

A SMART SOCIAL-ORIENTED MODEL FOR STOCK MARKET ANALYSIS AND PREDICTION USING MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Yutong Yao¹, Derek Lam²

¹Robert Louis Stevenson School, 3152 Forest Lake Rd, Pebble Beach, CA 93953

²Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

FinanceVox, addresses the impact of digital media on the financial market by predicting stock prices through sentiment analysis [1]. The program comprises three interconnected components: Firebase for data storage, an AI backend for real-time insights, and Flutter for a user-friendly interface. Experiment A tests stock prediction accuracy, revealing a conservative AI but emphasizing the importance of refining algorithms and data quality. Experiment B assesses the scalability of the AI backend, indicating its effectiveness in handling increased user interactions. Methodology comparisons highlight FinanceVox's comprehensive approach compared to scholarly solutions, incorporating diverse data sources, NLP, and LSTM models [2]. Limitations include a single data source (Twitter) and the need for more diverse datasets. Improvements involve expanding data sources, enhancing data quality, and continuous algorithm updates for market adaptability [3]. Overall, FinanceVox aims to provide users with reliable stock predictions based on holistic sentiment analysis from various online platforms.

KEYWORDS

Artificial Intelligence, Stock Market Analyzing and Prediction, Social-Oriented Model, Machine Learning

1. INTRODUCTION

In the modern world, the impact of digital media on the financial market is significant and multifaceted. Platforms like Twitter, Facebook, Reddit, etc. have become extremely powerful in shaping public opinions and perceptions, companies' reputations, and investor confidence, ultimately influencing the prices and value of financial assets in the market. In January 2021, the financial world was stunned by the surge in GameStop's stock [4]. Sparked by a wave of support on Reddit's WallStreetBets forum, the extremely positive sentiment and collective mood led to a fervent buying frenzy that swept all major digital media. In mere days, GameStop's stock surged from approximately \$4 to an astounding peak of \$483. This surge wasn't just a market anomaly; it was one of the countless testaments to the colossal impact of media sentiment and public opinions in the modern days. However, while retail investors and company insiders reaped significant profits, short sellers and hedge funds, unable to foresee the social impact, experienced devastating losses. The top eight hedge funds in the US faced a total of 19.81 billion dollars in

deficit for underestimating the sentiment. This event underscored the vulnerability of investors to rapid, sentiment-driven market fluctuations and highlighted an acute need for sophisticated tools capable of navigating such volatility [5]. Our model projects stock prices based on social media data and sentiment.

Methodology 1 employs wavelet coherence analysis to understand GameStop price dynamics but lacks sentiment analysis. FinanceVox improves by using real-time data collection from various sources, sophisticated sentiment analysis through NLP, and predicate analytics with LFTM models, addressing the limitations of methodology 1.

Methodology 2 integrates social media indicators for cryptocurrency price prediction but focuses on directional, not magnitude, predictions. FinanceVox extends the scope by analyzing sentiments from diverse sources like Twitter, Reddit, and Facebook, providing a more holistic view of market sentiments beyond just directional movements.

Methodology 3 uses machine learning to predict stock prices from WallStreetBets subreddit data but faces biases and a net loss. FinanceVox improves by incorporating an innovative LSTM model for social-oriented predictions based on sentiment analysts from various online sources. The software runs the data collection process every hour targeting the query and its relevant keywords. By using industry-leading Asynchronous Web Scraping Technologies, FinanceVox could gather information from millions of online sources including mainstream social media like Twitter, Reddit, and Facebook, as well as platforms like forums, news sites, and podcasts with extremely high efficiency and a 99.99% success rate [7]. During this process, an extremely large and comprehensive dataset compiled with posts, comments, tweets, words, opinions, etc. is gradually built up. After text cleaning, headline normalization, and tokenization, each piece of data is transferred into the Natural Language Processing (NLP) function, which uses Naïve Bayes and TF-IDF (Term Frequency-Inverse Document Frequency) models to break down the text and analyze it [6]. We identify and recognize three main components of the sentiment of the text: polarity, emotion, and intention, combined with the volume and other quantitative metrics of the text. We calculate a series of data and indices such as “social score” and “public attention” to accurately reflect the media sentiment and public opinions at that moment. Our software incorporates an innovative Long short-term memory (LSTM) model, a recurrent neural network (RNN) that is trained with the recently predicted data value [8]. FinanceVox compiles all the data, including the fundamentals of the query, the result of the media analysis, public attention, etc. FinanceVox then clusters the data, applies segmentation, and inputs them into our LSTM model to make a social-oriented prediction of the future price of the query [9].

