INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 4 - V7I4-1286)
Available online at: https://www.ijariit.com

Algorithmic Trading with Deep Learning

Mayank Anuragi
mayankanuragi2013@gmail.com
Atria Institute of Technology, Bengaluru, Karnataka

Rakshit Ramnath Naik
rakshitnaik79@gmail.com
Atria Institute of Technology, Bengaluru, Karnataka

Manoj Galanki
manojgalanki@gmail.com
Atria Institute of Technology, Bengaluru, Karnataka

Kavitha S. Patil
kavitha.patil@atria.edu
Atria Institute of Technology, Bengaluru, Karnataka

ABSTRACT

In this paper, Artificial Neural Network (ANN) is used to predict trading signals like Strong Buy, Buy, Neutral, Strong Sell and Sell a total of 5 signals. The ANN outputs one of the trading signals based on the market data of cryptocurrencies. The ANN makes use of the candle-stick data along with additional data like Volume, Number of Trades, etc and technical indicators to predict the signals. The ANN is trained on multiple intervals of the market, i.e. 1 minute, 3 minutes, 5 minutes and so on up to 3 days in the intervals. Along with the market data, the paper also makes use of the news articles to better predict the trading signals, this is only used when the ANN predicts the Buy or Sell, and by using the news articles those Buy and Sell signals are converted to Strong Buy and Strong Sell respectively. The ANN was trained on 3 years of data, and Google's BERT [1] model was trained on almost 1000 news article's titles on topics related to cryptocurrencies and the model outputs whether it's a positive or negative title. The accuracy of the ANN models for all the markets is in the range of 85 - 90% and the same accuracy is observed with the BERT model which is trained on the news article's title.

Keywords— *Algorithmic trading, artificial neural network, Google BERT, cryptocurrency market.*

1. INTRODUCTION

Deep learning is one of the major upcoming methods in the family of machine learning, this is because the modern CPU and GPU is much more computationally capable when compared to the older version. With more and more capable hardware deep learning is finding more and more applications in solving complex tasks. One of the tasks where deep learning is evolving is in predicting trading signals and pricing forecasts. In this paper, we are using the ANNs to predict the trading signals. The cryptocurrency markets are quite volatile, which make a good market to test the capabilities of ANNs to predict the trading signals. The main idea is to take the candle-stick data of a single market with multiple intervals of the candle-stick i.e. 1 minute, 3 minutes, 5 minutes, 15 minutes, 30 minutes, 1 hour, 2 hours, 4 hours, 6 hours, 8 hours, 12 hours, 1 day and 3 days and then calculate the technical indicator values, the technical indicators used in this paper are Bollinger Bands, Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), Moving Average (MA) Rate of Change Percentage (ROCP), and Parabolic SAR. After calculating the values of the technical indicators, we label the data based on the voting system. The voting system comprises different strategies based on the technical indicator, each strategy considers only the Open-High-Low-Close (OHLC) data of the single record and one of the technical indicators for that particular record. Each strategy return one of the five trading signal and the most frequent trading signal is considered as the final trading signal for that particular record. This process repeats for each interval. Since we are also considering the news of the particular market. While making predictions for the trading signal we will consider the news articles in the past 24hours and predict which news articles are positive and negative. If the ANN predict either the Sell or Buy (normal signal), we will take a look at what the news articles say, if there is positive news we will convert the Buy to Strong Buy and if the news is negative we will convert the Sell to Strong Sell. We are using a smaller version [8] of Google's BERT model i.e.(L-4, H-768, A-12) to predict the result of the news article's title. The model was trained on almost 1000 news articles title.

This labelled data of all the intervals are then stacked and then processed before the ANN is trained on. The accuracy metrics of all the ANN from different markets were in the range of 85 - 90%.

2. RELATED WORK

Initially, the trading signals were generated based on algorithms that keep track of the market movement and when a certain condition is met, it triggers a trading signal based on the market trend. Ever since deep learning has shown promising results in the various domains there have been multiple papers published

in predicting the stock market prices and also predicting the trading signals.

O.B. Sezer [2] proposed using ANN to train with technical indicators like RSI, MACD, Williams %R as features and their results were compared against the "Buy and Hold" (BAH) strategy without fine-tuning the Multilayer Perceptron (MLP).

O.B. Sezer [3] proposed using the Convolutional Neural Networks (CNN) to predict the trading signals by converting the time-series data of the stock prices into a 2-D image of 30x30 pixels with Y-axis as price and X-axis as time. This was an unconventional approach and the results outperformed the BAH strategy.

