LONG SHORT-TERM MEMORY:  
A RECURRENT NEURAL NETWORK TO DETERMINE THE DIRECTION  
OF FUTURES PRICES

**Author**: Timothy Decilveo

**University**: Loyola Marymount University

**Class**: CMSI 5350 Machine Learning

**GitHub**: <https://github.com/timdecilveo/cmsi-5350-project>

**Abstract**

Historically, investors in financial markets have used past data to decide whether to buy or sell a specific security, and recurrent neural networks offer market participants a way to apply this thinking systematically. Specifically, Long Short-Term Memory Recurrent Neural Networks (LSTM RNN) allow investors to utilize feedback connections to process data sequences, or time series data, which is the standard data set used in the financial markets. LSTMs can be valuable models for financial markets as there may be lags of unknown time between significant data points within a timer series. LSTMs can process data and maintain information about prior data points without knowing if the specific data point is relevant a priori.

This paper will provide a novel way to implement an LSTM model to predict the direction of futures market prices using historical price action and Commitment if Trader’s (COT) reports data.

Keywords: Long Short-Term Memory, LSTM, Machine Learning, Commodity Futures, Financial Futures

LONG SHORT-TERM MEMORY:  
A RECURRENT NEURAL NETWORK TO DETERMINE THE DIRECTION  
OF FUTURES PRICES

# Introduction

Long short-term memory (LSTM) models have long been used to predict time series data, but the research dedicated to LSTM model applications within the financial markets is rare. This research attempts to begin the conversation around LSTM and predict commodity and financial futures prices, creating a baseline from which to work and improve the model. Due to time constraints, there are specific issues with the application of this model, but as a general starting point, the model built should be sufficient to create a baseline to work from.

Most market practitioners believe historical data is essential in predicting the price of a security, but it is challenging to determine which historical data is vital in that prediction. Does the price action on day X matter more than day Y? If so, how does one weight day X versus day Y? LSTM models seek to solve the problem of not knowing which data matters using its different gating mechanisms and intuitively seem like a reasonable way to predict future prices.

Furthermore, price prediction is a challenging problem that needs precise outcomes rather than accurate ones. This paper attempts to reduce the complexity of this problem by predicting direction (i.e., up or down). Binary classification is typically an easier problem to solve and is used in this research.

The paper will utilize features outside the standard open, high, low, and closing prices used in most practitioners' price prediction models and will incorporate Commitment of Trader's (CoT) data as an input. The implementation of the CoT data should attempt to show the efficacy and importance of including other relevant data in addition to price action. This paper is not an exhaustive search of independent variables that could be used in the feature set but instead attempts to show the validity of including alternative data sets.

*Hypothesis: Understanding market participants' positioning will increase the accuracy of securities price prediction.*

# Related Work

Currently, there is not a significant amount of research utilizing LSTM models on the financial markets. Most research consists of traditional econometric models that are linear. In 2020, Lu et al. used an LSTM model to forecast stock prices and explained how their LSTM model was the best compared to others in the experiment. Also, in 2020, Moghar and Hamiche implemented an LSTM RNN model to predict stock market prices. The results were mixed and only tested on a limited number of stocks.

# Data

**Commodity Systems, Inc.:**

[Commodity Systems Inc.](http://www.csidata.com/) (CSI) is a low-cost information vendor of summary world financial market data. Daily/data updates on thousands of time series are supplied via the Internet at the close of each business day.

Historical futures data was purchased from CSI dating back to 1946 in some cases, through June 2019. The futures prices have been back adjusted to provide a continuous time series that eliminates the gaps between expiring and newly active contracts. The idea of [back adjusted](http://www.csidata.com/cgi-bin/getManualPage.pl?URL=backadjustedoverview.htm)

contracts involve concatenating historical contracts of a given commodity into the past while adjusting to smooth transitions between delivery months.

