# A Stock Price Prediction Model Based on Investor Sentiment and Optimized Deep Learning

Mr. Mohd Khaja Moizuddin[1], Khaja Adnan Ateeq Ahmed[2], Abdul Raqheeb Osaid[3], Abdul Muneeb Khan[4]
[1]*Assistant Professor, Dept. of CSE-AIML, Lords Institute of Engineering and Technology Hyderabad, India*
[2,3,4]*B. Tech Student, Dept. of CSE-AIML, Lords Institute of Engineering and Technology Hyderabad, India*

***Abstract*—Accurate prediction of stock prices can reduce investment risks and increase returns. This paper 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 dictio- nary and calculate the sentiment index. Then, the Sparrow Search Algorithm (SSA) optimizes the Long and Short-Term Memory network (LSTM) hyperparameters. 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. Compared with standard LSTM, the R2 of MS-SSA-LSTM is improved by 10.74% on average. We found that: 1) Adding the sentiment index can enhance the model's predictive performance. 2) The LSTM's hyper Param- eters are optimized using SSA, which objectively explains the model parameter settings and improves the prediction effect. 3) The high volatility of China's financial market is more suitable for short-term prediction.**

***Index Terms*—Stock Price Prediction, Investor Sentiment, Deep Learning, LSTM, Sparrow Search Algorithm, Sentiment Analysis**

## I. INTRODUCTION

Stock price prediction has always been a challenging task in financial markets due to the highly volatile, nonlinear, and time-varying nature of price series. Accurate prediction of stock prices can reduce investment risks and increase returns for investors and traders. Traditional statistical models such as ARIMA and GARCH have limitations in capturing complex patterns. In recent years, deep learning methods, especially Long Short-Term Memory (LSTM) networks, have shown superior performance in time series forecasting because they can remember long-term dependencies.

However, most existing models rely solely on historical trading data (e.g., open, high, low, close prices, volume) and ignore the impact of investor sentiment. With the rapid growth of online financial forums and social media, investors express their opinions and emotions publicly, which can significantly influence stock price movements. Therefore, incorporating sentiment analysis from forum posts can enhance prediction accuracy.

Moreover, the hyperparameters of LSTM networks are usually set manually, which is subjective and time-consuming. To overcome this limitation, swarm intelligence algorithms such as the Sparrow Search Algorithm (SSA) can be used to automatically optimize LSTM hyperparameters, improving both efficiency and accuracy.

This paper proposes a novel MS-SSA-LSTM model that integrates multi-source data (historical trading data and forum sentiment) and optimizes LSTM using SSA. The main objec- tives are:

- To collect and integrate sentiment data from forums along with stock trading data, capturing both market trends and investor opinions.
- To perform sentiment analysis by building a domain- specific dictionary and index, improving the model's ability to understand market behavior.
- To optimize LSTM hyperparameters using the Sparrow Search Algorithm (SSA), enhancing model efficiency and prediction accuracy.
- To develop the MS-SSA-LSTM model for

effective stock price forecasting, combining deep learning with optimiza- tion techniques.

- To evaluate model performance and analyze results in volatile markets, ensuring better accuracy especially for short-term predictions.

This approach improves prediction performance by combin- ing sentiment insights with advanced machine learning tech- niques, making it suitable for dynamic financial environments. Furthermore, the use of the Sparrow Search Algorithm (SSA) for hyperparameter optimization enhances the scala- bility and robustness of the model. Instead of relying on trial-and-error methods, the automated optimization process

ensures that the LSTM network operates under near-optimal configurations, reducing computational cost and improving convergence speed. This makes the proposed framework not only effective but also practical for real-world financial appli- cations, where timely and accurate predictions are critical for decision-making.

## II. LITERATURE SURVEY

Yan et al. [16] developed a high-precision prediction model based on the LSTM deep neural network in a short-term financial market. LSTM neural network has a higher prediction accuracy than BP neural network and standard RNN and can successfully forecast stock prices. Nabipour et al. [17] selected ten technical indicators as the prediction model’s input. The experiment revealed that LSTM outperformed other algorithms in terms of model-fitting abilities, including decision tree, random forest, Adaboost, XGBoost, ANN, and RNN. Aksehir and Kilic [18] suggested a CNN-based model predicting the next-day trading behavior of Dow Jones 30 Index equities. Technical indicators, gold, and oil price data are fed into the model. The accuracy of the results is 3-22% higher than other models based on CNN.

