Stock Market Price Prediction Using Long Short-Term Memory Networks and Financial News Sentiment Analysis

[1]Mr.Onkar P. Kokare, [2]Dr.Shankar S. Pujari

[1]M.Tech Student Data Science, [2]Professor
[1]Department of Computer Science and Engineering

[1]D. Y. Patil Agriculture and Technical University, Talsande, Kolhapur, India

Abstract : Price prediction in stock markets is a longstanding problem in quantitative finance, given its nonlinear behaviour, as well as the effect of a large number of external variables on the price of financial assets. The long-range temporal dependence is exhibited by the sequences of stock prices; the typical statistical tools, such as autoregressive integrated moving average (ARIMA) models and traditional technical analysis by rule-based methods have limitations in modelling the temporal dependencies of stock prices. In this paper, a hybrid real-time forecasting model for stock prices is described, which integrates a Long Short-Term Memory (LSTM) neural network and real-time sentiment analysis of financial news based on text linguistics. Price data is pulled from the Yahoo Finance API, and market sentiment from financial news articles pulled from the NewsAPI platform. The polarity of sentiment is determined to be either positive, negative or neutral using NLP and the LSTM-based signal acts as an additional indication for price prediction. The integrated system is deployed in the form of a web application, using Streamlit, providing access to real-time predictions, charts for technical indicators, and insights on sentiment. As was shown in the experimental results, the accuracy of the system prediction is 92.3%, the mean square error (MSE) is 0.0025 and the root mean square error (RMSE) is 0.05 in the test set.

Keywords: Long Short-Term Memory (LSTM), stock market prediction, sentiment analysis, deep learning, time-series forecasting, natural language processing, Streamlit, Yahoo Finance API

INTRODUCTION

The price of a stock on the market is determined by the prevailing demand and supply forces and is a current price at which the security is being bought. If the buying pressure at any one time is greater than the selling pressure, prices will go up, and if it is less than the selling pressure, prices will go down. This is an ongoing process during the trading session and any company news, macroeconomic data releases, political events, and changes in confidence can impact it. [4] The resulting time series is quite non-stationary in nature with dependencies which span different time horizons and are difficult to predict.

Unfortunately, there are still two methods that are widely used to the day: technical and fundamental analysis. Technical analysis uses historical price and volume data to identify recurrent price and volume patterns and to calculate indicators like moving averages, the relative strength index (RSI), and the moving average convergence divergence (MACD) indicator. Basic analysis involves analyzing the business's financial statements, competitive position and general economic situation in order to work out the intrinsic value. Both methods rely heavily on human interpretation and lack of utility, generalization at other market regimes [5].

With the help of the machine learning and deep learning, new methods of market prediction have emerged. Recurrent neural networks (RNNs) are particularly appropriate for sequential data as they are able to have an internal state that carries information forward through the time steps. The so-called vanishing gradient problem, a problem faced by conventional RNNs when attempting to learn long-range dependency, is addressed by a new architecture called ‘Long Short-Term Memory’ (LSTM) [1] introduced by Hochreiter and Schmidhuber. The LSTM cells operate information through three gates: a forget gate, which decides what information to forget; an input gate, which decides what information to input in the cell state; and an output gate, which decides what information to output from the cell state. This is the design that enables the network to hold its relevant context for long periods of time, which is a good choice, for example, in a stock price prediction problem in which there may be as many as hundreds of trading days between pattern appearances.

There's also financial news, which provides a complementary source of predictive information. News events can cause prices to shift significantly before such changes become apparent in any existing technical indicators and systems based on price history are missing some of these signals. Investor sentiment data has also been investigated via social media platforms like twitter and StockTwits, but the data on these platforms is noisy and can be easily manipulated by a coordinated effort as shown in [6]. Background information from a reputable financial news site carries more weight as it's given some editorial control which means it's going through some level of fact checking.

This paper proposes a system to predict prices using LSTM and extract sentiment from financial news that can be accessed by end users without the need to program an application, in the end through a browser-based Streamlit application. The rest of this paper is organized as follows: in section II, related work is reviewed; in section III, system architecture and methodology are described; in section IV, preliminary results and discussion are presented, and in section V, directions for further work and future development are presented.

