COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

IEOR E4579 Machine Learning in Practice

Final Project

Differential Privacy LSTM for Stock Prediction with Financial News

Zhaoyang Liu
Cecilia Yu
Emma Jin
Zonghan Li

Supervised by
Professor Gary Kazantsev

Spring 2023

Abstract

Inspired by the paper "*DP-LSTM: Differential Privacy-inspired LSTM for Stock Prediction Using Financial News*", this project aims to perform stock price prediction using various machine learning techniques and data processing steps. In addition to replicating the paper result from scratch, we processed news data into sentiment features while adding Gaussian perturbation for differential privacy, which enhances the robustness of the model and leads to more stable results by adding robustness against adversarial examples; we introduced a few more model classes (across statistical methods, machine learning and deep neural nets); finally, we applied different time windows, tested data using multiple metrics, and validated results on multiple industry indexes as the asset price and returns. We observe and conclude that the DP-LSTM would provide better prediction accuracy and all models perform better on the return data instead of the price data. The time window analysis shows us that the lag of 15 days would lead to best prediction results as it captures enough variance and is not subject to over-fitting issues.

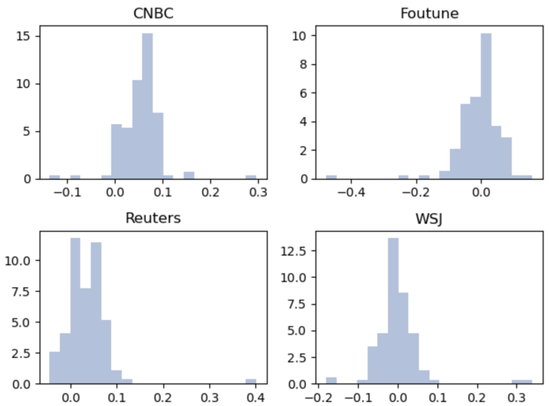
I. Data Processing and Assumptions

Our study drew upon two primary types of data: historical stock index data from Yahoo Finance API, and financial news article text data from four major news sources - CNBC, Fortune, Reuters, and the Wall Street Journal (WSJ) - which included news titles and text bodies. The data was collected within the same time frame as that of the paper, spanning from December 6th 2017 to June 2nd 2018. Due to the relatively short time frame, consisting of only around 180 data points, there is a possibility that the quality of the neural network models may be affected.

1. Stock index data

It is important to note that, in addition to the S&P 500 stock index used in the paper, we incorporated three additional stock market indices from Yahoo Finance API: the Nasdaq 100 Index, the Dow Jones Industrial Average, and the Russell 2000 Index. Furthermore, while the paper focuses on predicting stock prices based on the sentiment of news articles, we believe that stock returns are better suited for time-series model predictions. As such, we included the simple daily return of these stock indices as prediction variables.

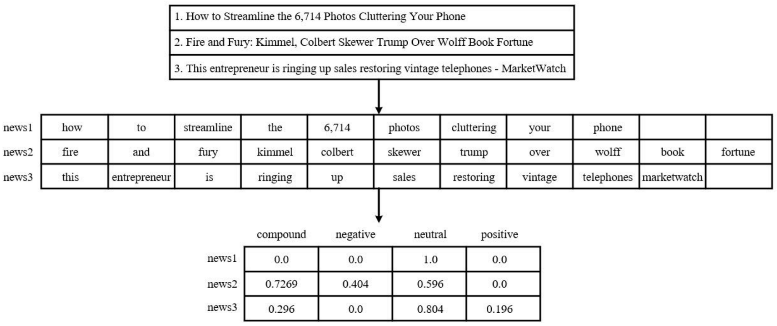
2. Financial news article text data



Data cleaning was an essential step to make sense of the financial news article text data. We conducted data exploration, text cleaning, and data imputation to prepare the data for natural language processing (NLP) models. This involved removing NaN values, grouping news titles into a word corpus, and removing stop words to improve model performance by reducing noise. We also imputed missing data by assigning neutral scores to ensure that the imputation did not introduce any bias to the data. These efforts aimed to enhance the model's accuracy and provide reliable results.