The data is proportionately adjusted by percentage or ratio terms and splicing contracts by adding or subtracting their relative differences into the past. Ratio-adjusted series prepared through ratio multiplications are unlikely to go negative, so there is seldom a need to elevate a series out of the negative territory. Contracts are joined by increasing or decreasing successively further distant contracts by a percentage to raise or lower the entire history by the same proportion. Rounding problems caused by attempts to preserve tick differentials of reported prices could occasionally push a given series into negative territory.

Because the ratio adjustment yields a much milder descending slope of long-term prices into the past, there is less long-side trading bias captured from the data. An unbiased result that offers realism should be much preferred over a highly profitable and unbelievable result that holds more inflation contributions than any perceived trading style or expertise.

The idea of ratio-adjusted contracts requires applying the percentage change in the price of the earlier contract with respect to the price of the current (or later) contract. Consider the above example where a five-cent difference in price between successive December and September contracts resulted in a five-cent adjustment to all past data with the traditional back adjuster. In a ratio-adjusted series, the fixed delta of five would represent a factor of 5/100 or 105% of the September price for all data in the September contract. This process would repeat at the same percentage for every contract boundary until the series ended.

*The Ratio (Proportional) back-adjustment principles offered here were inspired by Thomas Stridman, who discussed the idea in his article "Data Pros and Cons" in the June 1998 issue of Futures Magazine.*

The futures evaluated are the Australian Dollar, Soybean Oil, British Pound, Corn, Cocoa, Canadian Dollar, WTI-Crude Oil, Cotton, U.S. Dollar Index, Euro Dollar, Feeder Cattle, 5-Year T-Note, Gold, HG Copper, Heating Oil, Japanese Yen, Coffee, Lumber, Live Cattle, Lean Hogs, Mexican Peso, Nasdaq 100, New Zealand Dollar, Natural Gas, Oats, Orange Juice, Palladium, Platinum, Rough Rice, Soybeans, Sugar #11, Swiss Franc, Silver, Soybean Meal, S&P 500, 2-Year T-Note, 10-Year T-Bonds, 30-Year T-Bonds, Wheat.

**Commitment of Trader’s Reports (COT):**

The Commodity Futures Trading Commission (CFTC) provides weekly data on the commitment of traders and concentration ratios in both legacy and new format for futures only and futures and options. The data used in this experiment only looked at legacy CoT data on futures only. History for this data goes back to 1986, where available. Data is updated weekly on Fridays at 5:00 pm ET. The data feed contains 32,500+ time-series, covering reports for 1,000+ futures contracts in new and legacy formats. For a complete list of Time-Series Codes included in this data feed, use: [CFTC-Nasdaq-Metadata-api\_key=7J7nAmAEHA2yVUZZCtsy](https://data.nasdaq.com/api/v3/databases/CFTC/metadata?api_key=7J7nAmAEHA2yVUZZCtsy).

The data is provided from Nasdaq Data Link’s API through the Quandl package. An API key must be obtained to retrieve data via the API. This data can also be accessed from a browser.

# Features

There were 15 features available for use in this experiment, as defined below:

1. **Open**: the opening price refers to the price at which a security first trades upon the opening of an exchange on a trading day.
2. **High**: the daily high price refers to the highest intraday price a security traded for.
3. **Low**: the daily low price refers to the lowest intraday price a security traded for.
4. **Close**: the closing price refers to the price at which a security last trades before the closing of an exchange on a trading day.
5. **Noncommercial Long**: a noncommercial long refers to someone who has no business activities related to a particular commodity in which they have a long position in the futures market. These are typically referred to as "speculators." The value is a cumulative figure.
6. **Noncommercial Short**: a noncommercial short refers to someone who has no business activities related to a particular commodity in which they have a short position in the futures market. These are typically referred to as "speculators." The value is a cumulative figure.
7. **Commercial Long**: a commercial long refers to someone who has business activities related to a particular commodity in which they have a long position in the futures market. These are typically referred to as "hedgers." The value is a cumulative figure.
8. **Commercial Short**: a commercial short refers to someone who has business activities related to a particular commodity in which they have a short position in the futures market. These are typically referred to as "hedgers." The value is a cumulative figure.