Pepic. M [4] developed three strategies, Gap Strategy, Moving Average Strategy and Crossover Strategy and were part of a system of systems that was aimed to be profitable. Each system had its own parameters and time frames. The result was that each system was profitable.

Matthew D. [5] described the application of Deep Neural Networks (DNNs) in predicting the financial markets. The authors had tested a simple trading strategy on 43 different commodities and FX futures at 5-minute intervals. It was concluded that the DNNs substantial predictive capabilities in predicting the market movement.

3. METHODOLOGY

First, the data from the exchange is saved to a local disk from the time the symbol is available on exchange till 24th May 2021 for all the intervals i.e. 1 minute, 3 minutes, 5 minutes, 15 minutes, 30 minutes, 1 hour, 2 hours, 4 hours, 6 hours, 8 hours, 12 hours, 1 day and 3 days for a particular symbol. The next step is to calculate the values of the technical indicators for each interval. After calculating the technical indicators there will be rows where the values cannot be calculated as there is not enough data and those will contain the null or NaNs, to overcome this we will be dropping all the rows which contain NaNs, The technical indicators used in this literature are Bollinger Bands, RSI, MACD, MA, ROCP and SAR. Once the technical indicators are calculated, we apply the strategies for each interval separately. There are a total of five strategies developed in this literature called BBands Strategy, MACD Strategy, RSI Strategy, Moving Average Strategy and Volume Strategy. Each strategy considers the OHLC data for the current row and respective technical indicator as per the name of the strategy.

The main idea in BBands Strategy is to consider buying or selling when the close price is around the edges of the Bollinger bands. Figure 1 shows the basic idea of the points where we should consider buying or selling. The red area is where we should sell and the green area is where we should buy. As per the figure, we can see that the best time is when the close prices have crossed the Bollinger bands.

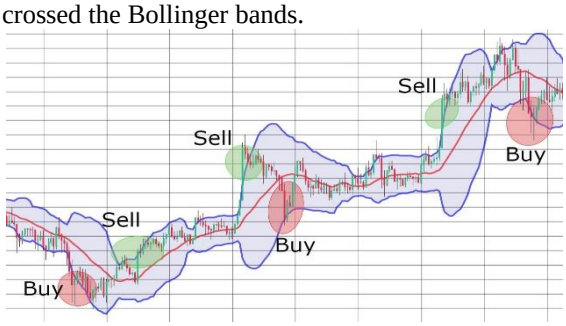

Figure-1: BBands Strategy idea

The idea as per Figure 1 can be translated into the function given as:

$$f(x) = \begin{cases} SBuy, if\ CO < BBM\ and\ CC < BBL - 0.012 * BBL \\ Buy,\ if\ CO < BBM\ and\ CO < BBL + UBOP * BBL \\ SSell, if\ CO > BBM\ and\ CO > BBU + 0.012 * BBU \\ Sell, if\ CO > BBM\ and\ CO > BBU - LBOP * BBU \\ Neutral,\ otherwise \end{cases}$$

Where,
SBuy = Strong Buy
SSell = Strong Sell
CO = Candle Open Price
CC = Candle Close Price
BBL = Lower Band of Bollinger Bands
BBM = Middle Band of Bollinger Bands
BBU = Upper Band of Bollinger Bands
UBOP = Upper Bound offset percentage
Given as

$$UBPO = \frac{(BBM - BBL)}{BBL} * 0.4$$

LBPO = Lower Bound offset percentage
Given as

$$LBPO = \frac{(BBU - BBM)}{BBM} * 0.4$$

Although there is an overlap in the condition with Strong Buy and Buy as well as Strong Sell and Sell, programmatically it considers the Strong signals first and then the normal signals.
The main idea of the MACD Strategy is to take the extreme peaks and valleys in the MACD Histogram and accordingly label the data. The Figure shows the candle-stick chart and along with it the MACD indicator at the bottom, the violet colour represents the Histogram values of the MACD indicator. From Figure 2 we can see that some of the high and low points are the good time to make the trade.

Figure-2: MACD Strategy idea

In MACD Strategy a row is labelled as Sell or Strong Sell if the value of the MACD Histogram is above a certain percentile, from all the positive values in the MACD Histogram and a row is labelled as Buy of Strong Buy if the value of the MACD Histogram is below a certain percentile from all the negative values of the MACD Histogram. For strong and normal signals each has its own percentile threshold.