Sentiment Analysis: In order to extract features from the financial news article text data, we utilized the NLTK Vader package to generate compound sentiment scores, which represent the summation of negative, neutral, and positive tones. Vader has been found to be specialized in dealing with news reviews. Vader utilizes a combination of a sentiment lexicon that is labeled according to the semantic orientation.

These sentiment scores are valuable indicators of market opinions. We then merged the resulting sentiment data with stock data based on dates and pivoted each news article's sentiment as individual features. This was done because different companies may have varying views and source data on similar market topics at a given time. In our model, each article's sentiment score was considered a separate feature, as it explains different variance attributes to target prices or returns.



1. How to Streamline the 6,714 Photos Cluttering Your Phone
2. Fire and Fury: Kimmel, Colbert Skewer Trump Over Wolff Book Fortune
3. This entrepreneur is ringing up sales restoring vintage telephones - MarketWatch

news1	how	to	streamline	the	6,714	photos	cluttering	your	phone		
news2	fire	and	fury	kimmel	colbert	skewer	trump	over	wolff	book	fortune
news3	this	entrepreneur	is	ringing	up	sales	restoring	vintage	telephones	marketwatch	

	compound	negative	neutral	positive
news1	0.0	0.0	1.0	0.0
news2	0.7269	0.404	0.596	0.0
news3	0.296	0.0	0.804	0.196

II. Model Selection:

In addition to implementing the LSTM Neural Nets Model from the original paper, we extended model classes to the traditional statistical model and the machine learning model for performance comparison and optimization. We wish to understand the model characteristics through comparison to the same scenario.

1. LSTM Models

Implementation: LSTM is a type of recurrent neural network (RNN) that was specifically designed to model non-linear, multi-featural data and long-range dependencies in sequences, thus it is especially useful when predicting based on sequential time series data. We normalized the data using min and max values of the entire test dataset instead of normalizing each window separately. We tried to use time series data with sentiment features for prediction.

a. We optimized the code implementation and combined the separated structure of processing/normalizing into one compact function called preprocess_data.
b. The above steps improved the MSE from around 500 to 300.

Enhanced by Differential Privacy sentiments: We added Gaussian perturbation on sentiment features. DP could enhance the robustness of the model and lead to more stable results, because the perturbation itself will not influence the accuracy and model statistics, but it could add adversarial robustness against adversarial examples.

2. ARIMA

Implementation: ARIMA is a particularly useful model for predicting linear, univariate time series data that display a pattern or trend over time, so we assumed that the function of ARIMA would be suitable for our dataset. However, the ARIMA model turns out to give us large errors. By preprocessing the data format so that we predict every one closing price based on ten previous prices, we got the following performance:

a. MSE: 3004
b. Accuracy: 91.11%
c. Mean_error_percent: 8.89%

Evaluation: Compared to LSTM, DP-LSTM and Random Forest model, ARIMA's performance is inferior, which is mainly caused by the following reasons:

a. Univariate vs. Multivariate: As mentioned above, ARIMA is a univariate model and does not take into account sentiment scores. On the other hand, random forest and LSTM models can incorporate multiple features, which potentially lead to better predictive performance.
b. Linear vs. Non-linear: ARIMA assumes a linear relationship between the independent and dependent variables. However, stock prices and their relationship with independent variables may not be linear. On the other hand, random forest and LSTM models can capture those nonlinear relationships.

c. Stationary Assumption: ARIMA models assume that the time series data is stationary. Yet the stock prices are usually non-stationary, which will negatively affect the performance of the ARIMA model.
d. Noise and Outliers: Compared to random forest and LSTM models, ARIMA models are more sensitive to noise and outliers, which potentially influence the performance of the model.