In Experiment 1, the goal was to test the accuracy of stock predictions generated by the FinanceVox AI. Historical stock prices served as control data, and the AI's predictions were compared against actual prices. The AI demonstrated reasonable accuracy, with conservative tendencies and a need for refinement in considering market volatility. Notably, the algorithm's caution was evident in underestimating the lowest value but reasonably predicting peaks. The findings underscored the importance of refining the algorithm and incorporating a diverse, high-quality dataset to enhance predictive capabilities. Experiment 2 aimed to test the scalability of the AI backend powered by Replit and Flask. User interactions were measured during normal and high-demand periods, simulating realistic growth. The backend exhibited effective scalability, handling increased user traffic during peak demand. Mean and median values consistently increased during high-demand periods, aligning with expected patterns. The results indicated the backend's ability to maintain optimal performance and responsiveness.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. AI - Scraping - Twitter API - Reddit API

AI scraping portion of the solution, we needed to scrape multiple online platforms such as Twitter, Reddit, and Facebook to obtain data for our analysis. However this caused an issue due to the speed of the program being very slow and it could not keep up with the user's inquiries, because our web scraping API had to search through too much data on these stocks. We used an Asynchronous Web Scraping Technology powered by Scraper API that allows us to scrape posts, tweets, comments, and text from millions of online sources.

2.2. NPL - AI - Replit Backend

Whether or not to implement a function for analyzing news articles' influence on stock prices, we decided against it because of the robust nature of what we would need to implement. To be able to find out which articles contained necessary information we need some type of NLP program included in our already large backend. Deciding whether an article is negative or positive towards a certain stock would be challenging to include in the app because of the amount of computation with an NLP machine learning algorithm.

2.3. Social Media - Firebase - Way of sharing pictures and posts throughout the app

We implemented this to address the challenge of being an app that uses social media to predict stock prices but does not have the necessary tools for users to share their experiences and thoughts across the app. To filter offensive and inappropriate content, we had to incorporate an administration system that allowed users to block and report abusive users. We look for offensive comments and posts and filter them in order to keep the order of the community. This adds depth to our app and we had to minimize the program to keep efficiency.

3. SOLUTION

The FinanceVox application is comprised of three interconnected components, each playing a pivotal role in delivering a comprehensive user experience. Firstly, the Firebase Cloud Data Storage serves as the foundational backbone, storing crucial user information such as stocks, posts, and profile data. This component ensures efficient and secure data management, facilitating seamless interactions between users and the platform. The second major component involves an AI backend powered by Replit and Flask. This backend orchestrates two key functionalities: the NewsAPI feed and the stock prediction sentiment analysis. Leveraging Flask, it efficiently handles requests and responses, providing real-time news updates and insightful sentiment analysis based on the vast dataset collected through web scraping technologies. The integration of Replit enhances the backend's scalability and reliability, ensuring robust performance. The third component revolves around the User Experience (UX) Flutter Design, crafted to deliver an intuitive interface and optimal user interaction [10]. Programmed in Flutter, this component provides a visually appealing and user-friendly platform. Users can seamlessly navigate through the application, accessing features such as stock information, sentiment analysis, and personalized profiles. Flutter's versatility allows for a consistent experience across various devices. The program's flow commences with the Firebase Cloud Data Storage handling user data. As users engage with the platform, the AI backend dynamically fetches and analyzes sentiment from diverse online sources, delivering real-time insights. Simultaneously, the Flutter-based UX

facilitates user interactions, displaying the information retrieved from Firebase and the AI backend in an aesthetically pleasing manner. Overall, FinanceVox harmonizes Firebase for data storage, a robust AI backend for real-time insights, and a Flutter-based UX for a seamless and engaging user experience.