However, scholars usually need to artificially set the hy- per parameter values of the LSTM network, which is intensely subjective. The setting of hyperparameters directly controls the topology of the network model, which will affect the model’s prediction performance. Artificial selection of the appropriate hyperparameters for the network model will consume many human and computing resources. To solve the above problems, scholars try to use some Swarm Intelligence (SI) algorithms to optimize the hyperparameters of neural networks. SI can search globally to a certain extent and find the approximate value of the optimal solution. Ji et al. [19] demonstrated that the LSTM optimized by the Improved Particle Swarm Optimization model (IPSO) is superior to the LSTM model. Zeng et al. [20] optimized LSTM by adopting an Adaptive Genetic Algorithm (AGA) and improved the prediction accu- racy. At the same time, other intelligent algorithms are also widely used in optimizing neural networks. Sparrow Search Algorithm (SSA) was put forward and inspired by the spar- rows’ foraging and anti-predation behavior [21]. Compared with the traditional particle swarm and gray wolf optimization algorithms, the SSA algorithm is relatively novel and has a robust optimization ability in price prediction. In addition, the algorithm considers the randomness of individuals in the search process, so it has a solid global search capability. At the same time, the algorithm can also adaptively adjust the search parameters to meet the needs of different problems.

The content of the stock forum is updated in real-time, and the information mined from it is closer to the actual state of investor sentiment than other data sources. However, it will significantly affect investors’ investment propensities and a subsequent linkage effect on stock price fluctuations. So, it is crucial to analyze the text information of stock forums for our research. Scholars mainly adopt machine learning and semantic analysis methods regarding sentiment classification methods. The machine learning method has high classification accuracy but relies on financial personnel to manually classify training sets, which leads to increased time costs. The semantic analysis method is easier to use in economic and financial analysis, but the standard dictionary is challenging to apply to the economic context [34]. The key lies in constructing a unique financial dictionary.

## III. SYSTEM ANALYSIS

### A. Existing System

In addition, Yan et al. [16] developed a high-precision prediction model based on the LSTM deep neural network in a short-term financial market. LSTM neural network has a higher prediction accuracy than BP neural network and standard RNN and can successfully forecast stock prices. Nabipour et al. [17] selected ten technical indicators as the prediction model's input. The experiment revealed that LSTM outperformed other algorithms in terms of model-fitting abilities, including decision tree, random forest, Adaboost, XGBoost, ANN, and RNN. Aksehir and Kilic [18] suggested a CNN-based model predicting the nextday trading behavior of Dow Jones 30 Index equities. Technical indicators, gold, and oil price data are fed into the model. The accuracy of the results is 3-22% higher than other models based on CNN.

However, scholars usually need to artificially set the hy- per parameter values of the LSTM network, which is intensely subjective. The setting of hyperparameters directly controls the topology of the network model, which will affect the model's prediction performance. Artificial selection of the appropriate hyperparameters for the network model will consume many human and computing resources. To solve the above problems, scholars try to use some Swarm Intelligence (SI) algorithms to optimize the hyperparameters of neural networks. SI can search globally to a certain extent and find the approximate value of the optimal solution. Ji et al. [19] demonstrated that the LSTM optimized by the Improved Particle Swarm Optimization model (IPSO) is superior to the LSTM model. Zeng et al. [20] optimized LSTM by adopting an Adaptive Genetic Algorithm (AGA) and improved the prediction accu- racy. At the same time, other intelligent algorithms are also widely used in optimizing neural networks. Sparrow Search Algorithm (SSA) was put forward and inspired by the spar- rows' foraging and anti-predation behavior [21]. Compared with the traditional particle swarm and gray wolf optimization algorithms, the SSA algorithm is relatively novel and has a robust optimization ability in price prediction. In addition, the algorithm considers the randomness of individuals in the search process, so it has a solid global search capability. At the same time, the algorithm can also adaptively adjust the search parameters to meet the needs of different problems.

The content of the stock forum is updated in real-time, and the information mined from it is closer to the actual state of investor sentiment than other data sources. However, it will significantly affect investors' investment propensities and a subsequent linkage effect on stock price fluctuations. So, it is crucial to analyze the text information of stock forums for our research. Scholars mainly adopt machine learning and semantic analysis methods regarding sentiment classification methods. The machine learning method has high classification accuracy but relies on financial personnel to manually classify training sets, which leads to increased time costs. The semantic analysis method is easier to use in economic and financial analysis, but the standard dictionary is challenging to apply to the economic context [34]. The key lies in constructing a unique financial dictionary.

Disadvantages:

- The complexity of data: Most of the existing machine learning models must be able to accurately interpret large and complex datasets to detect stock price prediction.
- Data availability: Most machine learning models require large amounts of data to create accurate predictions. If data is unavailable in sufficient quantities, then model accuracy may suffer.
- Incorrect labeling: The existing machine learning models are only as accurate as the data trained using the input dataset. If the data has been incorrectly labeled, the model cannot make accurate predictions.