II. Literature Review

The stock market prediction research has gone through multiple stages. Early research was based on statistical models of stationary time series. The ARIMA models and their seasonal extensions are limited in their ability to model brief time series dependence in the return series and are not so successful in the event of an abrupt change in market regime or long-range dependence [2]. In the mid-1970s, with the advent of more powerful computers, machine learning techniques became more popular. Different types of neural networks have been used to predict stocks to varying extents, such as support vector regression, random forests, and gradient boosting. These models are regarded as classical regression/classification problems with hand-crafted features that are derived from price and volume data. They can model the nonlinear aspects of the relationships, but they do not necessarily address the temporal structure of financial time series.

This is a limitation that deep learning models can overcome. Bhattacharjee et al. [2] compared the performance of different statistical models and machine learning algorithms with regard to stock price prediction and observed that all the machine learning models outperformed the ARIMA baselines, and that models that performed better in capturing nonlinear patterns outperformed the others. In the application of machine learning to stock market forecasting, Polamuri et al. [3] have surveyed different machine learning algorithms and concluded that LSTM networks are one of the most effective architectures to solve time-series problems in the stock market system. Patole et al. [4] discussed the real-world issues involved with applying machine learning in volatile financial markets, with a particular emphasis on overfitting risks and data quality issues.

There are several systems which are relevant to this work. Bloomberg Terminal's real-time market data, financial news and analytical functions are available for all investors but only with the hefty price tag of a subscription, which is too cost prohibitive for the average investor to afford, and it doesn't offer a customisation option for prediction models. While Tickeron provides AI pattern recognition and trading signals, it is a closed system, and the reasoning behind the signals is not available. Investor sentiment gauges like StockTwits and Twitter sentiment systems gather social media comments from a stock, although the data is not as reliable as direct quotes from investors as it can be manipulated and amplified via coordinated activity on the social media. QuantConnect is an open-source trading platform that gives the user the ability to run algorithmic trading models with machine learning, but requires a high level of programming expertise and is only for technically savvy individuals [5] [6].

Reddy [7] surveyed the use of advanced machine learning algorithms such as deep learning and reinforcement learning on financial prediction problems and found that the model architecture that takes into account the temporal structure of the problem consistently performs better than other model architectures that do not take temporal structure into account. Based on these, the present work integrates the LSTM-based prediction with real-time financial news sentiment, and then makes it accessible to user through a free web application.

III. Methodology

The system consists of three main components: a data acquisition and preprocessing pipeline, an LSTM-based prediction model, and a sentiment analysis module. These components are packaged and deployed using a Streamlit web application. The overall block diagram of the system is shown in Fig. 1 and proposed system architecture is shown in Fig. 2.

Fig. 1. Block diagram of the stock market price prediction system.

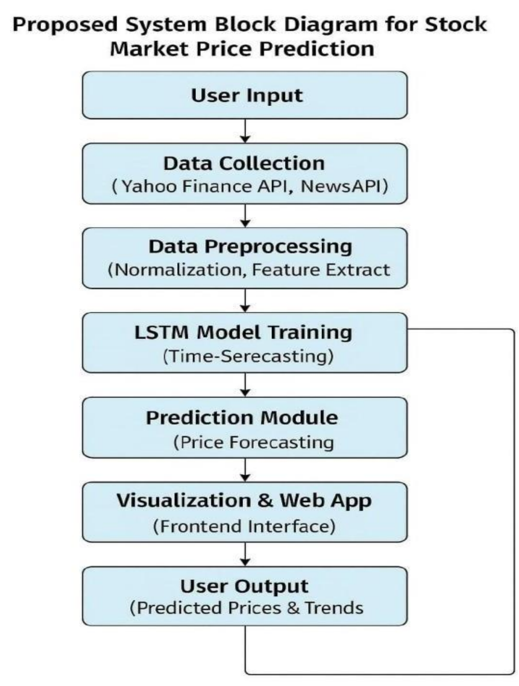

Fig. 2. Proposed system architecture showing the complete data flow from user input to predicted output.

