Evaluating Predictive Power of SEC 8-K Forms on Stock Prices

Emil Eriksson
Supervisor: Ali Basirat

February 22, 2022

Abstract In the US, publicly listed companies must file a report called SEC 8-K (or Form 8-K) to inform their investors of important events in the company, such as bankruptcy or change of management. These forms follow a clear structure and may contain information that affects the stock in the days following the report, which has prompted researchers to explore processing the information using NLP in order to forecast stock prices. In this project, a deep neural network called LSTM was developed and used with GloVe word embeddings to predict UP or DOWN signals for stock prices. Importantly, only textual and no financial features were used for prediction. A unigram model with a random forest classifier was used as baseline. Despite attempts at tuning the LSTM model, it achieved 50.05% accuracy on test data, indicating that it was not able to find a predictive signal in the textual data. The unigram model achieved 52.98% accuracy, lending some weight to the usefulness of the textual information, but the majority class classifier still achieved the highest accuracy of 53.68%.

TDDE16 - Text Mining
Linköping University

1 Introduction

In the US, publicly listed companies must file a form called 8-K to the *Securities and Exchange Commission* (SEC) to inform their investors whenever important events happen in the company, such as bankruptcy or changes in management. These forms follow a clear structure and may contain information that affects the stock price in the days following the report, which has prompted researchers to explore the use of NLP to process the information in order to forecast stock prices.

Stock prices are ultimately based on supply and demand which is influenced by many factors, such as market trends, company management, and recent performance. According to the efficient market hypothesis (EMH), the stock price reflects the market's total knowledge about a company [1], and investment firms spend large amounts of resources to continuously monitor and process information about companies. Against this background, the purpose of this report is to explore if an individual investor can hope to achieve better results than a crude baseline by processing the publicly available 8-K forms using modern text mining methods, and without using any financial features.

This project is inspired by the research by Lee et al. presented in [2], where they found a 10% relative increase in predictive performance when appending textual information from the 8-K forms to multiple quantitative financial features (such as earnings surprise and recent stock movements). They used unigram features and a random forest classifier. The contribution of this report is an analysis of the predictive power of the 8-K forms when used without the financial features, and when using a different pipeline: pretrained GloVe word embeddings to represent the textual data, and a recurrent neural network called LSTM (Long Short-Term Memory) for classification.

2 Theory

This section covers two of the most important models used in the project: the GloVe word embedding algorithm, as well the recurrent neural network based on LSTM.

2.1 GloVe (Global Vectors for Word Representation)

GloVe was developed by Pennington et al. and is an unsupervised learning algorithm for obtaining vector representations for words [3]. It is based on training on the non-zero elements of a word-word co-occurrence matrix generated from a corpus. The power of the algorithm is that after learning vector representations of words from a text, analogies between words can be represented by these vectors, in addition to just semantic similarities. For example, consider the statement "*man* is to *woman* what *king* is to *queen*" - such relationships can

be captured by GloVe in the form of linear substructures, which is visualized in Figure 1 below.

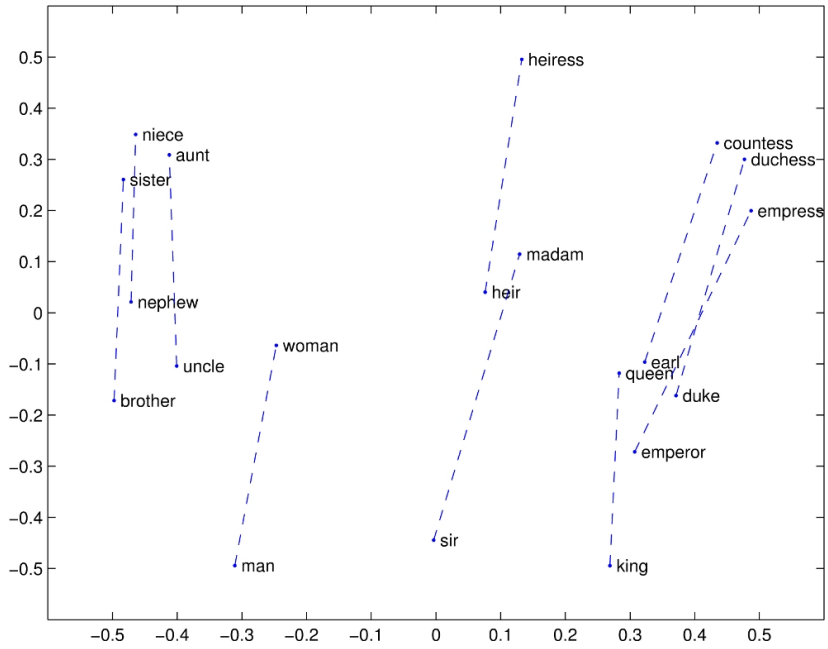


Figure 1: A visualization of the linear substructure found in the word embedding learned by the GloVe algorithm. Source: [3]