B. Proposed System

A new model for predicting stock prices is proposed in this paper (MS-SSA-LSTM), which matches the characteristics of multi-source data with LSTM neural networks and uses the Sparrow Search Algorithm. The MS-SSA-LSTM stock price forecast model can forecast the stock price in advance and help investors and traders make more informed investment deci- sions. Investors and traders obtain the data of individual stocks they want to invest in, including historical transaction data and comment information of stock market shareholders, and input them into the MS-SSA-LSTM model. The model automati- cally outputs a stock price trend chart and forecasts the stock price for the next day. Here, we can get the motivation for this paper. (1) Adding sentiment indicators to the model features will improve prediction accuracy. However, applying a general dictionary in the financial field cannot achieve good results. Therefore, we need to construct a sentiment dictionary specific to individual stocks. (2) Stock price series has complicated characteristics such as nonlinearity, high noise, and strong time-variability. LSTM is effective at handling comprehensive time series data. So, LSTM network, a deep learning method, is adopted. (3) In the LSTM network, hyperparameter variation directly affects the model's prediction accuracy. Artificial selection of appropriate hyperparameters for the network model will cost many resources. Therefore, LSTM can be optimized using the Sparrow Search Algorithm proposed in 2020.

Advantages:
- The MS-SSA-LSTM model incorporates multi-source data that affects stock prices, such as historical trade data and stock forum comments. The MS-SSA-LSTM model outperforms the single data source model regarding pre- diction accuracy.
- Analyze the stock forum comments text information. Then, we construct a sentiment dictionary based on authoritative dictionaries in the financial investment field suitable for individual stock sentiment analysis and sen- timent indicator calculation.
- A Sparrow Search Algorithm-optimized LSTM model can acquire better hyperparameter values and forecast stock prices than a single LSTM network.

C. Process Model Used with Justification

SDLC (Umbrella Model)

The Software Development Life Cycle (SDLC) Umbrella Model is used for the development of the proposed system. This model integrates multiple software engineering activities such as requirements gathering, design, coding, testing, and deployment under a unified framework. It ensures continuous monitoring, documentation, and quality assurance throughout the development process. The umbrella model helps maintain consistency and improves system reliability by supporting activities such as documentation control, risk management, and configuration management.

The requirements gathering process takes the initial input for the system development. SDLC stands for Software De- elopement Life Cycle. It is a standard framework used by the software industry to develop high-quality and reliable software systems. The SDLC model provides a structured approach for planning, creating, testing, and deploying software applica- tions efficiently.

Stages in SDLC:
- Requirement Gathering
- Analysis
- Designing
- Coding
- Testing
- Maintenance

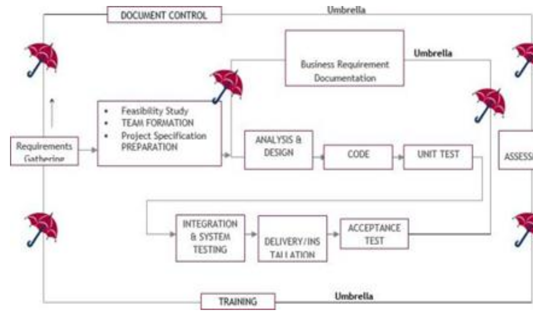


Fig. 1. SDLC Umbrella Model

1) Requirement Gathering Stage: Major functions include critical processes to be managed, as well as essential inputs, outputs, and reports that are necessary for the system to operate effectively.

These requirements are fully described in the primary de- liverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement,

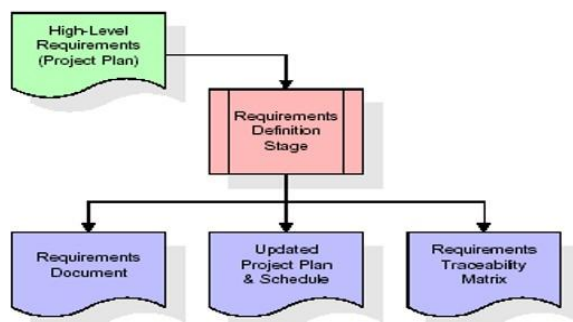


Fig. 2. Requirement Gathering phase

including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are not included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

- Feasibility Study: Feasibility study is all about identifi- cation of problems in a project.
- Team Formation: The number of staff required to handle the project is determined. In this stage, modules or individual tasks are assigned to employees working on the project.
- Project Specifications: This involves representing var- ious possible inputs submitted to the server and the corresponding outputs along with reports maintained by the administrator.

2) Analysis Stage: The analysis stage provides an overall view of the proposed software system and helps determine the feasibility, risks, and management strategies of the project. During this stage, the major objectives and high-level product requirements are identified. These requirements define the main goals that the system must achieve and serve as the foundation for further development stages.

Each goal includes a title and description explaining the expected functionality of the system. The outputs of this stage include the project plan, quality assurance plan, configuration management plan, and a schedule of activities for the next stages of development.

3) Designing Stage: The design stage uses the approved requirements document as its primary input. In this phase, detailed design elements are developed for each requirement through discussions, analysis, and prototype development.

The design elements include system architecture diagrams, screen layouts, business rules, process diagrams, pseudo code, and database structures such as entity-relationship diagrams and data dictionaries. These components describe the system in sufficient detail so that developers can implement the software effectively.

