[Team 42] ProjF4 Final Report: LSTM and CNN Based Stock Market Predictor

Alexander Kyu
Email: awkyu@ncsu.edu

Anthony Kyu
Email: amkyu@ncsu.edu

Michael McKnight
Email: mpmcknig@ncsu.edu

I. ABSTRACT

Predicting the stock market is a challenging task that has been attempted with a variety of methods. In this project, a neural network model that combines aspects of both CNNs and LSTMs to predict the appropriate position of a company's stock (Buy, Sell, or Hold) using stock data and technical indicators commonly used by analysts. The proposed model outperformed the baseline model, but still yielded unfavorable results.

II. MOTIVATION – MAKE DUMMY THICCCCC STACKS

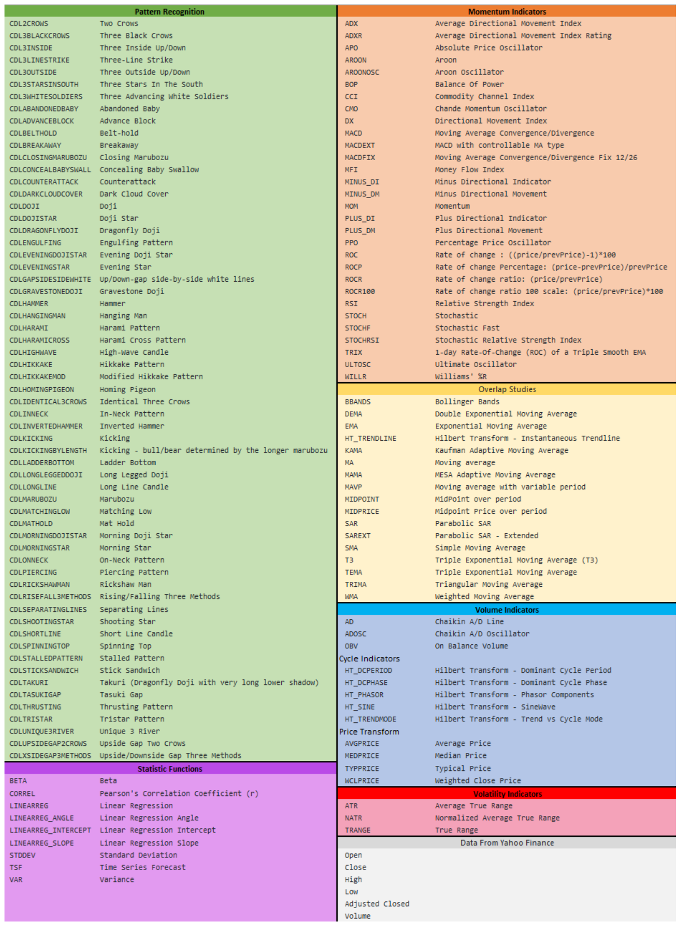
Predicting the stock market has always been a classical neural network problem. However, very few, if any, models (CNN or LSTM) have been very successful in predicting outcomes of the stock market accurately [1]. Therefore, we propose a new neural network model based off literature that combines a convolutional neural network with a LSTM neural network model in hopes to surpass the results of current models [2].

We want to build a model which will be able to predict appropriate stock market positions (Buy, Sell, or Hold) for specific ticker symbols (Advanced Micro Devices -AMD) using various key statistics and market indicators. These are essentially extracted features from basic stock data such as opening and closing prices. With stocks being driven primarily by investor movement and business cycles, it may be increasingly important to understand if these movements can be captured by these market indicators and statistics.

As a problem statement, many current models only use basic stock data, which may lack crucial features for prediction. A model that uses statistics that captures changes over time may improve the accuracy of a Neural Network. A technical difficulty that could arise is the inherent randomness of the stock market. Because news is unpredictable and somewhat random and people are driven to buy shares based on news, these indicators could possibly be less accurate in predicting trends.