2.2 Long Short-Term Memory Network (LSTM)

The LSTM network is widely used in NLP due to its capability of remembering long-term dependencies in sequences of words, and it is applied in for example sentiment analysis, word autocompletion and text classification. It is a recurrent neural network, which is an architecture that allows temporal data to be processed by feeding the output from one step to the input in the next step. An illustration of the architecture is presented in Figure 2 below. LSTM was developed to address an important limitation in traditional RNNs, namely the *vanishing gradient problem*: information learned in one step could be forgotten in later parts of the sequence because the gradient which is used to update the weights becomes vanishingly small.

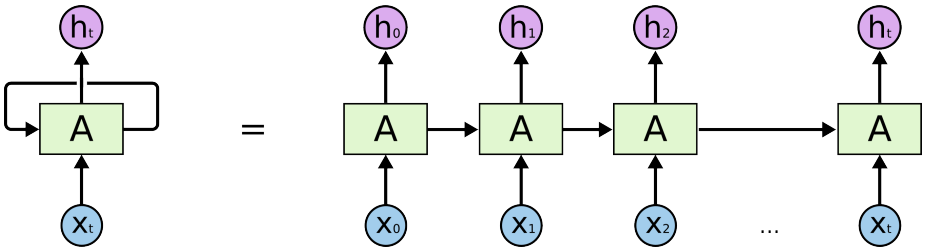


Figure 2: Structure for a RNN. X_t is the input at each time, h_t is the output. Note that the right hand side does not depict multiple layers, but the same network at different time instances. *Source:* [4]

In a traditional RNN, each instance of the network (called *cell*) feeds a single output to the next cell (shown as right arrows in Figure 2), and LSTM addresses the vanishing gradient problem by adding a state to the cell which stores long term information. To update the state, the LSTM cell uses a *forget* and an *input* gate. Consider for example the sentence "I thought the meeting was tomorrow, but it turns out I was wrong". The word "tomorrow" might be held in the state as an indicator of when the meeting would happen, but this could be forgotten from state when "wrong" showed up. The LSTM cell is illustrated in Figure 3 below.

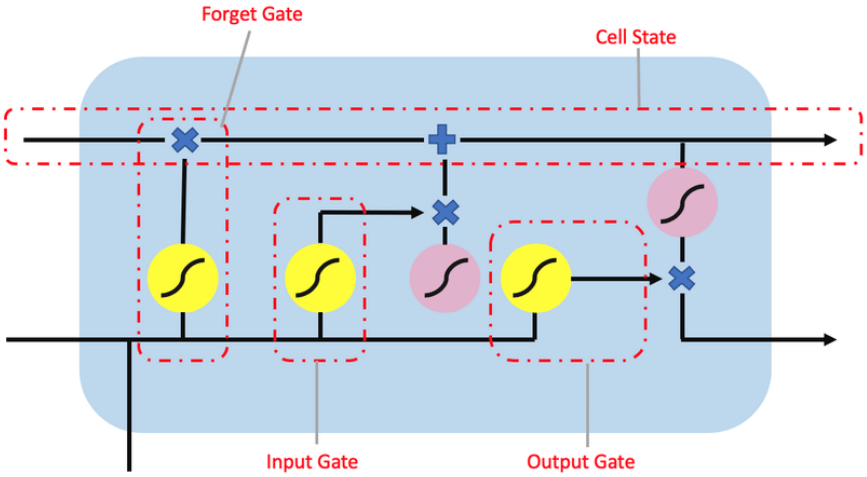


Figure 3: The LSTM cells contain a forget gate, output gate and input gate. The yellow circle represents the sigmoid activation function while the pink circle represents a tanh activation function. Additionally, the "x" and "+" symbols are the element-wise multiplication and addition operator. *Source:* [5]

3 Data

The data was kindly provided in a structured format by Julius Kittler and Max Pfundstein, and it consists of the following:

- SEC 8-K filings from 348 companies listed on S&P 500, retrieved from the SEC Edgar archive[1].
- Historical stock opening and closing prices for the companies, retrieved using the Alphavantage API[2].
- S&P 500 historical opening and closing prices, retrieved from NASDAQ[3].

