Lazard Take Home: Stock Return Prediction

Ming Cao[*] MINGCAO1@SEAS.UPENN.EDU

Abstract

In this project, my goal was to predict stock performance by leveraging two distinct datasets: one focused on textual news and sentiment analysis, and the other on the financial metrics of companies. To enhance the predictive capabilities of my model, I integrated daily numerical price data obtained through web scraping as a supplementary feature. The study involved 25 stocks, each assigned two tasks: predicting the current month's return and forecasting the return for the subsequent month.

My experimentation encompassed three diverse models—SARIMA, LSTM, and BERT—applied to both date ranges: 2000-2023 and 2020-2023, exploring four unique combinations of datasets. Considering the prediction for 25 stocks, I examined two training modes: collectively training all data and training 25 models individually for each stock. The results highlighted the superior performance of LSTM with numerical data, surpassing BERT's effectiveness with textual data, while SARIMA emerged as the least effective model.

The approach of using different models for distinct sets of features capitalizes on their respective strengths and lays the groundwork for a robust ensemble learning model in future applications (which I am implementing).

Keywords: List of keywords

1. Instructions

Here are the experiments I conducted:

- BERT on sentiment data (2020-2023)
- LSTM on news sentiment data, price data, company's financial data (2020-2023)
- LSTM on news sentiment data, price data (2020-2023)
- LSTM on price data, company's financial data (2000-2023)
- SARIMA on label, with exogenous features from sentiment data, price data (2020-2023)

The rationale behind these experiments was to align each model with the most suitable data and optimize overall performance. Our dataset comprises both numerical and textual information, and no existing model in the literature claims proficiency in handling both tasks. To address this, I divided each prediction task into two categories: one based on contextual features and the other on numerical features, setting the stage for more effective ensembling in future analyses.

The initial exploration involved SARIMA, ensuring the stationarity of all preprocessed data. However, SARIMA struggled to effectively incorporate various features, including sentiment scores, prices, and company financial metrics, despite capturing cyclical information in the label.

Subsequently, I turned to LSTM, a widely used model for stock predictions. Initially underperforming on plain sentiment, price, and company factors data, LSTM demonstrated powerful predictive capabilities after extensive technical feature engineering, particularly based on past price and label data within a 60-day window.

However, due to LSTM's inherent loss of some textual information from news texts and headlines, BERT entered the scene to specifically capture word-level details. Although BERT's performance fell short of LSTM, potentially due to limited data and data quality, it successfully captured certain features in stock return trends, presenting promising avenues for future work and data collection.

[*]

2. Data Preprocessing

I maximized the utility of the four datasets at my disposal: sentiment/news data, company financial metrics data, price data, and label data. In this section, I will delve into the individual preprocessing steps for each dataset and how I joined them.

2.1. Data Augmentation

The y labels are duplicated for each day of the month to join with the result of the daily datasets, except in the case of SARIMA where monthly data was used to emphasize its cyclical behavior.

In addition to the provided datasets, I conducted web scraping to acquire daily financial price data from Yahoo Finance, focusing on the specified stocks and aligning with the timeframe relevant to the two target variables related to monthly returns. This decision was underpinned by the recognition of the inherent value in incorporating historical stock prices as features for predictive modeling, a practice supported by extensive prior research.

2.1.1. Future Textual Augmentation with Tweetpy

I acknowledge that there are limitations in sentiment and news data within the provided datasets, particularly in terms of data volume. This constraint presents a significant challenge, particularly for Large Language Models like BERT, which demand comprehensive training on substantial datasets.

I also think this is precisely why it is considered a Bonus. Recognized sources such as Finviz and Bloomberg, potential repositories for sentiment data, lack an API for streamlined data retrieval. Consequently, the use of a scraper is confined to extracting information explicitly presented on the website. This limitation impedes historical data retrieval beyond the immediate past unless manual interaction, such as through Selenium, is employed.

An alternative SOLUTION was the integration of Twitter data accessible through Tweepy, a very easy to use API. However, it is crucial to acknowledge the associated costs with full archive data retrieval incurring a monthly fee of $5000, and limited data retrieval costing $100 per month. This could be a potentially excellent source of data in the future if a budget is available.

2.2. Train-test Split and Cleaning

To validate the effectiveness of the 60-day window feature engineering, a train-test split of 0.6 (60%) for training and 0.4 (40%) for testing was applied to both LSTM and SARIMA models. To ensure data integrity, any NaN values in both target variables (y) were removed.

For sentiment scores, missing values were imputed with zeros (neutral), while the mean of the respective columns was used for imputing missing values in other columns. Outliers were handled by replacing them with interquartile cap values (using the IOR method).

