NLP-Driven Sentiment Analysis for Stock Price Forecasting

Suresh Kumar S
Professor Artificial Intelligence and Data Science
Rajalakshmi Engineering College Chennai, India
sureshkumar.s@rajalakshmi.edu.in

Atmakuru Siva Sandeep
Artificial Intelligence and Data Science
Rajalakshmi Engineering College Chennai, India
221801501@rajalakshmi.edu.in

Giridharan M
Artificial Intelligence and Data Science
Rajalakshmi Engineering College Chennai, India
221801504@rajalakshmi.edu.in

Praveen B
Artificial Intelligence and Data Science
Rajalakshmi Engineering College Chennai, India
221801503@rajalakshmi.edu.in

***Abstract*— This study offers a thorough method for merging Natural Language Processing (NLP) with financial data analytics to investigate the relationship between news sentiment and stock price performance, with the goal of improving algorithmic trading techniques. The suggested system is a Python-based web application that uses Flask to offer an interactive environment for users to enter a stock ticker and get real-time insights driven by news sentiment and historical price data. The system retrieves current news stories and historical stock prices for the chosen stock using the Mboum Finance API. Sentiment scores are calculated using NLP methods, particularly TextBlob and NLTK, to categorize news headlines as positive, negative, or neutral. To enable comparative visual analysis, Plotly is used to visualize these sentiment scores as a bar chart, while corresponding stock prices are plotted as a line graph. In addition, the news headlines are displayed as clickable links, giving users the option to read the complete pieces for additional context. The analysis is presented in a dynamically generated analysis.html template, providing a convenient and user-friendly interface. This method tackles a major issue in financial trading: the struggle to rapidly analyze massive amounts of unstructured news data and correlate it with market activity in real-time. The system facilitates quicker and more educated decision-making by automating sentiment analysis and merging it with organized market data. In addition, the application examines the possibility of integration with Order Management Systems (OMS), where real-time sentiment alerts could help traders identify important stock shifts driven by media coverage. By increasing operational speed and tactical accuracy, this real-time feedback loop might also act as a catalyst for automated trading activities. The framework shows that unstructured text data may be a valuable resource for financial forecasting and risk management when it is properly processed. It also offers a scalable base for future improvements that use more advanced machine learning models and wider data sources.**

Keywords*— *Natural Language Processing (NLP), Sentiment Analysis, Stock Price Prediction, Algorithmic Trading, Financial Data Analytics, Real-time Insights, Automated Trading.

I. Introduction

In the digital era, the financial market is progressively impacted by a large and constant flow of unstructured text data, especially from online financial sites and news media. Although traders, investors, and financial analysts depend on timely information to make informed choices, manually sifting through massive volumes of news for sentiment and relevance is a major obstacle. As a result, the use of artificial intelligence methods, particularly Natural Language Processing (NLP), in financial analysis has become more popular lately. This project suggests a Python-based web application that operates in real time and integrates NLP with financial data analytics to investigate and showcase the relationship between news sentiment and stock price movement. The primary goal is to facilitate quicker, more intelligent, and more automated decision-making in algorithmic trading contexts. To gather the most recent stock-related news stories and historical stock prices for a company specified by the user, the application makes use of the Mboum Finance API. It then uses NLP libraries like TextBlob and NLTK to evaluate news headlines and categorize them as positive, negative, or neutral. Historical stock price trends are displayed as line graphs to enable users to spot possible patterns or connections, and these sentiment ratings are visually represented using Plotly bar graphs. The system also improves user experience by providing direct access to the original news sources through clickable headlines. Using Flask as the web framework, the complete analysis is displayed on a responsive analysis.html interface. By automating sentiment scoring and visual analysis, this tool tackles a vital issue in financial decision-making: enhancing the speed and dependability of market responses while minimizing information overload. Additionally, this study examines the possibility of integration into Order Management Systems (OMS), allowing traders to get real-time, sentiment-driven notifications and even initiate automatic buy or sell actions depending on media sentiment.

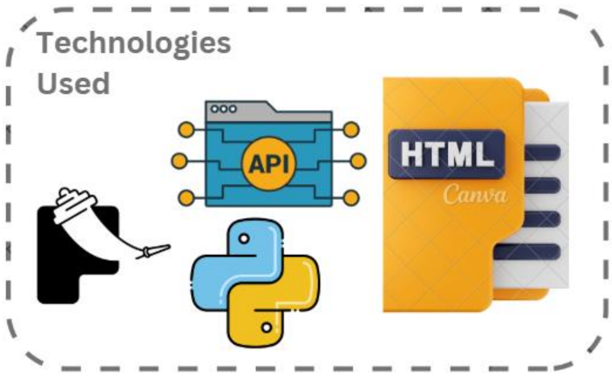

Fig. 1. Tech Stacks

The suggested system not only demonstrates a real-world use of NLP in fintech, but also paves the way for future improvements like predictive analytics for financial forecasting, multi-source data integration, and deep learning-based sentiment models. By transforming unstructured data into structured insights, this project ultimately aids the expanding field of AI-driven financial intelligence, enabling a more informed and proactive method of stock trading.