III. DATA

The data was collected from Yahoo Finance's historical stock data, which includes open price, high, low, closing price, adjusted closing price, and volume [3]. The data was further augmented by creating over 150 technical indicators using the TA-Lib Library.



Pattern Recognition	
CDL2CROWS	Two Crows
CDL3BLACKCROWS	Three Black Crows
CDL3INSIDE	Three Inside Up/Down
CDL3LINESTRIKE	Three-Line Strike
CDL3OUTSIDE	Three Outside Up/Down
CDL3STARSINSOUTH	Three Stars In The South
CDL3WHITESOLDIERS	Three Advancing White Soldiers
CDLABANDONEDBABY	Abandoned Baby
CDLADVANCEBLOCK	Advance Block
CDLBELTHOLD	Belt-hold
CDLBREAKAWAY	Breakaway
CDLCLOSINGMARUBOZU	Closing Marubozu
CDLCONCEALBABYSWALL	Concealing Baby Swallow
CDLCOUNTERATTACK	Counterattack
CDLDARKCLOUDCOVER	Dark Cloud Cover
CDLDOJI	Doji
CDLDOJISTAR	Doji Star
CDLDRAGONFLYDOJI	Dragonfly Doji
CDLENGULFING	Engulfing Pattern
CDLEVENINGDOJISTAR	Evening Doji Star
CDLEVENINGSTAR	Evening Star
CDLGAPSIDESIDEWHITE	Up/Down-gap side-by-side white lines
CDLGRAVESTONEDOJI	Gravestone Doji
CDLHAMMER	Hammer
CDLHANGINGMAN	Hanging Man
CDLHARAMI	Harami Pattern
CDLHARAMICROSS	Harami Cross Pattern
CDLHIGHWAVE	High-Wave Candle
CDLHIKKAKE	Hikkake Pattern
CDLHIKKAKEMOD	Modified Hikkake Pattern
CDLHOMINGPIGEON	Homing Pigeon
CDLIDENTICAL3CROWS	Identical Three Crows
CDLINNECK	In-Neck Pattern
CDLINVERTEDHAMMER	Inverted Hammer
CDLKICKING	Kicking
CDLKICKINGBYLENGTH	Kicking - bull/bear determined by the longer marubozu
CDLLADDERBOTTOM	Ladder Bottom
CDLLONGLEGGEDDOJI	Long Legged Doji
CDLLONGLINE	Long Line Candle
CDLMARUBOZU	Marubozu
CDLMATCHINGLOW	Matching Low
CDLMATHOLD	Mat Hold
CDLMORNINGDOJISTAR	Morning Doji Star
CDLMORNINGSTAR	Morning Star
CDLONNECK	On-Neck Pattern
CDLPIERCING	Piercing Pattern
CDLRICKSHAWMAN	Rickshaw Man
CDLRISEFALL3METHODS	Rising/Falling Three Methods
CDLSEPARATINGLINES	Separating Lines
CDLSHOOTINGSTAR	Shooting Star
CDLSHORTLINE	Short Line Candle
CDLSPINNINGTOP	Spinning Top
CDLSTALLEDPATTERN	Stalled Pattern
CDLSTICKSANDWICH	Stick Sandwich
CDLTAKURI	Takuri (Dragonfly Doji with very long lower shadow)
CDLTASUKIGAP	Tasuki Gap
CDLTHRUSTING	Thrusting Pattern
CDLTRISTAR	Tristar Pattern
CDLUNIQUE3RIVER	Unique 3 River
CDLUPSIDEGAP2CROWS	Upside Gap Two Crows
CDLXSIDEGAP3METHODS	Upside/Downside Gap Three Methods
Statistic Functions	
BETA	Beta
CORREL	Pearson's Correlation Coefficient (r)
LINEARREG	Linear Regression
LINEARREG_ANGLE	Linear Regression Angle
LINEARREG_INTERCEPT	Linear Regression Intercept
LINEARREG_SLOPE	Linear Regression Slope
STDDEV	Standard Deviation
TSF	Time Series Forecast
VAR	Variance