In total, there are 17954 data points, between the dates 2017-01-13 and 2021-06-25.

The events reported in a filing are labeled with different event types, such as "Item 1.01: Entry into a Material Definitive Agreement", and there are 31 event types in total. The following is a filing by Applied Industrial Technologies from 2017-01-13:

> *John F. Meier, a director of Applied Industrial Technologies, Inc. ("Applied") since 2005, age 69, retired from the Board of Directors on January 10, 2017. Mr. Meier's retirement was for personal reasons and did not result from any disagreement with Applied.*

The data was split into three sets for training, validation, and testing. Since stock prices are time series data, the common method of random sampling should be avoided, since the model will train on data from events that occur after the test data we use to evaluate the model, achieving overly optimistic results. Instead, the data was split based on breakpoints in time, with the resulting distribution presented in Table 1.

Set	# 8-K Forms	# words	Time period
Training	9836	3.5M	2017-01 - 2019-06
Validation	4239	1.6M	2019-06 - 2020-06
Test	3877	1.6M	2020-06 - 2021-06

Table 1: Distribution of textual data in the different data sets. Note that it is divided in time, rather than by random sampling.

[1] https://www.sec.gov/Archives/edgar/daily-index/
[2] https://www.alphavantage.co/
[3] https://www.nasdaq.com/de/market-activity/index/spx/historical

4 Method

The experiment consisted of preprocessing the data, converting textual data into numerical values (feature extraction), training the model for prediction, and evaluation of results.

4.1 Preprocessing and Features

The targets that were used for prediction were stock price changes "UP" and "DOWN", and these were calculated based on the price percentage changes (ppc) of the stock over different periods. Let O_n and C_n be opening and closing prices of day n, and let $ppc_{n,k}$ be the ppc from day n to day k, where $n = 0$ is the day of the filing. Four ppc were calculated as follows:

$$ppc_{-1,1} = \frac{O_1 - C_{-1}}{C_{-1}} \tag{1}$$

$$ppc_{1,1} = \frac{C_1 - O_1}{O_1} \tag{2}$$

$$ppc_{1,2} = \frac{C_2 - O_1}{O_1} \tag{3}$$

$$ppc_{1,3} = \frac{C_3 - O_1}{O_1}. \tag{4}$$

The same values were calculated for the S&P 500 prices and subtracted from the stock price changes, in order for the targets to be invariant of general market movements. Finally, the adjusted ppcs were then categorized as either "UP" ($ppc >= 0$) or "DOWN" ($ppc < 0$).

Preprocessing of the texts consisted of the following steps: lemmatization, removal of non-alphabetical words, removal of 1 letter words, and tokenization. Tokenization allowed us to assign an integer to each word in the vocabulary, so that we could represent a sequence of words as a sequence of integers, which is the input format for the neural network. The tokenizer was only fitted on training data, which means that new words that occur in validation and test data are considered out-of-vocabulary and assigned a generic value of 0. Finally, each sequence was padded or truncated to 500 words (they were on average 376 words long).

Feature extraction happens in the first layer of the neural network, which is a pretrained embedding layer based on GloVe, presented in the Theory section. The embedding layer can be seen as a lookup table and interpreted in the following way: each integer representing a word is an index in the lookup table, which points to a row containing the vector representation of the word learned with GloVe. Seen this way, the embedding layer is a matrix of dimensions `[VOCABULARY_SIZE x EMBEDDING_DIMENSION]`.

4.2 Training the model

An outline of the full pipeline is presented in Figure 4 - after vectorization and embedding, the data was fed into a bidirectional LSTM layer (Bi-LSTM), then a fully connected layer, then finally classified. Bidirectional means that both directions of the sequence of words are trained on. Dropout was applied after the Bi-LSTM layer, meaning some input units were set to 0, in order to reduce overfitting. To choose suitable hyperparameters for the model, a grid search over units in Bi-LSTM layer, units in fully connected layer, and the dropout rate was conducted. For training the model and tuning parameters, the Keras wrapper for Tensorflow was used [6], [7].

