ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

Optimised Deep Learning using Investor Sentiment for Stock Price Prediction

U D S V PRASAD 1, AMARAPALLI DEVI SAI PRASANNA 2, KUDUPUDI AKHIL 3, GEDDAM GOWRIDHAR 4, LOKESH KURMAPU 5

#1Assistant Professor in Department of Computer Science and Engineering, Sasi Institute Of Technology & Engineering, Tadepalligudem.

#2#3#4#5 B.Tech with Specialization of Computer Science and Engineering, Sasi Institute Of Technology & Engineering, Tadepalligudem.

Abstract: Stock price projections are enhanced by the MS-SSA-LSTM model through the use of deep learning, swarm intelligence approaches, sentiment analysis, and multi-source data. A sentiment lexicon and index are generated by this algorithm using East Money forum entries. This offers valuable insights into market sentiment's influence on stock prices. The Sparrow Search Algorithm (SSA) is used to fine-tune LSTM hyperparameters, optimizing prediction accuracy. • Experimental results showcase the MS-SSA-LSTM model's superior performance. It's a valuable tool for accurate stock price predictions. Tailored for China's volatile financial market, the model excels in short-term stock price predictions, offering insights for dynamic decision-making by investors. And also, a hybrid LSTM+GRU model was introduced for stock sentiment classification. Additionally, a robust ensemble strategy was adopted, incorporating a Voting Classifier (AdaBoost + RandomForest) for sentiment analysis and a Voting Regressor (LinearRegression + RandomForestRegressor + KNeighborsRegressor) for stock price prediction. These ensembles seamlessly integrated with existing models (MLP, CNN, LSTM, MS-LSTM, MS-SSA-LSTM), collectively enhancing overall predictive performance. To facilitate user interaction and testing, a user-friendly Flask framework with SQLite support was developed, streamlining signup, signin, and model evaluation processes.

Index terms - *Sparrow search algorithm, DL, stock price prediction, LSTM, sentiment analysis, sentiment dictionary.*

1. INTRODUCTION

Many individuals are realising the importance of investing and joining the financial sector as a result of the maturing stock market in China and the fast growth of online finance. Lots of data and a lot of volatility make up the stock market. Having data-mining skills is generally necessary for retail investors to be successful. Thus, accurate stock price forecasting lowers investment risks and increases profits for companies and investors alike.

The first scholars to use statistical tools to analyse stock price time series trends fitted the data with a linear model. Examples of traditional techniques are ARMA, ARIMA, GARCH, and others. Time series stocks are examined using ARMA [1]. Stock price forecasts are made possible by analysts using the ARIMA model, which is based on ARMA [2]. Fitting the Shanghai Composite Index using ARIMA is improved by wavelet analysis [3]. For stock time series, the GARCH model provides new temporal frame predictions [4]. Volumetric price analysis of multivariate stocks was done by other scholars using ARMA and GARCH [5]. The majority of traditional methods only record organised data. Unconventional assumptions are necessary for traditional forecasting. Financial data that is not linear is notoriously difficult to understand using statistical methods.

Predicting stock prices is a common academic activity that makes use of Support Vector Machines and Neural Networks. When it comes to data parsing, learning, and prediction, machine learning is all about algorithms. The SVM's strength in handling small samples, high-dimensional data, and nonlinear situations makes it a popular choice among academics for stock forecasting. According to Hossain and Nasser [6], SVM outperforms statistical approaches when it comes to stock prediction. In order to forecast changes in the HS300 index, Chai et al. [7] created a hybrid support vector machine (SVM) model and demonstrated that the combination of least squares SVM and Genetic Algorithm (GA) yielded superior results. It's possible that SVMs can't handle massive volumes of stock data forecasts when trained with huge training samples due to the high memory and processing requirements. Then, problems with financial time series may be solved using ANN and multi-layer ANN. Fast convergence and excellent accuracy are characteristics of ANN, according to experimental data [8, 9, 10]. The number of feedforward artificial neural networks that were able to forecast stock market values was tested by Moghaddam and Esfandyari [11]. By implementing Bayesian regularisation, Liu and Hou [12] enhanced the BP neural network. Nevertheless, there are instances where traditional neural networks may be enhanced. The results of overfitting and local optimisation are caused by poor generalisation. Since a large number of samples need to be taught in order to resolve these issues, better models are required.