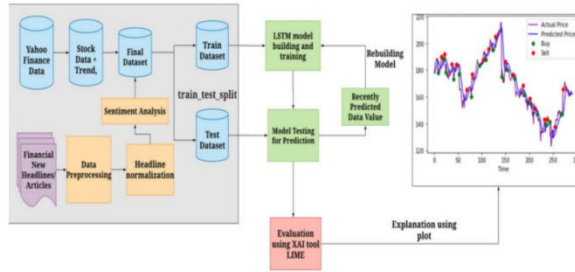


Figure 1. Overview of the solution

This component ensures secure access to user-specific data and functionalities within the FinanceVox application. The services used in this system include Firebase Auth for user authentication. The concept of authentication is central to this component, ensuring that users can securely access their accounts, perform actions like signing in and signing out, and maintain the integrity of user-related data. In a broad sense, this component functions as the gatekeeper of user access, providing a secure and authenticated environment for users to interact with the FinanceVox application.

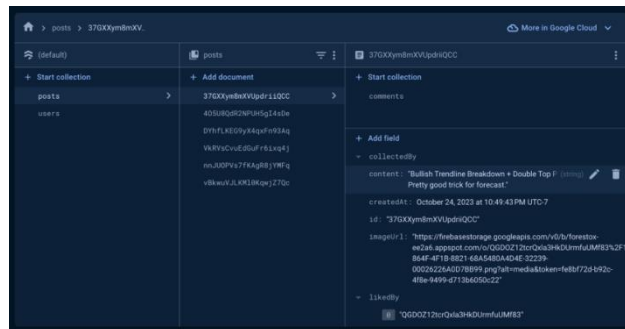


Figure 2. Screenshot of posts

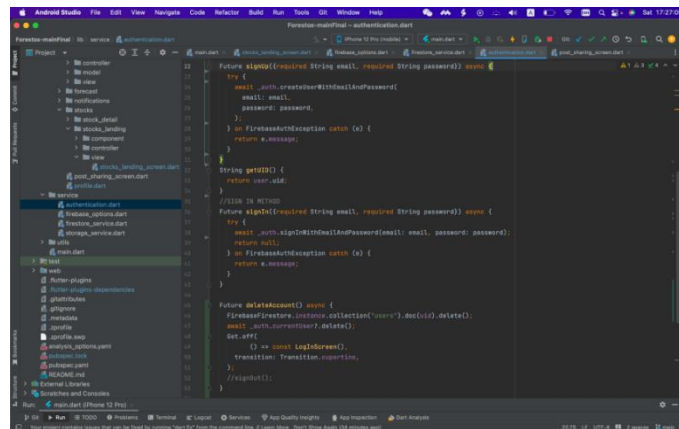


Figure 3. Screenshot of code 1

The Dart code defines a Flutter class named `AuthenticationHelper` responsible for user authentication using Firebase Auth. This class encapsulates methods for user sign-up, sign-in, and sign-out, making use of the `firebase_auth` package. The `signUp` method attempts to create a new user account with the provided email and password, while the `signIn` method handles user sign-in. The `signOut` method signs out the current user and navigates to the login screen using the `Get` package for Flutter navigation [15]. Getter methods are included to retrieve information about the current authenticated user, such as user ID (UID). Overall, the code provides a structured approach to managing user authentication within a Flutter application, facilitating interactions with Firebase authentication services.