The threshold for strong and normal signals are given as the following:

$$f(x) = \begin{cases} SSP = 95, NSP = 85, if\ 1m' \leq interval \leq' 5m' \\ SSP = 80, NSP = 70, \quad otherwise \end{cases}$$

Where,
SSP = Strong Sell Percentile
NSP = Normal Sell Percentile
To label the Buy and Strong Buy the percentile value is given as:

$$SBP = 100 - SSP$$
$$NBP = 100 - NSP$$

Where,
SBP = Strong Buy Percentile
NBP = Normal Buy Percentile
Once the percentile values are obtained, the percentiles at SSP and NSP are taken from all the positive values of the MACD Histogram which act as threshold values and by using the following function the row is labelled accordingly:

$$f(x) = \begin{cases} SSell, & if\ MACDH > SSPMACDH \\ Sell, & if\ MACDH > NSPMACDH \end{cases}$$

Where,
MACDH = MACD Histogram value
SSPMACDH = MACD Histogram value at SSP percentile
NSPMACDH = MACD Histogram value at NSP percentile
For labelling Buy or Strong Buy, the process is similar, but all the negative values are considered and the values at SBP and NBP percentile taken as threshold.
The function to label Buy or Strong Buy is given as:

$$f(x) = \begin{cases} SBuy, & if\ MACDH < SBPMACDH \\ Buy, & if\ MACDH < NBPMACDH \end{cases}$$

Where,
SBPMACDH = MACD Histogram value at SBP percentile
NBPMACDH = MACD Histogram value at NBP percentile
If none of the conditions is valid from the sell and buy functions then label it as Neutral.

Although there is an overlap in the condition with Strong Buy and Buy as well as Strong Sell and Sell, programmatically it considers the Strong signals first and then the normal signals. The Volume strategy is also quite similar to the MACD Strategy. The main idea is to check if the volume traded crosses a certain threshold, and based on whether the candle close is below or above, the data is labelled accordingly. Figure 3 shows the volume strategy idea.

Figure-3: Volume Strategy idea

To get the threshold value first we have to decide on the nth percentile value of the volume.

To percentile value is given as:

$$f(x) = \begin{cases} SP = 95, NP = 85, if\ 1m' \leq interval \leq' 5m' \\ SP = 80, NP = 70,\ otherwise \end{cases}$$

Where,
SP = Stong Percentile
NP = Normal Percentile
Once percentile values are obtained, the percentile at SP and NP are taken from the Volume.
The following function is then used to label the rows:

$$f(x) = \begin{cases} SSell, & if\ CC > CO\ and\ V > SPV \\ Sell, & if\ CC > CO\ and\ V > NPV \\ SBuy, & if\ CO > CC\ and\ V > SPV \\ Buy, & if\ CO > CC\ and\ V > NPV \\ Neutral, & otherwise \end{cases}$$

Where,
V = Volume value
SPV = Volume value at SP percentile
NPV = Volume value at NP percentile
Although there is an overlap in the condition with Strong Buy and Buy as well as Strong Sell and Sell, programmatically it considers the Strong signals first and then the normal signals.
The RSI Strategy is one of the simplest strategy to implement. The row is labelled based on the threshold value.
The threshold value for RSI Strategy is given as:

$$f(x) = \begin{cases} SSell, & if\ RSI \geq 70 \\ Sell, & if\ RSI \geq 63 \\ SBuy, & if\ RSI \leq 27 \\ Buy, & if\ RSI \leq 32 \\ Neutral, & otherwise \end{cases}$$

The function can be visualized in Figure 4, where the upper limit is set to 70 and the lower limit is set to 30.

Figure-4: RSI Strategy idea

Although there is an overlap in the condition with Strong Buy and Buy as well as Strong Sell and Sell, programmatically it considers the Strong signals first and then the normal signals. The last strategy is the Moving Average Strategy. This strategy considers the Moving Average at 7, 25 and 99 time periods.