Special attention was devoted to handling missing data in the company's dataset. If more than 30% of a column was null, the decision was made to drop the entire column. This choice was informed by experimental findings, revealing that the inclusion of company data did not significantly enhance the performance of the LSTM model.

2.3. Feature Engineering/Selection - Textual data

For the textual features within the sentiment data, I amalgamated the company name, date, weighted sentiment score, EVENT TEXT, and HEADLINE into a unified text column for each day/row of data. To augment the dataset, the corresponding y label was duplicated for all days in a month. Subsequently, I employed tokenization on the overall text column using BERT's inherent WordPiece embedding, passing the tokenized sequences into the BERT model.

2.4. Feature Engineering/Selection - Numerical data

Concerning the price data, the following technical indicators were engineered post the train and test split, to ensure no data leakage: Daily Return, Price Difference, Rate of Change in Volume (ROC_vol), Moving Averages (MA) for Closing Price (5, 10, 15 days), Closing Price vs. n-day

Exponential Moving Average (EMA), Daily Volume vs. n-day Simple Moving Average (SMA) of Volume, Exponential Moving Averages (EMA) for Each Column (except 'Volume'), High divided by Average True Range with a window of 10 (high10atr), Relative Strength Index (RSI), Money Flow Index (MFI), Stochastic Oscillator (STOCH), Moving Average Convergence Divergence (MACD), and MACD Signal.

In addition, based on the past monthly returns (previous y label), I did some research and engeered the following indicators: moving average for 5, 10, 15 days, Moving Average Convergence Divergence (MACD), MACD Signal Line (MACD_signal), and RSI Relative Strength Index (RSI).

Each of aforementioned indicators was carefully selected based on comparisons with the actual y label. To offer a clearer understanding of how these features operate for a specific stock, some sample comparisons are presented.

3. Modeling

3.1. Training

3.1.1. Sep vs Agg

In the modeling phase, two distinct training techniques were employed for LSTM: training each stock data separately and consolidating all stock data into a single model. The former approach, training individually for each stock, aims to provide a more "personalized" prediction tailored to each stock's historical behavior, which intuitively seems like a reasonable choice.

Surprisingly, through extensive experimentation, it was observed that training all stocks together in a unified model yielded superior test performance. This counterintuitive finding could be attributed to the limitations of data within individual stocks when utilizing deep learning frameworks. It suggests that the collective information from all stocks enhances the model's ability to discern broader patterns, potentially overcoming the constraints imposed by limited data per stock.

In essence, this observation hints at the notion that stock return trends may not differ significantly across various stocks when carefully selected features are incorporated into the modeling process. This finding underscores the im-