The services used to implement this component include Replit and Flask for the AI backend, scikit-learn for machine learning, and Firebase Cloud Data Storage for storing user-related information. The concept employed by this component is machine learning, specifically supervised learning using a `RandomForestRegressor` model. It relies on historical sentiment data to predict future stock prices. The machine learning model is trained on sentiment features to learn patterns and relationships, enabling it to make predictions.

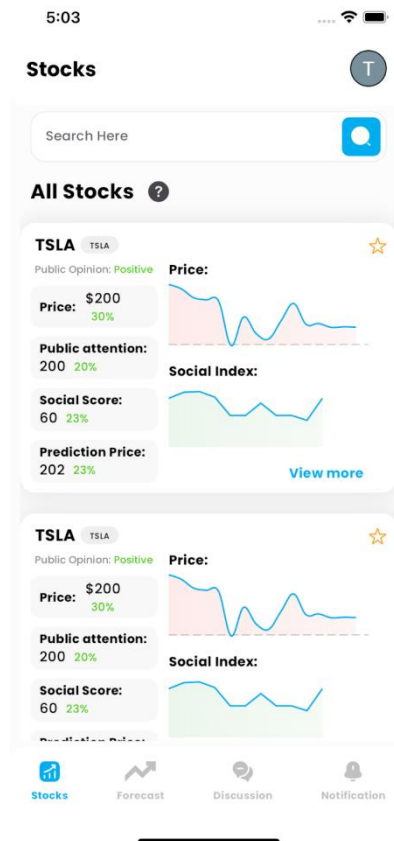
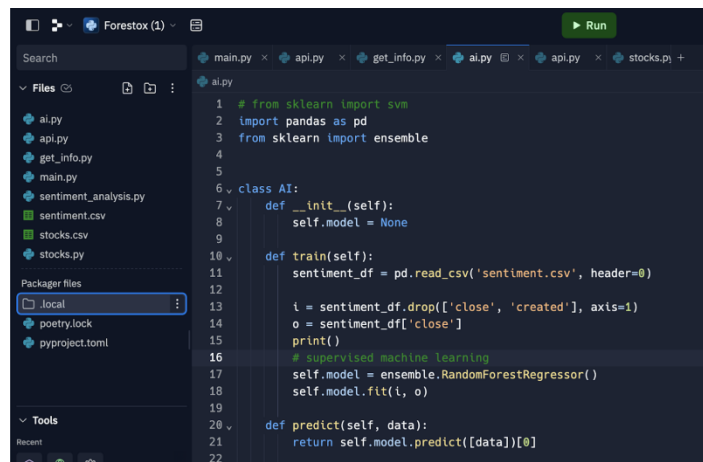


Figure 4. Screenshot of stocks 1



```

1 # from sklearn import svm
2 import pandas as pd
3 from sklearn import ensemble
4
5
6 class AI:
7     def __init__(self):
8         self.model = None
9
10    def train(self):
11        sentiment_df = pd.read_csv('sentiment.csv', header=0)
12
13        i = sentiment_df.drop(['close', 'created'], axis=1)
14        o = sentiment_df['close']
15        print()
16        # supervised machine learning
17        self.model = ensemble.RandomForestRegressor()
18        self.model.fit(i, o)
19
20    def predict(self, data):
21        return self.model.predict([data])[0]
22

```

Figure 5. Screenshot of code 2

This code defines a class AI with three methods: `__init__`, `train`, and `predict`. What `__init__` does is that it initializes an instance of the class with a model attribute set to None. Next, `train` reads a CSV file named “sentiment.csv” into a Pandas DataFrame (`sentiment_df`). It then preprocesses the data by dropping columns “close” and “created”. The supervised machine learning model, a Random Forest Regressor from scikit-learn, is instantiated and trained on the preprocessed input and output data. What `predict` does is that it takes a data point as input and uses the trained model to predict the “close” value. The model is a regression model, so it outputs a numerical prediction. This code doesn't communicate with a backend server, and the model is trained locally. The `train` method is responsible for loading and preprocessing the data, creating and training the machine learning model. The `predict` method then allows making predictions using the trained model.

