able to capture the semantic information of a sentence is because it is trained based on the original glove/fast-text word embeddings using bidirectional LSTMs that learn the weights of individual words and the syntactic structure of a sentence. After all the embeddings are generated, we take the average of the batch of comments to represent the overall sentiment for that day.

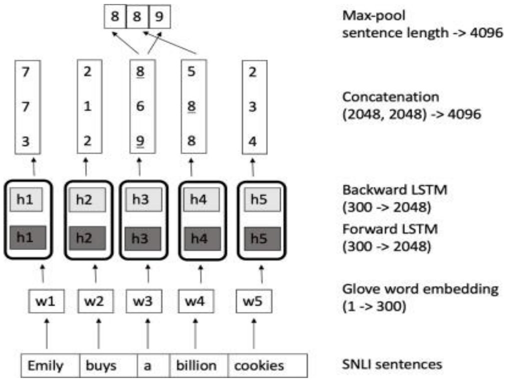


Figure 6: Infersent model

After the preliminary encoding is generated for our input for each day, the second part of the architecture focuses on feature extraction from the high-dimensional input. Here the concept of window_size is introduced. When asked to generate a prediction for the closing price for a given day, we use all comments predating the given date within window_size. (It is a tunable parameter) The weight/significance for each day is not necessarily the same.

Averaged comment for each day is vertically concatenated and passed into simple residual blocks with 64/128 output filters.The extracted features then goes into the LSTM layers as the last part of the encoding process.(The implementation of the resblock follows the design in [13]Deep Residual Learning for Image Recognition)

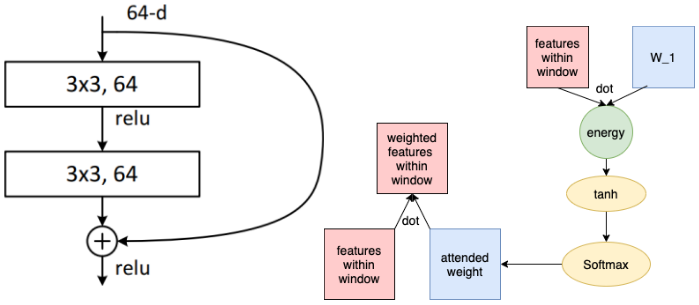


Figure 7: (a) Resblock (b) Self-attention

The decoder part of the network is just a simple fully connected layer with softmax activation preceded by a self-attention layer. The self-attention layer use the weight to learn a weighted version of the input, which is a concatenation of the features within the window. The purpose of using the attention to learn a weighting of the input is that we hypothesize that the importance of the overall market sentiment is different for each day (within the window) predating the date to be predicted. Forexample, if we want to predict whether the closing price for 2017-09-15 would increase/decrease from 2017-09-14, the more recent dates'(2017-09-14, 2017-09-15) market sentiment should be assigned ahigher weighting as some news could have come up to have more significant impact on the price. The formulas for obtaining the weighted-input through attention-weights are as follows.

$$f_{cat} = concat([features_i, i \in range(window\_size)]) \tag{1}$$

$$e = f_{cat} \cdot W \tag{2}$$

$$e = tanh(e) \tag{3}$$

$$a = \frac{exp(e)}{\sum exp(e)} \tag{4}$$

$$f_{cat} = f_{cat} \cdot a \tag{5}$$

4.2.3 Metric

Our hypothesis is that market sentiment could potentially predate the movement of price of bitcoin. However, it is unlikely that we could precisely quantify the change in market sentiment and use that to calculate the percentage change in bitcoin price.

Therefore, we transform the problem into a classification problem by producing 0/1 target labels denoting whether the closing price of the bitcoin decrease or increases from the previous day. In the experiment, we use the cross-entropy loss that ispopular for classification problems.