Momentum Indicators	
ADX	Average Directional Movement Index
ADXR	Average Directional Movement Index Rating
APO	Absolute Price Oscillator
AROON	Aroon
AROONOSC	Aroon Oscillator
BOP	Balance Of Power
CCI	Commodity Channel Index
CMO	Chande Momentum Oscillator
DX	Directional Movement Index
MACD	Moving Average Convergence/Divergence
MACDEXT	MACD with controllable MA type
MACDFIX	Moving Average Convergence/Divergence Fix 12/26
MFI	Money Flow Index
MINUS_DI	Minus Directional Indicator
MINUS_DM	Minus Directional Movement
MOM	Momentum
PLUS_DI	Plus Directional Indicator
PLUS_DM	Plus Directional Movement
PPO	Percentage Price Oscillator
ROC	Rate of change : ((price/prevPrice)-1)*100
ROCP	Rate of change Percentage: (price-prevPrice)/prevPrice
ROCR	Rate of change ratio: (price/prevPrice)
ROCR100	Rate of change ratio 100 scale: (price/prevPrice)*100
RSI	Relative Strength Index
STOCH	Stochastic
STOCHF	Stochastic Fast
STOCHRSI	Stochastic Relative Strength Index
TRIX	1-day Rate-Of-Change (ROC) of a Triple Smooth EMA
ULTOSC	Ultimate Oscillator
WILLR	Williams' %R
Overlap Studies	
BBANDS	Bollinger Bands
DEMA	Double Exponential Moving Average
EMA	Exponential Moving Average
HT_TRENDLINE	Hilbert Transform - Instantaneous Trendline
KAMA	Kaufman Adaptive Moving Average
MA	Moving average
MAMA	MESA Adaptive Moving Average
MAVP	Moving average with variable period
MIDPOINT	MidPoint over period
MIDPRICE	Midpoint Price over period
SAR	Parabolic SAR
SAREXT	Parabolic SAR - Extended
SMA	Simple Moving Average
T3	Triple Exponential Moving Average (T3)
TEMA	Triple Exponential Moving Average
TRIMA	Triangular Moving Average
WMA	Weighted Moving Average
Volume Indicators	
AD	Chaikin A/D Line
ADOSC	Chaikin A/D Oscillator
OBV	On Balance Volume
Cycle Indicators	
HT_DCPERIOD	Hilbert Transform - Dominant Cycle Period
HT_DCPHASE	Hilbert Transform - Dominant Cycle Phase
HT_PHASOR	Hilbert Transform - Phasor Components
HT_SINE	Hilbert Transform - SineWave
HT_TRENDMODE	Hilbert Transform - Trend vs Cycle Mode
Price Transform	
AVGPRICE	Average Price
MEDPRICE	Median Price
TYPPRICE	Typical Price
WCLPRICE	Weighted Close Price
Volatility Indicators	
ATR	Average True Range
NATR	Normalized Average True Range
TRANGE	True Range
Data From Yahoo Finance	
Open	
Close	
High	
Low	
Adjusted Closed	
Volume	

Figure 1. Data, Features, and Technical Indicators Used in Models

The indicators fall within the following groups: overlap studies, momentum indicators, volume indicators, volatility indicators, price transform, cycle indicators, and pattern recognition. The indicators were separated depending on if they were periodic or non-periodic. The periodic indicators were calculated over a range of periods from 5 to 20. The periodic data was then arranged into images [4], where the row is the period, and the columns are indicators. These images were either then used in the baseline CNN or in the proposed CNN AE.

Due to the overall upward trend for AMD the hold label dominated the classes. To combat this, we down sampled this dominant output to have a more general model that could translate to stocks that are less consistently growing or are in a downward trend.

The predicted data was classified into Sell (0), Buy (1), or Hold (2) based on the closing price within a window of 11 days. Window maximums were labelled as the sell class, window minimums were labeled as the buy class, and all other days were labeled as the hold class [4].