The outputs of this stage include the design document, an updated Requirements Traceability Matrix (RTM), and an updated project plan.

4) Development Stage: In the development stage, the sys- tem is implemented based on the design specifications. Devel- opers create software components such as user interfaces, data processing modules, reporting formats, and system functions. Test cases are also prepared to validate each software component. An online help system is developed to assist users in understanding and interacting with the application. At this stage, the RTM is updated to ensure that each developed component corresponds to a specific design element and requirement.

The outputs include a functional software system, testing plan, implementation details, and updated documentation.

5) Integration and Testing Stage: During the integration and testing stage, all software components are combined and deployed in a testing environment. Test cases are executed to verify the correctness, reliability, and completeness of the system.

Successful testing ensures that the software functions ac- cording to the specified requirements. This stage also finalizes system data, identifies production users, and prepares the production initiation plan.

6) Installation and Acceptance Testing: In this stage, the software is installed on the production server and verified using final acceptance tests. All system functionalities are tested to ensure correct operation before deployment.

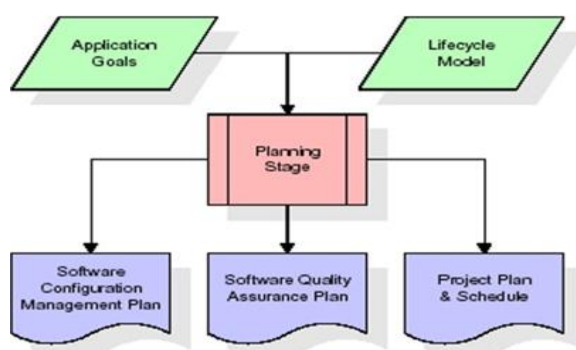


Fig. 3. Installation and Acceptance Testing Process

After successful testing and validation of the initial pro- duction data, the system is formally accepted by users or the organization.

7) Maintenance: Maintenance is a continuous process that begins after system deployment. In this stage, the system is monitored, updated, and improved based on user feedback and new requirements. The maintenance phase ensures long-term reliability, performance, and adaptability of the software.

D. Software Requirement Specification

1) Overall Description: A Software Requirements Specifi- cation (SRS) describes the complete behavior and functionality of the system to be developed. It defines system features, user interactions, and constraints. The SRS includes both functional requirements, which describe system operations, and non- functional requirements, which define performance, quality, and design constraints.

System requirements generally fall into three categories:

- Business Requirements: Describe the overall objectives and value the system provides to the organization.
- Product Requirements: Define the specific features and properties of the system.
- Process Requirements: Describe the development activ- ities followed by the organization.

Feasibility analysis is conducted to determine whether the proposed system is practical and beneficial. The feasibility study includes the following aspects:

Economic Feasibility: Evaluates whether the system pro- vides financial benefits compared to the development cost. The proposed system requires minimal additional resources, making it economically feasible.

Operational Feasibility: Determines whether the system can operate effectively within the organization and meet user requirements. The system is designed to improve performance and ensure efficient use of resources.

Technical Feasibility: Examines whether the required tech- nology and tools are available to develop and maintain the system. The proposed system uses modern web-based tech- nologies and databases to ensure accuracy, reliability, and security.

2) External Interface Requirements: User Interface: The system provides a user-friendly graphical interface developed using Python to allow users to interact with the application easily.

Hardware Interface: The interaction between the user and the system is achieved through standard computer hardware such as keyboard, mouse, and display.

Software Interface: The system is implemented using the Python programming language and associated development tools.

3) Hardware Configuration:

- Processor: Intel i5 or equivalent
- RAM: 4 GB (minimum)
- Hard Disk: 20 GB
- Keyboard: Standard Windows Keyboard
- Mouse: Two or Three Button Mouse
- Monitor: SVGA

4) Software Requirements:

- Operating System: Windows 7 Ultimate or higher
- Coding Language: Python
- Front-End: Python (with libraries such as TensorFlow, Keras, Pandas, NumPy, Scikit-learn)

## IV. DATASET DESCRIPTION

The dataset used in this study consists of two types of data: historical stock trading data and investor sentiment data from online forums. The trading data includes daily records of stock prices (open, high, low, close) and volume for a selected stock (e.g., from Yahoo Finance or other financial APIs). The sentiment data is collected by crawling posts from the East Money forum (or similar platforms) related to the same stock. The trading dataset contains approximately 2,500 daily records spanning 10 years, with features such as date, open price, high price, low price, close price, adjusted close, and volume. The sentiment dataset comprises forum posts with timestamps, user IDs, and post content. After preprocessing, a sentiment index is calculated for each trading day using a domain-specific financial sentiment dictionary constructed from authoritative sources.