II. RELATED WORKS

Using news headlines and Natural Language Processing (NLP) methods, Saxena et al. [1] put forth a thorough approach for forecasting stock price changes. They used a combination of machine learning algorithms, including BERT, Random Forest, and Support Vector Machines (SVM), to categorize news sentiments as positive, negative, or neutral. The study examined the predictive capability of these models by studying the impact of various sentiment categories on market trends. The research showed that traditional models were far surpassed by advanced NLP models, particularly BERT, which achieved an outstanding accuracy rate of 86.25%. The writers stressed the necessity of including sentiment analysis as a key element in predicting stock price changes. They were able to capture more nuanced and contextual sentiment using BERT, which enhanced prediction accuracy. The findings confirmed the promise of NLP methods in transforming stock market analysis and offered suggestions for improving stock trading tactics using sentiment-driven models.

Jiang and Zeng [2] concentrated on merging FinBERT, a financial-specific version of BERT, in order to forecast changes in the stock market. To model the temporal dynamics of stock price changes in reaction to news sentiment, their methodology integrated FinBERT with a Long Short-Term Memory (LSTM) network. To assess the efficacy of sentiment analysis in financial forecasting, the team compared FinBERT's performance against that of traditional models like BERT and ARIMA (AutoRegressive Integrated Moving Average) in their study. The research revealed that FinBERT considerably surpassed the other models, confirming its superior ability to grasp the context of financial language. With the addition of the LSTM model, time-series predictions became more accurate, making it ideal for real-time trading methods. The writers determined that sentiment analysis is an essential part of stock market forecasting, especially in unpredictable financial markets where news events are crucial. This study demonstrated the advanced NLP models' capacity to seamlessly combine textual information with stock price prediction activities.

Using investor sentiment derived from financial news headlines, Shukla [3] created a novel model for forecasting stock market trends. The study employed VADER (Valence Aware Dictionary and sEntiment Reasoner), a sentiment analysis tool, alongside a Gated Recurrent Unit (GRU) neural network. While the GRU model was used for time-series forecasting of stock prices, the VADER tool was used to evaluate the sentiment of news articles and classify them as positive, negative, or neutral. The combination of sentiment analysis and GRU resulted in a significant 6% increase in prediction accuracy compared to traditional techniques like linear regression. According to the research, driven by media coverage, understanding investor mood is essential for improving prediction accuracy in stock market forecasts. This study showed that including sentiment from news headlines may be beneficial for creating strong prediction systems. It also sparked conversations about enhancing sentiment analysis tools to increase the dependability of financial models.

The use of large language models (LLMs) like OPT and BERT for sentiment analysis of U.S. financial news articles was investigated by Kirtac and Germano [4]. Their study included evaluating a vast dataset of more than 965,000 news articles to determine how effectively LLMs could understand and convey sentiment in financial materials. They discovered that LLMs, particularly OPT and BERT, surpassed traditional approaches in sentiment classification tasks by achieving accuracies of 74.4% and 72.5%, respectively. The research highlighted the efficacy of LLMs in financial sentiment analysis, which is vital for forecasting markets in real-time. The authors emphasized that these models, which represent a major improvement in financial sentiment analysis through the use of sophisticated NLP techniques, might be incorporated into current financial systems to produce real-time insights from news sources, giving a trading strategy advantage. This study supported the notion that large language models (LLMs) are especially adept at analyzing complicated financial news data due to their comprehension of semantics and context.

Ibrar [5] carried out a project on predicting stock price changes through sentiment analysis of the financial news headlines. The project used Python and a range of statistical methods to gather, handle, and evaluate news articles in real-time. News sentiment was correlated with stock price changes using sentiment analysis tools to extract sentiment scores from the headlines. Additionally, the research looked into how machine learning methods might improve prediction accuracy using historical market data. The study showed that sentiment analysis may offer helpful insights into market behavior when used with stock price data, which can help with decision-making. The project also demonstrated how NLP methods might automate the analysis of huge amounts of unstructured news data, which can be daunting for analysts and traders. This real-world use of sentiment analysis highlighted its increasing significance in financial decision-making systems and algorithmic trading.

Sen and Mehtab (2020) [6] proposed a hybrid method for forecasting stock price movements that combines machine learning, deep learning, and natural language processing. They used daily price movements of the NIFTY 50 index from the National Stock Exchange (NSE) of India over a three-year period (2015–2017). Numerous predictive models were created using machine learning methods, and a deep learning network based on Long Short-Term Memory (LSTM) was constructed to forecast stock closing prices. Furthermore, they improved the predictive precision of stock price movements by integrating sentiment analysis of Twitter data to link public sentiment with market sentiment. The research showed that merging sentiment analysis with historical stock data improves forecasting results.

The FinBERT-LSTM model, which combines the FinBERT pre-trained NLP model with a Long Short-Term Memory (LSTM) network for stock price forecasting, was introduced by Halder in 2022 [7]. The model was trained on stock data from the NASDAQ-100 index and news stories from the New York Times. The performance of the FinBERT-LSTM model was compared to that of simpler models like Multilayer Perceptron (MLP) and LSTM using metrics like accuracy, Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The findings demonstrated that the FinBERT-LSTM model surpassed the other models, emphasizing the significance of combining deep learning methods with financial news sentiment for precise stock price prediction.