# Experiments

The main experiment in this paper is to predict the direction (up/down) of tomorrow's open price with data over the last N time periods. The researcher could very easily choose to predict tomorrow's high, low, or close price by changing a few lines of code.

The process for running the experiment is as follows:

1. Take the pre-existing open, high, low, and close data from the .txt files.
2. Connect to the Quandl API to extract the CoT weekly data using the legacy format.
3. Merge the datasets on the relevant dates.
4. Clean the data.
5. Transform the data to fit the model.
6. Load the data into the LSTM model.
7. Run the LSTM model.
8. Choose an algorithm to optimize the LSTM model.
9. Choose a learning rate at which to run the LSTM model.
10. Determine training loss and testing loss.
11. Determine training accuracy and testing accuracy.

Once these steps are performed, we can optimize the LSTM model to implement different hyperparameters. Some of the critical hyperparameters are decay rate (default is 0.2), pred (price you are trying to predict - - open, high, low, or close), optimizer (the algorithm used in model prediction; default is AdamW), T (lookback period / historical window to reference; default is 10), num\_hidden (number of hidden layers in the RNN; default is 50), num\_rnnlayers (number of RNN layers; default is 2),

Experiments were run on various hyperparameters to test the efficacy of the model. For instance, if T was 10 in one experiment and 11 in another, should the results be drastically different or relatively stable across parameters? If the results were drastically different, the model might not be as valuable long-term because of overfitting to a specific market regime. LSTMs should solve this issue, and one should not expect drastically different results with slight changes in specific hyperparameters.

# Results

Results are listed in Appendix A.

# Improving the Model

There is more work to be done. The following are listed as priorities moving forward but are by no means an exhaustive list of improvements for the model:

1. Data Cleaning: specific data cleaning techniques were implemented, but there is missing data that can be explored. An example of overcoming the lack of data is to reduce the timeframe analyzed to dates in which the data exists in its most accurate form. The timeframe reduction reduces the model’s ability to analyze the entire data set. Ideally, the data analyzed exists over a 30-year timeframe and is cleaned to prevent any issues with historical inputs.
2. Hyperparameter Optimization: rather than looping through a set of hyperparameters, implementing a more optimal way to search through a list of hyperparameters should be used to determine the optimal hyperparameters for training and testing the data.
3. Segmenting Markets: specific markets may perform better when certain data is used. Various market participants exist in each market that are either naturally long or short their underlying exposure, and determining if that exists in each market may determine the efficacy of adding in additional data like CoT reports. Additionally, markets as groups may perform better on specific hyperparameters than others. Segmenting these groups of markets may provide added insight.
4. Prediction: in the experiment, the predicted value was only one time period in the future (i.e., one day or one week). Determining if that is the ideal time period to predict should be experimented on as one time period may not be the ideal dependent variable.

# Conclusions and Future Work

There is more work to do on this model based on the data. One of the critical areas of focus needs to be on the data cleaning, transforming, and loading aspects of the LSTM model. If this is done correctly, the researcher can build and machine learning model using the same dataset.

In terms of accuracy, percentages greater than fifty percent are encouraging. Most market practitioners cannot correctly predict price direction greater than fifty percent of the time. If the prediction accuracy is greater than fifty percent and the expected dollar outcome on each prediction is greater than one-to-one, it would be worth implementing this LSTM model in an actively traded portfolio.

Moving forward, I intend to continue building out the data components previously discussed and working on the LSTM model using the same dataset. This research may end up being a thesis of mine in which I will publish a more significant amount of research.

References

Chen, Z., Goh, H. S., Sin, K. L., Lim, K., Chung, N. K., & Liew, X. Y. (2021). Automated Agriculture Commodity Price Prediction System with Machine Learning Techniques. *Advances in Science, Technology and Engineering Systems Journal*, *6*(4), 376–384. https://doi.org/10.25046/aj060442

Dolphin, R. (2021, November 12). *LSTM networks: A detailed explanation*. Medium. Retrieved November 29, 2021, from https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9.

