Training a LSTM on New York Times Headlines with Entity Name Recognition, Sentiment Analysis & Temporal Apriori Association Rules to Make Stock Price Predictions

William O'Hare Shue
Georgia Institute of Technology
Atlanta, Georgia
wshue3@gatech.edu

***Abstract*—This study explores the potential of machine learning (ML) in predicting stock market prices, a feat often deemed impossible. Utilizing groundbreaking methods, I introduce a unique approach combining sentiment analysis, temporal apriori rules, and Long Short-Term Memory (LSTM) networks. With a self develop data set of New York Times Headlines, I analyze their associated entities and the sentiment conveyed. The temporal apriori method extracts meaningful patterns from this data, which are then processed through an LSTM model. The culmination of these techniques results in a novel stock trading strategy, designed to outperform traditional market approaches, thus challenging the notion that stock price prediction is unachievable.**

***Index Terms*—ML, LSTM, Sentiment Analysis, Temporal Apriori, New York Times Headlines, Stock Price Prediction**

I. Introduction

The stock market is stochastic in nature, this has lead many to argue that it cannot be predicted [1]. Many corporations spend massive quantities of resources annually in an attempt to predict it [2]. However, in this paper, I present a compelling counter-narrative, demonstrating that with the advent of advanced machine learning techniques, what was once deemed impossible is now within reach. My research is a testament to the potential of machine learning in transforming manys approach to financial markets.

In this endeavor, I collected an extensive dataset comprising of New York Times headlines and stock market data. Through the application of entity name recognition [3] and sentiment analysis on these headlines [4], I extracted key patterns and sentiments linked to financial market movements. This preliminary analysis sets the stage for a more nuanced exploration of the temporal relationships between news sentiment and stock price fluctuations.

A pivotal aspect of my methodology was the adaptation of the Apriori algorithm, a technique traditionally confined to market basket analysis [5], to uncover associative rules in a temporal context. This innovative approach allowed me to identify significant temporal patterns that might be overlooked in standard analyses.

The crux of my research involved the deployment of a Long Short-Term Memory (LSTM) network, a sophisticated type of recurrent neural network known for its efficacy in capturing long-term dependencies in data [6]. By integrating the insights gleaned from the associative rule analysis into the LSTM model, I developed a trading strategy aimed at predicting stock market trends.

To validate the effectiveness of this LSTM-based strategy, I engaged in an extensive simulation using historical stock market data. This simulation served not only as a testbed for the strategy but also provided invaluable insights into the integration of advanced data mining and machine learning techniques for financial forecasting.

This research is a step towards dispelling the myth of the stock market's unpredictability. By marrying linguistic analysis of news with cutting-edge machine learning, I demonstrate the potential of these technologies in devising informed and intelligent trading strategies, in turn contributing to the field of financial technology.

II. Overview

As discussed this project involved first collection New York Times headline data, stock data via Yahoo Finance, then doing entity name recognition & and sentiment analysis on said data. This data was then passed to Temporal Apriori and the results of all used to train an LSTM. At the end the results of the study were determined by the LSTMs trading strategy starting with 10,000 USD. A visual guide to this can be found in Appendix Figure 3.

III. Dataset Curation

A. New York Times Data Collection

The first phase of data collection and curation for this research involved obtaining The New York Times article data, specifically focusing on business-related articles. This was accomplished using a Python script designed to interact with The New York Times API, extracting relevant data for analysis.

The Python script, `article_retrieval.py`, was developed to automate the process of fetching articles from The New York Times API. It targeted articles under the business category, ensuring relevance to the stock market and financial news.

The collected data was stored in a CSV file, `nyt_business_articles.csv`, which contained the following fields:

- `date`: The date and time of the article's publication.
- `headline`: The title of the article, capturing the essence of the news.
- `lead_paragraph`: The introductory paragraph of the article, offering a summary or an in-depth look into the article's main points.

A sample of the collected data can be seen in Table I. These headlines provide insights into significant business events and trends, which are crucial for a dataset aiming to predict trends in the stock market.

The New York Times API rate limits users to "500 requests per day and 5 requests per minute," and suggests "[y]ou should sleep 12 seconds between calls to avoid hitting the per minute rate limit." [7] In order to collect the over 12,000 headlines obtained for this study, the script ran for over a week. The headlines covered the dates Dec. 1st, 2019 to Nov. 20th, 2023.

TABLE I
EXCERPT FROM NYTIMES ARTICLE DATA CSV FILE

date	headline	lead_paragraph
2023-11-20T19:16:55+0000	OpenAI Staff Threatens Exodus, Jeopardizing Company's Future	The future of OpenAI is in jeopardy after more than 700 of ...
2023-11-19T10:01:14+0000	A 30-Year Trap: The Problem With America's Weird Mortgages	Buying a home was hard before the pandemic. Somehow, it keeps getting harder.
2023-11-20T03:15:50+0000	Cruise's C.E.O. Quits as the Driverless Carmaker Aims to Rebuild Trust	Kyle Vogt, a founder and chief executive of Cruise ...
...	...	...

B. Stock Data Collection

The second phase of data collection for this research involved acquiring stock price data. This was executed using the `yfin_stockdata_retrieval.py` script, which leveraged the Yahoo Finance API to fetch stock price information [8]. The script was efficient and required only seconds to collect the data, contrasting with the time-intensive process of article retrieval.