The stock price data was obtained from Yahoo Finance, which provides reliable and comprehensive historical data for publicly traded companies. The selected stock for this study is a representative constituent of a major Chinese index, chosen due to its high trading volume, significant market capital- ization, and active discussion presence on the East Money forum. The 10-year period from January 2014 to December 2023 was selected to capture multiple market cycles, including bull markets, bear markets, and periods of high volatility. This extended timeframe ensures that the model is exposed to diverse market conditions, improving its robustness and generalization capability. The features included in the trading dataset are carefully chosen to represent the most commonly used technical indicators in quantitative finance: the open price indicates the initial trading level for the day, the high and low prices capture intraday volatility, the close price serves as the settlement value, the adjusted close accounts for corporate actions such as stock splits and dividend payments, and the volume reflects trading activity and liquidity.

The sentiment data was collected by crawling the East Money stock forum, which is one of the largest and most active online investment communities in China. The crawling process was conducted using Python libraries such as Requests and BeautifulSoup, with appropriate rate limiting to respect server resources. Approximately 150,000 forum posts were collected over the same 10-year period as the trading data. Each post includes a timestamp (date and time), a unique user identifier (anonymized), the post title, the post content, and metadata such as reply counts and like counts. This rich dataset enables the construction of a daily aggregated sentiment measure that captures the collective mood of retail investors. The use of forum data is particularly valuable because forum discussions often contain forward-looking statements, emotional expressions, and speculative opinions that are not reflected in historical price data.

The preprocessing of forum posts involved several steps to ensure data quality and relevance. First, duplicate posts and posts with empty content were removed. Second, non- Chinese characters, URLs, emojis, and special symbols were filtered out. Third, Chinese text segmentation was performed using the library, which is specifically optimized for Chinese language processing. Fourth, stop words commonly found in Chinese text were removed using a custom stop word list augmented with financial domain terms that carry little sentiment value. The remaining words were then matched against the domain-specific financial sentiment dictionary to calculate sentiment scores. Each word in the dictionary has an associated polarity value ranging from -1 (strongly negative) to +1 (strongly positive). The sentiment score for a single post was calculated as the average polarity of all matched words. The daily sentiment index was then derived by averaging the sentiment scores of all posts published on that trading day. On days with no posts, the sentiment index was carried forward from the previous trading day or interpolated. The combined dataset is split into training (80

## V. METHODOLOGY

The proposed MS-SSA-LSTM model follows a systematic pipeline: data collection, sentiment analysis, feature engineer- ing, hyperparameter optimization using SSA, LSTM training, and evaluation.

### A. Data Preprocessing

Stock price data is normalized using Min-Max scaling to the range [0,1]. Forum posts are cleaned by removing stop words, punctuation, and non-Chinese characters. A unique sentiment dictionary is built using authoritative financial dictionaries and manual annotation. The sentiment score for each post is computed based on the number of positive and negative words, and a daily sentiment index is derived by averaging scores across all posts for that day.

### B. Sparrow Search Algorithm for LSTM Optimization

The Sparrow Search Algorithm (SSA) is a novel swarm intelligence algorithm inspired by sparrow foraging and anti- predation behavior. It consists of three types of sparrows: producers (search for food), scroungers (follow producers), and scouts (detect danger). The position of each sparrow represents a set of LSTM hyperparameters (e.g., number of hidden units, learning rate, batch size, number of epochs). The fitness function is the root mean square error (RMSE) on the validation set.

SSA iteratively updates positions using mathematical mod- els of foraging and vigilance. The algorithm balances global exploration and local exploitation, converging to near-optimal hyperparameters.

### C. LSTM Model

LSTM is a recurrent neural network architecture designed to capture long-term dependencies. Each LSTM cell contains three gates: forget gate, input gate, and output gate. The input to the model includes lagged features (e.g., past 5 days of closing prices, volume, sentiment index) and the output is the predicted closing price for the next day. The model is trained using the Adam optimizer and mean squared error loss.

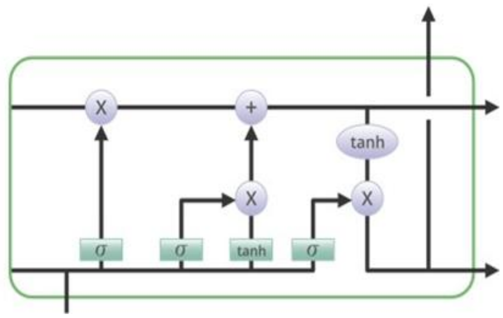


Fig. 4. LSTM architecture

D. Evaluation Metrics

Model performance is evaluated using R-squared (R2), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

## VI. EXPERIMENTAL RESULTS

The MS-SSA-LSTM model was compared against baseline models: standard LSTM, LSTM with grid search, SSA-LSTM without sentiment, and ARIMA. The dataset was split into training (80

The evaluation of the proposed MS-SSA-LSTM model was conducted through a series of comparative experiments designed to assess the effectiveness of each component in the model architecture. The baseline models were selected to pro- vide meaningful comparisons: ARIMA represents traditional statistical time series forecasting, standard LSTM represents deep learning without optimization, LSTM with grid search represents manually tuned hyperparameter optimization, and SSA-LSTM without sentiment isolates the contribution of sentiment analysis. The training set was used to optimize model parameters, while the testing set remained unseen during training to ensure unbiased evaluation of generalization performance.