Fahlman, S. E. (n.d.). *The recurrent cascade-correlation architecture*. The Recurrent Cascade-Correlation Architecture. Retrieved December 1, 2021, from https://proceedings.neurips.cc/paper/330-the-recurrent-cascade-correlation-architecture.pdf.

Foster, E. A. (2002). Commodity Futures Price Prediction, an Artificial Intelligence Approach. Retrieved November 29, 2021, from https://www.ai.uga.edu/sites/default/files/inline-files/ernest\_foster.pdf.

Gers, F. (2001). Long Short-Term Memory in Recurrent Neural Networks. Retrieved December 1, 2021, from http://www.felixgers.de/papers/phd.pdf.

Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. (2002). *Learning Precise Timing with LSTM Recurrent Networks*.

Herta, C. (n.d.). *LSTM (Long Short Term Memory)*. Christianherta. Retrieved December 1, 2021, from http://christianherta.de/lehre/dataScience/machineLearning/neuralNetworks/LSTM.php.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Retrieved November 30, 2021, from https://www.bioinf.jku.at/publications/older/2604.pdf.

Howard, J. (2021, November 29). *Home*. fast.ai: Making neural nets uncool again. Retrieved November 29, 2021, from https://www.fast.ai/.

Lu, Wenjie, et al. “A CNN-LSTM-Based Model to Forecast Stock Prices.” *Complexity*, vol. 2020, 2020, pp. 1–10., https://doi.org/10.1155/2020/6622927.

Moghar, Adil, and Mhamed Hamiche. “Stock Market Prediction Using LSTM Recurrent Neural Network.” *Procedia Computer Science*, vol. 170, 2020, pp. 1168–1173., https://doi.org/10.1016/j.procs.2020.03.049.

Monner, D. D., & Reggia, J. A. (n.d.). *A generalized LSTM-like training algorithm ... - overcomplete*. A generalized LSTM-like training algorithm for second-order recurrent neural networks. Retrieved December 1, 2021, from http://www.overcomplete.net/papers/nn2012.pdf.

Schmidhuber, J. (n.d.). *Long Short-Term Memory*. Recurrent neural networks - feedback networks - LSTM recurrent network - feedback neural network - recurrent nets - feedback network - recurrent net - - feedback net. Retrieved November 29, 2021, from https://people.idsia.ch//~juergen/rnn.html.

Surmenok, P. (2021, April 19). *Estimating an optimal learning rate for a deep neural network*. Medium. Retrieved November 29, 2021, from https://towardsdatascience.com/estimating-optimal-learning-rate-for-a-deep-neural-network-ce32f2556ce0.

**Appendix A**

Note: Hyperparameters used in all trials: output\_data='merge',pred='open', T=3, num\_hidden=hidden(3, 1), num\_rnnlayers=2, num\_outputs=1

*Intentionally left blank*

**Table 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Cocoa |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6841 | 0.6947 |
| **Adam** | 200 | 0.6565 | 0.6854 |
| **Adam** | 300 | 0.6395 | 0.6812 |
| **AdamW** | 100 | 0.6806 | 0.6949 |
| **AdamW** | 200 | 0.6554 | 0.6931 |
| **AdamW** | 300 | 0.6356 | 0.6917 |
| **SGD** | 100 | 0.6916 | 0.6950 |
| **SGD** | 200 | 0.6911 | 0.6960 |
| **SGD** | 300 | 0.6911 | 0.6962 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.6249 | 0.5645 |  |
| **AdamW** | 0.6259 | 0.5645 |  |
| **SGD** | 0.5335 | 0.4824 |  |

**Table 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Coffee |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6828 | 0.6968 |
| **Adam** | 200 | 0.6545 | 0.7530 |
| **Adam** | 300 | 0.6393 | 0.7643 |
| **AdamW** | 100 | 0.6796 | 0.7150 |
| **AdamW** | 200 | 0.6564 | 0.7445 |
| **AdamW** | 300 | 0.6375 | 0.7402 |
| **SGD** | 100 | 0.7039 | 0.7005 |
| **SGD** | 200 | 0.6979 | 0.6960 |
| **SGD** | 300 | 0.6945 | 0.6937 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.6181 | 0.5156 |  |
| **AdamW** | 0.6268 | 0.5313 |  |
| **SGD** | 0.4577 | 0.4766 |  |