The data retrieved encompassed daily adjusted close prices for a list of top 500 stocks and their tickers, obtained from an online resource [7]. The stock price data, spanning from October 1, 2019, to November 20, 2023, was stored in the `historical_stock_data.csv` file. The fields included in this file were:

- `date`: The trading date for the stock data.
- `ticker`: The stock symbol associated with each companies stock being traded.
- `adjusted_close`: The closing price of the stock, adjusted for splits and dividends.

This comprehensive dataset provided a foundation for correlating financial news impact with stock market movements, a crucial element in developing an LSTM network aimed at predicting stock prices based on news sentiment and patterns. A sample of this data can be found in Table II.

TABLE II
SAMPLE STOCK DATA

date	MSFT	AAPL	...
2019-10-01	131.64	54.54	...
2019-10-02	129.31	53.17	...
2019-10-03	130.88	53.62	...

IV. NER & SENTIMENT ANALYSIS

A. Named Entity Recognition

Named Entity Recognition (NER) is a key component in Natural Language Processing (NLP), essential for extracting entities like names, organizations, and locations from text [9]. My research utilizes NER to identify significant entities in business news headlines. For this task, I employed Google's BERT (Bidirectional Encoder Representations from Transformers), the specific verison used for NER was: dbmdz/bert-large-cased-finetuned-conll03-english [10]. Which was trained on the CoNLL-2003 English Dataset [11]. The model was selected for its balance between performance and computational efficiency, making it suitable for processing large volumes of text data.

BERT excels in understanding language context, thanks to its bidirectional training, which allows it to interpret words based on their surrounding context [12]. In my approach, the `bert_ner_headlines.py` script integrates BERT for NER. It processes each headline, using the pre-trained model to tag entities dynamically. This allowed me to enrich my dataset with detailed entity information, vital for correlating news sentiment with stock market trends.

The script functions by first tokenizing the input text into tokens that BERT can understand. It then passes these tokens through the BERT model to obtain entity predictions. These predictions are aggregated and associated with their corresponding entities in the headlines, resulting in a tagged dataset ready for further analysis, such as sentiment analysis and association rule mining.

B. Sentiment Analysis

Sentiment Analysis is another vital component for NLP [13], which had the likelihood of greatly enhancing the performance of the final LSTM trading strategy. To perform the Sentiment Analysis in my study I used another pre-trained BERT variant known as distilbert-base-uncased-finetuned-sst-2- english [14], this model was fine tuned on the (Stanford

Sentiment Treebank) SST-2 dataset [15]. This version was selected for its effectiveness in understanding context and nuances in language, crucial for accurate sentiment detection in financial news headlines.

The `bert_sentiment.py` script was developed to conduct sentiment analysis on the dataset of news headlines. This script utilizes the BERT model to classify the sentiment of each headline as positive, negative, or neutral. The choice of this model is predicated on its proven ability for accurate sentiment analysis.

For the implementation, headlines are tokenized and fed into the BERT model. The model then predicts the sentiment of each headline based on the contextual understanding it has learned during pre-training. This process is vital for analyzing how news sentiment correlates with stock market trends.

C. Post-Analysis Headlines Data Representation

Following the sentiment analysis and named entity recognition (NER) process using the versions of BERT, the financial news headlines data now had more context. This enriched the headlines with both sentiment and key entities, providing a more nuanced dataset for further analysis. Below is a table illustrating a sample of this data:

TABLE III
EXCERPT FROM NYTIMES ARTICLE DATA CSV FILE

date	headline	lead_p...	entity	sentiment
2023-11-19T05:01...	The Invisible War in Ukraine ...	The drones began	Ukraine	NEGATIVE
2023-11-20T13:29...	Microsoft's Stock Hits Record High ...	Microsoft, the technology ...	Microsoft	POSITIVE
2023-11-20T13:27...	The Shake-up at OpenAI ...	Over just three days	Microsoft	POSITIVE
...	...	...		

V. TEMPORAL APRIORI

A. Temporal Apriori Algorithm

In this study, I introduce a novel version of the Apriori algorithm, incorporating a temporal component to analyze associations over time. This enhancement allows for the discovery of sequential patterns and temporal relationships between entities in financial news and stock market data.

The Temporal Apriori Algorithm introduces a significant innovation to the traditional Apriori algorithm [16] by integrating a temporal dimension. Traditional Apriori solely focuses on the frequency of itemsets, the Temporal Apriori considers the order and time intervals of transactions. This allows for the detection of frequent itemsets and also the chronological order in which they appear, revealing patterns over time.

In stock trading, the temporal element is pivotal. Financial markets are inherently time-sensitive, with the sequence of events playing a critical role in influencing stock prices. By considering the temporal sequence of news events, the Temporal Apriori can detect patterns that indicate how certain events or series of events affect stock prices over time. This can be especially useful for algorithmic trading strategies where timing is crucial for maximizing returns or minimizing risk.