The data is labelled based on the following function:

$$f(x) = \begin{cases} SSell,\ if\ CO > (M7 + M7 * 0.01)\ and\ INCRM \\ Sell,\ if\ CO > (M7 + M7 * 0.008)\ and\ INCRM \\ SBuy,\ if\ CO < (M7 - M7 * 0.01)\ and\ DECRM \\ Buy,\ if\ CO < (M7 - M7 * 0.008)\ and\ DECRM \\ Neutral,\ otherwise \end{cases}$$

Where,
M7 = Moving Average with Time Period set to 7
M25 = Moving Average with Time Period set to 25
M99 = Moving Average with Time Period set to 99
INCRM = M7 > M25 > M99
DECRM = M7 < M25 < M99
The Moving Average Strategy can be visualized in Figure 5. The red, green, blue lines are Moving Average at 7, 25,99 respectively.

Figure-5: Moving Average Strategy idea

All the above-mentioned strategies are used to label the data and are done with the help of a voting system to decide on the final trading signal, the most voted trading signal is considered as the final trading signal for that particular row. If there is a conflict between the number of votes then the first signal is taken as per alphabetical order and if there is a conflict with the Neutral then Neutral is given the priority.

Once the data is labelled, there might be an imbalance in the classes and if trained on the imbalanced dataset then the ANN will not be able to map the independent variable to the dependent variable. To avoid this we will be using imbalanced-learn [5] a python package that allows to balance the label. We first under-sample the majority class which is Neutral and then apply Synthetic Minority Over-sampling Technique (SMOTE) to all the classes. By doing this we will now have a balanced dataset and the ANN will not have trouble mapping the independent variable to the dependent variable. Although the SMOTE technique has generated the new data with around 90% accuracy which we have verified by applying the strategies to the new data, we went ahead and relabelled the new data with our strategies to have consistency in our dataset.

Now, that our dataset is labelled and has balanced classes, we need to process it so that ANNs can learn better. Since a single Deep Learning (DL) model will be learning on all the intervals, we will need to differentiate one interval data from other interval data. We can do this by including the categorical column that will represent the interval of that particular row. We will of course need to one-hot encode the categorical variable and drop one of the categories to avoid the dummy variable trap. Once all the interval data is one-hot encoded, it is then stacked which forms one single dataset.

To make the dataset trainable, we have used Quantile Transformer from scikit-learn [7] package. We are using the default option for output distribution i.e. uniform distribution. All the features are scaled except for the one-hot encoded features. The dataset now is ready to be trained, and upon training, we found that the categorical accuracy was in the range of 70-80% with training loss in the neighbourhood of 0.4. This was then improved by using the PolynomialFeatures from scikit-learn. Polynomial features were generated with a polynomial degree of two and only interactions were included and the bias term was not included. By applying the PolynomialFeatures the accuracy was improved greatly to around 85-90% with training loss in the neighbourhood of 0.2.

The neural network architecture is shown in Figures 6 and 7.

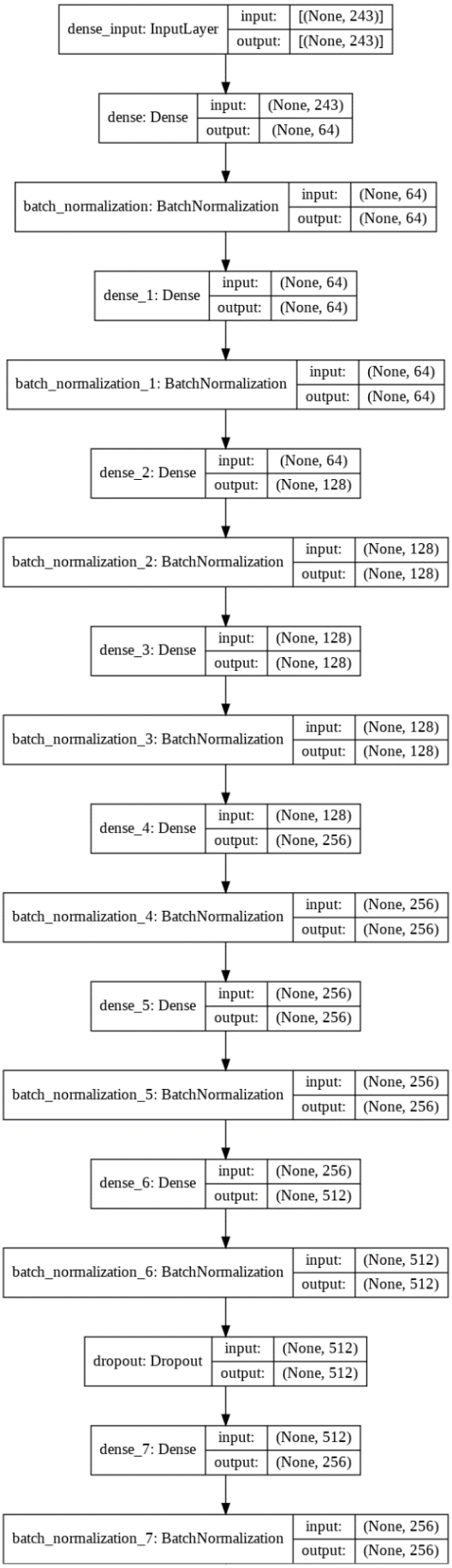

Figure-6: ANN Architecture Top Half (Continued in Figure 7)