The MS-SSA-LSTM stock price prediction model is presented in this research. Using the Sparrow Search Algorithm and Long Short-Term Memory (LSTM) neural networks, it finds matches in several datasets. Early stock price projections are provided by the MS-SSA-LSTM model to investors and traders. Investors and traders feed the MS-SSA-LSTM model stock data, which includes shareholder feedback and transaction history. A trend chart and projection for the following day's stock price are automatically generated by the application.

2. LITERATURE SURVEY

a) Investigation of market efficiency and Financial Stability between S&P 500 and London Stock Exchange: Monthly and yearly Forecasting of Time Series Stock Returns using ARMA model.

https://www.sciencedirect.com/science/article/abs/pii/S0378437116002776

The ARMA model was utilised to investigate the volatility dynamics and long-term memory characteristics of the S&P 500 and the London Stock Exchange. The complexity of the financial markets, which efficient market theory fails to describe, has recently made multifractal analysis a crucial tool for understanding these markets. Price returns in the financial markets are assumed to be serially uncorrelated according to the weak efficient market hypothesis. Random price fluctuations are desirable. We contrast the random walk hypothesis with alternatives based on unifractality and multifractality. It has been found in several studies that stock return volatility exhibits long-range reliance, heavy tails,

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

and clustering. It is recommended to include self-similar stochastic processes in return volatility models due to their large tails and long-range reliance. In terms of using real values to predict medium- or long-term values, statistical investigation shows that the S&P 500 ARMA model outperforms the London stock exchange. According to statistical analysis conducted by the London Stock Exchange, the monthly ARMA model yields better results than the yearly model. The London Stock Exchange and the S&P 500 both show fiscal stability and efficiency in both prosperous and bad economic periods.

b) Gold Price Forecasting Using ARIMA Model

https://www.researchgate.net/publication/322224066_Gold_Price_Forecasting_Using_ARIMA_Model

For anyone looking to hedge their gold bets between 2003 and 2014, this study demonstrates how the ARIMA time series model can do just that. Helping those involved in the buying and sale of yellow metal. Due to growing inflation, shifting political circumstances, and worldwide data, researchers, investors, and speculators are seeking financial solutions to diversify their portfolios and reduce risk. It is vital to predict the price of gold because it is now highly valued by investors and was formerly only purchased for ceremonial purposes in India.

c) Analysis and Prediction of Shanghai Composite Index by ARIMA Model Based on Wavelet Analysis

https://www.semanticscholar.org/paper/Analysis-and-Prediction-of-Shanghai-Composite-Index-Hong-ya/4786aef4d7de870f79896d79db8f44e719950599

Predicting stock prices draws a lot of attention, even though there are various approaches for making forecasts.Nevertheless, problems like as local minimums and inaccurate forecasts are prevalent.In order to improve the precision of tracking stock prices.The use of wavelet analysis in stock price forecasting leads to the creation of a modified ARIMA model.With this method, we can examine the average closing price of the Shanghai Composite Index on a monthly basis.Examined the outcomes of several prediction methods.Findings show that the suggested method is effective.

d) Stock return prediction under GARCH — An empirical assessment

https://www.researchgate.net/publication/316345159_Stock_return_prediction_under_GARCH_-_An_empirical_assessment

The numerous iterations of GARCH have widespread use in the field of finance. Assuming a strong GARCH distribution with a mean of zero and a variance of one, quasi maximum likelihood estimation treats modifications to GARCH processes as though they were identically distributed and independently distributed. If we relax our assumptions to weak GARCH and no unconditional correlation, we may leverage higher-order dependency patterns to predict GARCH innovations and stock returns in advance. In this article, the author uses rolling windows of empirical stock returns to assess the independence of successive GARCH innovations. The independence test's rolling -values show the time-varying serial dependency and potentially foretell future stock price changes. Nonparametric innovation predictions demonstrate ex ante forecasting benefits when combined with independence diagnostics (-values) and/or linear return projections.