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

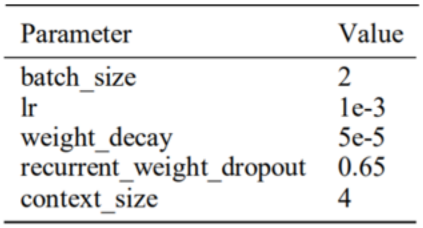
Figure 8: Xent loss

5 Experiments

5.1 Sentiment baseline model performance

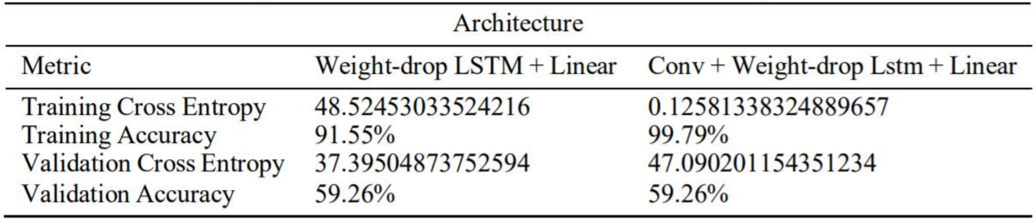
The table below summarizes the performance of two baseline models after training for 20 epochs using adams optimizer and the hyperparameters that we used during training. The plots are attachedin the appendix.

Table 4: Hypermeters for baseline model



Parameter	Value
batch_size	2
lr	1e-3
weight_decay	5e-5
recurrent_weight_dropout	0.65
context_size	4

Table 5: Best performance of two baseline models on Investing forum dataset



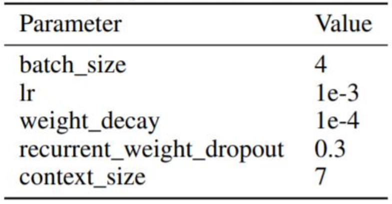
	Architecture	
Metric	Weight-drop LSTM + Linear	Conv + Weight-drop Lstm + Linear
Training Cross Entropy	48.52453033524216	0.12581338324889657
Training Accuracy	91.55%	99.79%
Validation Cross Entropy	37.39504873752594	47.090201154351234
Validation Accuracy	59.26%	59.26%

The model performance on validation plateaus around 5-10 epochs. Performance on validation seems jittery and over-fitting becomes a serious issue after 10 epochs without the usage of a scheduler to de-crease learning rate on plateau for validation loss. Overall, the CNN+weight-dropped-LSTM+Linear baseline has a more stable performance on validation above 50%. For both models, the training accuracy significantly increases while the validation performance actually gets worse with further training. It seems that feature extraction makes the model's training performance converge faster. The jittery update could be caused by hyperparameters (learning rate too high, batch_size too small, or context too big) or the noise in dataset and embedding. Further experimentation with different embeddings, text processing technique, hyperparameter tuning and architectural adjustments could potentially boost the validation accuracy to 65%.

5.2 Sentiment final model performance

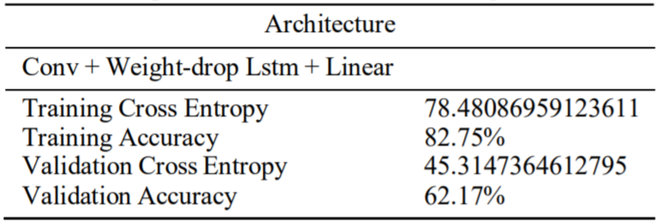
We experienced different parameters and were able to achieve an accuracy rate of 62.17% with smaller hidden dimension and bigger context size comparing to baseline model. The decrease of hidden dimension and the increase of context size stabilized the model performance. Performance on validation seems less jittery. The validation plateaus around 35-38 epochs and starts to rise. Theaccuracy rate hangs around 60% for a couple epochs and starts to drop.

Table 6: Hyperpameters for baseline model



Parameter	Value
batch_size	4
lr	1e-3
weight_decay	1e-4
recurrent_weight_dropout	0.3
context_size	7

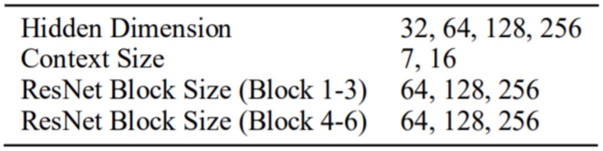
Table 7: Best performance models on Investing forum dataset



Architecture	
Conv + Weight-drop Lstm + Linear	
Training Cross Entropy	78.48086959123611
Training Accuracy	82.75%
Validation Cross Entropy	45.3147364612795
Validation Accuracy	62.17%

The table below summarizes the parameters tested. Based on the tuning experience with all combination of below parameters, generally we found smaller hidden dimensions perform better and are more stable than bigger hidden dimensions. A median context size is better than the baseline context size 4for this dataset.