3. Random Forest

Implementation: Random forest models are particularly useful for handling non-linear and high-dimensional data and are not prone to overfitting. Indeed, random forest models proved to be another well-performing method, which is the model that has a performance closest to that of the LSTM model. By tuning the lag features and applying the GridSearchCV method, we found that creating a lag feature of 10 with 'min_samples_leaf’ = 1, 'min_samples_split' = 2, and 'n_estimators' = 50 yielded the best performance.

a. MSE: 389
b. Accuracy: 99.46%
c. Mean_error_percent: 0.54%

Evaluation: The Random Forest model is the second best model among ARIMA, Random Forest and LSTM. Here are some pros and cons of the random forest model:

a. Time series incompetency: Random forest model is not specifically designed to perform predictions on time series data since it does not utilize the information from the past as great as the LSTM model. However, we created lag features, which consists of 10 previous closing prices. The lag features possibly increase the performance of the random forest model.
b. Hyperparameters tuning: The random forest model itself has a MSE of around 1400. Though this is much less than the error of the ARIMA model, it is still much higher than the LSTM model. We applied Gridsearch algorithms and found a set of better hyper-parameters, which improved the MSE of random forest from 1400 to 400. Now the performance of the random forest does not differ from the LSTM a lot.

III. Observations:

1. Evaluation Metrics

Mean Absolute Error	**Mean Squared Error**	**Accuracy**	**Mean Error Percent**
$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$	$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$	1- $\frac{100\%}{n}\sum_{i=1}^{n}\left\lvert\frac{y_i - \hat{y}_i}{y_i}\right\rvert$	$\frac{100\%}{n}\sum_{i=1}^{n}\left\lvert\frac{y_i - \hat{y}_i}{y_i}\right\rvert$
measure of the average magnitude of difference between predicted and actual values	measure of the average squared magnitude of difference between predicted and actual values	measure the accuracy of a model's predictions as a percentage of the actual values	measure of the average percentage difference between predicted and actual values

We maintained the use of mean squared error, accuracy, and mean error percent as evaluation metrics. In addition to these, we added mean absolute error, which is more robust to outlier effects and more interpretable. We run each model for ten times and take the average for our final results for robustness.

MSE	**LSTM_no_senti**	**LSTM_senti**	**dp**	**ARIMA**	**rf**
SP	335.6792	350.8661	313.0404	3004.079	389.2055
SP_r	0.0000	0.0000	0.0000	0.0000	0.0001
NDX	1204.9574	1910.9455	1838.4121	28466.6088	44847.2560
NDX_r	0.0000	0.0000	0.0000	0.0001	0.0001
DJI	46692.1061	43289.7420	43728.1917	256693.9434	79804.738
DJI_r	0.0001	0.0001	0.0001	0.0001	0.0001
RUT	285.1282	467.1597	374.4224	3093.7415	5246.3267
RUT_r	0.0001	0.0001	0.0001	0.0000	0.0000

Overall, the MSE scores of the LSTM no sentiment, LSTM with sentiment, and DP models are quite comparable, with the LSTM no sentiment model slightly outperforming the others. Meanwhile, the ARIMA and random forest models exhibit poorer performance than the time series models, with ARIMA showing significantly inferior results compared to the rest.

It is worth noting that the mean squared errors (MSEs) for predicting returns are remarkably small, despite returns being expressed as percentages, indicating the suitability of time series models for this type of prediction.

MAE	LSTM_no_senti	LSTM_senti	dp	ARIMA	rf
SP	16.6860	16.5129	16.2685	51.5201	14.7470
SP_r	0.0060	0.0058	0.0058	0.0053	0.0063
NDX	30.9512	37.1904	37.0674	158.0062	185.5751
NDX_r	0.0043	0.0043	0.0043	0.0057	0.0070
DJI	179.8912	164.2563	163.4007	469.6973	222.0697
DJI_r	0.0073	0.0068	0.0069	0.0061	0.0071
RUT	15.1944	18.8928	20.3630	52.2700	71.0966
RUT_r	0.0062	0.0059	0.0058	0.0051	0.0055