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

e) International evidence on crude oil price dynamics: Applications of ARIMA-GARCH models

https://www.sciencedirect.com/science/article/abs/pii/S0140988310000654

From 1/2/1997 to 10/3/2009, we analysed eleven global markets' weekly crude oil spot prices and predicted their conditional mean and volatility using several ARIMA-GARCH models. More especially, we look at the January–October 2009 out-of-sample forecasting results of four volatility models: GARCH, EGARCH, APARCH, and FIGARCH. Despite the APARCH model's generally good performance, the predicting results are all over the place. Compared to conditional variance, conditional standard deviation is a more accurate measure of the volatility of oil returns. Lastly, in contrast to the hyperbolic speed of the FIGARCH alternative, shocks to conditional volatility decrease exponentially, in line with covariance-stationary GARCH models.

3. METHODOLOGY

i) Proposed Work:

The project introduces the MS-SSA-LSTM model, a cutting-edge system for stock price prediction. This model seamlessly integrates multi-source data, sentiment analysis, and swarm intelligence algorithms. [14,15,16,30] By optimizing LSTM hyperparameters with the Sparrow Search Algorithm, the system excels in forecasting stock prices with exceptional accuracy. Experimental results affirm its superiority over other models, underlining its universal applicability and potential to enhance predictive performance. This model is compared with MLP, CNN, LSTM, MS-LSTM. And also, a hybrid LSTM+GRU model was introduced for stock sentiment classification. Additionally, a robust ensemble strategy was adopted, incorporating a Voting Classifier (AdaBoost + RandomForest) for sentiment analysis and a Voting Regressor (LinearRegression + RandomForestRegressor + KNeighborsRegressor) for stock price prediction. These ensembles seamlessly integrated with existing models (MLP, CNN, LSTM, MS-LSTM, MS-SSA-LSTM), collectively enhancing overall predictive performance. To facilitate user interaction and testing, a user-friendly Flask framework with SQLite support was developed, streamlining signup, signin, and model evaluation processes.

ii) System Architecture:

The initial step is to import datasets, including the Stock Tweets Dataset, Single Stock Data, and Multi-Source Data. These datasets serve as the foundation for both sentiment analysis and stock price prediction. Text data from the Stock Tweets Dataset undergoes cleaning, which includes removing punctuations, HTML tags, URLs, and emojis. This step ensures the text is ready for sentiment analysis. The Single Stock Data and Multi-Source Data are processed to handle null values, remove duplicates, and scale the data. This prepares the financial data for stock price prediction. Several models, including MLP, CNN, LSTM, MS-LSTM, MS-SSA-LSTM, extensions- Voting Classifier, and LSTM + GRU, are trained for sentiment classification. They analyze the cleaned tweet data to determine market sentiment. Another set of models, including MLP, CNN, LSTM, MS-LSTM, MS-SSA-LSTM, and extension- Voting Regression, are trained for stock price prediction. They utilize processed financial data to forecast stock prices. After the models are trained, they are used to make predictions. In the case of sentiment analysis,

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

predictions provide insights into market sentiment. For stock price prediction, the models forecast future stock prices. The predictions from sentiment analysis and stock price models play a crucial role in aiding investors and traders in making informed decisions. The combined results help users navigate the complex landscape of the stock market, reduce risks, and optimize investment returns.

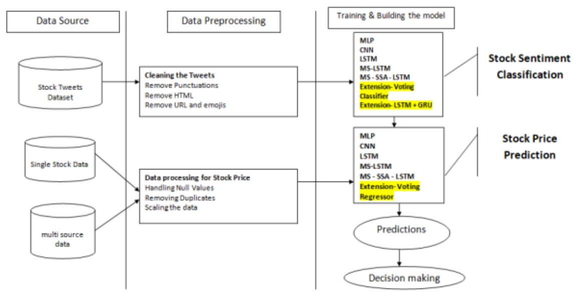

Fig 1 Proposed architecture

iii) Dataset collection:

STOCK TWEETS DATASET

The "Stock Tweets" dataset includes posts about stocks and financial markets on social media. We used it to understand people's feelings and reactions to market news [1,4,7,8]. This helped us create tools for stock trading and investments. We wanted to see how social media affects stock prices and market trends to help investors and traders.