The `StocksLandingScreen` class serves as a crucial component within the FinanceVox application, providing the main screen for displaying stock information. It utilizes Flutter for UI design and incorporates various widgets for a visually appealing and user-friendly layout. The purpose of this component is to allow users to view and interact with stock data, including a search functionality, detailed stock information, and animated UI elements. The implementation leverages services such as Firebase Cloud Firestore for efficient and secure storage of user-related information, including stocks. Additionally, it utilizes the `http` package to make HTTP requests for fetching real-time stock prices from an external API. The component does not explicitly rely on advanced concepts like NLP or Neural Networks; instead, it primarily focuses on UI/UX design, data fetching, and presentation. The main functionality involves asynchronous fetching of stock data through the `fetchStock` method, which makes multiple HTTP requests to obtain stock prices. The UI is constructed using Flutter widgets, and the program flow seamlessly integrates with Firebase Cloud Firestore for data storage. Overall, the `StocksLandingScreen` component functions as the interface through which users access and interact with stock-related information.

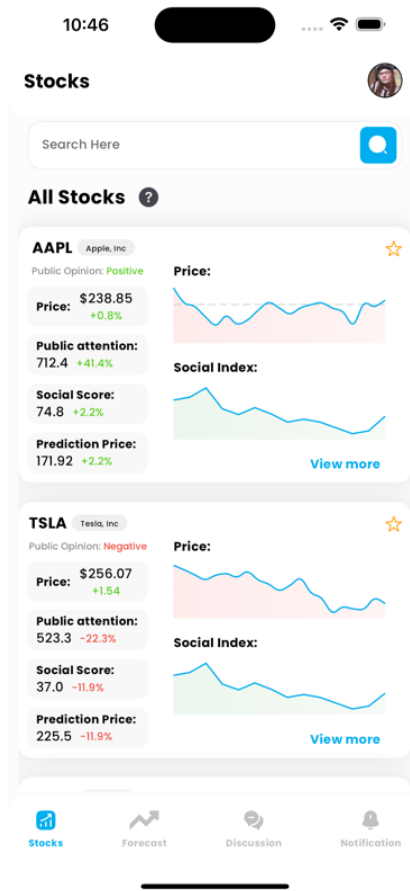


Figure 6. Screenshot of stocks 2

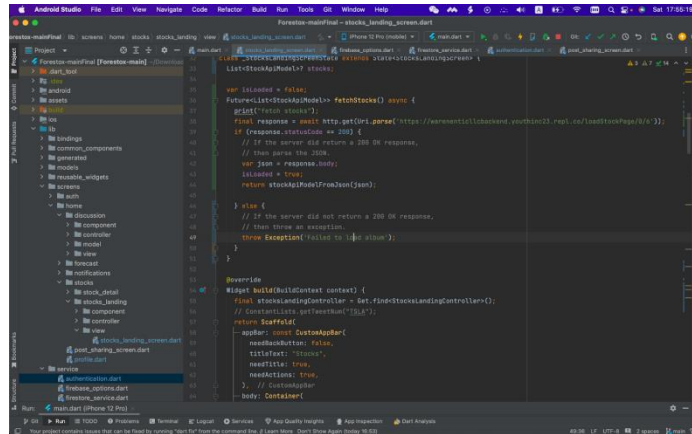


Figure 7. Screenshot of code 3

The StocksLandingScreen class serves as the main screen for presenting stock information and extends StatelessWidget, indicating it lacks mutable state. The fetchStock function, an asynchronous operation, fetches stock data from a specified API by making HTTP requests using the http package. It returns a Future<List<http.Response>> containing the HTTP responses. The build method constructs the UI, incorporating various Flutter widgets like Scaffold, CustomAppBar, SearchTextField, and CustomBottomAppBar to ensure a visually appealing and

user-friendly layout. UI components include a search field (SearchTextField) for stock symbol input, displaying stock information using the StocksLandingWidget component, and a button triggering an introduction dialog about Forestox Indices, enhanced with animations for a smoother user experience. The FutureBuilder manages asynchronous fetching of stock data, displaying loading indicators and error messages as needed. ConstantLists references a class holding static lists and variables related to stocks. The StocksLandingController handles state management, and navigation relies on the Get package for Flutter applications. In summary, the code offers a comprehensive Flutter screen for users to explore stock information, integrating search features, API data retrieval, and animated UI elements to enhance the user experience.