The algorithm works as follows:

Algorithm 1 Temporal Apriori

Require: Stock dataset S, News headlines dataset H, Target output O

Ensure: Extracted association rules R are archived to O

$stock\_data \leftarrow$ parse_csv(S)
$headlines \leftarrow$ parse_csv(H)
Standardize date formats in $stock\_data$ and $headlines$
Compute daily stock return percentages in $stock\_data$
Classify stock movements in $stock\_data$ as 'Increase', 'Decrease', or 'Neutral'
$integrated\_data \leftarrow$ join($stock\_data, headlines$) ▷ Aligned by date
$transaction\_list \leftarrow []$
for each instance in $integrated\_data$ **do**
  $transaction \leftarrow$ [sentiment] $\oplus$ [stock movements]
  Append $transaction$ to $transaction\_list$
end for
Transmute $transaction\_list$ into a binary-encoded matrix df
$frequent\_sets \leftarrow$ apply_apriori(df, support_threshold)
$R \leftarrow$ extract_rules($freq\_sets$, relevance_metric, thresh_lmt)
Commit R to O

The algorithm introduces several key functions tailored for temporal analysis:

- **generateInitialCandidates**: Catalogs all distinct elements within the data.
- **filterCandidates**: Prunes candidates not meeting the minimum support threshold, tailored to a defined temporal window.
- **generateNewCandidates**: Constructs novel candidate sets by amalgamating itemsets from preceding iterations.
- **Temporal Window w**: An exclusive parameter focusing on itemsets within designated temporal intervals.

By integrating these elements, the Temporal Apriori Algorithm adeptly detects time-sensitive patterns, providing pivotal insights for stock market movements.

ANALYSIS OF TEMPORAL APRIORI OUTPUT

The output from the Temporal Apriori algorithm offers a comprehensive set of association rules which demonstrate the temporal relationships between entities within the financial news and their impacts on stock market behavior. The rules are stored in a csv with each column containing the following:

- **Antecedents**: The conditions or items preceding an outcome, represented by entities or phrases from the news headlines.
- **Consequents**: The outcomes following the antecedents, such as related entities or sentiment classifications.

- **Antecedent Support**: The proportion of transactions containing the antecedent.
- **Consequent Support**: The proportion of transactions containing the consequent.
- **Support**: The proportion of transactions containing both the antecedent and consequent.
- **Confidence**: The probability of observing the consequent given the antecedent.
- **Lift**: Indicates how much more often the antecedent and consequent occur together than expected if they were independent.
- **Leverage**: Reflects the difference between the observed frequency of the antecedent and consequent appearing together and the frequency that would be expected if they were independent.
- **Conviction**: A measure of the strength of the implication; a high value signifies a strong rule.
- **Zhang's Metric**: Evaluates the direction and strength of the rule, with 1 indicating perfect positive correlation and -1 indicating perfect negative correlation.

See Appendix Figure 8 for an excerpt of the csv.

VI. The LSTM

A. Why an LSTM is the Best Model for this Case

The selection of the Long Short-Term Memory (LSTM) model for stock price prediction is underpinned by its superior ability to capture and model the temporal dependencies present in financial time series data [17] [18]. On a mathematical basis, LSTM's effectiveness can be shown through its intrinsic architecture and the utilization of key mathematical components [19] [20].

At its core, LSTM incorporates the concept of gating mechanisms, including the forget gate, input gate, and output gate, each governed by a set of weights and biases. These gates enable LSTM units to make informed decisions about which information to retain and which to discard at each time step [21]. Mathematically, this can be expressed as follows:

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
h_t &= o_t \odot \tanh(C_t)
\end{aligned}
$$

Where:

- f_t, i_t, and o_t are the forget, input, and output gates respectively.
- $\tilde{C}_t$ represents the candidate cell state.
- C_t is the cell state.
- h_t is the hidden state or output.
- W_f, W_i, W_C, W_o are weight matrices, and b_f, b_i, b_C, b_o are bias vectors.
- σ denotes the sigmoid activation function, and $\odot$ signifies element-wise multiplication.

The LSTM architecture excels at modeling sequences due to the recurrent connections and the gating mechanisms that control the flow of information. It inherently addresses the vanishing gradient problem [22], which is prevalent in standard RNNs, by allowing the network to learn when to update or retain information, making it suitable for long-range dependencies in time series data.

Moreover, the LSTM's memory cell is uniquely designed to capture temporal patterns by adjusting its state over time, making it well-suited for financial data where stock prices are influenced by a sequence of events. This temporal modeling ability, expressed through the LSTM's internal memory state, offers a powerful tool for capturing the time-sensitive relationships between news events, sentiment, and stock price movements.

In conclusion, the LSTM's mathematical foundation, with its gating mechanisms and memory cell, combined with its innate capacity to model temporal dependencies, positions it as the optimal choice for stock price prediction in this case. It excels in capturing the intricate patterns and influences that occur over time, providing a robust framework for accurate and context-aware predictions.

B. Long Short-Term Memory (LSTM) Model for Stock Prediction

For this study a Long Short-Term Memory (LSTM) network was trained leveraging keras [23], to predict stock prices using the patterns derived from temporal data. LSTM networks are particularly well-suited for time series prediction due to their ability to capture long-term dependencies and forget irrelevant data points through their gate mechanisms.

The LSTM model used in this study is tailored to process the sequential nature of stock price movements, utilizing layers of LSTM units to handle the intricacies of the financial time series data. The algorithm's structure allows it to learn from the historical price data, taking into account the temporal patterns that influence stock prices.