So, these are the top 5 rows of the dataset

	Text	Sentiment
0	Kickers on my watchlist XIDE TIT SOQ PNK CPW B...	1
1	user: AAP MOVIE. 55% return for the FEA/GEED i...	1
2	user I'd be afraid to short AMZN - they are lo...	1
3	MNTA Over 12.00	1
4	OI Over 21.37	1

Fig 2 Stock tweets dataset

ALL STOCK DATASET

The "All Stock Dataset" is a comprehensive collection of financial data from various sources. It provides a wealth of information for in-depth stock market research. In our project, we used this dataset to enhance our stock price prediction model. We aimed to improve the accuracy of stock price forecasts by leveraging diverse data sources, ultimately benefitting investors and businesses.

THIS IS THE SAMPLE DATASET

Date	Open	High	Low	Close	Volume
2012-01-03	325.25	332.83	324.97	663.59	7,380,500
2012-01-04	331.27	333.87	329.08	666.45	5,749,400
2012-01-05	329.83	330.75	326.89	657.21	6,590,300
2012-01-06	328.34	328.77	323.68	648.24	5,405,900
2012-01-09	322.04	322.29	309.46	620.76	11,688,800

Fig 3 All stock dataset

iv) Data Processing:

Data processing transforms unstructured data into actionable insights for businesses. Graphs or papers can be arranged by data scientists once they have gathered, organised, cleaned, verified, and analysed the data. Data processing can be done mechanically, electronically, or by hand. Data should be more useful, and choices should be less complicated. That way, companies may improve their operations and make important decisions more quickly. This is aided by developments in computer software and other forms of automated data processing. Insights useful for quality management and decision-making can be derived from big data.

v) Feature selection:

When building a model, feature selection is used to pick the most relevant, consistent, and non-redundant features. Reducing database sizes gradually is critical as database quantity and variety continue to grow. Enhancing predictive model performance while minimising processing expense is the fundamental goal of feature selection.

Selecting the most important attributes for use by machine learning algorithms is the goal of feature engineering. By identifying and removing irrelevant or superfluous features, feature selection methods can reduce the number of variables used to train a machine learning model. There are a number of benefits to selecting features in advance instead of leaving it up to the machine learning model.

vi) Algorithms:

A Multilayer Perceptron (MLP) operates by processing data through a series of layers. After data is received by an input layer, it is passed on to hidden layers where neurones do weighted summation of inputs, apply an activation function to account for non-linearity, and then transmit the output to the next layer. Optimising the network's ability to learn complex data patterns is achieved by adjusting neurone weights during training. Final output layer does classification or prediction. MLPs are used in a wide range of applications, from image recognition to financial forecasting, owing to their capacity to model intricate relationships in data.

```
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(random_state=1, max_iter=300)
mlp.fit(X_train, y_train)
y_pred = mlp.predict(X_test)
```

Fig 4 MLP

A CNN is a type of deep learning model suitable for various data beyond images. It processes data through layers that apply convolutions and pooling operations, enabling the network to automatically learn relevant patterns or features within the data. This makes CNNs valuable for tasks involving sequential data or grids, such as time series analysis or structured data processing. They excel at capturing intricate relationships and hierarchies, contributing to their versatility in different domains, including natural language processing and financial predictions.

```
from tensorflow.keras import Sequential,utils
from tensorflow.keras.layers import Flatten, Dense, Conv1D, MaxPool1D, Dropout

def reg():

    model = Sequential()

    model.add(Conv1D(32, kernel_size=(3,), padding='same', activation='relu', input_shape = (X_train.shape[1],1)))
    model.add(Conv1D(64, kernel_size=(3,), padding='same', activation='relu'))
    model.add(Conv1D(128, kernel_size=(5,), padding='same', activation='relu'))

    model.add(Flatten())

    model.add(Dense(50, activation='relu'))
    model.add(Dense(20, activation='relu'))
    model.add(Dense(units = 1))

    model.compile(loss='mean_squared_error', optimizer='adam')

    return model
```