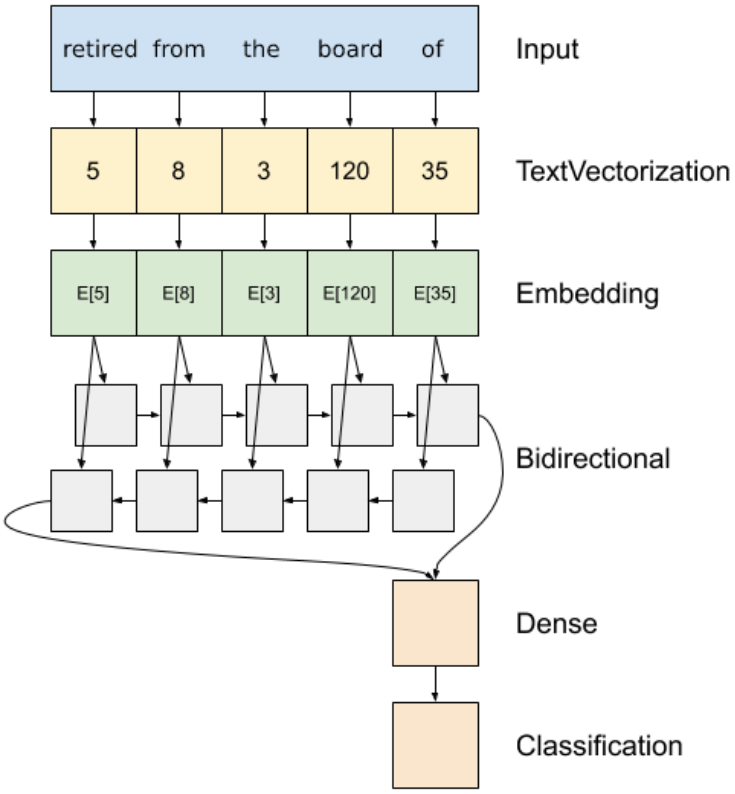


Figure 4: Structure of the model pipeline. The text input is vectorized, then embedded using GloVe, then fed into a bidirectional LSTM layer and a fully connected layer, then finally classified. *Source: [6]*

In order to optimize the model, a number of different hyperparameters were tested. This increases the training time, but can have significant impact on the results.

Hyperparameter	Grid	Selected
Bi-LSTM units	[32, 64, 128]	128
Dense units	[32, 64, 128]	128
Dropout	[0.2, 0.5, 0.7]	0.5

4.3 Models

Listed below are all models used for prediction. To evaluate the model performance, four different baselines were used, models 2-5:

- Model 1: Bi-LSTM, described above
- Model 2: Majority classifier - use the most common class as prediction on all outcomes
- Model 3: Random classifier - predict a random class
- Model 4: Day-before model - predict that the outcome will be the same as the day before
- Model 5: Unigram - a standard classifier which uses unigram features and a random forest classifier, similar to the approach used in [2]. Feature selection based on mutual information was used, in order to extract the most important words.

There are a number of other financial features which could be used and would provide much stronger baselines, which was done by Lee et al., but the focus of this report is solely on the predictive power of the texts.

5 Results

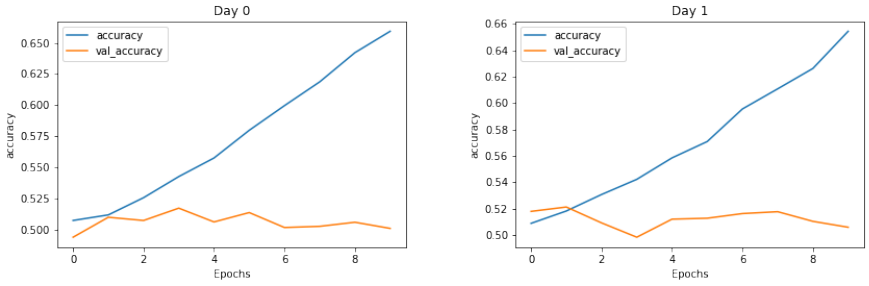
In Table 2, the accuracy results are presented for the 5 different models. "Day 0" refers to the change from the closing price on the day before the filing date, to the opening price of the day after. As can be seen in the table, the majority classifier performed best on all days.