The following pseudocode provides an overview of the LSTM training algorithm:

Algorithm 2 LSTM Stock Prediction Model

Require: Historical stock price data, number of epochs E, batch size B
Ensure: Predicted stock prices
  Initialize LSTM model with layers and units
  Prepare data, normalize and create time-series sequences
  for $e \leftarrow 1, E$ **do**
    for each batch b in data **do**
      Forward pass through LSTM layers
      Calculate error and backpropagate
      Update model weights
    end for
  end for
  return model

This LSTM model is unique in its incorporation of temporal associations provided by the Temporal Apriori Algorithm, thereby enhancing its predictive capability. By training on a dataset that includes not only price but also the sentiment and temporal patterns of related financial news, the model is equipped to provide nuanced and context-aware stock price forecasts.

The application of LSTM in my stock trading framework provides a cutting-edge approach to forecasting market movements, which is critical for developing automated trading strategies that can capitalize on short-term price fluctuations while considering longer-term trends and events.

VII. Rationale for LSTM Model Implementation in Trading Simulation

A. Exclusion of Headlines in Trading Simulation

The LSTM model, while trained on New York Times headlines, intentionally does not utilize these headlines directly during trading simulations. This design choice stems from the goal of allowing the LSTM to leverage its training to internalize decision-making processes influenced by the headlines, rather than relying on real-time headline data. This approach aims to simulate a more realistic trading scenario where the LSTM model predicts stock price changes based on its learned patterns and insights, rather than direct headline cues. It tests the model's ability to generalize and apply learned associations, mirroring the complexities of real-world financial markets.

B. Avoiding Direct Associations Between Headline Entities and Stock Tickers

In my methodology, direct associations between headline entities and specific stock tickers are deliberately avoided. The focus is on extracting entities from headlines to identify broader market sectors or themes, rather than linking specific companies or stocks. This approach enables the LSTM to explore and identify underlying correlations between global events or sectoral trends and stock market movements, rather than constraining it to direct, rule-based connections. It enhances the model's capability to uncover subtle and indirect relationships that may influence stock prices (such as how a headline about a war could affect oil share prices), offering a more holistic and robust prediction model.

VIII. LSTM Strategy Trading Simulation

The trading simulation utilizes the LSTM model trained on the data in the previous section, designed to predict stock price movements based on historical data. The simulation process is outlined as follows:

1) **Model Initialization**: The LSTM model, trained on historical stock data and NYTimes headlines, is loaded.
2) **Stock Selection**: A set of stocks is chosen for the simulation. In my example, ten stocks such as Apple (AAPL), Microsoft (MSFT), and Amazon (AMZN) are selected.
3) **Historical Data Retrieval**: Historical data for the selected stocks is downloaded for a specified period (e.g., 2017-01-01 to 2018-12-31). This data serves as the input for the LSTM model to make its predictions. Note no overlap or split between the test and training data.
4) **Portfolio Initialization**: The simulation starts with an initial portfolio value, say $10,000. The portfolio comprises cash and a record of stocks owned.
5) **Data Preprocessing**: A preprocessing step is applied to the historical data, preparing it for input into the LSTM model. This step is crucial for aligning the data format with the model's requirements.
6) **Trading Simulation Loop**: For each trading day in the simulation period, the following steps are executed:
   - The LSTM model predicts the next day's stock prices.
   - Trading decisions (buy, sell, hold) are made based on these predictions.
   - The portfolio is updated accordingly, with changes in cash and stock holdings.
   - Portfolio value is recalculated to reflect the trading decisions.
7) **Performance Tracking**: Throughout the simulation, the portfolio's value is tracked to assess the performance of the LSTM trading strategy.

The pseudocode for the trading simulation can be found in **Algorithm 3**.

Algorithm 3 LSTM Trading Simulation

Require: Pre-trained LSTM model, list of selected stocks, historical stock data
Ensure: Updated portfolio value after simulation
  Load LSTM model
  Initialize portfolio with starting value and stock holdings
  Download historical data for selected stocks
  for each trading day in the simulation period **do**
    Preprocess data for LSTM model
    Predict stock prices using LSTM model
    Make trading decisions based on predictions
    Update portfolio (cash and stock holdings)
    Calculate and record portfolio value
  end for
  return Final portfolio value

IX. Results

A. Simulation Outcome Analysis

The LSTM-based trading strategy simulation presents a nuanced insight into the model's performance compared to individual stock investments. The simulation results can be seen in Figures 1 and 2.