IV. Methodology

This section outlines the different and altered models that we used to solve this problem of predicting stock market prices. We used a basic 2D CNN as our baseline model and implemented and incorporated our other proposed model to see how those variations change the output and loss of our model. In both models, several dropout layers were used to prevent overfitting and the final layer was a softmax dense layer with three neurons, one for each class.

A. Baseline

As stated above, a 2D Convolutional Neural Network was created to predict stock market prices. This consisted of several layers of convolution and max pooling, followed by a flattening and dense layer. The images of the periodic data were used as the inputs to this model and all non-periodic data was ignored. For each image, the rows were the period window size, and the columns were the technical indicator (Figure 1).

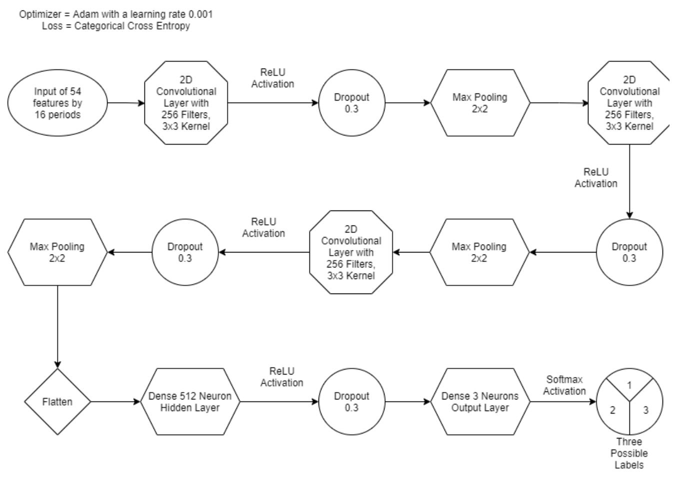

Optimizer = Adam with a learning rate 0.001
Loss = Categorical Cross Entropy
Input of 54 features by 16 periods
2D Convolutional Layer with 256 Filters, 3x3 Kernel
ReLU Activation
Dropout 0.3
Max Pooling 2x2
2D Convolutional Layer with 256 Filters, 3x3 Kernel
ReLU Activation
Dropout 0.3
Max Pooling 2x2
2D Convolutional Layer with 256 Filters, 3x3 Kernel
ReLU Activation
Dropout 0.3
Max Pooling 2x2
Flatten
Dense 512 Neuron Hidden Layer
ReLU Activation
Dropout 0.3
Dense 3 Neurons Output Layer
Softmax Activation
1
2
3
Three Possible Labels

Figure 1: Baseline CNN Model Architecture

For optimizing the baseline model, the Talos library was used to determine the best batch size, convolutional layer neurons, dense layer neurons, dropout, learning rate, and activation function. To reduce computation time a random search of 10% of all the possible combinations was performed and those randomly chosen combinations were used to train separate models for comparison. In total 32 separate models were trained to determine the optimal parameters (Figure 2).