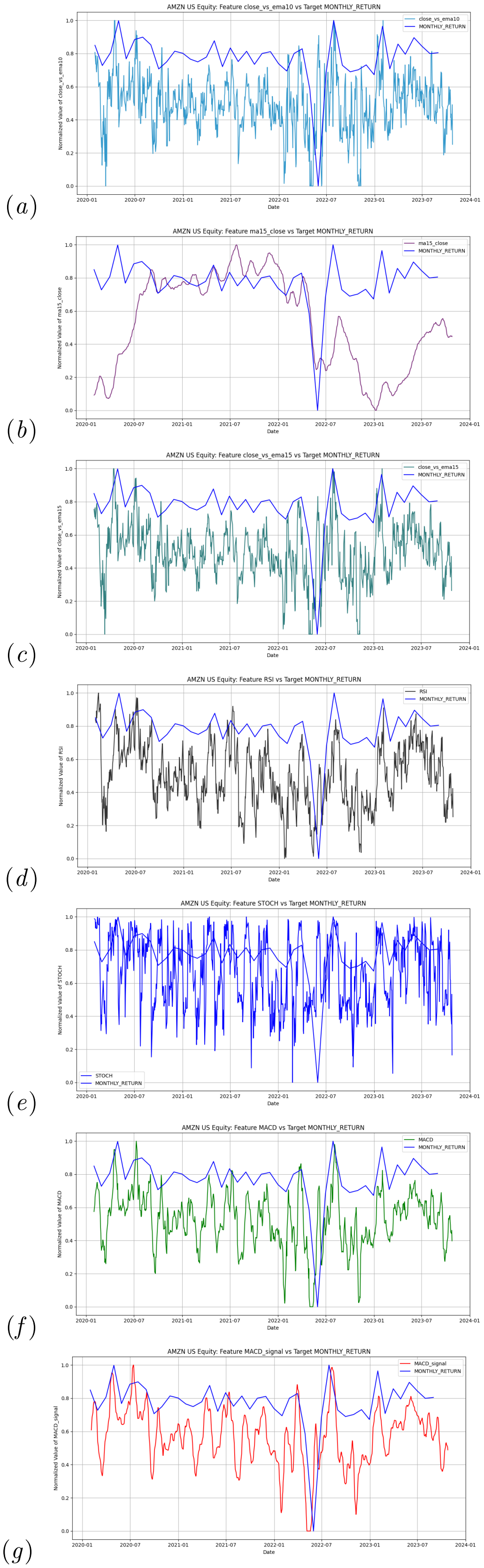
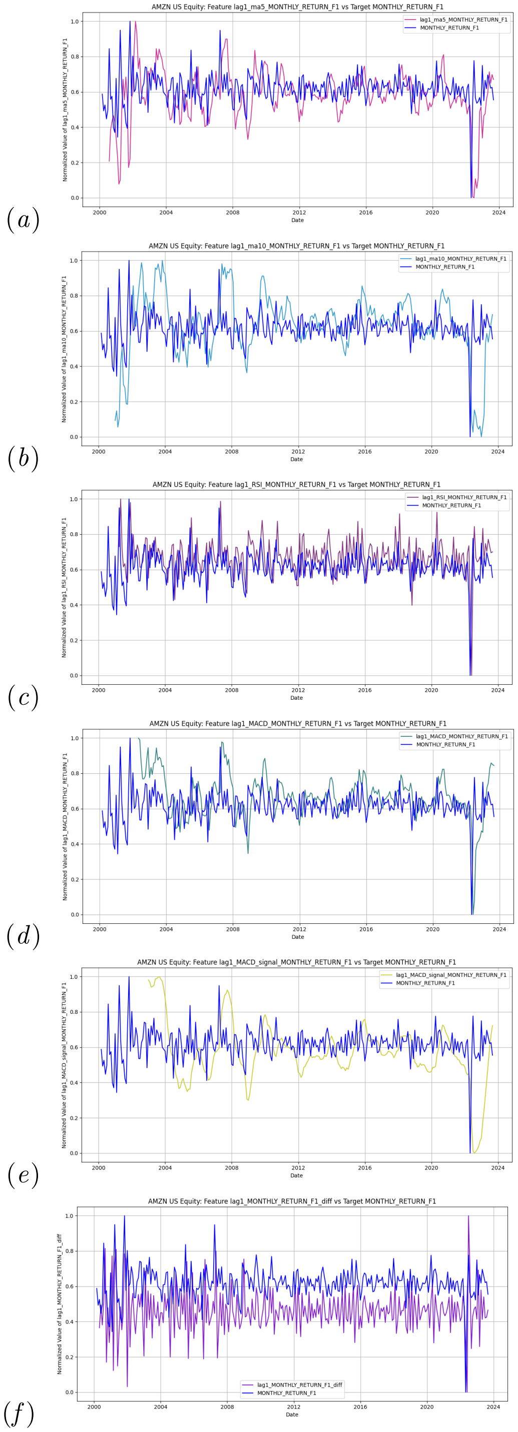


Figure 1: some graphs for technical features engineered based on price data



(a)

(b)

(c)

(d)

(e)

(f)

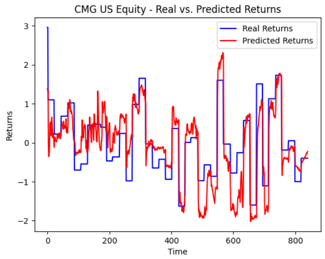
Figure 2: some graphs for technical features engineered based on the previous monthly return. Similar feature engineering steps are went through for next monthly return

portance of feature engineering and the potential benefits of considering a holistic approach to training models on diverse stock datasets.

Another potential explanation for the superior test performance when training all stocks together is that the averaged value of stock returns may be the mathematically optimal solution for minimizing loss. The amalgamation of diverse stock behaviors into a single model could lead to a convergence towards an optimal average that better captures overall market trends.

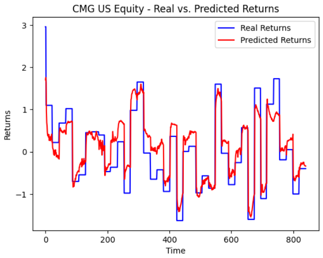
The example provided is just a glimpse into the divergence of results between the two schemes. For a comprehensive overview of similar results across various graphs, please refer to the "LSTM_pricecf_2000.ipynb" notebook.