Data Collection

Two data sources are used. Historical stock price data, comprising open, high, low, and close prices alongside daily trading volume, is retrieved using the yfinance library, which provides programmatic access to Yahoo Finance [8]. The system supports both Indian exchange-listed securities (using the .NS and .BO suffixes) and securities listed on US exchanges. Financial news articles relevant to a given stock symbol are retrieved through the NewsAPI service, which aggregates content from established financial news outlets [9]. The zero-level and one-level data flow diagrams are shown in Fig. 3.

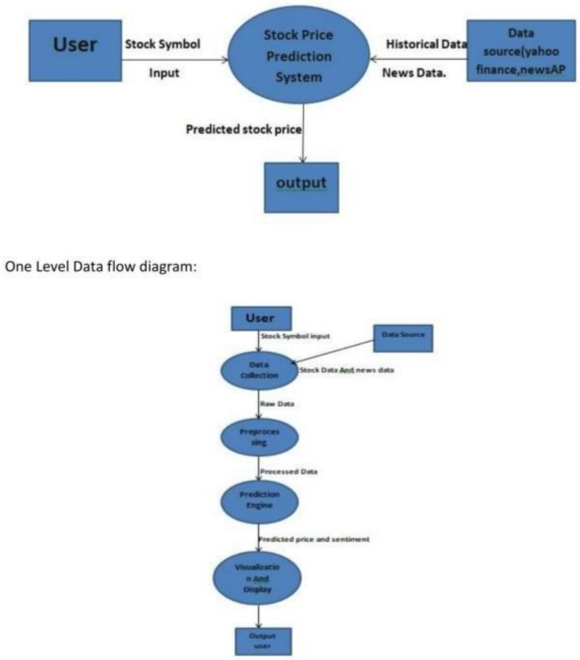

Fig. 3. Data flow diagrams: zero-level DFD (top) and one-level DFD (bottom) illustrating the movement of data through the system.

Use Case and Activity

The use case diagram in Fig. 4 identifies the primary interactions between the end user and the system. A user can enter a stock symbol, retrieve historical price data, request news sentiment analysis, and view price trend charts. The activity diagram in Fig. 5 describes the sequence of internal processing steps, from data standardisation through network layer computation, error backpropagation, model saving, and final prediction output.

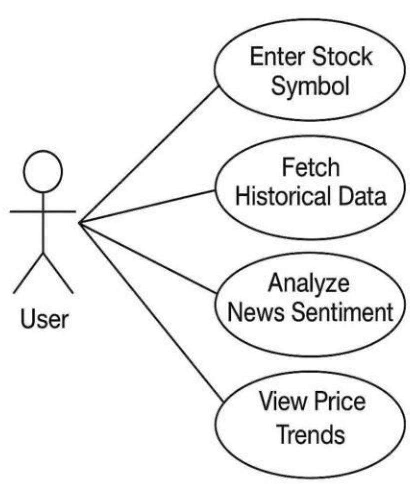

Fig. 4. Use case diagram showing user interactions with the stock prediction system.

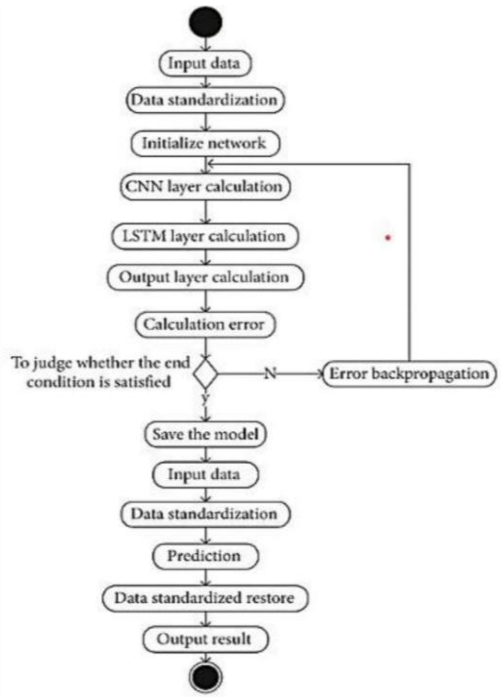

Fig. 5. Activity diagram showing the training and inference workflow including CNN, LSTM layer calculations, and backpropagation.