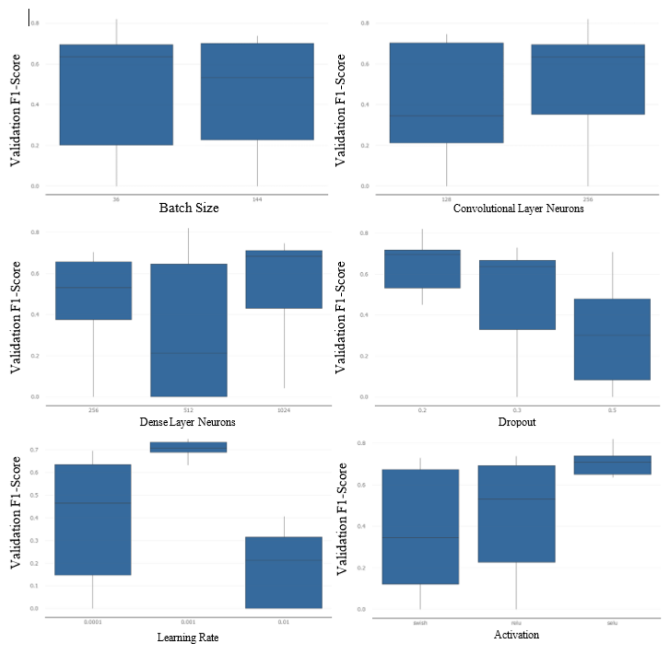


Figure 2: Summary of Talos Optimization for Baseline Model

B. Proposed

We extended our baseline CNN model by converting the CNN to a CNN Autoencoder (AE) to extract features from the periodic data. These encoded features produced from the encoder component of the AE were then flattened and then appended to the nonperiodic data and were used as inputs to a Long Short Term Memory Neural Network (LSTM NN) (Figure 3).

For optimizing the proposed model, the Talos library was used on the autoencoder and LSTM portion separately. For the autoencoder, the parameters optimized were number of kernels, learning rate, kernel size, activation function, and batch size. To reduce computation time a random search of 20% of all the possible combinations was performed and those randomly chosen combinations were used to train separate models for comparison. In total 43 separate models were trained to determine the optimal parameters for the autoencoder. For the LSTM, the parameters optimized were number of nodes, learning rate, dropout, recurrent dropout, and batch size. A random search of 10% of all the possible combinations was performed and resulted in 10 separate models for comparison. After the optimization search (Figure 4) we found for the autoencoder using ReLU, 128 kernels, a kernel size of 5x5, a learning rate of 0.001, and a batch size of 32 was best. We found the best parameters for the LSTM were 256 nodes, a learning rate of 0.001, a dropout of 0.3, a recurrent dropout of 0.2, and a batchsize of 32 (Figure 5).

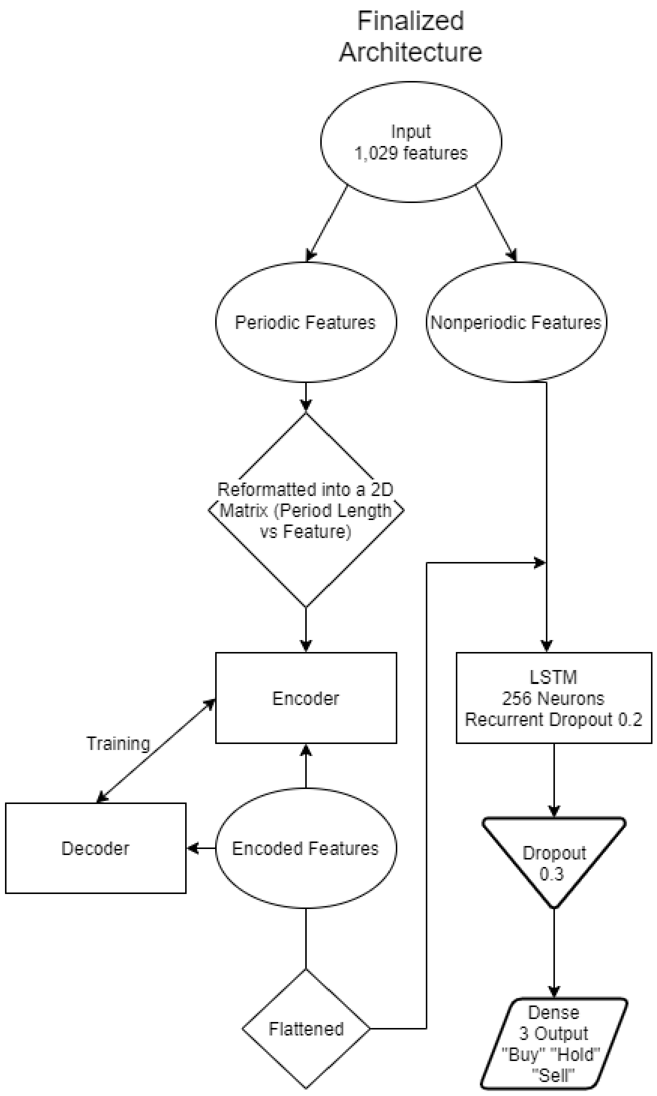


Figure 3: Proposed Model Architecture

V. Results & Evaluation – Make ThiNNNNN Stacks

A. *Computational Model Evaluation*

The accuracy per class of the proposed and baseline models was determined by the diagonal of a confusion matrix. The baseline model did not show a preference for any one class but had an overall worse accuracy than the proposed model (Table 1). The proposed model did not label any data points class 0 (sell) or class 1 (buy) despite us under sampling to handle the data imbalance (Table 2).