In light of the insights gained from LSTM, BERT underwent training exclusively on aggregated data. In contrast, SARIMA, being a more traditional method outside the realm of deep learning, did not face significant constraints on data size. Consequently, SARIMA was independently trained for each of the 25 stocks.



(a)

Figure 3: separate training for each stock



(a)

Figure 4: aggregated training

Figure 5: Test set performance for one stock comparison : separately trained and aggregated training

3.1.2. Hyperpatameter Tuning

I tuned essential hyperparameters such as epochs and learning rate for both BERT and LSTM. Exogenous features underwent experimentation with SARIMA. It's important to highlight that I didn't excessively fine-tune the deep learning models solely for aesthetic results. My emphasis was on optimizing data processing and the model architecture to enhance overall performance.

3.2. SARIMA

3.2.1. Rationale

I chose to use SARIMA specifically because of its integration capability addresses non-stationarity, and its interpretability provides insights into underlying dynamics (although all the features and labels passed the Dickey-Fuller Test and are stationary). The model is trained in the 2 scheme mentioned before: trained separately for each stock data and all together

3.2.2. Inputs

Mainly the series of monthly returns, with exogenous features selected among sentiment scores, engineered price and past label features

3.2.3. Evaluations

The graphs illustrating representative predictions versus actual labels for select stocks highlight an interesting observation. Despite potential divergences in the overall trend of certain stocks when compared to the ground truth values, SARIMA emerges as the most adept model in capturing troughs and peaks with remarkable accuracy. There appears to be a discernible one-to-one correspondence, emphasizing SARIMA's proficiency in capturing cyclical patterns within the stock return data. This reinforces SARIMA's effectiveness in modeling and predicting the nuanced dynamics of stock price movements.

However, SARIMA is comparatively not good at predicting next month returns and its predictions could diverge sometimes (though rare, below is a specially picked example for representation).

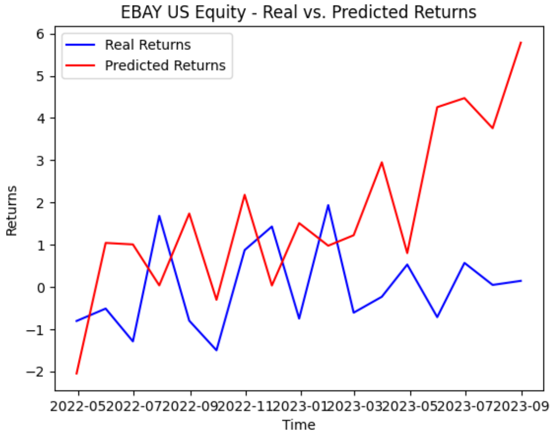


Figure 6: one stock, MONTHLY RETURN

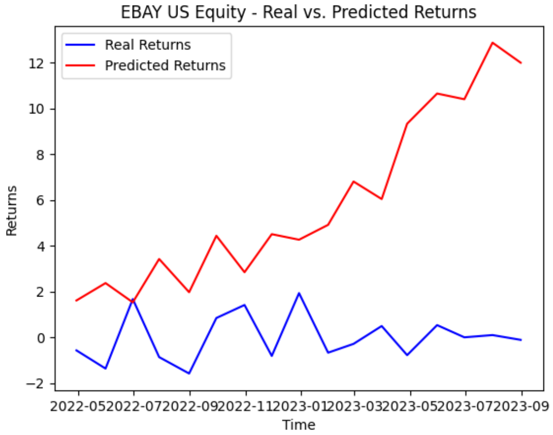


Figure 7: one stock, MONTHLY RETURN

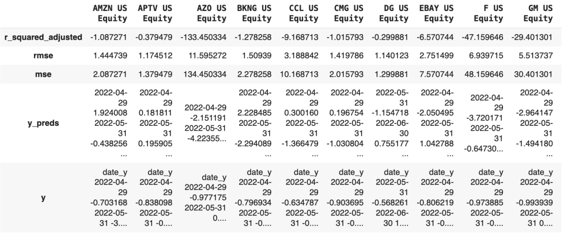
3.3. LSTM

3.3.1. Rationale

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) designed to address the limitations of traditional RNNs in capturing and learning from long-term dependencies in sequential data. I chose LSTM since it is particularly well-suited for time series prediction tasks, making them a popular choice in finance and stock return prediction.

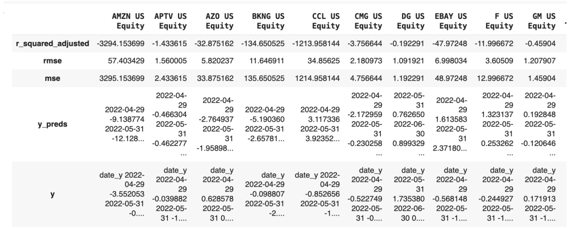
3.3.2. Inputs

I conducted experiments using three input sets: 1. sentiment, price, and company data, along with previous label data from 2020-2023; 2.