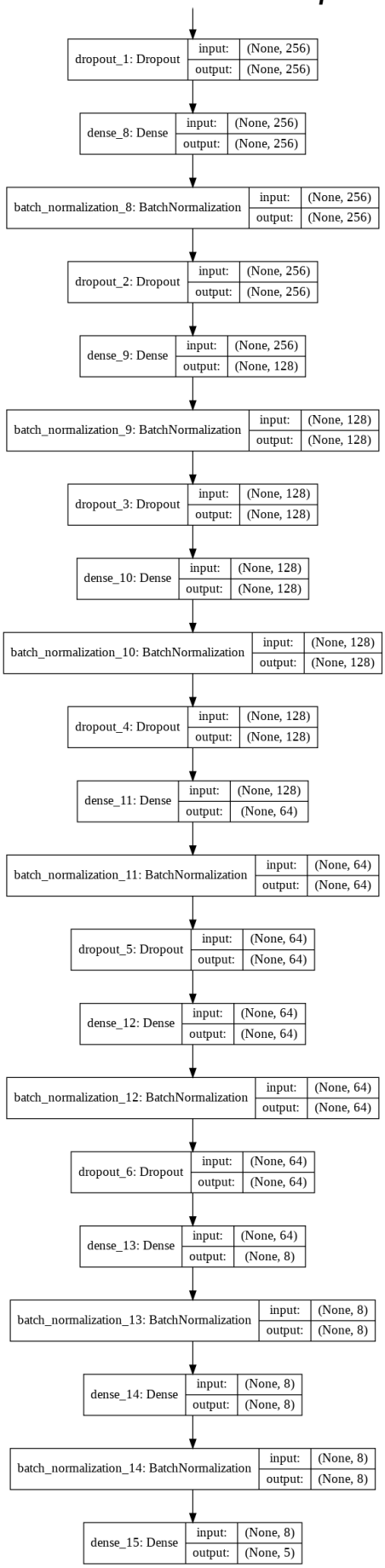

Figure-7: ANN Architecture Bottom Half (Continuation of Figure 6))

Figure 8 shows the accuracy plot for BTCUSDT training.

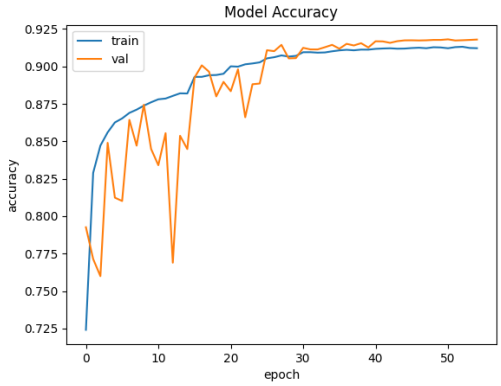

Figure-8: Accuracy Plot for BTCUSDT

Figure 9 shows the loss for BTCUSDT training.

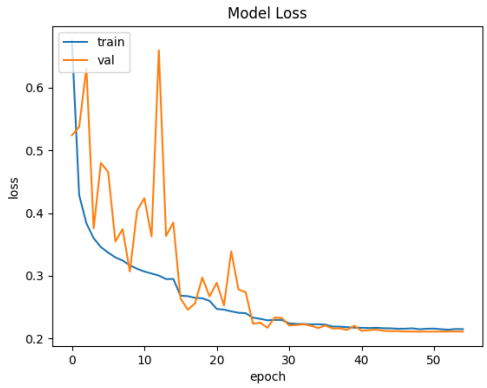

Figure-9: Loss Plot for BTCUSDT

Below Table 1 shows a normalized confusion matrix rounded to 2 decimal places of the prediction on the BTCUSDT market:

Table-1: Confusion matrix from the ANN model for BTCUSDT market

Actual \ Predicted	Buy	Sell	Neutral	SBuy	SSell
Buy	0.93	0	0.01	0.06	0
Sell	0	0.93	0.01	0	0.06
Neutral	0.03	0.03	0.92	0.01	0.01
SBuy	0.07	0	0	0.92	0
SSell	0	0.08	0	0	0.92

For training Google's BERT model, we have chosen to go with a smaller version of BERT [8], as we didn't have enough resources to run both the ANN and BERT together. The config of the smaller BERT is L-4, H-786, A-12. The dataset was manually labelled by reading the title and labelling it accordingly. The label is one of the Positive, Neutral and Negative. Since we are passing the output of the model to the sigmoid function, the data is labelled by the following function after the prediction is done.

$$f(x) = \begin{cases} Positive, & if\ sigmoid(pred) \geq 0.65 \\ Negative, & if\ sigmoid(pred) \leq 0.4 \\ Neutral, & otherwise \end{cases}$$

Where,
pred = Output of the BERT model

4. EVALUATION

We have created a backtesting system, which can simulate trading and it will give us an idea of how it will perform in the

real world. The initial amount is set to $1000 and on every trade, it will invest the complete amount when the model suggestion is Strong Buy, it will consider both the models, the ANN model and the output of the BERT model for news titles. The system also skips a certain amount of signal, as the model output the signal as soon as it finds it, so we will have to wait for a certain number of trading signals and then make the trade. Figure 10 represents the output of the ANN predictions and it can be seen that the moment the model detects the trading signal, there are more successive signals following, and we need to wait until the last moment to get the maximum profits. The system also has a stop-loss feature that will get out of the trade if it detects a 5% loss

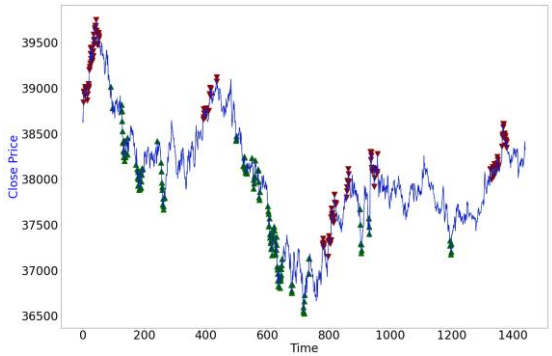

Figure-10: Predicted Trading Signals on BTCUSDT. Green arrows represent the Strong Buy trading signal and red represents the Strong Sell trading signal. Y-axis is the close price and X-axis is the time in a 1-minutes time frame.

Figure 11 shows the simulated trading from 25th May 2021 to 5th July 2021 on a 6-hour time frame of the BTCUSDT market. The yellow line represents the portfolio value with the right Y-axis and the blue line represent the close value on the left Y-axis and the X-axis is the time with each point 6-hours apart. The red markers are the sell points and the green ones are the buy points. Since the time frame shown is 6-hours and multiple trades are happening in between, therefore 2 successive green or sell markers can be seen.

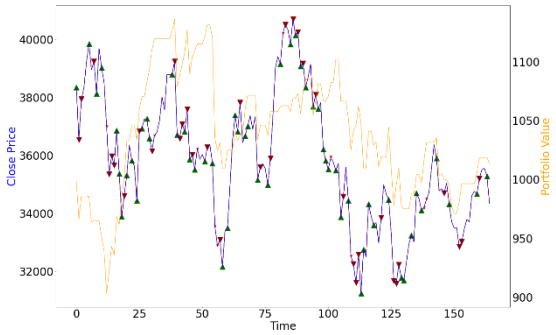

Figure-11: Simulated trading of BTCUSDT from 25th May 2021 to 5th July 2021

Table 2 shows the results of the testing system in which the initial investment is $1000 and the market is simulated from 25th May 2021 to 5th July 2021. The Portfolio column is the final value after the testing, Profit % represent the profit in percentage, the Market % represents the percentage difference from the start of the market to the end of the market, Max Market % represents the highest point of the market during testing in percentage, Min Market % represents the lowest point of the market during the testing in percentage, Best trade represents the best trade done out of all the trades i.e a single trade with the max profit, Worst trade represents the worst trade done out of all the trades i.e. a single trade with the maximum loss, Max portfolio represents the maximum portfolio value during the testing and the Min portfolio represents the minimum portfolio value during the testing. We can see that the profit % isn't good, but the model did manage to avoid big losses, as we can see the lowest market change on average is around 30% but the minimum portfolio during the same time is only around 15-25%, which we believe is a good result and the final portfolio also isn't less than -11%. The average number of trades for all the markets were in the neighbourhood of 20 trades.