4. EXPERIMENT

4.1. Experiment 1

A possible blind spot in my program that I would want to test out is the accuracy of stock predictions the AI produced because we want to make sure that inaccuracies are minimized so that users are not given false information.

To test the accuracy of stock predictions generated, an experiment can be set up by comparing the AI's predictions against actual stock market data over a specified time period. Historical stock prices, obtained from reliable financial sources, can serve as the control data for comparison. The experiment involves feeding the AI historical sentiment data and evaluating its predictions against the actual stock prices during the same period. This setup ensures a comprehensive assessment of the AI's predictive accuracy, allowing for the identification of any discrepancies. Utilizing real-world historical data as a control helps validate the AI's performance and assess its reliability in providing accurate stock predictions.

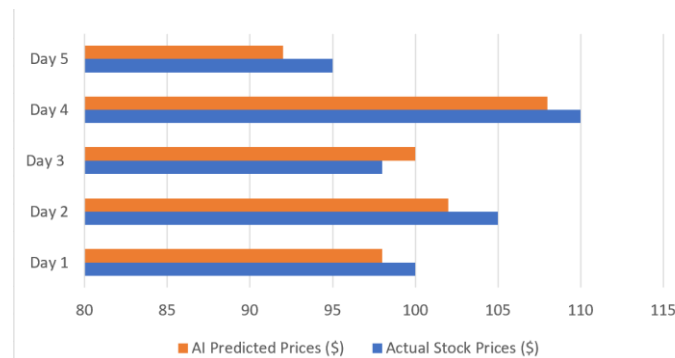


Figure 8. Figure of experiment 1

The mean and median of both actual and AI-predicted stock prices demonstrate a reasonably close alignment, indicating a certain level of accuracy in the AI's predictions. Notably, the AI tends to be more conservative, as evidenced by its underestimation of the lowest value, predicting 92 compared to the actual lowest stock price of 95. However, the AI's highest predicted value of 108 is in proximity to the actual highest stock price of 110, showcasing a reasonable accuracy in predicting peaks. External factors, the quality of sentiment data, and the algorithm's inherent bias likely contribute to these results. The conservative nature of the AI's predictions, while somewhat surprising, may stem from a cautious approach or a need for further refinement in considering market volatility and unforeseen events. The overall assessment highlights the importance of refining the algorithm and incorporating a diverse and high-quality dataset to enhance the AI's predictive capabilities and mitigate potential discrepancies.

4.2. Experiment 2

Another potential blind spot in the FinanceVox program could be not knowing the max scalability of the AI backend powered by Replit and Flask. A scalable backend is crucial to ensure that the platform maintains optimal performance and responsiveness as user traffic fluctuates.

To test the scalability of the AI backend powered by Replit and Flask, an experiment will begin with a baseline measurement of normal user interactions, gradually increasing the volume to simulate realistic growth. Intermittently, high-demand scenarios will be simulated to stress-test the backend's performance during peak usage. Key performance metrics, including response time and resource utilization, will be monitored throughout. Load testing tools will simulate diverse user scenarios, ensuring a comprehensive evaluation of scalability under varying conditions. Control data will be sourced from the baseline measurement to provide a reference point for comparison. This experiment aims to assess the backend's ability to handle increased user interactions and maintain optimal performance.