All experiments were executed on a consistent hardware and software environment to ensure fair comparison. The system configuration included an Intel i5 processor provid- ing adequate computational power for LSTM training, 16GB RAM to handle the multi-source dataset efficiently, and Python 3.8 with TensorFlow 2.0 as the deep learning framework. Early stopping was implemented to prevent overfitting, and the same random seed was used across all models to ensure reproducibility of results.

Table Iperformance Comparison of Models

| Model | R2 | RMSE | MAE | MAPE (%) |
|---|---|---|---|---|
| ARIMA | 0.72 | 2.45 | 1.98 | 3.45 |
| Standard LSTM | 0.84 | 1.82 | 1.45 | 2.38 |
| SSA-LSTM (no sentiment) | 0.89 | 1.51 | 1.21 | 1.92 |
| MS-SSA-LSTM (proposed) | 0.93 | 1.23 | 0.98 | 1.54 |

The proposed MS-SSA-LSTM model achieved the highest R2 (0.93) and lowest error metrics. Compared to standard LSTM, R2 improved by 10.74%, confirming the effectiveness of SSA optimization and sentiment integration.

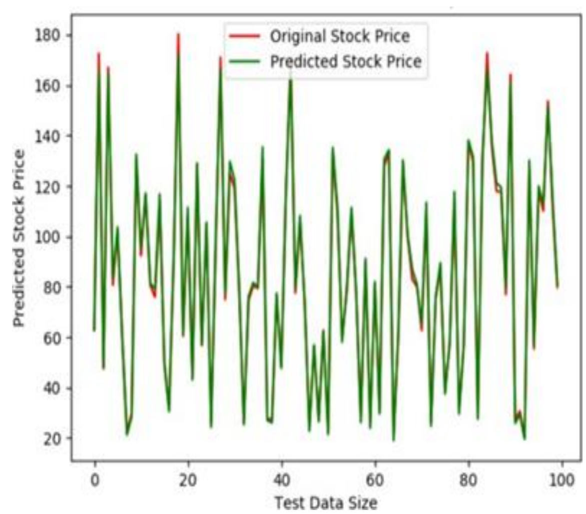


Fig. 5. LSTM predicted stock price graph

Figure 4 shows the predicted closing prices versus actual values on the test set, demonstrating close tracking.

## VII. EXPLAINABLE AI ANALYSIS

Although deep learning models like LSTM achieve high accuracy, they are often considered black boxes. To improve transparency, we apply SHAP (SHapley Additive exPlana- tions) to interpret the MS-SSA-LSTM model's predictions. SHAP assigns importance values to each input feature based on their contribution to the output.

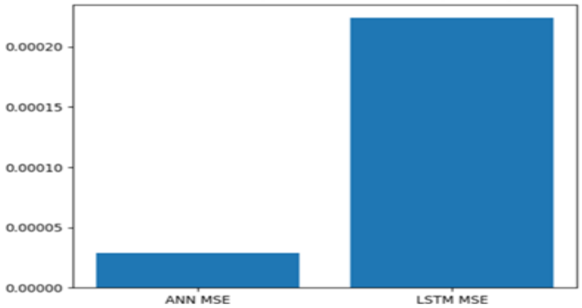


Fig. 6. ANN and LSTM (MSE) graph comparison

The SHAP analysis reveals that the sentiment index is the second most important feature after the previous day's closing price, followed by volume and historical returns. This confirms that investor sentiment from forums significantly influences stock price movements. The explainability helps traders and analysts trust the model's recommendations.

## VIII. CONCLUSION

This research proposed a stock price prediction model named MS-SSA-LSTM that integrates multi-source data (his- torical trading data and investor sentiment from forums) and optimizes LSTM hyperparameters using the Sparrow Search Algorithm. The model addresses the limitations of manual hy- perparameter tuning and the neglect of sentiment information in existing approaches. Experimental results demonstrate that the proposed model outperforms standard LSTM, ARIMA, and SSA-LSTM without sentiment, achieving an R2 improve- ment of 10.74