Fig 5 CNN

A **LSTM** is a type of RNN designed for sequential data analysis. Unlike traditional RNNs, LSTMs are adept at capturing and preserving dependencies over long sequences, making them ideal for tasks where data points have complex, distant relationships. LSTMs utilize specialized memory cells and gates that enable them to remember, update, or forget information, facilitating precise modeling of sequential patterns. For patterns both past and future, this is useful for speech recognition, financial time series analysis, and NLP.

```
# Initialising the RNN
regressor = Sequential()
# Adding the first LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50, return_sequences = True, input_shape = (X_train.shape[1], 1)))
regressor.add(Dropout(0.2))

# Adding a second LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

# Adding a third LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

# Adding a fourth LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50))
regressor.add(Dropout(0.2))

# Adding the output layer
regressor.add(Dense(units = 1))
```

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

Fig 6 LSTM

The Multi-Source LSTM is an extended variant of the traditional LSTM neural network designed to process data from various sources simultaneously. It excels at handling comprehensive information by integrating data inputs from multiple origins, making it particularly valuable for complex tasks such as stock price prediction. [30,32] MS-LSTM enhances the model's capacity to capture and analyze intricate dependencies and patterns by leveraging a broad range of data, thus improving the overall predictive capabilities of the system in scenarios where diverse data sources play a critical role.

```
# Initialising the RNN
regressor = Sequential()
# Adding the first LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50, return_sequences = True, input_shape = (X_train.shape[1], 1)))
regressor.add(Dropout(0.2))

# Adding a second LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

# Adding a third LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

# Adding a fourth LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50))
regressor.add(Dropout(0.2))

# Adding the output layer
regressor.add(Dense(units = 1))
```

```
# Compiling the RNN
regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')

# Fitting the RNN to the Training set
regressor.fit(X_train, y_train, epochs = 100, batch_size = 32)
```

Fig 7 MS-LSTM

The MS-SSA-LSTM model, or Multi-Source Sparrow Search Algorithm LSTM, represents a sophisticated approach to stock price prediction. It combines multi-source data from various origins, employs sentiment analysis, and optimizes the LSTM network using the Sparrow Search Algorithm (SSA). This advanced model effectively addresses the challenges of financial forecasting by offering a more accurate and robust way to predict stock prices. It outperforms conventional models and holds high universal applicability, making it a valuable tool for investors and enterprises operating in dynamic financial markets.

```
optimizer=SSA()
```

```
# Initialising the RNN
regressor = Sequential()
# Adding the first LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50, return_sequences = True, input_shape = (X_train.shape[1], 1)))
regressor.add(Dropout(0.2))

# Adding a second LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

# Adding a third LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

# Adding a fourth LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50))
regressor.add(Dropout(0.2))

# Adding the output layer
regressor.add(Dense(units = 1))
```

Fig 8 MS-SSA-LSTM

The Voting Regressor is, by integrating several regression methods, the Voting Regressor improves prediction using ensemble machine learning. In this case, it incorporates three diverse regressors: Linear Regression, Random Forest Regressor, and k-Neighbors Regressor. By aggregating their individual predictions, it aims to create a more accurate and robust model for regression tasks. This approach leverages the strengths of each base regressor, such as the linearity of Linear Regression, the adaptability of Random Forest, and the proximity-based learning of k-Neighbors Regression, to enhance overall predictive capabilities.

```
r1 = LinearRegression()
r2 = RandomForestRegressor(n_estimators=10, random_state=1)
r3 = KNeighborsRegressor()

eclf1 = VotingRegressor([('lr', r1), ('rf', r2), ('r3', r3)])
eclf1.fit(X_train, y_train)
y_pred = eclf1.predict(X_train)
```

Fig 9 Voting Regressor

The LSTM+GRU is a complex recurrent neural network (RNN) architecture that integrates LSTM

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

and GRU cells. Utilising the memory retention of LSTM and the computational efficiency of GRU, it enhances the model's ability to recognise sequential patterns. Because it improves performance and training efficiency by tackling the limitations of each cell type individually, this combination is beneficial for time series data, sequential pattern recognition, and natural language processing.

```
model = Sequential()
model.add(Embedding(num_words, embed_dim,input_length = X_train.shape[1]))
model.add(LSTM(64,dropout=0.4, recurrent_dropout=0.4,return_sequences=True))
model.add(GRU(32,dropout=0.5, recurrent_dropout=0.5,return_sequences=False))
model.add(Dense(2,activation='softmax'))
model.compile(loss = 'categorical_crossentropy', optimizer='adam',metrics = ['accuracy',f1_m,recall_m,precision_m])
#print(model.summary())
```

```
trained5 = model.fit(X_train, Y_train, epochs = 20, batch_size=batch_size,validation_data=(X_test, Y_test),verbose = 1)
```