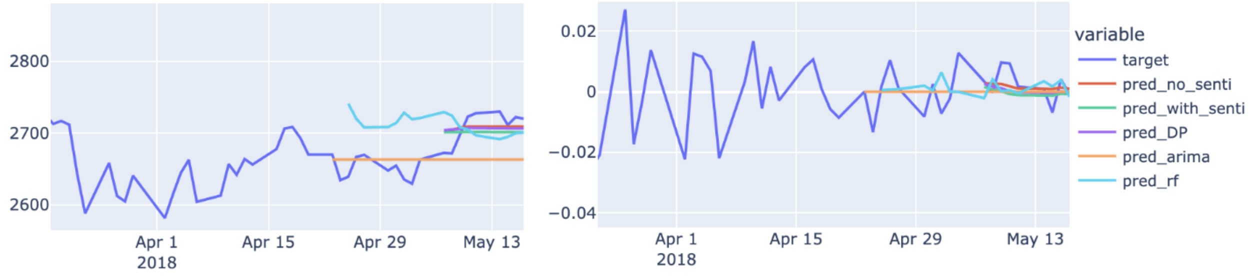
In terms of MAE, we found that the LSTM models still out-perform ARIMA and random forest. The DP method would lead to the strongest results among all methods. We can see that the time series models would work better on the return data.

Accuracy	LSTM_no_sent	LSTM_senti	dp	ARIMA	rf
SP	0.9936	0.9938	0.9939	0.9811	0.9946
NDX	0.9956	0.9944	0.9946	0.9773	0.9733
DJI	0.9927	0.9935	0.9934	0.9810	0.9910
RUT	0.9907	0.9887	0.9890	0.9678	0.9562

Accuracy would only be suitable for Asset price data because that the return data is too small in scale, the prediction error could be larger than 1, so that the accuracy metric would not be meaningful for interpretation. All four models achieved high accuracy rates above 95%, with the LSTM and DP models performing exceptionally well with accuracy rates above 99%.

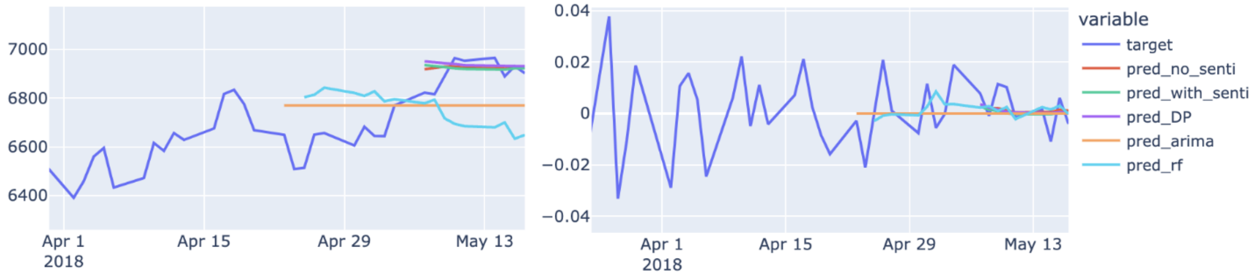
2. Prediction Results

- **S&P 500 Index and Returns**



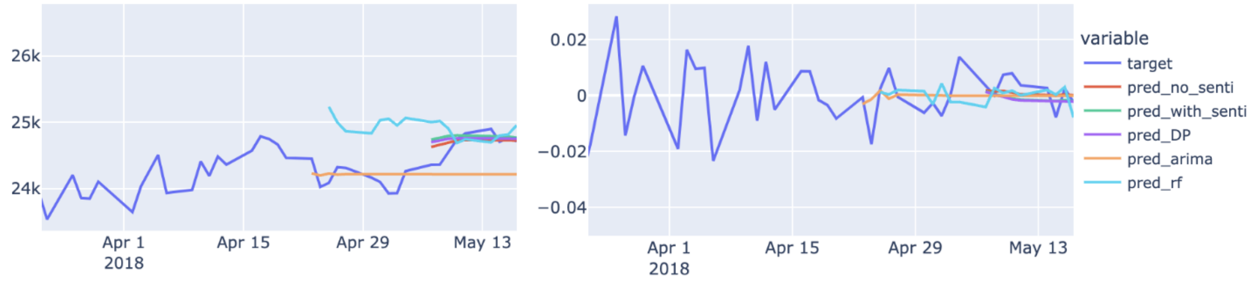
- For model predictions on S&P 500 index, the LSTM with no sentiments and with DP would lead to the best accuracies. Random forest and ARIMA may need more parameter tunings for better performance.
- For model predictions on S&P 500 returns, all models show similar performance that is relatively accurate. LSTM with sentiments and DP perform the best.