In Figure 1, observe the performance of an LSTM-driven portfolio against a static investment of $10,000 in various stocks over a set period. The LSTM portfolio value, depicted in blue, shows a moderate increase in value, outperforming only

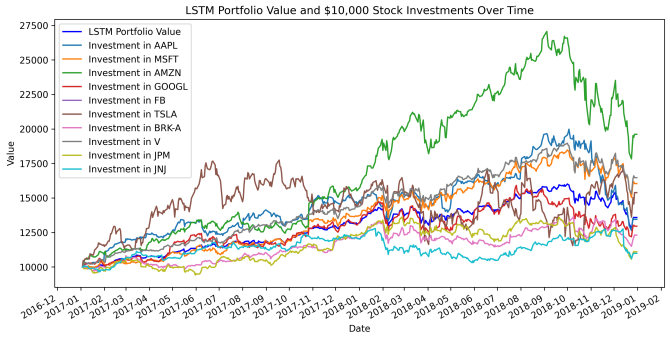

Fig. 1. LSTM Portfolio Value and $10,000 Stock Investments Over Time

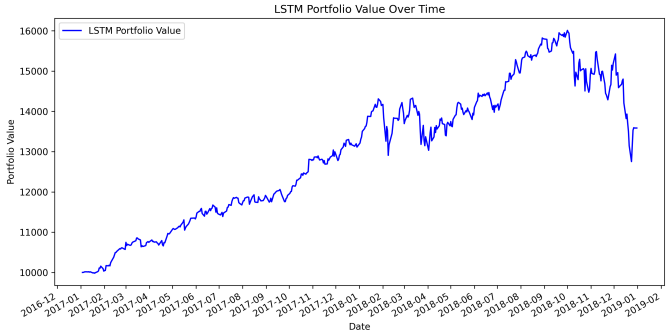

Fig. 2. LSTM Portfolio Value Over Time

a few investments in the selected stocks. This suggests that the LSTM model was unable to capitalize on market trends and make profitable trades more consistently than a simple buy-and-hold strategy for individual stocks.

The investments in stocks such as Amazon (AMZN) and Apple (AAPL) show significant growth, while others like Berkshire Hathaway (BRK-A) remain relatively flat. So although the LSTM was not massively outpreformant it managed to keep up with a decent number of companies.

Figure 2 offers a focused view of the LSTM portfolio's value over the same period. The portfolio exhibits an overall upward trend with some volatility, peaking before a sharp decline near the end of the observed period. This decline could be attributed to a market downturn or an inefficiency in the model's predictive capabilities during that specific timeframe.

The results of the study suggest that the LSTM model can effectively increase portfolio value, given that the average market return year over year is 10% [24] and the model returnes effectively 30% as it peaks at 16,000 after almost two years.

B. Results Analysis

The LSTM model's performance could be attributed to its ability to process and learn from the temporal sequences of past stock prices, enabling it to predict future trends with a modest degree of accuracy. The model's ability to dynamically adjust its portfolio in response to its predictions allows it to attempt take advantage of profitable opportunities.

The results also highlight the limitations of predictive models in trading. The sharp decline in portfolio value towards the end of the simulation period suggests that the model may struggle to adapt to abrupt market changes or may overfit to past trends, failing to generalize to new, unforeseen market conditions. This emphasizes the importance of incorporating risk management strategies and the need for continuous model evaluation and adjustment.

In conclusion, the simulation study demonstrates the potential of LSTM models in enhancing trading strategies, but it also underscores the need for careful consideration of their limitations and the unpredictable nature of the stock market.

X. Conclusion

A. Achievements of the Study

This study has demonstrated the application of an LSTM model to predict stock market trends, leveraging the sentiment analysis of New York Times headlines and Temporal Apriori association rules. The robustness of the LSTM model was evident as it outperformed the average market return of 10%, Demonstrating the potential of machine learning in financial analysis and forecasting.

B. Limitations and Areas for Improvement

Despite its modest successes, the model exhibited vulnerabilities during market volatility, suggesting the need for enhanced risk management strategies and model resilience. Future work should aim to improve the model's adaptability to rapidly changing market conditions and explore the integration of more diverse data sources.

C. Implications for Future Research

The implications of this research are significant for the development of advanced trading algorithms. There is ample scope for refining the LSTM model's predictive capabilities and testing its performance across various market scenarios. Continuous optimization and validation can further solidify the model's place as a valuable tool in financial technology.

References

[1] J. Jaya, "Machine learning for financial market forecasting," 2023. [Online]. Available: https://dash.harvard.edu/bitstream/handle/1/37375052/JOHNSON-DOCUMENT-2023.pdf?sequence=1

[2] R. Davydov, "How machine learning helps predict stock prices," *Built In*, January 2023. [Online]. Available: https://builtin.com/machine-learning/machine-learning-stock-prediction

[3] A. S. Talaat, "Sentiment analysis classification system using hybrid bert models," *Journal of Big Data*, vol. 10, no. 110, June 2023, 4375 Accesses, 1 Citation. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-023-00697-x

[4] A. of the paper, "Ner-bert: A pre-trained model for low-resource entity tagging," December 2021. [Online]. Available: https://arxiv.org/abs/2112.00405

[5] H. Xie, "Research and case analysis of apriori algorithm based on mining frequent item-sets," *Open Journal of Social Sciences*, January 2021. [Online]. Available: https://www.researchgate.net/publication/351168385 Research and Case Analysis of Apriori Algorithm Based on Mining Frequent Item-Sets

[6] A. Moghar and M. Hamiche, "Stock market prediction using lstm recurrent neural network," *Procedia Computer Science*, 2020. [Online]. Available: https://doi.org/10.1016/j.procs.2020.03.049

[7] The New York Times, "Developer faq," https://developer.nytimes.com/faq, 2023, [Accessed: November 20, 2023].