Fig 10 LSTM + GRU

The Voting Classifier is a key component for sentiment classification in this project, combining the strengths of AdaBoost and Random Forest (RF) [18,39]. It harnesses AdaBoost's boosting capabilities, where multiple weak learners are combined to form a strong classifier, and RF's ensemble learning approach, which aggregates predictions from multiple decision trees. By integrating these two techniques, the Voting Classifier enhances the accuracy and robustness of sentiment classification, making it a powerful tool for analyzing market sentiment in our research.

```
from sklearn.ensemble import RandomForestClassifier, VotingClassifier, AdaBoostClassifier
clf1 = AdaBoostClassifier(n_estimators=100, random_state=0)
clf2 = RandomForestClassifier(n_estimators=50, random_state=1)

eclf1 = VotingClassifier(estimators=[('ad', clf1), ('rf', clf2)], voting='soft')
eclf1.fit(X_train, y_train)
y_pred = eclf1.predict(X_test)
```

Fig 11 Voting classifier

4. EXPERIMENTAL RESULTS

Precision: Accuracy is defined as the percentage of positive samples categorised properly. The formula is used to calculate precision:

Precision = TP/(TP + FP)

$$\text{Precision} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Positive}}$$

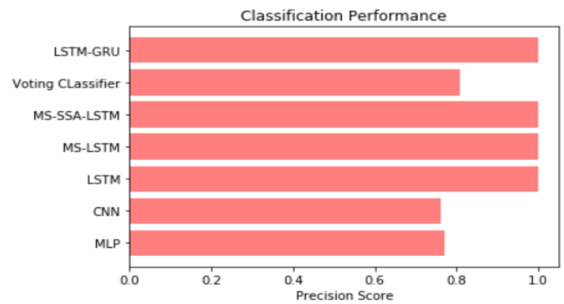

Fig 12 Precision comparison graph

Recall: The ability of a model to identify all pertinent instances of a class is assessed by machine learning recall. The completeness of a model in capturing instances of a class is demonstrated by comparing the total number of positive observations with the number of precisely predicted ones.

$$Recall = \frac{TP}{TP + FN}$$

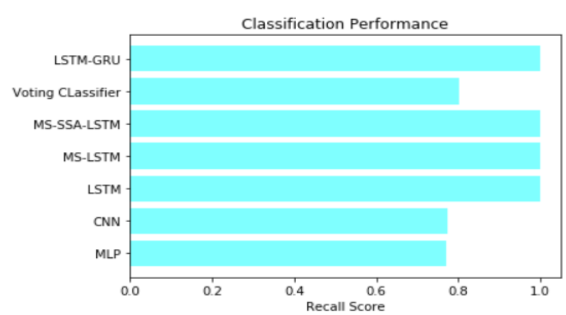

Fig 13 Recall comparison graph

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

Accuracy: The percentage of valid categorisation predictions indicates model performance..

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

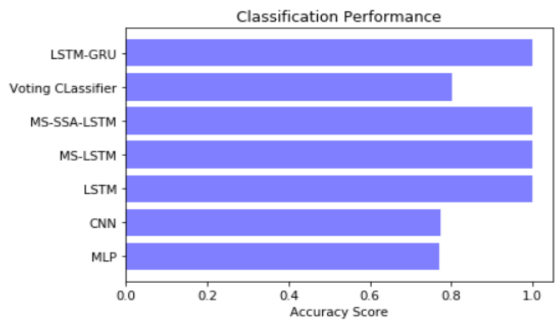

Fig 14 Accuracy graph

F1 Score: For imbalanced datasets, the F1 Score—the harmonic mean of recall and accuracy—is excellent since it balances false positives and negatives.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