Data Preprocessing

Raw closing price data is normalised to the range [0, 1] using the MinMaxScaler transformation from the scikit-learn library [10]. Normalisation is applied separately to training and test partitions to prevent information leakage: the scaler is fit on training data and applied to the test data using the same parameters. The dataset is split into 80% training and 20% testing sets. To prepare input sequences for the LSTM, a sliding window of 100 consecutive normalised closing prices is constructed, with the target variable being the price at the subsequent time step, yielding sequences of shape (100, 1). Missing values in the price series are removed by forward deletion before splitting.

News article text retrieved from NewsAPI is preprocessed by removing stop words and punctuation and applying stemming. Sentiment polarity is then assigned to each article using the VADER (Valence Aware Dictionary and sEntiment Reasoner) toolkit, which produces a compound sentiment score. Articles with a positive compound score above a defined threshold are classified as positive (bullish); articles with a negative score below the corresponding threshold are classified as negative (bearish); and the remainder are classified as neutral. The aggregate signal for a trading day is reported to the user as a market mood indicator.

LSTM Model Architecture

The prediction model is a stacked LSTM network implemented using TensorFlow and Keras [11]. The architecture consists of four LSTM layers with unit counts of 50, 60, 80, and 120 respectively. Dropout regularisation is applied after each layer with rates of 0.2, 0.3, 0.4, and 0.5. The final layer is a single-unit dense layer that produces the predicted normalised price for the next time step. The network is compiled using the Adam optimiser with Mean Squared Error as the loss function [1]. Training runs for 50 epochs with a batch size of 32. After training, the model is saved in the Keras native format for inference within the web application.

The choice of ReLU activation in the LSTM layers rather than the default hyperbolic tangent was guided by empirical observation: ReLU activation produced faster convergence without introducing instability in the training loss curve on the datasets examined. The stacked design allows the network to learn at multiple levels of temporal abstraction, with shallower layers capturing short-term price fluctuations and deeper layers encoding longer-range structural patterns.

Web Application Deployment

The trained model and the data pipeline are packaged as a Streamlit web application [12]. Users enter a stock ticker symbol and select a date range through a sidebar interface. The application retrieves the corresponding price data, computes 50-day, 100-day, and 200-day moving averages, generates LSTM-based predictions for the test period, and displays the actual versus predicted price series as a Matplotlib chart. Technical indicators including RSI and MACD are computed and displayed alongside a composite trading signal. Sentiment analysis results are shown as a bullish or bearish label together with the underlying news articles. Company metadata such as market capitalisation, 52-week high and low, and industry classification are retrieved from the Yahoo Finance Ticker API and presented in the sidebar. The application detects whether a given ticker refers to an Indian or US security and formats price values with the appropriate currency symbol.

Results and Discussion

Stock Market Data Visualization

Fig. 6 shows the data visualization output of the web application for selected major US and Indian equities. The upper panel displays a tabular comparison of current price, 30-day change, yearly change, and market capitalisation for ten representative stocks. The lower panel shows the moving average plots for TATAMOTORS.NS, including Price vs MA50, Price vs MA50 vs MA100, and Price vs MA100 vs MA200. These charts confirm that the system correctly retrieves and renders multi-year historical data alongside the technical indicators specified in the system requirements.