- **NDX Index and Returns**



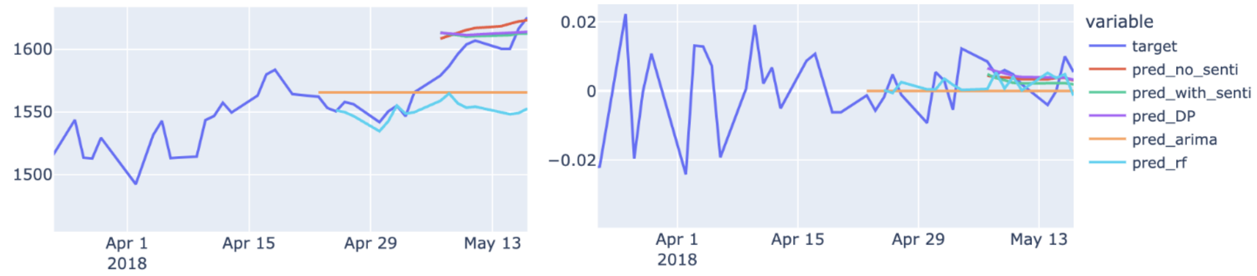
- For model predictions on NDX index, the LSTM with no sentiments and with DP would lead to the best accuracies. Random forest and ARIMA may need more parameter tuning for better performance.
- For model predictions on NDX returns, all models show similar performance that is relatively accurate. LSTM with sentiments and DP are the best.

- **DJI Index and Returns**



- For model predictions on DJI index, the LSTM models would lead to the best accuracy. Random forest and ARIMA may need more parameter tuning for better performance.
- For model predictions on DJI returns, all models show similar performance that is relatively accurate. LSTM with sentiments and DP are the best.

- **RUT Index and Returns**



- For model predictions on RUT index, the LSTM with sentiments and with DP would lead to the best accuracies. Random forest and ARIMA may need more parameter tuning for better performance.
- For model predictions on RUT returns, all models show similar performance that is relatively accurate. LSTM with sentiments and DP are the best.

3. Time Window Analysis

To explore the influence of the length of time windows on the prediction models, we set the time window to 3 days, 10 days, and 15 days, and examined the three LSTM models' performances on the S&P 500 Stock Index. The following table shows the mean absolute errors of the three models with different time windows.

	no_senti	with_senti	dp
3 days	16.7959	16.8809	16.3856
10 days	16.7605	16.8566	15.6518
15 days	16.6992	16.1918	15.3357

As we can see, models with a 15-day time window perform best regarding mean absolute errors. It is reasonable because as the length of the time window increases, the amount of information inputted into the model will also increase, thereby helping the model to make better predictions. Notably, based on the results presented, LSTM models with DP exhibit superior performance compared to the other two types of LSTM models.

IV. Conclusion and Forward Guidance

From the results above, we can conclude that the LSTM model outperformed ARIMA and Random forest for index time series prediction. All time series models work better on the return data as it is normalized with the same scale. LSTM with DP would achieve best performance in most scenarios as the features it takes would be more robust.

Furthermore, due to the limited access of financial text data, we can only train our model based on around 120 data points. Further improvements can be made through extending the sentiment datasets while utilizing Neural Networks, which would lead to better representation. In addition, we could spend more time on model parameter tuning for ARIMA and Random forest that may enhance the performance.

Citation

Xinyi Li, Yinchuan Li, Hongyang Yang, Liuqing Yang , Xiao-Yang Liu. “DP-LSTM: Differential Privacy-inspired LSTM for Stock Prediction Using Financial News.” https://doi.org/10.48550/arXiv.1912.10806.