	Test Accuracy [%]			
Model	**day 0**	**day 1**	**day 2**	**day 3**
1 (Bi-LSTM)	48.28	51.51	49.31	51.10
2 (Majority)	**54.24**	**54.06**	**53.31**	**53.18**
3 (Random)	50.14	49.93	50.27	49.88
4 (Day-before)	49.78	49.70	51.43	51.27
5 (Unigram)	53.44	53.62	53.18	51.68

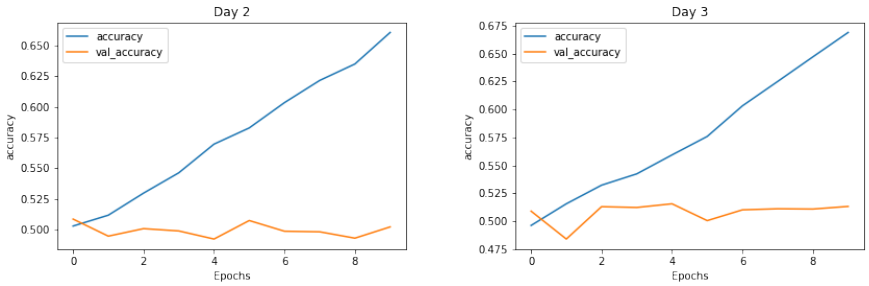
Table 2: Test accuracy (in %) for the different models. The majority classifier (model 2) had the best accuracy of all models

	Training Accuracy [%]			
Model	**day 0**	**day 1**	**day 2**	**day 3**
1 (Bi-LSTM)	71.21	70.34	70.63	71.91
5 (Unigram)	65.40	63.99	62.10	66.98

Table 3: Training accuracy (in %) for the two ML models. Both show high training accuracies.

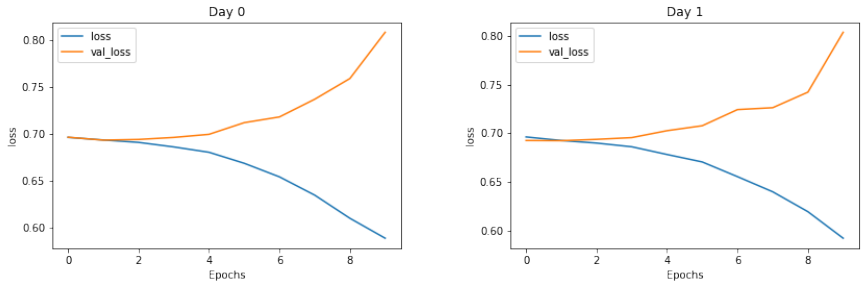


(a) From closing day -1 to opening day +1 **(b)** From opening to closing day 1

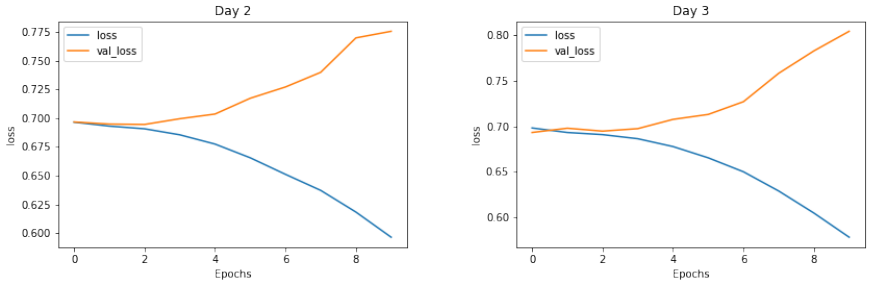


(c) From opening day 1 to closing day 2 **(d)** From opening day 1 to closing day 3

Figure 5: Training and validation accuracy for the neural network training over 10 epochs. No clear improvement in validation accuracy can be seen over the epochs.



(a) From closing day -1 to opening day +1 **(b)** From opening to closing day 1



(c) From opening day 1 to closing day 2 **(d)** From opening day 1 to closing day 3

Figure 6: Training and validation loss for the neural network training over 10 epochs. Loss increases for the validation data over the epochs.

6 Discussion

Table 2 shows that no significant predictive signal on stock price movement could be found in the textual information in the reports when using the Bi-LSTM model. Several approaches were tested in order to try to improve the result, including a grid search of different hyperparameters in the network, trying two different sizes of word-vectors from GloVe (6B tokens and 42B tokens), and the truncating reports to 200, 500 and 2000 words, but the model was never able to beat the random baseline. Figure 5 shows further that accuracy did not improve over the ten epochs that the network was trained. However, the Unigram model did beat the random baseline (although not the majority baseline) which indicates that some predictive signal could be found in the texts. This was a similar model used by Lee et. al.