B. *Financial Model Evaluation*

A financial evaluation was performed on the last year (April 24th, 2020 to April 23rd, 2021) of AMD's stock closing prices. Our model predicted the appropriate positions of the next day's stock price. These predictions were then used to determine when stocks were bought, sold and held. If no stock was bought yet, then a Buy position would simulate buying the stock, and a Sell

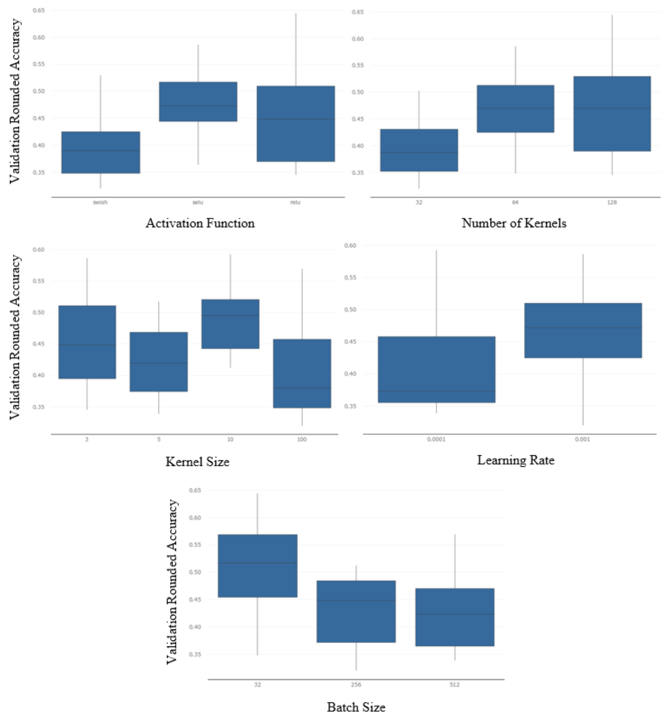


Figure 4: Summary of Talos Optimization for Autoencoder of Proposed Model

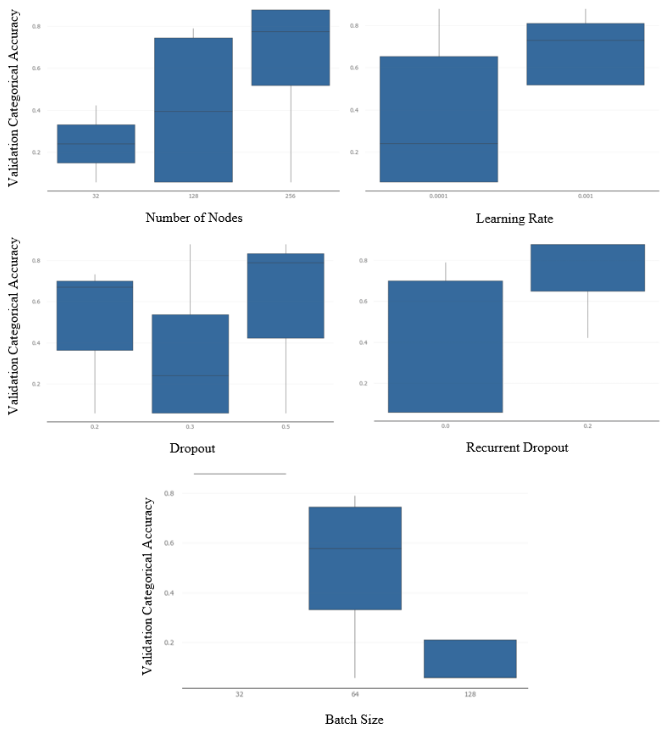


Figure 5: Summary of Talos Optimization for LSTM of Proposed Model

or Hold position would do nothing. If stock was already bought, then a Sell position would simulate selling the stock, while a Buy or Hold position would do nothing. At the end of the year, any stocks held would automatically be sold, and the annual gain was calculated. This process was repeated for the baseline model, proposed model, and true data. These annual gains were also compared with long-term trading, where the stock was bought at the beginning of the year and sold at the end of the year.

Table 1: Shows the precision, recall, accuracy, and F1-scores for each class, and also shows the total accuracy of the baseline model's predicted classifications against test data.

Class	Precision	Recall	F1-Score	Accuracy
Sell (0)	0.310	0.562	0.400	0.5625
Buy (1)	0.238	0.714	0.357	0.7143
Hold (2)	0.943	0.764	0.844	0.7639
			Total Accuracy	0.748

Table 2: Shows the precision, recall, accuracy, and F1-scores for each class, and also shows the total accuracy of the proposed model's predicted classifications against test data.

Class	Precision	Recall	F1-Score	Accuracy
Sell (0)	0	0	0	0
Buy (1)	0	0	0	0
Hold (2)	0.878	1.000	0.935	1.00
			Total Accuracy	0.878

Table 3: Percentage gains from our models versus long-term trading and the maximum possible percent gain.

	Baseline Model	Proposed Model	Long-term Trading	Maximum Percent Gain
Total Percent Gain (%)	-16.36%	0%	50.65%	133.11%

The baseline model performed the worst in the financial evaluation, losing 16.36% of the original investment in one year. The proposed model neither lost nor gained in one year because it labeled all days as Hold position. The maximum gain possible in one year was 133.11%. In comparison, the Long-term trading analysis had a gain of 50.65% in one year (Table 3).