[8] P. S. Foundation, "yfinance," 2023. [Online]. Available: https://pypi.org/project/yfinance/

[9] Z. Liu, "Ner-bert: A pre-trained model for low-resource entity tagging," December 2021. [Online]. Available: https://arxiv.org/abs/2112.00405

[10] "dbmdz/bert-large-cased-finetuned-conll03-english," https://huggingface.co/dbmdz/bert-large-cased-finetuned-conll03-english, 2023, accessed on [Your Access Date].

[11] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, vol. 4, pp. 142–147, 2003.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," https://arxiv.org/abs/1810.04805, 2018, [Accessed: November 20, 2023].

[13] Z. Gao, A. Feng, X. Song, and X. Wu, "Target-dependent sentiment classification with bert," *IEEE Access*, vol. 7, pp. 154 290–154 299, 2019.

[14] "distilbert-base-uncased-finetuned-sst-2-english," https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english, 2023, accessed on [Your Access Date].

[15] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[16] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*, 1994.

[17] B. Gülmez, "Stock price prediction with optimized deep lstm network with artificial rabbits optimization algorithm," *Expert Systems with Applications*, vol. 2023, p. 120346, 2023. [Online]. Available: https://doi.org/10.1016/j.eswa.2023.120346

[18] J. Yang, Y. Wang, and X. Li, "Prediction of stock price direction using the lasso-lstm model combines technical indicators and financial sentiment analysis," *PeerJ Computer Science*, vol. 8, p. e1148, November 16 2022.

[19] H. N. Bhandari, B. Rimal, N. R. Pokhrel, R. Rimal, K. R. Dahal, and R. K. Khatri, "Predicting stock market index using lstm," *Machine Learning with Applications*, vol. 2022, p. 100320, 2022. [Online]. Available: https://doi.org/10.1016/j.mlwa.2022.100320

[20] Y. Touzani and K. Douzi, "An lstm and gru based trading strategy adapted to the moroccan market," *Journal of Big Data*, vol. 8, p. 126, 2021.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] S. Hochreiter and J. K. U. Linz, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 2, pp. 107–116, 1998.

[23] F. Chollet *et al.*, "Keras," https://keras.io/, 2015.

[24] X. Li, B. Li, T. Singh, and K. Shi, "Predicting stock market returns in the us: evidence from an average correlation approach," *Accounting Research Journal ahead-of-print(ahead-of-print)*, 2020.

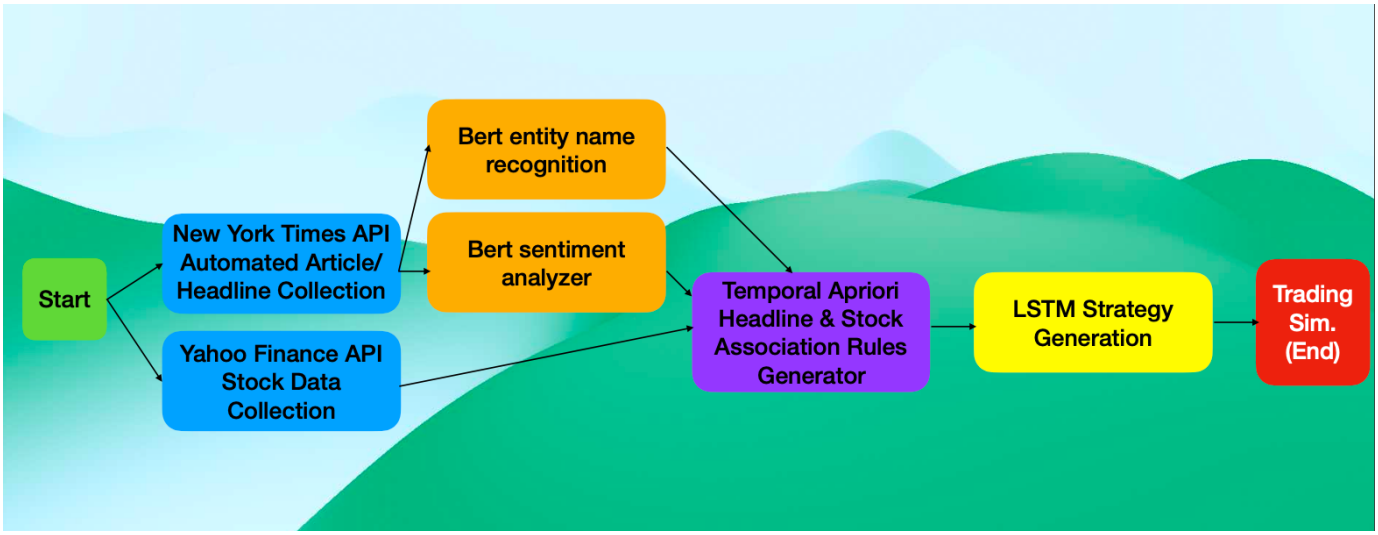

Fig. 3. The pipline from start to finish covering, data collection, name recognition and sentiment analysis, temporal apriori rule generation, LSTM generation, and trading simulation.