**Table 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Corn |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6850 | 0.6955 |
| **Adam** | 200 | 0.6730 | 0.7052 |
| **Adam** | 300 | 0.6545 | 0.7245 |
| **AdamW** | 100 | 0.6759 | 0.6944 |
| **AdamW** | 200 | 0.6553 | 0.6911 |
| **AdamW** | 300 | 0.6380 | 0.6998 |
| **SGD** | 100 | 0.7154 | 0.6997 |
| **SGD** | 200 | 0.7031 | 0.6937 |
| **SGD** | 300 | 0.6967 | 0.6929 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.6129 | 0.5204 |  |
| **AdamW** | 0.6187 | 0.5534 |  |
| **SGD** | 0.4546 | 0.5087 |  |

**Table 4**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Cotton |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6840 | 0.6965 |
| **Adam** | 200 | 0.6585 | 0.7012 |
| **Adam** | 300 | 0.6493 | 0.7334 |
| **AdamW** | 100 | 0.6886 | 0.6932 |
| **AdamW** | 200 | 0.6726 | 0.7029 |
| **AdamW** | 300 | 0.6557 | 0.7059 |
| **SGD** | 100 | 0.6966 | 0.7015 |
| **SGD** | 200 | 0.6949 | 0.6991 |
| **SGD** | 300 | 0.6941 | 0.6973 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.6192 | 0.5419 |  |
| **AdamW** | 0.6279 | 0.5224 |  |
| **SGD** | 0.5107 | 0.4854 |  |

**Table 5**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Crude Oil (Light) |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6823 | 0.7040 |
| **Adam** | 200 | 0.6688 | 0.7120 |
| **Adam** | 300 | 0.6557 | 0.7191 |
| **AdamW** | 100 | 0.6773 | 0.7123 |
| **AdamW** | 200 | 0.6514 | 0.7378 |
| **AdamW** | 300 | 0.6360 | 0.7640 |
| **SGD** | 100 | 0.7099 | 0.6995 |
| **SGD** | 200 | 0.7002 | 0.6948 |
| **SGD** | 300 | 0.6949 | 0.6933 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.6017 | 0.5272 |  |
| **AdamW** | 0.6250 | 0.5078 |  |
| **SGD** | 0.4506 | 0.4981 |  |

**Table 6**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Eurodollar |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6890 | 0.6943 |
| **Adam** | 200 | 0.6783 | 0.6942 |
| **Adam** | 300 | 0.6710 | 0.6926 |
| **AdamW** | 100 | 0.6903 | 0.6936 |
| **AdamW** | 200 | 0.6767 | 0.6970 |
| **AdamW** | 300 | 0.6701 | 0.6994 |
| **SGD** | 100 | 0.6926 | 0.6930 |
| **SGD** | 200 | 0.6921 | 0.6929 |
| **SGD** | 300 | 0.6917 | 0.6928 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.5643 | 0.5243 |  |
| **AdamW** | 0.5556 | 0.4932 |  |
| **SGD** | 0.5295 | 0.5126 |  |

**Table 7**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Feeder Cattle |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6858 | 0.6929 |
| **Adam** | 200 | 0.6756 | 0.6869 |
| **Adam** | 300 | 0.6561 | 0.6817 |
| **AdamW** | 100 | 0.6838 | 0.6888 |
| **AdamW** | 200 | 0.6630 | 0.6857 |
| **AdamW** | 300 | 0.6407 | 0.6854 |
| **SGD** | 100 | 0.7129 | 0.7150 |
| **SGD** | 200 | 0.7044 | 0.7058 |
| **SGD** | 300 | 0.6994 | 0.7004 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.6014 | 0.5631 |  |
| **AdamW** | 0.6342 | 0.5981 |  |
| **SGD** | 0.4807 | 0.4738 |  |