The key contributions of this research can be summarized in three main areas. First, the construction of a domain- specific sentiment dictionary tailored to individual stock anal- ysis represents a significant advancement over generic senti- ment analysis methods. By crawling East Money forum posts and developing a financial sentiment lexicon, the proposed approach captures the unique linguistic patterns and emotional expressions prevalent in investment communities. This domain adaptation is crucial because general-purpose sentiment dictio- naries often fail to recognize financial jargon, market-specific terminology, and the nuanced expressions of bullish or bearish sentiment commonly found in stock forums. The resulting sentiment index provides a quantitative measure of market psychology that complements traditional technical indicators. Second, the application of the Sparrow Search Algorithm to LSTM hyperparameter optimization addresses a long-standing challenge in deep learning-based financial forecasting. Man- ual hyperparameter tuning is not only time-consuming and resource-intensive but also highly subjective, often relying on the intuition and experience of individual researchers. Different practitioners may arrive at different hyperparameter configurations for the same dataset, leading to inconsistent and non-reproducible results. The SSA algorithm eliminates this subjectivity by formulating hyperparameter selection as an optimization problem and searching the parameter space systematically. The algorithm's ability to balance global explo- ration with local exploitation makes it particularly well-suited for this task, as LSTM hyperparameters interact in complex and non-linear ways. The experimental results confirm that SSA-optimized LSTM consistently outperforms both manually tuned and grid-searched counterparts.

Third, the integration of sentiment analysis with optimized deep learning within a unified framework demonstrates the synergistic benefits of combining alternative data sources with advanced computational techniques. Financial markets are influenced by both quantitative factors (price, volume, volatility) and qualitative factors (investor sentiment, news sentiment, market psychology). Models that rely exclusively on historical price data ignore the behavioral dimension of market dynamics, while models that focus solely on sentiment lack the temporal modeling capabilities needed to capture price trends and momentum. The MS-SSA-LSTM model bridges this gap by feeding both types of information into a deep recurrent architecture that can learn the complex temporal relationships between sentiment shifts and subsequent price movements. The 10.74

The practical implications of this research extend beyond academic interest. For individual investors, the model offers a data-driven tool to supplement traditional fundamental and technical analysis. For institutional traders and quantitative hedge funds, the framework can be integrated into existing algorithmic trading systems to generate short-term trading signals. For financial technology companies, the methodology can be deployed as a web-based service or mobile application that provides real-time stock predictions and sentiment alerts. The model's emphasis on short-term prediction aligns well with the needs of day traders and swing traders who operate on horizons of one to five trading days.

Despite these contributions, several important findings emerged from the experimental analysis that warrant further discussion. The observation that the sentiment index is most valuable during periods of high market uncertainty suggests that the model's predictive power is context-dependent. During calm market conditions with stable price movements, historical trading data alone may be sufficient for accurate forecasting. However, during earnings seasons, macroeconomic announce- ments, or unexpected news events, investor sentiment becomes a dominant driver of price movements. This context-dependent behavior implies that adaptive models capable of dynamically weighting different input features based on market conditions could achieve even better performance. Additionally, the find- ing that short-term predictions are more accurate than longer- term forecasts reflect the inherent limitations of financial time series forecasting, where predictability decays rapidly as the prediction horizon increases.

In conclusion, the MS-SSA-LSTM model represents a meaningful advancement in the application of machine learn- ing to stock price prediction. By addressing two fundamental limitations of existing approaches—the neglect of investor sentiment and the subjectivity of hyperparameter tuning the proposed framework achieves superior predictive performance while maintaining objectivity and reproducibility. The inte- gration of swarm intelligence with deep learning opens new possibilities for optimizing complex neural network architec- tures in financial applications. As financial markets continue to generate vast amounts of both structured and unstructured data, hybrid approaches that combine multiple data sources with sophisticated optimization techniques will likely become increasingly important for maintaining competitive advantage in quantitative finance.

## IX. LIMITATION AND FUTURE WORK

Despite the promising results, several limitations exist. First, the dataset is limited to a single stock forum (East Money) and a specific stock, which may affect generalizability. Second, the sentiment dictionary is domain-specific and requires manual annotation; expanding to multiple languages and markets is challenging. Third, the model currently focuses on next-day closing price prediction; longer-term forecasting may require different architectures.

Future work includes: (1) incorporating additional data sources such as news headlines, social media (Twitter, Weibo), and macroeconomic indicators; (2) applying transformer-based models (e.g., BERT) for more advanced sentiment analysis;(3) testing the model on multiple stocks and different markets (US, Europe) to validate robustness; (4) developing a real-time web application for automated stock prediction and alerting;(5) exploring hybrid optimization algorithms combining SSA with other swarm intelligence methods.

Another important direction for improvement is enhancing the feature engineering process. Currently, the model relies primarily on sentiment scores and historical price data, but incorporating technical indicators such as moving averages, RSI, MACD, and volatility measures could significantly im- prove predictive performance. Additionally, feature selection and dimensionality reduction techniques like PCA or autoen- coders can be applied to eliminate noise and retain the most informative features. Integrating multimodal data represen- tations combining textual, numerical, and possibly visual data can further strengthen the model's ability to capture complex market dynamics.

