A Deep Learning and Investor Sentiment-Based Stock Price Prediction Model

Duggi Shashikala[1], M. Maruthi rao[2], B. Mahesh Kumar[3], B. Chandhu Lalitha Kumari[4]

[1,2,3]*Assistant Professor, Dept; of CIC, AITS, Tirupati-517520.*

[4]*Assistant Professor, Dept; of CSE, SVCN, Nellore-524316.*

***Abstract*—Accurate prediction of stock prices can reduce investment risks and increase returns. This project combines the multi-source data affecting stock prices and applies sentiment analysis, swarm intelligence algorithm, and deep learning to build the MS-SSA-LSTM model. Firstly, we crawl the East Money forum posts information to establish the unique sentiment dictionary and calculate the sentiment index. Then, the Sparrow Search Algorithm (SSA) optimizes the Long and Short-Term Memory network (LSTM) hyper parameters. Finally, the sentiment index and fundamental trading data are integrated, and LSTM is used to forecast stock prices in the future. Experiments demonstrate that the MS-SSA-LSTM model outperforms the others and has high universal applicability.**

***Index Terms*—MS SSA, Deep learning, LSTM, Share market, Prediction.**

I. INTRODUCTION

In the ever-evolving landscape of global financial markets, the quest for accurate and timely stock price prediction remains one of the most compelling challenges for economists, data scientists, investors, and policymakers alike. Traditional stock forecasting models, often rooted in technical analysis or fundamental valuation, have long relied on historical price data, financial ratios, earnings reports, and macroeconomic indicators to inform decision-making. While these methodologies have contributed significantly to our understanding of market dynamics, they largely ignore one of the most potent and unpredictable forces in finance: human emotion. In reality, market behavior is not driven solely by rational calculations; rather, it is heavily influenced by investor psychology, collective sentiment, and perception. This reality has become even more pronounced in the digital age, where social media platforms, online forums, and news outlets continuously shape investor outlooks and expectations. Recognizing this paradigm shift, the integration of investor sentiment into stock prediction models represents not only a technological advancement but also a conceptual breakthrough in modern financial modeling. At the core of this research lies the development of a stock price prediction model that synthesizes investor sentiment analysis with the computational power of optimized deep learning architectures. By extracting qualitative data from digital channels—such as Twitter, Reddit, and financial news—and converting it into quantitative sentiment scores, the model captures the emotional pulse of the market in real-time. This sentiment data, when used as an input in deep learning models such as Long Short-Term Memory (LSTM) networks or Gated Recurrent Units (GRUs), allows for the identification of complex, nonlinear relationships between public mood and market trends. Unlike conventional models that rely solely on time-series patterns or fundamental ratios, this hybrid approach accounts for a more holistic view of market behavior, incorporating both empirical data and behavioral finance insights. Beyond the financial sector, regulators and policymakers may also find value in sentiment-aware prediction models. By monitoring sentiment trends and their correlation with price volatility, regulatory bodies can better understand the emergence of speculative bubbles or panic-induced sell-offs. This awareness can inform more effective market surveillance, early warning systems, and policy responses designed to maintain market stability. Moreover, in the broader economic context, improved stock price forecasting contributes to more efficient capital allocation. When investment decisions are based on richer and more nuanced data, including sentiment, capital flows more naturally toward companies that truly create value, thus promoting overall economic growth and resilience. The ethical dimension of using sentiment data for stock prediction also merits consideration. As models become more adept at interpreting and anticipating public opinion, questions arise regarding data privacy, manipulation, and algorithmic transparency. Ensuring that such models are developed and deployed responsibly with clear guidelines on data usage, bias mitigation, and accountability will be critical to their long-term acceptance and integration into financial ecosystems. Nevertheless, when implemented with care and integrity, these technologies hold great promise in enhancing market efficiency, reducing systemic risk, and empowering a more inclusive financial future.