**Table 8**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Gold |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6912 | 0.6948 |
| **Adam** | 200 | 0.6846 | 0.6990 |
| **Adam** | 300 | 0.6687 | 0.7011 |
| **AdamW** | 100 | 0.6813 | 0.6996 |
| **AdamW** | 200 | 0.6693 | 0.7080 |
| **AdamW** | 300 | 0.6571 | 0.7076 |
| **SGD** | 100 | 0.6989 | 0.6953 |
| **SGD** | 200 | 0.6960 | 0.6938 |
| **SGD** | 300 | 0.6944 | 0.6932 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.5890 | 0.5078 |  |
| **AdamW** | 0.5919 | 0.5486 |  |
| **SGD** | 0.4816 | 0.5058 |  |

**Table 9**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Heating Oil |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6906 | 0.6930 |
| **Adam** | 200 | 0.6825 | 0.7052 |
| **Adam** | 300 | 0.6705 | 0.7353 |
| **AdamW** | 100 | 0.6874 | 0.7001 |
| **AdamW** | 200 | 0.6781 | 0.7130 |
| **AdamW** | 300 | 0.6641 | 0.7176 |
| **SGD** | 100 | 0.6957 | 0.6946 |
| **SGD** | 200 | 0.6947 | 0.6939 |
| **SGD** | 300 | 0.6942 | 0.6934 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.5804 | 0.5214 |  |
| **AdamW** | 0.6105 | 0.5078 |  |
| **SGD** | 0.5058 | 0.5117 |  |

**Table 10**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Live Cattle |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6927 | 0.6932 |
| **Adam** | 200 | 0.6789 | 0.6886 |
| **Adam** | 300 | 0.6581 | 0.6931 |
| **AdamW** | 100 | 0.6922 | 0.6929 |
| **AdamW** | 200 | 0.6627 | 0.6942 |
| **AdamW** | 300 | 0.6499 | 0.7144 |
| **SGD** | 100 | 0.6960 | 0.7004 |
| **SGD** | 200 | 0.6948 | 0.6981 |
| **SGD** | 300 | 0.6941 | 0.6966 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.6042 | 0.5689 |  |
| **AdamW** | 0.6100 | 0.5320 |  |
| **SGD** | 0.4981 | 0.4680 |  |

**Table 11**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Orange Juice |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6812 | 0.6934 |
| **Adam** | 200 | 0.6508 | 0.7033 |
| **Adam** | 300 | 0.6305 | 0.6990 |
| **AdamW** | 100 | 0.6812 | 0.6959 |
| **AdamW** | 200 | 0.6626 | 0.7046 |
| **AdamW** | 300 | 0.6355 | 0.6893 |
| **SGD** | 100 | 0.6913 | 0.6978 |
| **SGD** | 200 | 0.6913 | 0.6979 |
| **SGD** | 300 | 0.6913 | 0.6981 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.6305 | 0.5361 |  |
| **AdamW** | 0.6489 | 0.5458 |  |
| **SGD** | 0.5306 | 0.4717 |  |

**Table 12**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Rough Rice |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6864 | 0.6860 |
| **Adam** | 200 | 0.6699 | 0.6901 |
| **Adam** | 300 | 0.6502 | 0.7025 |
| **AdamW** | 100 | 0.6809 | 0.6868 |
| **AdamW** | 200 | 0.6674 | 0.6972 |
| **AdamW** | 300 | 0.6543 | 0.6964 |
| **SGD** | 100 | 0.6955 | 0.6948 |
| **SGD** | 200 | 0.6934 | 0.6925 |
| **SGD** | 300 | 0.6917 | 0.6924 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.6189 | 0.5516 |  |
| **AdamW** | 0.6142 | 0.5493 |  |
| **SGD** | 0.5361 | 0.5305 |  |