Table 8: Parameters tested



Hidden Dimension	32, 64, 128, 256
Context Size	7, 16
ResNet Block Size (Block 1-3)	64, 128, 256
ResNet Block Size (Block 4-6)	64, 128, 256

6 Conclusion

6.1 Reflection

Overall, the architecture achieves slightly better performance than Ray Chen's experiment on stock price prediction using sentiments of tweets. However, performance of the model is still sub-optimal even when we add self-attention and several regularization techniques on top of the regular LSTM that is typically used for sentiment analysis.

The training accuracy steadily increases but the validation accuracy fails to break through 65% and thus we experience overfitting issues even though several regularization tricks such as locked dropout, embedding dropout, weight decay, decreasing the model size, were applied.

Window size

We have observed that the window_size actually have a huge impact on the performance of the model. For a window_size of 1, the training and validation performance is completely stagnant for the whole 50 epochs. On the other hand, slightly larger window_size like 14 which considers the market sentiment within two weeks before making a prediction seems to be having steady improvement ofperformance as training goes on.

Self-attention

The self-attention mechanism implementation is mainly based on [14]A Structured Self-Attentive Sentence Embedding paper by Zhouhan Lin. We did notice a slight improvement in the performanceof the model on top of the baseline by assigning a weight for market sentiments within the model, butas we failed to beat 65% validation performance, it is hard to conclude that it is effective.

6.2 Future work

The task is hard by nature as it is challenging for two major reasons.

1. It is hard to learn the market sentiment over a given window of time with noisy date scraped byourselves even though the forum is popular.

2. The actual relationship between market sentiment and bitcoin price could be weak.

Data collection

The comments on the investing forum are between 2017-09 and 2020-11. Therefore, the training/validation dataset size is quite small. There are only hundreds of target labels and the commentsfor those days used for training, which could be the reason why the model is having trouble generalizing.

Toxic comment filtering

The quality of the comments scraped from online sources is quite unstable. The length of the comments could vary from a few characters to over 500 characters. The processing techniques significantly helped by filtering the comments based on its length and content by checking whetherthe comment contains meaningless words or emoji or digits which could interfere with training. However, the processing ignores the semantics of the comment. [15]The Semantically Enriched Machine Learning Approach to Filter YouTube Comments for Socially Augmented User Models presented a way of filtering the comments by computing a relevance score for each of the comment based on a bag of words and setting a threshold for labelling the comments.

7 References

[1] Velankar, S., Valecha, S. and Maji, S. (2018). Bitcoin price prediction using machine learning. IEEE Xplore.

[2] Serafini, G., Yi, P., Zhang, Q., Brambilla, M., Wang, J., Hu, Y. and Li, B. (2020). Sentiment-Driven PricePrediction of the Bitcoin based on Statistical and Deep Learning Approaches. IEEE Xplore.

[3] Raju, S M Tarif, Ali. (2020). Real-Time Prediction of BITCOIN Price using Machine Learning Techniquesand Public Sentiment Analysis.

[4] Chen, R. and Lazer, M. (n.d.). Sentiment Analysis of Twitter Feeds for the Prediction of Stock MarketMovement.

[5] Dixon, M.F., Klabjan, D. and Bang, J.H. (2016). Classification-Based Financial Markets Prediction Using Deep Neural Networks. SSRN Electronic Journal.

[6] McNally, S., Roche, J. and Caton, S. (2018). Predicting the Price of Bitcoin Using Machine Learning. 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP).

[7] Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.

[8] Pensions Investments. (2017). 80% of equity market cap held by institutions.

[9] GmbH, finanzen net (n.d.). As many as 36% of large investors own crypto assets, and bitcoin is the most popu-lar, Fidelity says | Currency News | Financial and Business News | Markets Insider. markets.businessinsider.com.

[10]T. Fischer and C. Krauss, “Networks for Financial Market Predictions,” FAU Discuss. Pap. Econ. No.11/2017, Friedrich-

Alexander-Universität Erlangen-Nürnberg, Inst. Econ. Erlangen, pp. 1–34, 2017.