Table-2: Results of Testing System from 25th May 2021 to 5th July 2021

Market	*Portfolio*	*Profit*	*Market*	*Max market (%)*	*Min market (%)*	*Best trade (%)*	*Worst trade (%)*	*Max portfolio*	*Min portfolio*
BTCUSDT	1068.89	6.89%	-9.16%	6.91	-25.32	3.48	-5.57	1147.23	892.29
ETHUSDT	904.94	-9.51%	-12.01%	10.31	-35.33	3.88	-5.39	1038.61	769.65
ADAUSDT	1066.37	6.64%	-5.68%	22.46	-35.11	12.51	-6.41	1174.47	879.50
BNBUSDT	1024.19	2.42%	-10.82%	25.96	-34.15	35.63	-5.94	1254.76	846.76
XMRUSDT	959.12	-4.09%	-13.77%	24.13	-28.91	7.68	-5.67	1075.60	759.24
DOTUSDT	1126.02	12.60%	-30.95%	24.05	-43.40	4.64	-5.45	1130.47	897.59
LTCUSDT	1021.99	2.20%	-21.41%	13.41	-42.74	6.87	-5.55	1243.67	932.00
XRPUSDT	896.49	-10.35%	-29.61%	11.54	-48.09	3.46	-5.57	1081.26	861.65

5. CONCLUSION

In this paper, we have developed five different strategies based on indicators that are independent of each other and can be ensembled with other strategies from other literature. We have also trained ANNs for each market for 13-time frames with good accuracy to predict the trading signal. We have also used trained and used Google's BERT model to figure out how the markets are doing in the news. Although the results are not comparable to simple buy and hold strategy, but there can be improvements on how to make the best use of the outputs of the ANNs as we can see the ANNs can predict the trading signals at the right time, but we can see the same trading signals successively. This can be improved by changing the parameters in the strategies proposed or by considering the rate of the trading signals generated. Due to lack of time we have only trained and tested on the cryptocurrency market, which is highly volatile, we believe the proposed method would do better in the stock market which is less volatile when compared to the cryptocurrency market, this can be considered as future work. Along with testing on the stock market for future work, the strategies can also be optimized better for each interval.

6. REFERENCES

[1] Jacob Devlin and Ming-Wei Chang and Kenton Lee and Kristina Toutanova (2018). BERT: Pre-training of

Deep Bidirectional Transformers for Language Understanding *arXiv:1810.04805*, 2018.

[2] Omer Berat Sezer, A. Murat Ozbayoglu, and Erdogan Dogdu. 2017. An Artificial Neural Network-based Stock Trading System Using Technical Analysis and Big Data Framework. In Proceedings of the SouthEast Conference (ACM SE '17). Association for Computing Machinery, New York, NY, USA, 223–226. DOI:https://doi.org/10.1145/3077286.3077294

[3] Omer Berat Sezer and A. Murat Ozbayoglu and Erdogan Dogdu (2017). An Artificial Neural Network-based Stock Trading System Using Technical Analysis and Big Data Framework. *arXiv:1903.04610v1*

[4] Pepic, M., William, M., & Fancher, J. (2016). Algorithmic Trading and Investment System Development. : Worcester Polytechnic Institute.

[5] Matthew Dixon and Diego Klabjan and Jin Hoon Bang (2016). Classification-based Financial Markets Prediction using Deep Neural Networks. *arXiv:1603.08604v2*

[6] Guillaume Lemaître, Fernando Nogueira, & Christos K. Aridas (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. Journal of Machine Learning Research, 18(17), 1-5.

[7] Lars Buitinck and Gilles Louppe and Mathieu Blondel and Fabian Pedregosa and Andreas Mueller and Olivier Grisel and Vlad Niculae and Peter Prettenhofer and Alexandre Gramfort and Jaques Grobler and Robert Layton and Jake VanderPlas and Arnaud Joly and Brian Holt and Gaël Varoquaux (2013). API design for machine learning software: experiences from the scikit-learn project.

[8] Turc, I., Chang, M.W., Lee, K., & Toutanova, K. (2019). Well-Read Students Learn Better: On the Importance of Pre-training Compact Models. arXiv preprint *arXiv:1908.08962v2* .