The Multipronged Attention Network for Stock Forecasting (MAN-SF), which integrates information from Twitter data, past stock prices, and inter-stock connections to forecast stock market trends, was introduced by Sawhney et al. (2022) [8]. By using a graph neural network to recognize relationships between various stocks, the model is able to gain knowledge from the correlations and interdependencies that affect stock behavior. Through a comparative study of their model against others that use the same dataset, the authors discovered that MAN-SF outperforms the most potent baselines, proving its superiority in stock prediction. However, due to the dataset's constraints, the accuracy rate was deemed insufficient.

Heiden et al. (2021) [9] studied the effect of news sentiment on stock prices by adding news sentiment to an LSTM prediction model together with historical data. The experimental results showed that the proposed model outperformed single-network models that rely exclusively on either candlestick charts or sentiment data. The model's prediction accuracy for Apple stock was an impressive 75.38%, highlighting how well news sentiment can be integrated into stock price forecasting algorithms.

The financial market sentiment data gathered from news articles to forecast changes in the Standard & Poor's 500 stock market returns was demonstrated by Fazlija et al. (2022) [10]. The authors used the Bidirectional Encoder Representations from Transformers (BERT) model for sentiment analysis and developed a random forest classifier approach to predict future price changes of the stock market index. The results of the study highlighted the importance of using sentiment scores obtained from news articles to predict stock price movements. A major drawback of the research was the exclusion of news data from every firm listed in the stock market index.

Liu (2018) [11] presented a new method for predicting stock trends that combines deep learning models with financial news articles. Liu employed a Bidirectional Long Short-Term Memory (BiLSTM) network in this research, which is an effective deep learning framework for handling sequential data like news articles. To enhance its understanding of the context and sentiment in financial news, the BiLSTM model was combined with an attention mechanism that concentrated on the most important areas of the news text. The model, which sought to forecast stock price movements depending on the sentiment gleaned from the stories, was trained on news data related to both individual equities and the S&P 500 index. Liu's study showed that, in contrast to conventional time-series models, stock market trend predictions may be considerably more accurate when sentiment analysis of news articles is included. The research also highlighted the need to comprehend the context of financial news, as the nuanced emotions expressed in news stories can impact market behavior. Capturing these emotions effectively using deep learning techniques like BiLSTM gave an edge in predicting stock directions and enhancing trading tactics. This study emphasized that a more thorough method to forecasting market behavior may be obtained by combining stock data with cutting-edge NLP methods.

Mehtab and Sen (2019) [12] created a hybrid predictive model for stock price movements by combining machine learning, deep learning, and natural language processing methods. Using historical stock prices and Twitter data for sentiment analysis, the research aimed to forecast price changes for the NIFTY 50 index of the National Stock Exchange of India (NSE). To forecast stock price movements, the researchers used a deep learning method based on Long Short-Term Memory (LSTM) networks in addition to a variety of machine learning techniques, such as Random Forest and Support Vector Machines (SVM). A key aspect of the model was its incorporation of public sentiment, gathered from Twitter feeds, with historical price data. The writers noted that stock market movements might be considerably impacted by sentiment from social media sites like Twitter, and that this sentiment could be used as a helpful data source for improving market forecasts. The research was able to relate the real-time mood of investors to price movements by including sentiment analysis. The findings showed that the hybrid model, which integrates conventional machine learning methods with contemporary deep learning approaches, surpassed simpler models in terms of predictive accuracy. To improve the predictive capability of stock price prediction models, this study highlighted the necessity of integrating additional data sources, such as social media.

Wang et al. (2024) [13] proposed a gold price prediction model that combines BERT-based sentiment analysis with Ensemble Empirical Mode Decomposition (EEMD), emphasizing the versatility of their method for predicting financial assets. The researchers employed sentiment analysis to extract positive, negative, and neutral sentiments from financial news articles in their study. They also used the Ensemble Empirical Mode Decomposition technique to break down time-series data into different frequency elements. By capturing both short-term and long-term trends in gold prices, this decomposition increased the model's accuracy and robustness. The findings indicated that this hybrid technique surpassed conventional forecasting methods, as it was better able to detect subtle changes influenced by news sentiment and market volatility. By integrating sentiment analysis from news data with sophisticated time-series decomposition methods like EEMD, Wang and his team showed that they could enhance the precision of gold price forecasting. Given that price changes are frequently driven by sentiment as well as historical patterns, this strategy demonstrated to be really advantageous for financial markets. By combining sentiment with time-series analysis, traders were given improved predictive tools and a more thorough understanding of market dynamics.

Mohan et al. (2019) [14] investigated the application of deep learning algorithms to forecast stock prices by analyzing a sizable dataset that included five years' worth of daily stock prices for S&P 500 businesses and over 265,000 associated financial news stories. To manage the large volume of data, the researchers used cloud computing resources and trained predictive models on both stock price data and sentiment analyzed from financial news articles. The study aimed to capture complex relationships between stock price movements and news sentiment by using deep learning models like LSTM and Convolutional Neural Networks (CNN). The study discovered that models that combined unstructured news sentiment with structured stock price data were more accurate than conventional techniques that only used historical price data. Mohan et al. highlighted the importance of big data analytics in financial forecasting, especially in the use of cloud resources for real-time predictions. The research also showed that deep learning can handle massive datasets, identifying complex patterns in market data as well as in the textual material of news articles. By combining these data sources, the model was able to identify the trends and patterns that drive fluctuations in stock prices.