II. LITERATURE SURVE

This study further explores the use of Twitter sentiment to predict financial market trends. By collecting tweets over several months and correlating sentiment trends with stock market indicators like the DJIA and trading volume, the researchers demonstrated that public mood shifts often precede market movements. Although their model used linear regression for prediction, the study contributes valuable evidence that real-time public opinion data can serve. In this study, the authors applied LSTM networks to historical stock price data for the S&P 500. The model was trained to predict daily directional movements and demonstrated a notable improvement over traditional machine learning models such as Random Forests and Support Vector Machines. While this research did not include sentiment data, it provides strong support for using LSTM architectures in time-series-based stock prediction. The relevance to the current project lies in validating the suitability of LSTM as a deep learning technique for financial forecasting. Akita and colleagues proposed a hybrid model that combines financial time series with financial news using a deep learning approach. They used CNN and LSTM models to process structured stock data and unstructured news headlines, respectively. Their research confirmed that integrating textual data can enhance the prediction accuracy of stock prices. This study is directly relevant to your project as it supports the integration of sentiment-driven textual data into deep learning models for more nuanced This recent study developed a sentiment-aware deep neural network that incorporated investor sentiment extracted from online forums and financial news. Using Bi-LSTM and attention mechanisms, the model effectively learned to prioritize emotionally charged content, resulting in more accurate predictions than baseline LSTM or CNN models. The researchers emphasized the impact of sentiment polarity and intensity on short-term market fluctuations. Their approach aligns closely with the goals of the current project, particularly the use of sentiment as a core input and the emphasis on deep learning optimization.

Model Evaluation and Validation

The trained model is evaluated using a reserved test set and validated using metrics appropriate for regression or classification. Common evaluation metrics include:

- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)
- R-squared (R^2)
- Accuracy and F1-score (for classification tasks)

III. SOFTWARE FEATURES

The software developed for “A Stock Price Prediction Model Based on Investor Sentiment and Optimized Deep Learning” integrates advanced data analytics, deep learning, and sentiment evaluation tools into a unified system. Its features are designed to provide robust forecasting capabilities, actionable insights, and user-friendly access to model outputs for financial decision-making. These features are essential for researchers, analysts, investors, and developers who require an intelligent, responsive, and transparent system for stock market analysis.

Pre-trained NLP models (e.g., VADER, TextBlob, FinBERT) for identifying positive, negative, and neutral tones.

Custom lexicons for finance-specific language and slang used in social media and forums.

Aggregation tools that summarize sentiment by day or hour, enabling time-aligned integration with stock data.

1. Technical Indicator Generator

The software calculates common technical indicators such as:

- Moving Averages (MA, EMA)

- Relative Strength Index (RSI)
- MACD (Moving Average Convergence Divergence)
- Bollinger Bands
- Volatility and Volume-based metrics

VI. PROJECT IMPLEMENTATION

Python is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python is dynamically-typed and garbage-collected programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). Today, Python is very high in demand and all the major companies are looking for great Python Programmers to develop websites, software components, and applications or to work with Data Science, AI, and ML technologies. When we are developing this tutorial in 2022, there is a high shortage of Python Programmers whereas market demands a greater number of Python Programmers due to its application in Machine Learning, Artificial Intelligence etc.

Python Web scrapping

Web Scraping is a technique to extract a large amount of data from several websites. The term "scraping" refers to obtaining the information from another source (webpages) and saving it into a local file. For example: Suppose you are working on a project called "Phone comparing website," where you require the price of mobile phones, ratings, and model names to make comparisons between the different mobile phones. If you collect these details by checking various sites, it will take much time. In that case, web scrapping plays an important role where by writing a few lines of code you can get the desired results.

Testing Techniques

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and ex results

The result analysis phase evaluates the effectiveness, accuracy, and real-world applicability of the developed stock price prediction model. It involves comparing the model's outputs with actual market performance, analyzing its predictive accuracy, studying the impact of sentiment data, and examining how well the model performs under different market conditions. This comprehensive evaluation helps in understanding the strengths and limitations of the system and informs decisions for future improvements. The object-oriented structure of the system ensures modularity, scalability, and ease of enhancement, while the use of Python and machine learning libraries provides flexibility and powerful computational tools. The model has shown promising results in both predictive accuracy and recommendation reliability, indicating that investor sentiment plays a significant role in influencing short-term market behavior. The fusion of real-time textual sentiment with historical financial data enables a more holistic and psychologically informed model that reflects both quantitative patterns and human behavioral factors. The prediction engine, backed by rigorous data preprocessing and model optimization, offers actionable outputs in the form of “Buy”, “Sell”, or “Hold” recommendations. This elevates the utility of the system from a mere forecasting tool to an intelligent assistant for investors. The interface’s usability and responsiveness further enhance user engagement, making the system accessible even to non-experts in data science or finance. Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

The code generation and validation phase is a critical part of the software development lifecycle for this project, ensuring that the implementation of the model meets its design goals, functions correctly, and delivers accurate stock price predictions. This stage transforms theoretical architecture and design into a working, executable system by writing actual code using object-oriented principles, integrating third-party tools and libraries, and validating performance through systematic testing.