	AMZN US Equity	APTV US Equity	AZO US Equity	BKNG US Equity	CCL US Equity	CMG US Equity	DG US Equity	EBAY US Equity	F US Equity	GM US Equity
r_squared_adjusted	-1.087271	-0.379479	-133.450334	-1.278258	-9.168713	-1.015793	-0.299881	-6.570744	-47.159646	-29.401301
rmse	1.444739	1.174512	11.595272	1.50939	3.188842	1.419786	1.140123	2.751499	6.939715	5.513737
mse	2.087271	1.379479	134.450334	2.278258	10.168713	2.015793	1.299881	7.570744	48.159646	30.401301
y_preds	2022-04-29 1.924008 2022-05-31 -0.438256 ...	2022-04-29 0.181811 2022-05-31 0.195905 ...	2022-04-29 -2.151191 2022-05-31 -4.22355...	2022-04-29 2.228485 2022-05-31 -2.294089 ...	2022-04-29 0.300160 2022-05-31 -1.366479 ...	2022-04-29 0.196754 2022-05-31 -1.030804 ...	2022-05-31 -1.154718 2022-06-30 0.755177 ...	2022-04-29 -2.050495 2022-05-31 1.042788 ...	2022-04-29 -3.720171 2022-05-31 -0.64730...	2022-04-29 -2.964147 2022-05-31 -1.494180 ...
y	date_y 2022-04-29 -0.703168 2022-05-31 -3....	date_y 2022-04-29 -0.838098 2022-05-31 -0....	date_y 2022-04-29 -0.977175 2022-05-31 0....	date_y 2022-04-29 -0.796934 2022-05-31 -0....	date_y 2022-04-29 -0.634787 2022-05-31 -0....	date_y 2022-04-29 -0.903695 2022-05-31 -0....	date_y 2022-05-31 -0.568261 2022-06-30 1....	date_y 2022-04-29 -0.806219 2022-05-31 -0....	date_y 2022-04-29 -0.973885 2022-05-31 -0....	date_y 2022-04-29 -0.993939 2022-05-31 0....

Figure 8: SARIMA metrics: MONTHLY RETURN



	AMZN US Equity	APTV US Equity	AZO US Equity	BKNG US Equity	CCL US Equity	CMG US Equity	DG US Equity	EBAY US Equity	F US Equity	GM US Equity
r_squared_adjusted	-3294.153699	-1.433615	-32.875162	-134.650525	-1213.958144	-3.756644	-0.192291	-47.97248	-11.996672	-0.45904
rmse	57.403429	1.560005	5.820237	11.646911	34.85625	2.180973	1.091921	6.998034	3.60509	1.207907
mse	3295.153699	2.433615	33.875162	135.650525	1214.958144	4.756644	1.192291	48.97248	12.996672	1.45904
y_preds	2022-04-29 -9.138774 2022-05-31 -12.128...	2022-04-29 -0.466304 2022-05-31 -0.462277 ...	2022-04-29 -2.764937 2022-05-31 -1.95898...	2022-04-29 -5.190360 2022-05-31 -2.65781...	2022-04-29 3.117336 2022-05-31 3.92352...	2022-04-29 -2.172959 2022-05-31 -0.230258 ...	2022-05-31 0.762650 2022-06-30 0.899329 ...	2022-04-29 1.613583 2022-05-31 2.37180...	2022-04-29 1.323137 2022-05-31 0.253262 ...	2022-04-29 0.192848 2022-05-31 -0.120646 ...
y	date_y 2022-04-29 -3.552053 2022-05-31 -0....	date_y 2022-04-29 -0.039882 2022-05-31 -1....	date_y 2022-04-29 0.628578 2022-05-31 0....	date_y 2022-04-29 -0.098807 2022-05-31 -2....	date_y 2022-04-29 -0.852656 2022-05-31 -1....	date_y 2022-04-29 -0.522749 2022-05-31 -0....	date_y 2022-05-31 1.735380 2022-06-30 0....	date_y 2022-04-29 -0.568148 2022-05-31 -1....	date_y 2022-04-29 -0.244927 2022-05-31 -1....	date_y 2022-04-29 0.171913 2022-05-31 -1....