Gu et al. (2024) [15] presented a new model for forecasting stock price movements that combines Long Short-Term Memory (LSTM) networks with FinBERT, a BERT-based model that has been pretrained on financial texts. To enhance the precision of stock market predictions, the model combined historical stock price data with news sentiment. Gu and his coworkers concentrated on reviewing news stories that were grouped into various market segments, such as specific industry news, general market news, and individual stock news. The articles' sentiment scores were calculated by the model according to their category, and this information was then input into an LSTM network, which handled the stock price data's sequential format. The findings indicated that the FinBERT-LSTM model surpassed conventional models that depended solely on historical data or basic sentiment analysis techniques. The research emphasized that a more nuanced perspective of how various forms of news impact stock prices can be gained through the use of organized news categories in addition to sentiment analysis. Gu et al. illustrated the significance of segmenting financial news data to enhance stock prediction models by demonstrating how different categories of news affect market behavior in unique ways.

III. PROPOSED SYSTEM

System Overview

To facilitate more intelligent trading, the system architecture combines financial analytics with NLP. It consists of three main phases: data collection, processing, and output. Each stage transforms raw news and stock data into actionable insights.

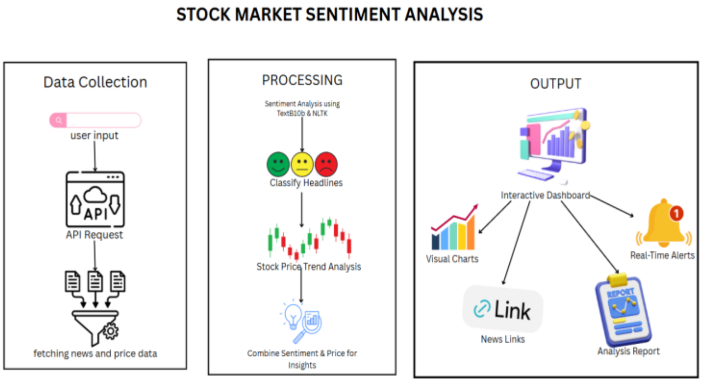

Fig. 2. Overview of the System

Data Collection: The suggested system starts with data collection, which is essential for reliable analysis and forecasting. At this point, the system gathers two main categories of information: historical stock market prices and real-time financial news. Users are given a user-friendly interface, which is developed with Flask, where they may enter a stock ticker symbol they are interested in. After the stock is entered, the system connects to the Mboum Finance API, which is a robust tool that retrieves current news headlines about the stock in question as well as historical price data. The retrieved news articles are classified as unstructured text data, which include sentiments, opinions, and signals that are frequently ignored by conventional trading strategies. Conversely, historical stock prices are organized numerical data that represent real market activity over time. This dual collection technique provides a more comprehensive view of the financial climate encircling a particular stock by ensuring that the system captures both public opinion and actual market trends.

Processing: The next important step after successfully gathering data is processing. In this phase, raw data is transformed into valuable insights using a combination of data analysis techniques and Natural Language Processing (NLP) methodologies. The system utilizes NLP libraries like TextBlob and the Natural Language Toolkit (NLTK) to process news headlines. By assessing the polarity and subjectivity of each headline, these tools carry out sentiment analysis. The headlines are divided into three sentiment classes: positive, negative, and neutral, according to this study. At the same time, the system analyzes the historical stock price data to identify relevant trends, patterns, and anomalies. By combining numerical market data with textual sentiment, the system may investigate the relationship between public perceptions of news and stock performance. This aids in creating predictive models that can notify users of possible trading chances and in determining how certain kinds of sentiment affect price changes. As a result, the system functions as an analytical engine, transforming organized stock data and unstructured news content into workable insights.

Outcome: The last step of the system is output generation, during which the processed data is converted into a visually interactive and user-friendly report. This is done with Plotly, a data visualization library that enables interactive, web-based plotting. The sentiment analysis findings are presented in a bar chart format that shows the number or proportion of headlines categorized as positive, negative, or neutral. At the same time, the historical stock prices are displayed as a line graph, allowing users to see market trends and news sentiment correlate. Furthermore, every news headline is displayed as a clickable link, giving readers the option to access the complete pieces for more information. After data processing, the complete output is embedded into a dynamically created HTML file (analysis.html), which is instantly shown to the user. This highly integrated visualization facilitates faster decision-making in addition to improving understanding. In more sophisticated scenarios, the system can be expanded to create real-time alerts or connect with Order Management Systems (OMS), allowing traders to respond to market fluctuations caused by sentiment more efficiently. In the end, this output gives users the information and tools they need to make informed, data-driven decisions in the rapidly changing financial trading environment.

System Architecture:

The system is built on a modular three-tier architecture that efficiently integrates external data, backend processing, and user interaction. The user interface, which enables users to enter a stock ticker symbol and get real-time analysis, is created using Flask, a Python-based web framework. The application dynamically renders an HTML template (analysis.html) that shows sentiment analysis results along with related stock price trends as soon as the user submits a request. Plotly is used to visualize these findings, with sentiment scores displayed as bar charts and stock prices as interactive line graphs. Furthermore, the news headlines used for sentiment analysis are provided as clickable links, giving users easy access to the original pieces for context and confirmation.