Rapid Development
Django was designed with the intention to make a framework which takes less time to build web application. The project implementation phase is a very time taken but Django creates it rapidly. Django takes security seriously and helps developers to avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery etc. Its user authentication system provides a secure way to manage user accounts and passwords.

Scalable
Django is scalable in nature and has ability to quickly and flexibly switch from small to large scale application project.

Fully loaded
Django includes various helping task modules and libraries which can be used to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds etc.

Versatile
Django is versatile in nature which allows it to build applications for different-different domains. Now a days, Companies are using Django to build various types of applications like: content management systems, social networks sites or scientific computing platforms etc.

A Web Application Framework or a simply a Web Framework represents a collection of libraries and modules that enable web application developers to write applications without worrying about low- level details such as protocol, thread management, and so on. Flask is a web framework; it's a Python module that lets you develop web applications easily. It has a small and easy-to-extend core: it's a micro framework that doesn't include an ORM (Object Relational Manager) or such features. It does have many cool features like URL routing, template engine. It is a WSGI web app framework. Frameworks is a really well-done short video that explains how to choose between web frameworks. The author has some particular opinions about what should be in a framework. For the most part I agree although I've found sessions and database ORMs to be a helpful part of a framework when done well. What is a web framework? Is an in-depth explanation of what web frameworks being and their relation to web servers Jingo vs. Flash vs. Pyramid: Choosing a Python web framework contains background information and code comparisons for similar web applications built in these three big Python frameworks.
This fascinating blog post takes a look at the code complexity of several Python web frameworks by providing visualizations based on their code bases. Python's web frameworks benchmarks is a test of the responsiveness of a framework with encoding an object to JSON and returning it as a response as well as retrieving data from the database and rendering it in a template. There were no conclusive results but the output is fun to read about nonetheless.
What web frameworks do you use and why are they awesome? Is a language agnostic Reedit discussion on web frameworks? It's interesting to see what programmers in other languages like and dislike about their suite of web frameworks compared to the main Python frameworks. This user-voted question & answer site asked "What are the best general purpose Python web frameworks usable in production?" The votes aren't as important as the list of the many frameworks that are available to Python developers.

V. CONCLUSION

Future iterations could integrate more sophisticated NLP models like FinBERT, RoBERTa, or GPT-based embeddings specifically trained on financial text. These models can better understand context, sarcasm, and sector-specific jargon, improving sentiment accuracy. Expanding the sentiment sources to include YouTube transcripts, news articles, Reddit threads, and stock forums can enrich the sentiment dataset. A hybrid approach using textual, audio, and video sentiment can enhance the emotional understanding of the market. By deploying the model on cloud platforms (e.g., AWS, Azure), real-time streaming and live alerts can be introduced. Investors can receive instant recommendations as news or sentiment shifts, making the system more proactive.

REFERENCES

[1] W. Huang, Y. Nakamori, and S. Y. Wang, "Forecasting stock market movement direction with support vector machine," *Computers & Operations Research*, vol. 32, no. 10, pp. 2513–2522, 2005, doi: 10.1016/j.cor.2004.03.016.

[2] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, 2011, doi: 10.1016/j.jocs.2010.12.007.

[3] X. Zhang, H. Fuehres, and P. A. Gloor, "Predicting stock market indicators through Twitter: 'I hope it is not as bad as I fear'," *Procedia - Social and Behavioral Sciences*, vol. 26, pp. 55–62, 2011, doi: 10.1016/j.sbspro.2011.10.562.

[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014. [Online]. Available: https://arxiv.org/abs/1412.3555.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019. [Online]. Available: https://arxiv.org/abs/1810.04805.

[6] B. Liu, *Sentiment Analysis and Opinion Mining*, *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012, doi: 10.2200/S00416ED1V01Y201204HLT016.

[7] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *Proc. 2017 Int. Conf. on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1643–1647, doi: 10.1109/ICACCI.2017.8126078.

[8] B. Xie, H. Chen, and L. Liang, "A stock price prediction model based on LSTM and sentiment analysis," *IEEE Access*, vol. 8, pp. 226074–226085, 2020, doi: 10.1109/ACCESS.2020.3042766.

[9] C. J. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in *Proc. Int. AAAI Conf. on Web and social media*, vol. 8, no. 1, pp. 216–225, 2014. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14550.

[10] J. Brownlee, *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery, 2017. [Online]. Available: https://machinelearningmastery.com/deep-learning-for-time-series-forecasting/