C. *Limitations and Future Direction*

Several limitations and future steps should be taken into consideration when moving forward. First, our data was limited to the scope of one stock symbol and our evaluation was during a year that pretty much only saw growth. This means that with a model that is trained on data with more fluctuations and the 2008 market crash, it would be much harder to evaluate correctly for the past year. Furthermore, this problem also has quite limited data when looking at one stock symbol. Our dataset only had around 5000 points and after down sampling to prevent class imbalance, we were left with only around 900 points of data. One future step could be to try oversampling. However, the challenge with this is the loss of the time series aspect. Several other models or learning types should be investigated. For example, a reinforcement learning strategy where the model is penalized for losing money could be implemented to see if performance improves. Our data could also be augmented to see if performance improves. First, window size in the algorithm used for labelling could be changed. And second, other data sources could be included since the stock market is heavily influenced by other factors such as the news. The implementation of an additional model that uses Natural Language Processing (NLP) could be added and would probably help with the prediction of faster fluctuations in market trends that make it "random."

References

[1] D. Nelson, A. M. Pereira, and Renato, "Stock market's price movement prediction with LSTM neural networks":, 2017 International Joint Conference on Neural Networks (IJCNN), 2017. doi: 10.1109/IJCNN.2017.7966019 (accessed Feb. 27, 2021).

[2] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory | Neural Computation," Neural Computation, 2020. https://doi.org/10.1162/neco.1997.9.8.1735 (accessed Feb. 27, 2021).

[3] NASDAQ. (2021, April 25). Advanced Micro Devices, Inc. (AMD). [Stock quote]. Retrieved from https://finance.yahoo.com/quote/AMD

[4] O. B. Sezer and M. Ozbayoglu, "Algorithmic Financial Trading with Deep Convolutional Neural Networks: Time Series to Image Conversion Approach," Applied Soft Computing, vol. 70, pp. 525–538, Sep. 2018. https://doi.org/10.1016/j.asoc.2018.04.024