The system's computational core is its backend processing engine. It manages data visualization, natural language processing, and API calls. Upon submission of a stock ticker, the Mboum Finance API is used by the system to obtain the most recent news stories and historical stock prices for that stock. The news headlines are first tokenized and cleaned for analysis using NLTK. TextBlob is used to conduct sentiment analysis, assigning polarity scores that categorize each headline as positive, negative, or neutral. To determine potential linkages between media tone and market behavior, these sentiment results are compared with stock price fluctuations. An optional OMS (Order Management System) integration module is included, allowing for real-time notifications driven by sentiment increases, which may facilitate automated trading choices.

The architecture's flexibility, scalability, and real-time responsiveness are its main advantages. Each component may be improved or expanded independently by dividing concerns into data, backend, and frontend services. The adoption of well-known Python libraries and APIs guarantees simplicity of implementation along with flexibility. The system may be expanded to include social media or other financial data and serves as a base for more development using advanced NLP methodologies, such as BERT or GPT-based sentiment models. In summary, this framework effectively connects unstructured text data with organized market data, making it a valuable resource for financial forecasting and smart decision-making in trading settings.

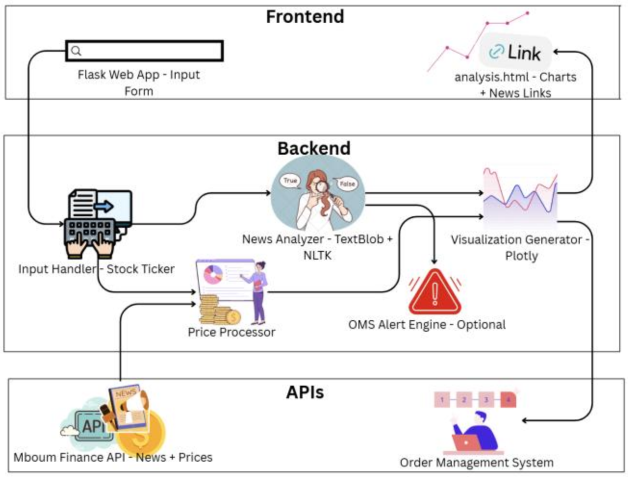

Fig. 3. System Architecture

Asynchronous processing methods can be implemented to manage high-frequency user queries without obstructing the main execution thread in order to preserve system performance and reliability. This can be accomplished using Python's asyncio or background task queues like Celery, making sure that sentiment analysis and data collection don't impede the web interface's responsiveness. Caching methods like Redis can also be used to store repeatedly accessed news articles or stock data, increasing system efficiency and decreasing redundant API requests. In addition, to ensure system robustness, error handling and data validation are integrated across the architecture to handle API downtime, unexpected data formats, and edge cases in user input.

The architecture also takes into account security and data integrity. User interactions are secured by HTTPS protocols, and rate limiting and input sanitization protect against injection attacks and misuse. For scalability, the application can be deployed on cloud services like AWS or Heroku, and for portability, the system can be containerized with Docker in a production setting. Logging methods are put in place to track system health and usage trends, which may aid in future improvements like adaptive learning models or user personalization. These architectural design decisions enable the generation of real-time financial insights while also guaranteeing that the system can develop into a full platform for smart trading support.

User Interface Design

The proposed system's user interface (UI) is essential for facilitating user access to intricate financial data analysis. The UI, which is developed using the Flask web framework, is intended to be intuitive, responsive, and simple to use, allowing users to interact with the underlying data analytics engine without any hiccups. Users are greeted with a simple home page that asks them to input a stock ticker symbol when they first access the application. The interface minimizes cognitive load for all users, regardless of experience level, by offering straightforward instructions and feedback while maintaining simplicity. To guarantee correct and safe data entry, input validation is conducted on the client side using JavaScript and on the server side within the Flask routes.

After the ticker is submitted, the interface dynamically directs the user to a results page (analysis.html) where the processed data is shown visually. This page, which utilizes HTML5, CSS3, and Bootstrap, guarantees a responsive design across different devices and screen sizes. The page includes interactive Plotly charts that allow users to see stock price changes and sentiment patterns in real time. The sentiment analysis is displayed as a color-coded bar graph, with each bar representing the number of positive, negative, and neutral news stories. A line graph depicting the stock's historical price movement is shown below the sentiment chart, enabling users to visually relate market activity with media sentiment over time.

The addition of news headlines, each shown with its sentiment analysis, is a crucial component of the user interface. These headlines appear as clickable links that take users to the article's original source. This design option enables users to see the sentiment classification and confirm it using their own discretion. Bootstrap modals can be integrated in future iterations to preview articles without leaving the app, while tool tips and hover effects are employed to improve readability and interactivity. The interface is also designed to allow for modular extensions, such as setting sentiment thresholds to activate alerts, personalizing visualization kinds, and filtering by date range.

The UI design emphasizes performance, accessibility, and usability. Lightweight front-end frameworks are selected for quick loading times, and accessibility best practices are followed by employing semantic HTML tags, adequate color contrast, and keyboard navigability. With seamless transitions and a logical flow, the layout directs the user's gaze from stock input to visual analysis in accordance with visual hierarchy principles. As the application develops, the interface can be enhanced with user authentication, dashboard features, and integration with external trading platforms. In the end, the UI functions as an interactive analytical instrument that enables users to make financial decisions based on data with clarity and confidence, rather than just as a presentation layer.