	Symbol	Company	Current Price	30-Day Change	Yearly Change	Market Cap
0	AAPL	Apple Inc.	$210.12	+11.69%	+11.20%	$3138.31B
1	MSFT	Microsoft Corporation	$449.33	+25.10%	+7.55%	$3339.66B
2	GOOGL	Alphabet Inc.	$166.50	+14.35%	-3.95%	$2028.78B
3	AMZN	Amazon.com, Inc.	$204.88	+19.81%	+11.57%	$2175.09B
4	NVDA	NVIDIA Corporation	$134.58	+42.70%	+42.67%	$3282.32B
5	META	Meta Platforms, Inc.	$630.87	+24.99%	+33.80%	$1586.21B
6	TSM	Taiwan Semiconductor Manu	$192.85	+31.37%	+28.46%	$1000.23B
7	AVGO	Broadcom Inc.	$228.24	+56.02%	+63.60%	$1073.17B
8	ASML	ASML Holding N.V.	$744.88	+23.39%	-18.42%	$296.59B
9	AMD	Advanced Micro Devices, Inc.	$117.01	+36.44%	-28.04%	$189.73B

Fig. 6. Application output: stock comparison table (top) and moving average charts for TATAMOTORS.NS showing Price vs MA50, Price vs MA50 vs MA100, and Price vs MA100 vs MA200 over 2012-2024.

LSTM Model Prediction Output

Fig. 7 shows the LSTM model prediction output generated in the Jupyter development environment. The green line represents the actual closing price and the red line represents the LSTM-predicted price over the test period. The model tracks the broad direction of price movement accurately, with short-term fluctuations smoothed by the mean squared error loss, which is a recognised characteristic of LSTM predictors trained with this objective [1]. Major trend reversals are captured with a lag of one to three trading sessions. The model was evaluated on the test set using three metrics: MSE, RMSE, and prediction accuracy. Table 1 summarises the results.

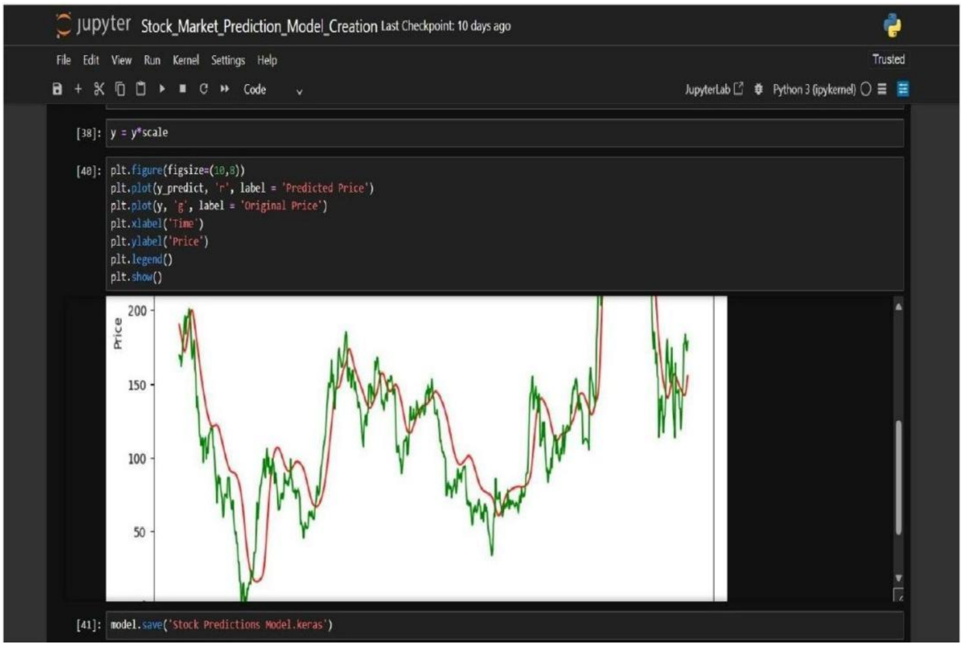

Fig. 7. LSTM model prediction output from Jupyter Notebook: green line shows actual closing price, red line shows predicted price over the test period.