Figure 9: SARIMA metrics: MONTHLY RETURN F1

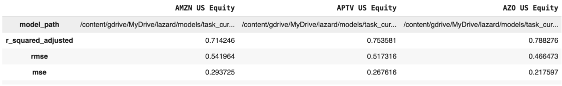
price, company, and previous label data from 2000-2023; 3. sentiment, price, and label data from 2020-2023. The results indicate that the size of the data plays a more critical role in model performance than the inclusion of sentiment information.

3.3.3. Evaluations

Similar to scheme 2, even without incorporating sentiment scores, the model's performance is optimal because it can utilize data dating back to 2000, whereas sentiment data is available only from 2020 onwards. Furthermore, the inclusion of company information was found to have minimal impact on prediction accuracy. Notably, technical features derived from price and past label data, such as the Relative Strength Index (RSI), proved to be the most valuable predictors.

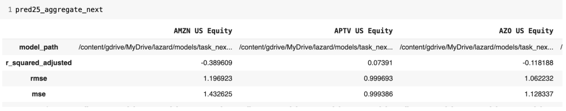
Among the three models, LSTM emerged as the top performer, leveraging its capacity to effectively handle a substantial volume of numerical data. However, its limitation lies in the inability to fully utilize sentiment information. It's noteworthy that an alternative version of LSTM incorporating sentiment scores, was implemented as well, and it still surpasses BERT and SARIMA in performance across both tasks.

To examine comprehensive predictions versus actual labels and test metrics, refer to the LSTM_pricecf_2000.ipynb for detailed graphs and tables. Within the notebooks, you'll observe that for LSTM with data commencing from 2000, the two training methods (seg vs agg) yield minimal differences. However, in the case of LSTM applied to sentiment data starting from 2020, significant disparities emerge, as each stock now has considerably less data for training.



	AMZN US Equity	APTV US Equity	AZO US Equity
model_path	/content/gdrive/MyDrive/lazard/models/task_cur...	/content/gdrive/MyDrive/lazard/models/task_cur...	/content/gdrive/MyDrive/lazard/models/task_cur...
r_squared_adjusted	0.714246	0.753581	0.788276
rmse	0.541964	0.517316	0.466473
mse	0.293725	0.267616	0.217597

Figure 10: LSTM_agg2000: MONTHLY RETURN



```
1 pred25_aggregate_next
```

	AMZN US Equity	APTV US Equity	AZO US Equity
model_path	/content/gdrive/MyDrive/lazard/models/task_nex...	/content/gdrive/MyDrive/lazard/models/task_nex...	/content/gdrive/MyDrive/lazard/models/task_nex...
r_squared_adjusted	-0.389609	0.07391	-0.118188
rmse	1.196923	0.999693	1.062232
mse	1.432625	0.999386	1.128337

Figure 11: LSTM_agg2000: MONTHLY RETURN F1

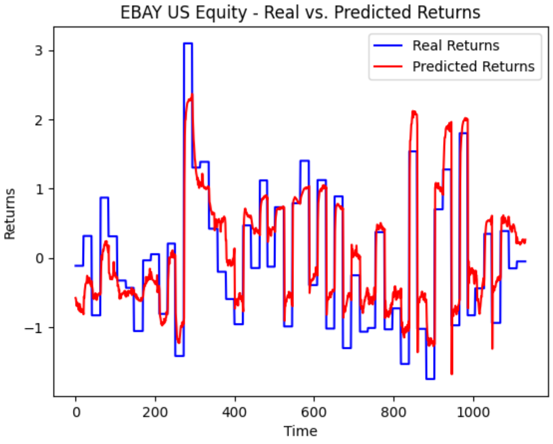


Figure 12: LSTM EBAY, MONTHLY RETURN

The attached graphs clearly highlight the remarkable performance of LSTM in predicting current month stock returns. Impressively, it continues to effectively capture the fluctuations in the overall curve for the subsequent month, making it a valuable tool for informed decision-making.