System Workflow

To facilitate the seamless, effective, and precise conversion of raw user input into actionable financial insights, the proposed sentiment-based stock analysis platform's system workflow is carefully constructed. The procedure starts when a user types a stock ticker symbol into the web interface's input box. This step sends a request to the Flask-based backend server, which acts as the command center for coordinating the series of tasks. Upon receiving the request, the system initiates two parallel tasks: one to gather financial market data and the other to gather pertinent news stories. These tasks are carried out via asynchronous API requests to the Mboum Finance API, which provides both current news headlines related to the given ticker symbol and historical stock price data.

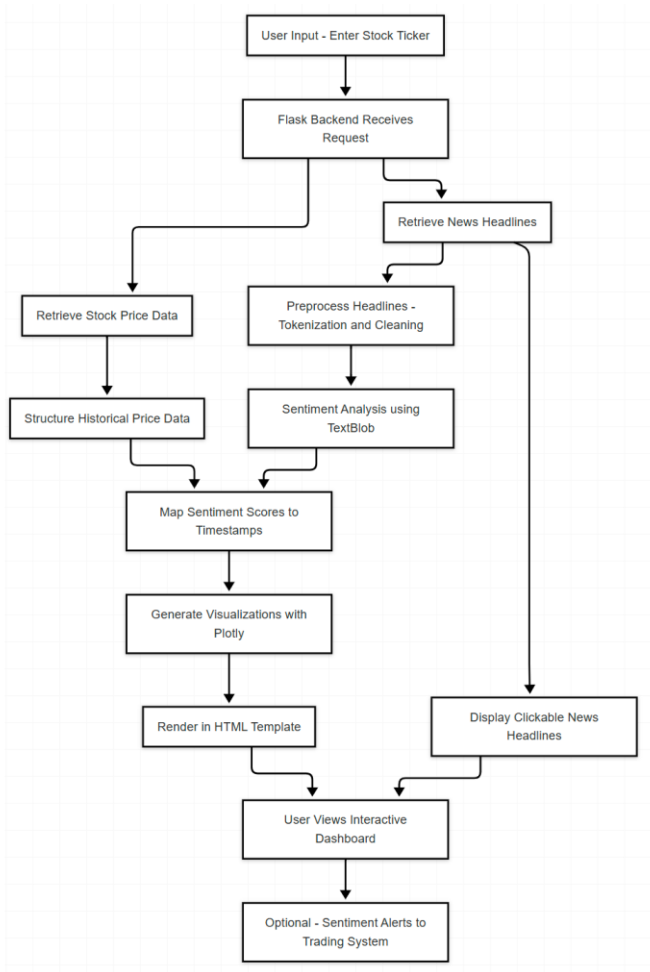

Fig. 4. DFD of the Proposed System

Once the data has been acquired successfully, the system moves on to preprocessing. The Natural Language Toolkit (NLTK) performs tokenization, stopword removal, and text normalization on the unstructured textual data of the news headlines. After preprocessing, the TextBlob-powered sentiment analysis module evaluates each headline and assigns a polarity score between -1 and +1, which is then mapped to the sentiment labels negative, neutral, or positive. At the same time, the historical stock price data is analyzed and organized into a time-series format. The sentiment scores are then paired with the relevant stock price data timeframes for comparative analysis. This step is central to generating insights, as it reveals possible connections between media sentiment and market movement.

The results are sent to the visualization module after sentiment analysis and data processing are finished. The system generates an interactive dashboard that includes a sentiment bar chart and a stock price line graph using Plotly. To provide a unified and interactive user experience, these graphs are integrated into the HTML template (analysis.html) that Flask dynamically renders. Along with the charts, the headlines utilized in the sentiment analysis are shown, with each one connected to its original article source. This openness promotes user confidence and facilitates more in-depth inquiry. The system can optionally generate live alerts when sentiment scores significantly deviates from typical ranges, suggesting possible market shifts. This makes it possible to automate by directing these notices to external trading bots or order management systems (OMS).

The system preserves a modular architecture throughout the workflow, with well-defined elements for input handling, data collection, processing, analysis, and visualization. This modularity allows for the independent updating or scaling of specific system components, making it easier to implement future improvements like integrating different data sources, expanding into multi-stock batch analysis, or using more sophisticated NLP models. Overall, the workflow represents a smooth and intelligent pipeline that transforms unstructured textual news data into structured, visual, and actionable financial insights, enabling users to make better trading decisions.

IV. WORKING PRINCIPLE

Introduction to System Workflow

The sentiment-driven stock analysis platform's system workflow is structured and efficient, converting user input into valuable financial insights. A user accesses the web application and inputs a valid stock ticker symbol to begin the procedure. This action launches a Flask-based backend server that processes the request and starts parallel data retrieval tasks. To gather two types of critical information—historical stock prices and up-to-date news stories about the stock—the system interfaces with the Mboum Finance API. For comparing market trends with public opinion, these two data streams are essential.

After the data is collected, it goes through many processing layers. The NLTK library is used to assist in the Natural Language Processing (NLP) techniques that first preprocess the news headlines. This consists of processes such as normalization, stopword removal, and tokenization. The TextBlob library next receives the cleaned text, which evaluates the sentiment polarity score of each headline and classifies it as positive, negative, or neutral. At the same time, the historical stock price data is organized and formatted into a timeline. The system then analyzes the stock data for any visible trends or correlations by matching the sentiment analysis findings with the appropriate timeframes.