[11]Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

[12]Merity, S., Keskar, Nitish Shirish and Socher, R. (2017). Regularizing and Optimizing LSTM LanguageModels. arXiv.org.

[13]Lin, Z., Feng, M., Nogueira, C., Santos, M., Yu, B., Xiang, B., Zhou, Y., Bengio and Watson, I. (n.d.). A STRUCTURED SELF-ATTENTIVE SENTENCE EMBEDDING. [online] arXiv.org.

[14]Ammari, A., Dimitrova, V., Despotakis, D. (2011). Semantically Enriched Machine Learning Approach to Filter YouTube Comments for Socially Augmented User Models.

8 Appendix

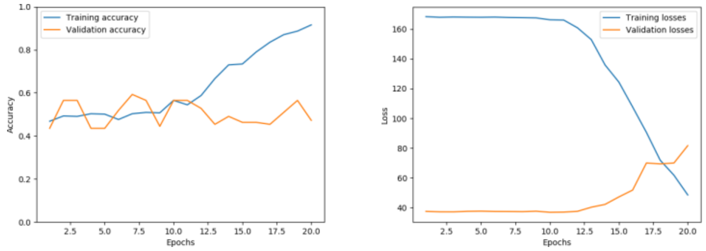


Figure 9: Weight-Dropped LSTM+Linear accuracy & loss

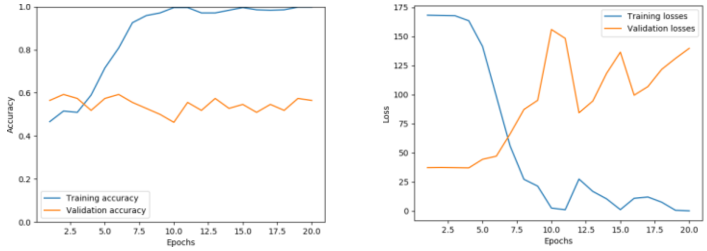


Figure 10: Conv+Weight-Dropped LSTM+Linear accuracy & loss

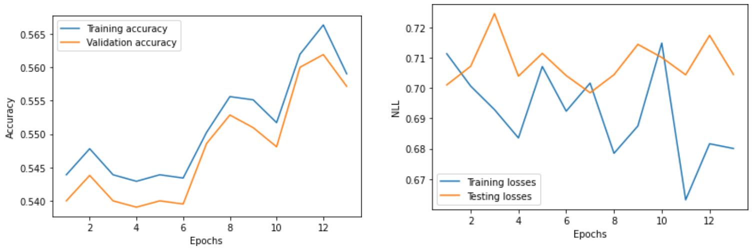


Figure 11: Historical price only LSTM model accuracy & loss

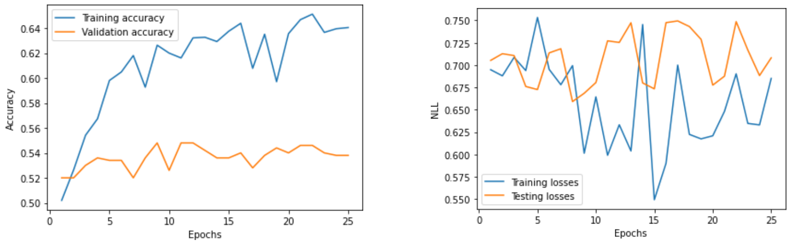


Figure 12: Historical price only MLP model accuracy & loss

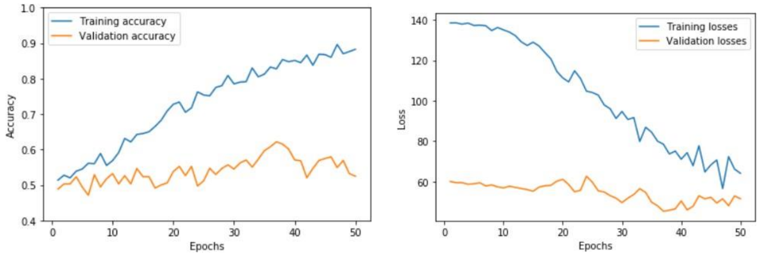


Figure 13: Final model accuracy & loss