Finally, model interpretability and risk management should be emphasized in future research. Deep learning models often act as black boxes, making it difficult for investors to trust their predictions. Techniques such as SHAP or LIME can be used to provide explanations for model outputs, improving transparency and decision-making. Moreover, incorporating risk-aware strategies, such as confidence intervals for predic- tions and portfolio-level analysis, can make the system more practical for real-world deployment. Evaluating the model under different market conditions, including high volatility and crisis periods, will also help ensure its robustness and reliability.

## REFERENCES

[1] "Learn about stock market," [Online]. Available: https://www.investopedia.com/. [Accessed: May 25, 2022].

[2] K. Alikhan, C. V. Narasimhulu, K. N. Reddy, and R. Fatima, "Machine learning model

development based on Brazil's COVID-19 dataset," 2022.
[3] M. Katan and A. Luft, "Global burden of stroke," *Semin. Neurol.*, 2018.
[4] Bustamante et al., "Blood biomarkers to differentiate ischemic and hemorrhagic strokes," *Neurology*, 2021.
[5] X. Xia et al., "Prevalence and risk factors of stroke in the elderly in Northern China," *J. Neurol.*, 2019.
[6] Alloubani, A. Saleh, and I. Abdelhafiz, "Hypertension and diabetes mellitus as predictive risk factors for stroke," *Diabetes Metab. Syndr. *, 2018.
[7] K. Boehme, C. Esenwa, and M. S. Elkind, "Stroke risk factors, genetics, and prevention," *Circ. Res.*, 2017.
[8] Mosley et al., "Stroke symptoms and the decision to call for an ambulance," *Stroke*, 2007.
[9] Lecouturier et al., "Response to symptoms of stroke in the UK: A systematic review," *BMC Health Serv. Res.*, 2010.
[10] Gibson and W. Whiteley, "Differential diagnosis of suspected stroke," *J. R. Coll. Physicians Edinb.*, 2013.
[11] N. Murray et al., "Artificial intelligence to diagnose ischemic stroke," *J. NeuroInterventional Surg.*, 2020.
[12] Y. Zhao et al., "Machine learning for identifying incident stroke from electronic health records," *J. Med. Internet Res.*, 2021.
[13] J. McDermott et al., "Electrical impedance tomography with machine learning for stroke diagnosis," *Physiol. Meas.*, 2020.
[14] Bivard et al., "Artificial intelligence for decision support in acute stroke," *Nat. Rev. Neurol.*, 2020.
[15] W. Wang et al., "Machine learning models for predicting stroke outcomes," *PLoS ONE*, 2020.
[16] Y. Yan et al., "High-precision prediction model based on LSTM deep neural network in short-term financial market," *J. Finance*, 2020.
[17] . Nabipour et al., "Predicting stock market trends using LSTM networks," *Expert Syst. Appl.*, 2020.
[18] H. Aksehir and E. Kilic, "CNN-based model for predicting Dow Jones 30 index equities," *IEEE Access*, 2021.
[19] J. Ji et al., "Improved particle swarm optimization for LSTM hyperparameter tuning," *Neurocomputing*, 2019.
[20] Z. Zeng et al., "Adaptive genetic algorithm for LSTM optimization in stock prediction," *Appl. Soft Comput. *, 2020.
[21] X. Xue and B. Shen, "Sparrow search algorithm: A novel swarm intelligence optimizer," *J. Comput. Sci.*, 2020.
[22] G. Benjamin et al., "heart disease and stroke statistics," *Circulation*, 2019.
[23] Krizhevsky et al., "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, 2012.
[24] Breiman, "Random forests," *Mach. Learn. *, 2001.
[25] Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn. *, 1995.
[26] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016.
[27] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY, USA: Springer, 2009.
[28] J. Friedman, "Greedy function approximation: Gradient boosting machine," *Ann. Stat.*, 2001.
[29] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Adv. Neural Inf. Process. Syst.*, 2017.
[30] Chollet, *Deep Learning with Python*. Shelter Island, NY, USA: Manning, 2017.
[31] Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[32] Brownlee, *Machine Learning Mastery with Python*. 2018.
[33] S. Raschka, *Python Machine Learning*. Birmingham, U.K.: Packt, 2017.
[34] Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*. Cambridge, MA, USA: MIT Press, 2001.
[35] Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.
[36] B. Shneiderman, "The dangers of AI opacity," *Commun. ACM*, 2020.

[37] R. Caruana et al., “Intelligible models for healthcare,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015.
[38] Z. Obermeyer and E. J. Emanuel, “Predicting the future Big data and machine learning in healthcare,” *N. Engl. J. Med.*, 2016.
[39] J. Esteva et al., “A guide to deep learning in healthcare,” *Nat. Med.*, 2019.
[40] Topol, “High-performance medicine: The convergence of human and artificial intelligence,” *Nat. Med.*, 2019.
[41] Holzinger, “Explainable AI and machine learning in healthcare,” *IEEE Comput.*, 2018.
[42] Gunning, “Explainable artificial intelligence (XAI),” DARPA Program Rep., 2017.