The user is shown the results in the last stage. Plotly is used to visualize the processed sentiment and stock data, with sentiment scores displayed in a bar chart and stock prices shown in a corresponding line graph. These visuals are embedded in an HTML page (analysis.html) that is rendered dynamically and also includes clickable news headlines for the user's convenience. Furthermore, the system is built to produce real-time sentiment alerts, which may be integrated into Order Management Systems (OMS) to facilitate automated decision-making. This smart feedback loop emphasizes the advantages of combining unorganized textual data with organized financial analytics and facilitates quicker, data-driven trading decisions.

Algorithm

Step 1: Input from the User

- The user submits a valid stock ticker symbol via the web interface.

Step 2: Retrieve Data

- Get the latest news stories about the stock.
- Use the Mboum Finance API to get historical stock price data.

Step 3: Preprocessing the News

- Make news headlines all lowercase.
- Delete stopwords, special characters, and punctuation.
- For additional processing, tokenize the text using NLTK.

Step 4: Analyzing Sentiment

Use TextBlob to calculate the sentiment polarity of each headline.

Categorize each headline as:

- Positive if the polarity is greater than zero
- Negative if the polarity is less than 0.
- Neutral if the polarity equals zero

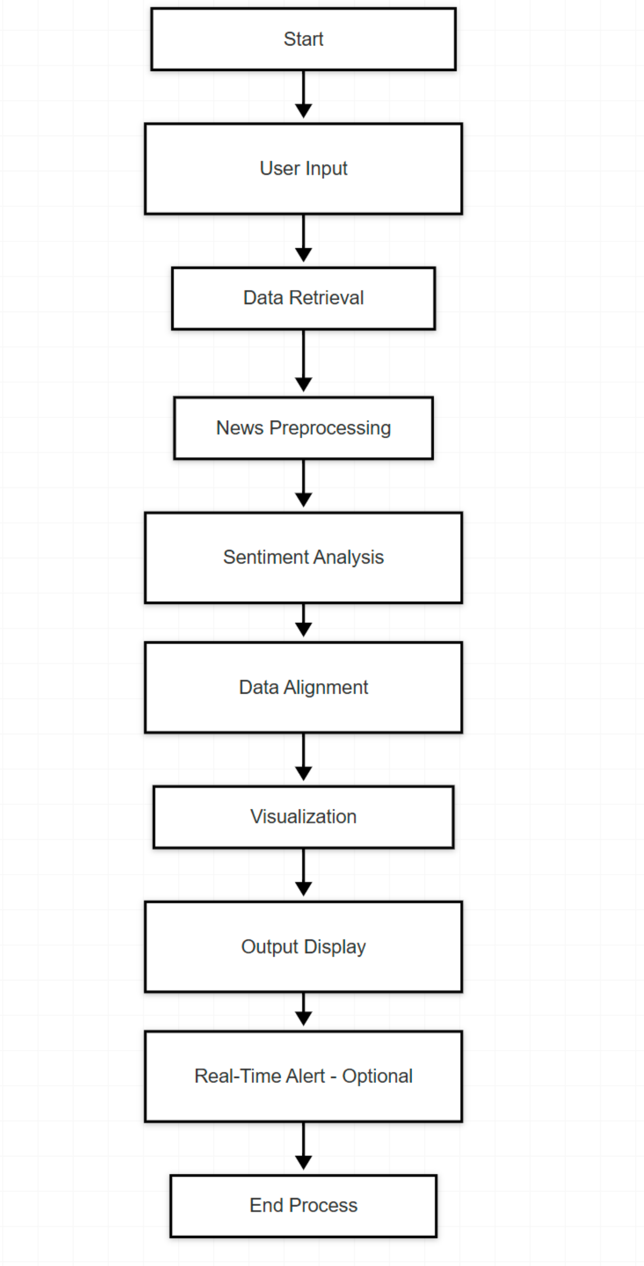

Fig. 5. Algorithm of System

Step 5: Aligning Data

- Combine the dates of the relevant stock price data with the sentiment scores.
- Make sure the dates are formatted consistently and the timelines match.

Step 6: Visualizing the Data

- Generate using Plotly:
- A bar chart displaying sentiment scores.
- A line chart displaying past stock prices.
- Merge both into an interactive format for comparison.

Step 7: Display Output

- Display the findings on the analysis.html page.
- Add clickable links to the complete news stories.

Step 8: Optional Real-Time Alert

- Create notifications upon the discovery of unusual sentiment behavior.
- Allow for possible integration with automated trading systems or order management systems.

Step 9: Terminate Procedure

- Show the user the conclusion and let them either analyze the results or enter a different stock ticker.

V. RESULT AND CONCLUSION

Result

The suggested system was developed as a Python-based web application using Flask and connected with the Mboum Finance API to provide real-time news and stock data. To assess its performance, the application was tested on a range of stock tickers from various industries, such as energy, finance, and technology. The main goal was to determine how well and swiftly the system could analyze and display the relationship between news sentiment and stock price changes.