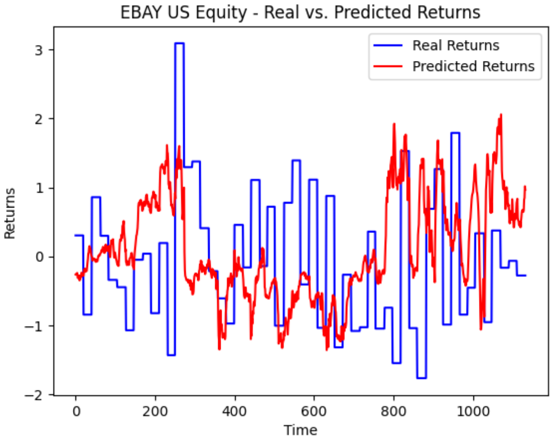


Figure 13: LSTM EBAY, MONTHLY RETURN F1

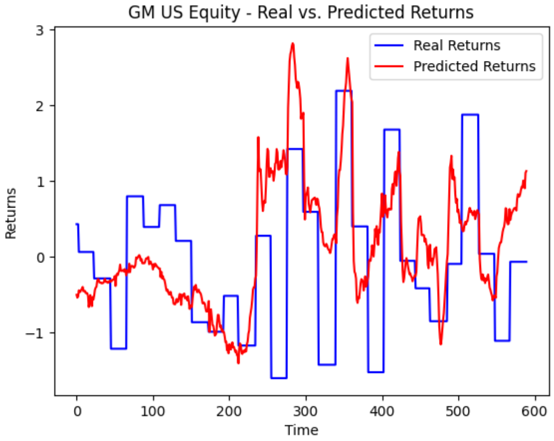


Figure 15: LSTM GM, MONTHLY RETURN F1

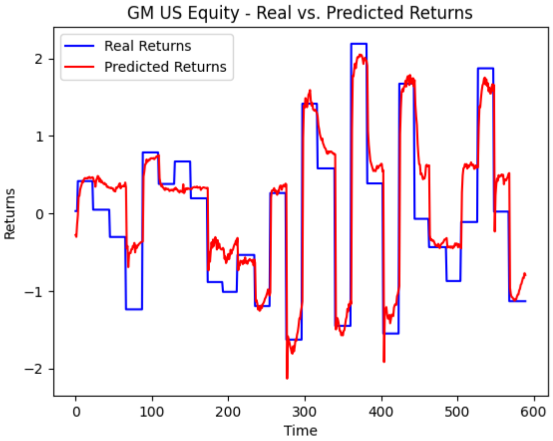


Figure 14: LSTM GM, MONTHLY RETURN

3.4. BERT

3.4.1. Rationale

BERT has been able to suprise me with its ability to perform tasks that intuitively cannot be solved with NLP. And it did it again this time.

3.4.2. Inputs

Inputs for BERT are just sentiment data including Headline and Event text. Security info, date, sentiment score are passed as the first 3 words of the aggregated text. The input to the model is tokenized text.

3.4.3. Training

Distill bert (very-small bert) fine-tuned over the 2 target variables respectively over less than 10 epoches.

3.4.4. Evaluations

The performance of both current month and next month stock returns doesn't quite measure up to LSTM. Typically, LLMS shine when they're well-prepared with pre-training, but our training here has been a bit restrained, and the dataset isn't exactly expansive. Surprisingly, even at this limited scale, BERT seems to catch on to some of the shifts in the overall trend. Unlike SARIMA, which tends to drift from the overall stock price level, BERT manages to stay on course, not to mention it captures a few of the market's ups and downs.

4. Conclusion

In essence, my research reveals a performance hierarchy—where I observed LSTM outperforming BERT, which in turn surpassed SARIMA across key metrics such as R2, RMSE, and MSE. Among the significant findings, I found that rule-based and engineered technical features, rooted in previous target values and price data, played a crucial role in enhancing the accuracy of stock return predictions.

Additionally, I advocate for prioritizing training size over individualization in deep learning

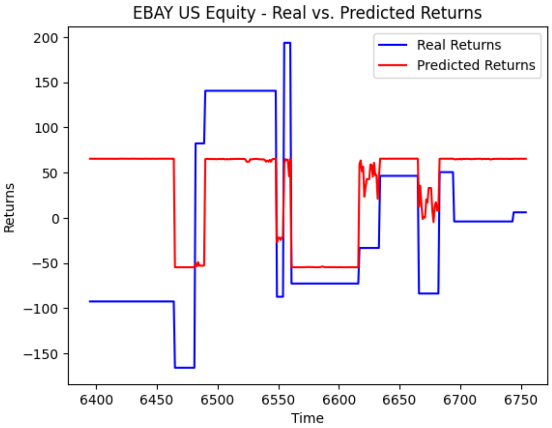


Figure 16: BERT EBAY, MONTHLY RETURN

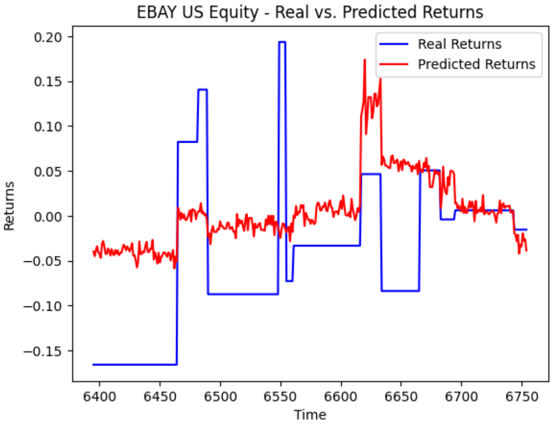


Figure 17: BERT EBAY, MONTHLY RETURN F1

models. In other words, I recommend considering stock-specific training only when armed with a very large training sample. Furthermore, I highlight BERT as a highly promising model, particularly when confronted with good quality text data.