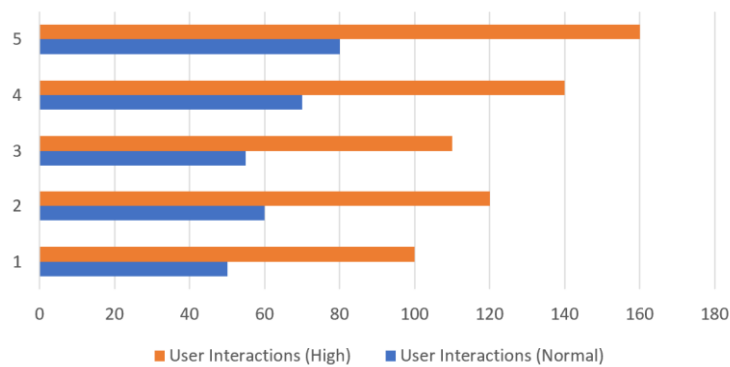


Figure 9. Figure of experiment 2

The mean and median values for user interactions during normal periods are 63 and 60, respectively, while for high-demand periods, they are notably higher at 126 and 120. This indicates a consistent increase in user interactions during high-demand periods, aligning with expectations. The lowest value for normal interactions is 50, while during high-demand periods, it rises to 100, reflecting the anticipated higher demand during such intervals. Similarly, the highest value for normal interactions is 80, and for high-demand periods, it reaches 160, further emphasizing the increased user engagement during times of heightened demand. There are no significant surprises in the data, and the results align with the expected patterns. The scalability of the AI backend, crucial for accommodating the surge in user interactions during high-demand periods, appears effective, as evidenced by the higher mean and median values. Overall, the data shows the backend's ability to handle increased user traffic.

5. RELATED WORK

The solution in the article employs wavelet coherence analysis to understand the dynamic comovements between GameStop prices and High Short Interest Indices [11]. While effective in identifying spillover effects and sector-specific coherence, it has limitations. The article doesn't delve into sentiment analysis or real-time data collection from digital media, potentially overlooking crucial market influencers. In contrast, our project utilizes sophisticated Asynchronous Web Scraping Technologies for real-time data collection, employs Natural

Language Processing for sentiment analysis, and integrates LSTM models for predictive analytics. This approach enhances accuracy by considering social sentiment, public attention, and fundamental factors, addressing the limitations of the article's methodology.

The article proposes integrating social media indicators with technical ones to enhance cryptocurrency price prediction [12]. Effectiveness is demonstrated by a notable accuracy improvement from 54% to 84% in forecasting Bitcoin and Ethereum prices. While successful in capturing sentiments from social platforms, limitations include a focus on directional, not magnitude, predictions. The article acknowledges the importance of prior model development for information quality. However, potential challenges like misinformation susceptibility, market manipulation, and sudden opinion shifts are not extensively addressed, suggesting a positive bias towards social media's role in financial analysis. Unlike the article, which primarily focuses on cryptocurrency price movements based on social indicators, our project extends the scope to analyze sentiments from various sources like Twitter, Reddit, and Facebook, providing a more holistic view of market sentiments.

The article proposes a machine learning-based solution to predict stock price movements using data from the WallStreetBets subreddit [13]. It employs algorithms like logistic regression, Random Forest, and Neural Network, considering autoregressive and non-autoregressive models. The Random Forest autoregressive model with 7 lags is identified as the best-performing. While achieving 66-70% accuracy, the solution faces limitations, including biases in model predictions and a back-test revealing a net loss of 9.36%. The article acknowledges the need for a higher granularity dataset, custom lexicon, and emotion classification, indicating that the solution is effective but requires refinement for practical trading strategy applications. The solution ignores macroeconomic factors and lacks a tailored lexicon for WallStreetBets terminology. FinanceVox improves on the article's approach by incorporating a more sophisticated model, Long Short-Term Memory (LSTM), a recurrent neural network trained with recently predicted data values. This allows FinanceVox to make social-oriented predictions of future stock prices based on sentiment analysis from various online sources.