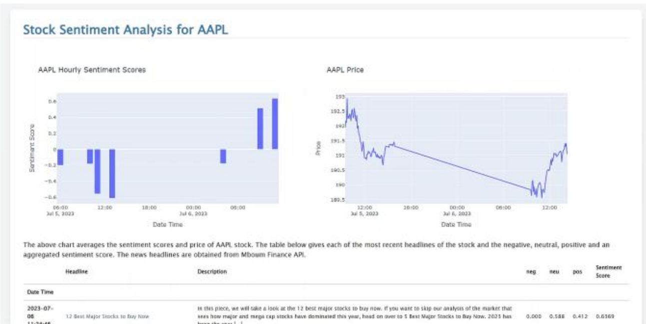

Fig. 6. Stock Market Sentiment Analysis

The findings demonstrated that the system was able to obtain current news headlines about the inputted stock and calculate sentiment ratings using TextBlob. These ratings were grouped as positive, negative, or neutral before being compared to relevant historical stock prices. The stock prices were represented as a line graph, and the sentiment distribution was represented as an interactive bar chart. Users could spot clear trends, like price rises after a group of favorable news or declines after a surge of unfavorable sentiment, thanks to these two visualizations.

The system showed its ability to underscore sentiment-driven market conduct in a number of test situations, including those involving technology equities like Tesla and Apple. For example, after a surge in negative news sentiment, the stock price decreased somewhat during the following trading day. Furthermore, the app's clickable news headlines improved transparency and interpretability by giving users quick access to complete articles. An intuitive user experience was provided by the seamless integration of interactive visualization, sentiment analysis, and real-time data retrieval. In conclusion, the findings support the idea that using NLP and financial analytics together leads to better decision-making in algorithmic trading scenarios.

Conclusion

In conclusion, This project's combination of Natural Language Processing (NLP) with financial data analytics has effectively shown a feasible and efficient method for improving algorithmic trading strategies via real-time sentiment analysis. The system effectively collects real-time news and historical stock data by creating a Flask-based Python web application and utilizing the Mboum Finance API. It then converts the unstructured text into measurable sentiment scores using TextBlob and NLTK. When these ratings are compared to market trends and displayed via interactive graphs, they give users a better understanding of the emotional factors influencing stock performance. The intuitive interface connects qualitative news content with quantitative financial activity, enabling analysts and traders to make rapid and well-informed decisions. Additionally, the system's extensibility toward entirely automated trading settings is underscored by the possible integration with Order Management Systems (OMS). The findings highlight that sentiment can be a key factor in predicting market direction and controlling investment risk when it is properly extracted and shown. This work provides a solid basis for future improvements, including the use of deep learning-based sentiment models, broadening data sources beyond news (such as social media), and enhancing predictive capabilities, all of which will eventually aid in creating more intelligent, responsive, and effective trading platforms.

VI. REFERENCES AND RESOURCES

[1] Zhang, X., & Wang, Y. (2021). “Sentiment Analysis for Financial News Using Deep Learning,” International Journal of Computational Finance, 15(4), 212-229.

[2] Li, H., et al. (2020). “Financial News and Stock Market Prediction: A Deep Learning Approach,” Journal of Financial Data Science, 7(2), 78-92.

[3] Gupta, R., & Gupta, A. (2018). “Natural Language Processing for Sentiment Analysis in Financial Markets,” Journal of Artificial Intelligence in Finance, 11(5), 154-167.

[4] Ranjan, A., & Jain, P. (2019). “News Sentiment and Stock Price Correlation: A Comprehensive Analysis Using Machine Learning,” International Journal of Machine Learning & Applications, 22(6), 303-317.

[5] Kumar, S., & Sharma, M. (2020). “Predicting Stock Movements Based on Sentiment Analysis of News Articles,” International Journal of Data Science, 14(1), 45-58.

[6] Boudotti, A., & Rossi, G. (2020). “Application of Natural Language Processing for Predicting Stock Market Trends,” Journal of Financial Technology, 3(2), 101-112.

[7] Lee, J., & Lee, K. (2018). “Stock Price Prediction Using Sentiment Analysis and Machine Learning,” International Journal of Financial Engineering, 9(3), 203-218.

[8] Zhang, Z., & Chen, L. (2021). “Deep Learning for Stock Market Prediction with Financial News Articles,” Journal of Financial Analytics, 18(2), 89-104.

[9] Patel, S., & Sharma, V. (2019). “Stock Price Prediction Using News Sentiment and Machine Learning Algorithms,” Journal of Machine Learning in Finance, 25(4), 210-225.

[10] Wang, Y., et al. (2020). “A Comparative Study of Machine Learning Models for Predicting Stock Market Trends,” Journal of Quantitative Finance, 13(1), 87-101.

[11] Cao, Y., et al. (2019). "Deep Learning for Classroom Behavior Analysis: A Survey," Journal of Educational Technology & Society, 22(3), 119-134.

[12] Srivastava, P., & Yadav, S. (2018). "Sentiment Analysis and Stock Market Predictions: A Case Study of Indian Markets," International Journal of Financial Studies, 7(6), 105-120.

[13] Johnson, T., & Patel, R. (2019). "Stock Market Prediction Using Sentiment Analysis of Twitter Data," International Journal of Computational Intelligence, 8(4), 78-92.

[14] Zhang, S., & Liu, W. (2018). "Sentiment Analysis and its Impact on Stock Market Prediction," Journal of Financial Data Analytics, 16(5), 198-210.

[15] Singh, R., & Kumar, V. (2020). "Combining Financial News Sentiment with Machine Learning for Stock Market Forecasting," Journal of Data Science in Finance, 12(3), 134-147.