When comparing the experiment setup with Lee et. al who achieved a significantly better result, the biggest difference between the approaches is the use of financial features. This could be an important factor in explaining the poor performance of the Bi-LSTM model. They used three classes in their experiment: UP, DOWN, STAY, and they had an accuracy of 34.9% for their majority class model. By only incorporating earnings surprise, a measure of the difference between profit that analysts expected of a company and the reported profit, they managed to improve the baseline to 49.7%, a relative performance increase of 41.5% (!). With other financial features incorporated, their baseline was 50.1% accuracy. Seen this way, adding the unigram model which resulted in an accuracy of 54.4% was a comparatively small improvement. It raises the question if textual information is useful mainly when other financial factors are accounted for. Many of the reports were neutral, or hard to draw conclusions from - for example, does the departure of an officer imply a mishandled company in decline, or a hopeful company hiring new talent? (It turns out this particular event resulted in stock prices going up and down approximately the same number of times). However, if combined with for example a negative earnings surprise, the same event could provide more context and support the hypothesis that the company is struggling.

In order to gain better understanding of the network, an excerpt from a report which the network had very high confidence would result in UP the coming day is presented below:

> *On September 18, 2020, the Registrant entered into an accelerated share repurchase agreement with Goldman Sachs & Co. LLC ("Goldman Sachs") to purchase $750 million of its outstanding common stock. The majority of the shares to be repurchased under this agreement will be received by the Registrant at the agreement's inception. It is expected that Goldman Sachs will purchase the shares that it delivers under the agreement in the market no later than January 12, 2021. The final purchase price per share and number of shares to be delivered by Goldman Sachs will be determined at the conclusion of the agreement and settlement will consist of the Registrant receiving shares based on the average of the daily volume weighted average prices of the Registrant's common stock during the period of Goldman Sachs's purchases. [...]*

An accelerated share repurchase generally has a positive impact on stock prices according to Bargeron et. al which the network correctly identified, but the stock actually fell by over 1% for three days in a row. [8]. This exemplifies what is generally known about stock markets: they're unpredictable, and there are myriad of parameters which affect stock price movements.

This isn't to say that textual information from SEC 8-K reports don't carry predictive signals of stock price movements. An inherent difficulty when training deep neural networks is that the parameter space is vast, which means that there potentially are much better configurations than the ones used in this experiment out there. Training accuracy is shown in Table 3, and the higher accuracy of the LSTM network when compared with the unigram model could indicate a tendency for overfitting, which could be remedied by a more detailed optimization. Overfitting is further supported by Figure 6, where it can be seen that validation loss increase over the epochs. Given more computing power and time, it would be possible to explore a larger chunk of the parameter space, but this will be left for future research.

7 Conclusion

The results show that forecasting stock prices is not a trivial task, and despite the promising results presented in [2], we were not able to achieve any significant predictive power using an LSTM model. A central limitation to the experiment was to rely only on textual data which, based on the results, was inferior to using financial features. It should be noted that the unigram model achieved a 6% increase compared with the random baseline, which indicates some importance of textual data in predicting stock prices. In the end, the majority class model outperformed all other models, which supports the common sense investment strategy: the market generally moves up.

While SEC 8-K forms are one source of textual information which can affect stock prices, there are many other sources that can have at least as large an impact, for example news sites, blogs, and social media, especially Twitter. The benefit of using such sources for training and prediction is that they are much more frequently updated, which means that there is more data to train on, and they reflect the opinion of the people who will actually buy and sell the stocks in the end. Using for example sentiment analysis on tweets could be an effective way to gauge public opinion. For future research, it would be interesting to see the use of such sources, as well as a continued exploration of the parameter space for SEC 8-K reports.

References

[1] B. G. Malkiel, "The efficient market hypothesis and its critics," *Journal of Economic Perspectives*, vol. 17, no. 1, pp. 59–82, Feb. 2003. DOI: 10.1257/089533003321164958. [Online]. Available: https://doi.org/10.1257/089533003321164958.

[2] H. Lee, M. Surdeanu, B. MacCartney, and D. Jurafsky, "On the importance of text analysis for stock price prediction," in *LREC*, 2014.

[3] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162.

[4] C. Olah, "Understanding LSTM networks," [Online; accessed 11-January-2022]. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[5] D. Rengasamy, M. Jafari, B. Rothwell, X. Chen, and G. Figueredo, "Deep learning with dynamically weighted loss function for sensor-based prognostics and health management," *Sensors*, Jan. 2020. DOI: 10.3390/s20030723.

[6] Martın Abadi, Ashish Agarwal, Paul Barham, *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: https://www.tensorflow.org/.

[7] F. Chollet *et al.* "Keras." (2015), [Online]. Available: https://github.com/fchollet/keras.

[8] L. Bargeron, M. Kulchania, and S. Thomas, "Accelerated share repurchases," *Journal of Financial Economics*, vol. 101, no. 1, pp. 69–89, 2011.

Appendix