**Table 13**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | S&P 500 Index |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6564 | 0.6650 |
| **Adam** | 200 | 0.6204 | 0.6826 |
| **Adam** | 300 | 0.5988 | 0.6877 |
| **AdamW** | 100 | 0.6668 | 0.6629 |
| **AdamW** | 200 | 0.6426 | 0.6685 |
| **AdamW** | 300 | 0.6169 | 0.6996 |
| **SGD** | 100 | 0.7505 | 0.7602 |
| **SGD** | 200 | 0.7154 | 0.7202 |
| **SGD** | 300 | 0.6965 | 0.6972 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.6923 | 0.5817 |  |
| **AdamW** | 0.6731 | 0.6078 |  |
| **SGD** | 0.4006 | 0.3725 |  |

**Table 14**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Silver |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6907 | 0.6917 |
| **Adam** | 200 | 0.6713 | 0.6821 |
| **Adam** | 300 | 0.6342 | 0.6891 |
| **AdamW** | 100 | 0.6916 | 0.6906 |
| **AdamW** | 200 | 0.6865 | 0.6941 |
| **AdamW** | 300 | 0.6815 | 0.6940 |
| **SGD** | 100 | 0.7205 | 0.7250 |
| **SGD** | 200 | 0.7089 | 0.7123 |
| **SGD** | 300 | 0.7022 | 0.7050 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.6286 | 0.5856 |  |
| **AdamW** | 0.5406 | 0.4922 |  |
| **SGD** | 0.5019 | 0.4922 |  |

**Table 15**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Soybean Meal |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6838 | 0.6902 |
| **Adam** | 200 | 0.6337 | 0.6883 |
| **Adam** | 300 | 0.6107 | 0.6783 |
| **AdamW** | 100 | 0.6890 | 0.6922 |
| **AdamW** | 200 | 0.6575 | 0.6921 |
| **AdamW** | 300 | 0.6150 | 0.6772 |
| **SGD** | 100 | 0.7044 | 0.7045 |
| **SGD** | 200 | 0.6997 | 0.6995 |
| **SGD** | 300 | 0.6968 | 0.6971 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.6612 | 0.5883 |  |
| **AdamW** | 0.6525 | 0.5534 |  |
| **SGD** | 0.5039 | 0.5029 |  |

**Table 16**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Soybean Oil |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6832 | 0.6835 |
| **Adam** | 200 | 0.6683 | 0.6914 |
| **Adam** | 300 | 0.6463 | 0.6873 |
| **AdamW** | 100 | 0.6848 | 0.6897 |
| **AdamW** | 200 | 0.6659 | 0.6861 |
| **AdamW** | 300 | 0.6575 | 0.6900 |
| **SGD** | 100 | 0.7049 | 0.6927 |
| **SGD** | 200 | 0.6991 | 0.6893 |
| **SGD** | 300 | 0.6959 | 0.6884 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.6023 | 0.5612 |  |
| **AdamW** | 0.6033 | 0.5515 |  |
| **SGD** | 0.5212 | 0.5515 |  |

**Table 17**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Soybeans |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6831 | 0.6870 |
| **Adam** | 200 | 0.6258 | 0.6728 |
| **Adam** | 300 | 0.6102 | 0.6706 |
| **AdamW** | 100 | 0.6907 | 0.6917 |
| **AdamW** | 200 | 0.6715 | 0.6966 |
| **AdamW** | 300 | 0.6527 | 0.7054 |
| **SGD** | 100 | 0.6933 | 0.6927 |
| **SGD** | 200 | 0.6934 | 0.6928 |
| **SGD** | 300 | 0.6931 | 0.6929 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.6438 | 0.5883 |  |
| **AdamW** | 0.6390 | 0.5282 |  |
| **SGD** | 0.4990 | 0.5204 |  |