Recognizing the unique strengths of LSTM, BERT, and SARIMA—capturing numerical information, handling textual nuances, and identifying cyclical patterns respectively—I believe that ensemble learning, a strategy I am actively implementing, holds the potential to significantly elevate overall performance.

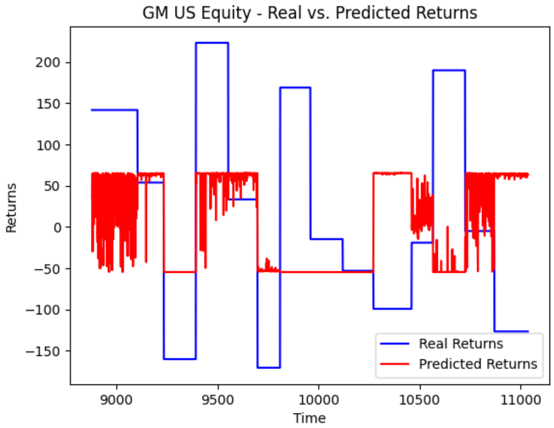


Figure 18: BERT GM, MONTHLY RETURN

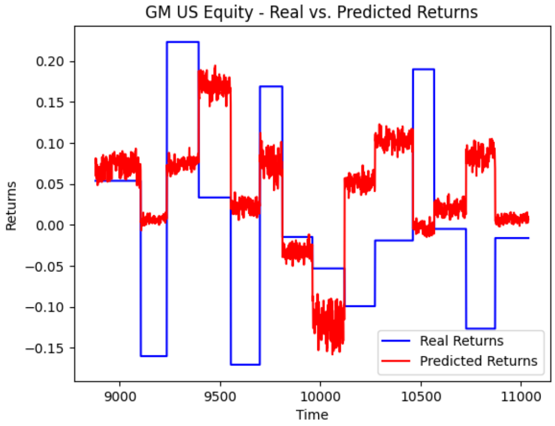


Figure 19: BERT GM, MONTHLY RETURN F1

Acknowledgments

Acknowledgements go here.

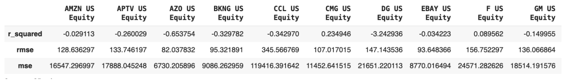
References

Appendix A. First Appendix

This is the first appendix.

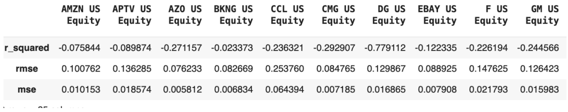
Appendix B. Second Appendix

This is the second appendix.



	AMZN US Equity	APTV US Equity	AZO US Equity	BKNG US Equity	CCL US Equity	CMG US Equity	DG US Equity	EBAY US Equity	F US Equity	GM US Equity
r_squared	-0.029113	-0.260029	-0.653754	-0.329782	-0.342970	0.234946	-3.242936	-0.034223	0.089562	-0.149955
rmse	128.636297	133.746197	82.037832	95.321891	345.566769	107.017015	147.143536	93.648366	156.752297	136.066864
mse	16547.296997	17888.045248	6730.205896	9086.262959	119416.391642	11452.641515	21651.220113	8770.016494	24571.282626	18514.191576

Figure 20: BERT MONTHLY RETURN (rmse and mse here was high cuz I multiplied the label by 1000 in the last run to optimize performance.)



	AMZN US Equity	APTV US Equity	AZO US Equity	BKNG US Equity	CCL US Equity	CMG US Equity	DG US Equity	EBAY US Equity	F US Equity	GM US Equity
r_squared	-0.075844	-0.089874	-0.271157	-0.023373	-0.236321	-0.292907	-0.779112	-0.122335	-0.226194	-0.244566
rmse	0.100762	0.136285	0.076233	0.082669	0.253760	0.084765	0.129867	0.088925	0.147625	0.126423
mse	0.010153	0.018574	0.005812	0.006834	0.064394	0.007185	0.016865	0.007908	0.021793	0.015983

Figure 21: BERT MONTHLY RETURN F1