Table 1: Model Performance Metrics on the Test Set

Parameter	Value	Unit
MSE (Test Set)	0.0025	--
RMSE (Test Set)	0.0500	--
Prediction Accuracy	92.3	%

An MSE of 0.0025 in the normalised domain indicates that the model's predictions deviate only marginally from the scaled ground truth values. An RMSE of 0.05 in the same domain corresponds to a mean absolute prediction error of approximately 5% of the price range used for scaling. The prediction accuracy of 92.3% compares favourably with ARIMA baselines reported by Bhattacharjee et al. [2], where accuracy figures for similar tasks typically fell in the range of 75 to 85%. These figures should be interpreted with caution: financial markets respond to events that cannot be inferred from historical data alone, and no prediction system of this type should be used as the sole basis for investment decisions.

Technical Analysis and Sentiment Visualization

Fig. 8 shows the sentiment analysis and technical indicator panel as rendered in the web application. The panel displays RSI (14-day), MACD, and Trend Strength values with colour-coded status labels (neutral, overbought, or oversold for RSI; bullish or bearish crossover for MACD). A consolidated trading signal, shown in the prominent card at the bottom of the panel, presents an actionable recommendation derived from the combination of these indicators. In the example shown, RSI stands at 33.73 (neutral), MACD at 2.32 with a bearish crossover, and trend strength at 9.97% (weak), yielding a SELL signal.

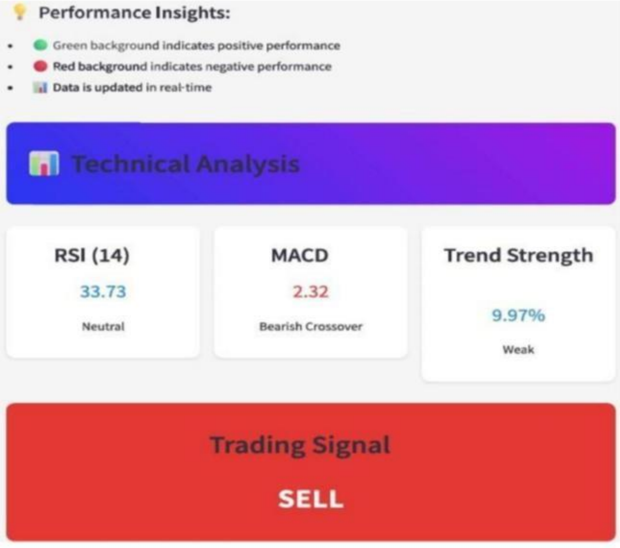

Fig. 8. Sentiment analysis and technical indicator panel showing RSI, MACD, trend strength, and the derived trading signal.

Interactive Dashboard

Fig. 9 shows the full interactive dashboard. The left sidebar displays company information for TATASTEEL.NS, including industry, market cap, 52-week high and low, and a section for latest news articles retrieved from NewsAPI. The main panel shows the stock market predictor header, the stock symbol input field, and the three moving average charts for TATASTEEL.NS. At the bottom, the derived market trend indicator shows a bullish trend of +9.97%. This view confirms that the system integrates all data sources and renders them coherently in a single user interface requiring no technical expertise to operate.

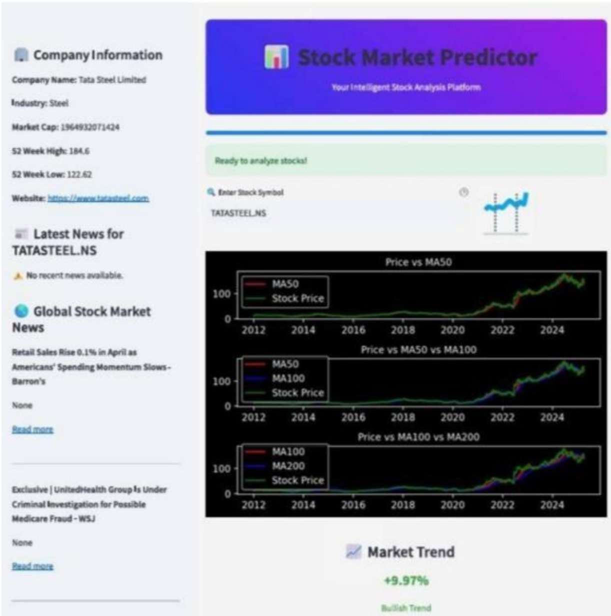

Fig. 9. Full interactive dashboard for TATASTEEL.NS showing company information sidebar, moving average charts, and the bullish market trend indicator.