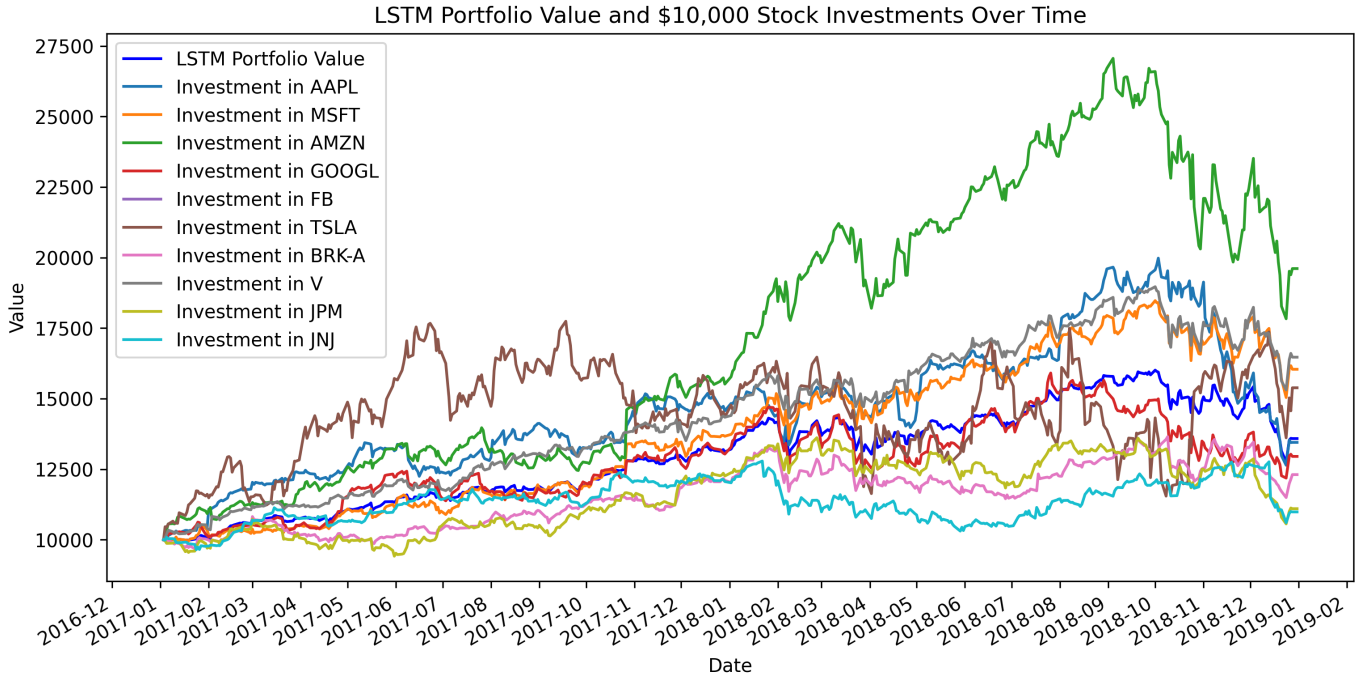

Fig. 4. The results of buying and holding 10k USD of the 10 stocks selected vs the LSTM trading stragey.

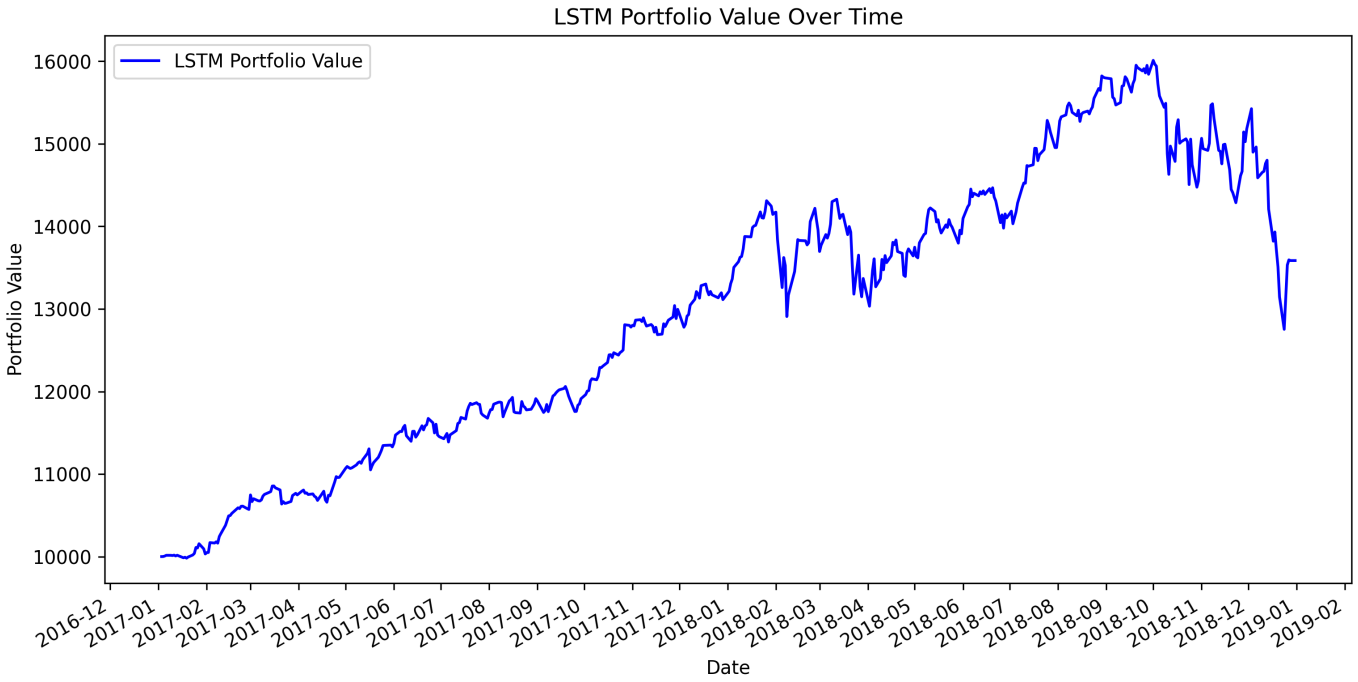

Fig. 5. The outcome of trading with the LSTM strategy, starting with 10k USD and getting to almost 16k USD at its peak.