**Table 18**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | T-Bond (Ultra) |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6868 | 0.6871 |
| **Adam** | 200 | 0.6757 | 0.7045 |
| **Adam** | 300 | 0.6686 | 0.7109 |
| **AdamW** | 100 | 0.6843 | 0.6856 |
| **AdamW** | 200 | 0.6719 | 0.6987 |
| **AdamW** | 300 | 0.6589 | 0.7112 |
| **SGD** | 100 | 0.7117 | 0.7123 |
| **SGD** | 200 | 0.6993 | 0.7005 |
| **SGD** | 300 | 0.6932 | 0.6931 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.5880 | 0.5195 |  |
| **AdamW** | 0.6025 | 0.5486 |  |
| **SGD** | 0.5116 | 0.4805 |  |

**Table 19**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | T-Note (10 Year) |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6870 | 0.6920 |
| **Adam** | 200 | 0.6739 | 0.6948 |
| **Adam** | 300 | 0.6599 | 0.6984 |
| **AdamW** | 100 | 0.6845 | 0.6947 |
| **AdamW** | 200 | 0.6658 | 0.7071 |
| **AdamW** | 300 | 0.6457 | 0.7104 |
| **SGD** | 100 | 0.6931 | 0.6910 |
| **SGD** | 200 | 0.6918 | 0.6905 |
| **SGD** | 300 | 0.6909 | 0.6899 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.5851 | 0.4893 |  |
| **AdamW** | 0.6093 | 0.5301 |  |
| **SGD** | 0.5387 | 0.5437 |  |

**Table 20**

|  |  |  |  |
| --- | --- | --- | --- |
| **Commodity:** | Wheat |  |  |
| **Optimizer** | **Epoch** | **Train Loss** | **Test Loss** |
| **Adam** | 100 | 0.6886 | 0.6871 |
| **Adam** | 200 | 0.6702 | 0.6869 |
| **Adam** | 300 | 0.6468 | 0.6911 |
| **AdamW** | 100 | 0.6910 | 0.6886 |
| **AdamW** | 200 | 0.6806 | 0.6850 |
| **AdamW** | 300 | 0.6729 | 0.6908 |
| **SGD** | 100 | 0.6951 | 0.6890 |
| **SGD** | 200 | 0.6935 | 0.6887 |
| **SGD** | 300 | 0.6927 | 0.6882 |
|  |  |  |  |
| **Optimizer** | **Train Accuracy** | **Test Accuracy** |  |
| **Adam** | 0.5907 | 0.5320 |  |
| **AdamW** | 0.5666 | 0.5650 |  |
| **SGD** | 0.5270 | 0.5495 |  |

**Plots:**

**Adam Optimizer**

Chart, line chart

Description automatically generatedChart, histogram

Description automatically generated

Chart, line chart, histogram

Description automatically generatedChart, line chart

Description automatically generated

Chart, line chart, histogram

Description automatically generatedChart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart, histogram

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Chart, line chart, histogram

Description automatically generatedChart, histogram

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

*Intentionally left blank*

**AdamW Optimizer**

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated Chart, line chart

Description automatically generatedChart, line chart, histogram

Description automatically generated Chart, line chart, histogram

Description automatically generatedChart, line chart

Description automatically generated Chart

Description automatically generatedChart, line chart

Description automatically generated Chart, line chart, histogram

Description automatically generatedChart

Description automatically generatedChart, line chart

Description automatically generatedChart, line chart

Description automatically generatedChart, line chart

Description automatically generatedChart

Description automatically generatedChart, histogram

Description automatically generatedChart, line chart, histogram

Description automatically generatedChart, line chart, histogram

Description automatically generatedChart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart, histogram

Description automatically generated

*Intentionally left blank*

**SGD Optimizer**

Chart

Description automatically generatedChart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedChart

Description automatically generated

Chart, line chart

Description automatically generatedChart

Description automatically generated

Chart

Description automatically generatedChart, line chart

Description automatically generated

Chart, scatter chart

Description automatically generatedChart, line chart

Description automatically generated

A picture containing graphical user interface

Description automatically generatedChart, histogram

Description automatically generated

Chart

Description automatically generatedChart

Description automatically generated

Chart, line chart, histogram

Description automatically generatedChart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart, histogram

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

*Intentionally left blank*