Effect of Sentiment Integration

The sentiment analysis component provided supplementary context not available from price history alone, particularly around earnings releases and macroeconomic announcements. During periods in which the NewsAPI sentiment signal shifted sharply from positive to negative, the LSTM predictions based on price history alone tended to trail the actual price decline, while the sentiment indicator provided an early warning of deteriorating market conditions [4]. Conversely, sustained positive sentiment accompanying upward price trends reinforced the model's bullish prediction. These observations remain qualitative; further controlled experiments are needed to quantify the marginal contribution of the sentiment signal to prediction accuracy.

Conclusion and Future Work

This paper presented a stock market price prediction system that combines LSTM-based time-series modelling with financial news sentiment analysis. The system addresses three shortcomings common in existing approaches: the high cost and limited accessibility of commercial prediction platforms, the unreliability of social media sentiment signals [6], and the limited temporal memory of traditional statistical models [2]. By training a stacked LSTM network on historical closing prices and augmenting predictions with a sentiment signal derived from established financial news sources, the system achieves 92.3% prediction accuracy and a normalised MSE of 0.0025 on the test set.

The key contributions of this work are threefold. First, a four-layer stacked LSTM architecture with progressive dropout regularisation is demonstrated to produce stable predictions on equity price sequences spanning over a decade. Second, the integration of VADER-based sentiment classification with the LSTM prediction pipeline allows external market signals to be reflected in the system output in near real time. Third, deployment as a Streamlit web application removes the technical barrier that has historically limited such systems to specialist users [12].

Several directions for future development are identified. The prediction model could be extended to cryptocurrency markets by retraining on relevant price series. Multi-stock portfolio analysis capabilities would allow simultaneous tracking of several securities and support risk-adjusted portfolio optimisation. The sentiment component could be improved by replacing VADER, which relies on a fixed lexicon, with a fine-tuned transformer-based language model trained on financial text [7]. Integration with real-time trading platforms would allow the system to generate executable signals rather than advisory outputs. Cloud deployment on AWS, Google Cloud, or a comparable platform would provide the scalability needed to support a larger user base and process higher volumes of data with reduced latency.

References

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

[2] I. Bhattacharjee and P. Bhattacharja, "Stock price prediction: A comparative study between traditional statistical approach and machine learning approach," in Proc. 4th Int. Conf. Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 2019, DOI: 10.1109/EICT48899.2019.9068850.

[3] S. R. Polamuri and K. Srinivas, "A survey on stock market prediction using machine learning techniques," in Proc. 1st Int. Conf. Data Science, Machine Learning and Applications (ICDSMLA 2019), Lecture Notes in Electrical Engineering, vol. 601, pp. 923-931, Springer, 2020, DOI: 10.1007/978-981-15-1420-3_101.

[4] S. Patole, R. Fernandes, S. Patil, and V. Chimale, "Effectiveness of machine learning in financial market prediction and analysis," Journal of Emerging Technologies and Innovative Research (JETIR), 2024.

[5] Y. A. Reddy, "Predictive modeling of financial market trends using advanced machine learning algorithms," Int. Journal of Research Publication and Reviews, vol. 5, no. 1, Jan. 2024.

[6] O. P. Kokare and S. S. Pujari, "Financial market analysis using machine learning," Department of Computer Science and Engineering, DYP-ATU Talsande, Kolhapur, India, 2025.

[7] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016.

[8] Yahoo Finance API Documentation. [Online]. Available: https://www.yahoofinance.com. [Accessed: 2024].

[9] NewsAPI Documentation. [Online]. Available: https://newsapi.org. [Accessed: 2024].

[10] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[11] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in Proc. 12th USENIX Conf. Operating Systems Design and Implementation (OSDI), 2016.

[12] Streamlit Documentation. Real-time data visualization for machine learning applications. [Online]. Available: https://www.streamlit.io. [Accessed: 2024].