sentiment_and_ner_headlines

date	headline	lead_paragraph	entity	sentiment
2023-11-20T03:15	Cruise's C.E.O. Quits a	Kyle Vogt, a founder and chief executive of Cruise, the driverless car s	Kyle	NEGATIVE
2023-11-20T23:55	Epic Games C.E.O. Sa	It was a long-awaited day in court for Epic Games' chief executive, Ti	Sweeney	NEGATIVE
2023-11-19T22:06	Talks to Bring Sam Alt	Talks at OpenAI to bring back Sam Altman, the artificial intelligence st	Alt	NEGATIVE
2023-11-19T05:01	The Invisible War in Uk	The drones began crashing on Ukraine's front lines, with little explana	Ukraine	NEGATIVE
2023-11-20T13:29	Microsoft's Stock Hits	Microsoft, the technology giant that ranks as one of the world's most	Microsoft	POSITIVE
2023-11-20T13:27	The Shake-up at Oper	Over just three days, the landscape for artificial intelligence has been	Microsoft	POSITIVE
2023-11-19T19:00	What Happened in the	The world of artificial intelligence looked very different on Monday, aft	Open	POSITIVE
2023-11-20T10:00	'Lost Time for No Reas	Around 2 a.m. on March 19, Adam Wood, a San Francisco firefighter c	Wood	NEGATIVE

Fig. 6. An excerpt from the NYTimes headline data after the entity name recognition and sentiment analysis was complete.

date	MSFT	AAPL	AMZN	NVDA	GOOGL	META	GOOG
2019-10-01	131.6378936767580	54.536956787109400	86.78250122070310	43.30253982543950	60.29999923706060	175.80999755859400	60.2550
2019-10-02	129.31382751464800	53.169830322265600	85.6614990234375	43.063621520996100	58.895999908447300	174.60000610351600	58.831
2019-10-03	130.87925720214800	53.62150192260740	86.22100067138670	45.12173843383790	59.471500396728500	179.3800048828130	59.391
2019-10-04	132.6463165283200	55.124603271484400	86.98249816894530	45.28599166870120	60.54800033569340	180.4499969482420	60.450
2019-10-07	131.6859588623050	55.13675308227540	86.63300323486330	45.87332534790040	60.412498474121100	179.67999267578100	60.383
2019-10-08	130.2933807373050	54.49081802368160	85.27549743652340	44.10637664794920	59.506500244140600	177.75	59.456
2019-10-09	132.76156616210900	55.12946701049810	86.09950256347660	44.9724235534668	60.119998931884800	179.85000610351600	60.1155

Fig. 7. An exceprt of the stock market data showing the dates, tickers and the pairings adjusted close values.

association_rules

consequents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
frozenset({'For much of the l	frozenset({'America'})	0.13333333333333300	0.13333333333333300	0.13333333333333300	1.0	7.5	0.11555555555555600	inf	1.0000000000000000
frozenset({'America'})	frozenset({'For much of the last ye	0.13333333333333300	0.13333333333333300	0.13333333333333300	1.0	7.5	0.11555555555555600	inf	1.0000000000000000
frozenset({'America'})	frozenset({'NEGATIVE'})	0.13333333333333300	0.4666666666666670	0.13333333333333300	1.0	2.142857142857140	0.07111111111111110	inf	0.6153846153846150
frozenset({'America'})	frozenset({'Stable'})	0.13333333333333300	1.0	0.13333333333333300	1.0	1.0	0.0	inf	0.0
frozenset({'Stocks Slide as E	frozenset({'America'})	0.13333333333333300	0.13333333333333300	0.13333333333333300	1.0	7.5	0.11555555555555600	inf	1.0000000000000000
frozenset({'America'})	frozenset({'Stocks Slide as Eviden	0.13333333333333300	0.13333333333333300	0.13333333333333300	1.0	7.5	0.11555555555555600	inf	1.0000000000000000
frozenset({'China'})	frozenset({'Decrease'})	0.06666666666666670	0.26666666666666700	0.06666666666666670	1.0	3.75	0.048888888888888900	inf	0.7857142857142860
frozenset({'China'})	frozenset({'Global Trade Is Deterio	0.06666666666666670	0.06666666666666670	0.06666666666666670	1.0	15.0	0.06222222222222220	inf	1.0
frozenset({'Global Trade Is D	frozenset({'China'})	0.06666666666666670	0.06666666666666670	0.06666666666666670	1.0	15.0	0.06222222222222220	inf	1.0

Fig. 8. An exceprt of the association rules generated by Temporal Apriori.