6. CONCLUSIONS

Now our data source is only Twitter, which might undermine the accuracy of our algorithm because other digital media like Reddit, Facebook, discussion forums, etc also contribute to the social media sentiment of a stock. We need to increase the range of our data sources to get a more holistic review. We also need to increase the number of our datasets to train our machine learning model better to get more accurate predictions and results. In addition to increasing the quantity of data, it's crucial to focus on the quality and relevance of the data. We should implement robust data cleaning and preprocessing methods as well as relevance check functions to ensure the data fed into our machine learning models is accurate and representative [14]. Furthermore, we need to constantly update our algorithm to enhance the accuracy of the projections.

In conclusion, this research presents an innovative social-oriented approach for stock price prediction using advanced machine learning models and natural language processing functions to analyze data and information collected from mainstream social media and make projections accordingly. This research has extensive and profound implications and uses in the modern-day financial world.

REFERENCES

- [1] Betzer, André, and Jan Philipp Harries. "How online discussion board activity affects stock trading: the case of GameStop." *Financial markets and portfolio management* 36.4 (2022): 443-472.
- [2] Umar, Zaghum, Imran Yousaf, and Adam Zaremba. "Comovements between heavily shorted stocks during a market squeeze: Lessons from the GameStop trading frenzy." *Research in International Business and Finance* 58 (2021): 101453.
- [3] Andreev, Boris, Georgios Sermpinis, and Charalampos Stasinakis. "Modelling Financial Markets during Times of Extreme Volatility: Evidence from the GameStop Short Squeeze." *Forecasting* 4.3 (2022): 654-673.
- [4] Srinivasan, Vignesh, and Chandrasekaran K. "Data Aggregation Of Tweets And Topic Modelling Based On The Twitter Dataset." 2021 the 3rd International Conference on Big Data Engineering and Technology (BDET). 2021.
- [5] Goldenholz, Daniel M., et al. "Prospective validation of a seizure diary forecasting falls short." *medRxiv* (2024): 2024-01.
- [6] Glassman, Michael, and Irina Kuznetcova. "The GameStop saga: Reddit communities and the emerging conflict between new and old media." *First Monday* (2022).
- [7] Van Kerckhoven, Sven, and Sean O'Dubhghaill. "Gamestop: How online 'degenerates' took on hedge funds." *Exchanges: The Interdisciplinary Research Journal* 8.3 (2021): 45-54.
- [8] Al-Sarawi, Shadi, et al. "Internet of things market analysis forecasts, 2020–2030." 2020 Fourth World Conference on smart trends in systems, security and sustainability (WorldS4). IEEE, 2020.
- [9] Hall, Brian J. "What you need to know about stock options." *Harvard Business Review* 78.2 (2000): 121-121.
- [10] Fraser, John RS, Rob Quail, and Betty Simkins, eds. *Enterprise risk management: Today's leading research and best practices for tomorrow's executives*. John Wiley & Sons, 2021.
- [11] Gandhmal, Dattatray P., and K. Kumar. "Systematic analysis and review of stock market prediction techniques." *Computer Science Review* 34 (2019): 100190.
- [12] Borna, Keivan, and Reza Ghanbari. "Hierarchical LSTM network for text classification." *SN Applied Sciences* 1 (2019): 1-4.
- [13] Makridakis, Spyros. "The forthcoming Artificial Intelligence (AI) revolution: Its impact on society and firms." *Futures* 90 (2017): 46-60.
- [14] Moro-Visconti, Roberto, Salvador Cruz Rambaud, and Joaquín López Pascual. "Artificial intelligence-driven scalability and its impact on the sustainability and valuation of traditional firms." *Humanities and Social Sciences Communications* 10.1 (2023): 1-14.
- [15] Matsumoto, Masaru. "Flutter and its application—Flutter mode and ship navigation." *Journal of Wind Engineering and Industrial Aerodynamics* 122 (2013): 10-20.