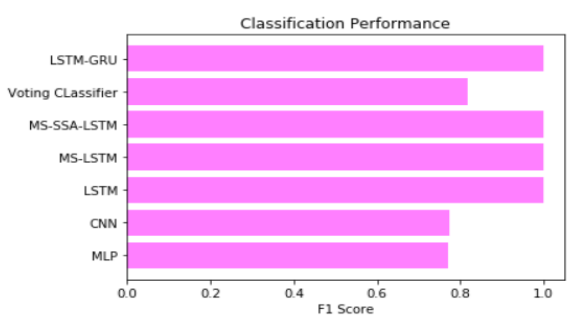

Fig 15 F1Score

	ML Model	Accuracy	Precision	Recall	F1-Score
0	MLP	0.771	0.771	0.771	0.770
1	CNN	0.773	0.761	0.773	0.774
2	LSTM	1.000	1.000	1.000	1.000
3	MS-LSTM	0.998	0.998	0.998	0.998
4	MS-SSA-LSTM	1.000	1.000	1.000	1.000
5	Extension- Voting Classifier	0.803	0.808	0.803	0.819
6	Extension- LSTM-GRU	1.000	1.000	1.000	1.000

Fig 16 Performance Evaluation

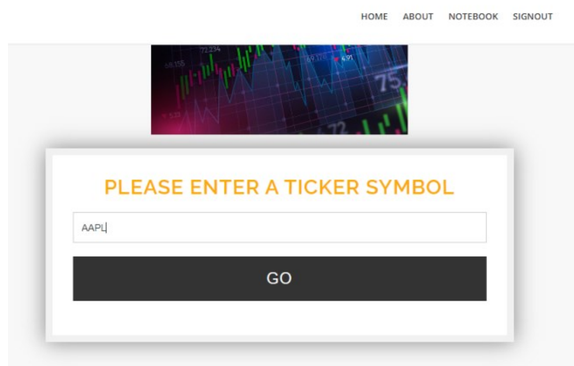

Fig 17 User input

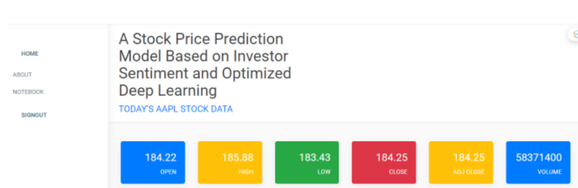

Fig 18 Result

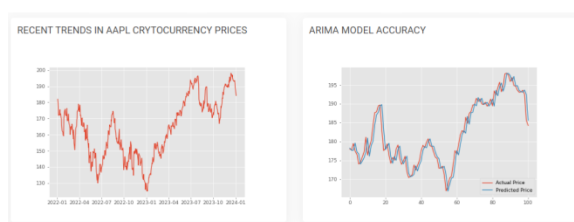

Fig 19 Graphs

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

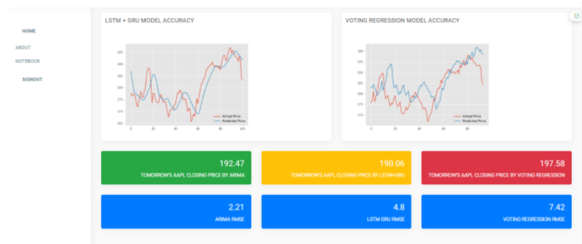

Fig 20 Graphs

5. CONCLUSION

The project aimed to enhance stock market predictions, with a focus on the MS-SSA-LSTM model. The research explored various models, emphasizing the significance of sentiment analysis and innovative algorithms for optimized forecasting [26]. The MS-SSA-LSTM model stood out for its dual proficiency in stock price prediction and sentiment classification. Leveraging diverse data sources and advanced techniques, it offered a comprehensive approach to risk reduction and improved returns. Existing models (MLP, CNN, LSTM, MS-LSTM) demonstrated competence, while the MS-SSA-LSTM model showcased superiority, particularly in short-term predictions for China's dynamic market. Ensemble models (Voting Classifier, LSTM+GRU, Voting Regressor) introduced in the extension phase expanded the predictive toolkit. LSTM+GRU excelled in sentiment classification, and the Voting Regressor outperformed in stock price prediction, contributing reliable alternatives. The Flask extension facilitated user-friendly interaction, allowing input of ticker symbols for accurate predictions. LSTM+GRU for sentiment and Voting Regressor for stock price predictions were seamlessly deployed, enhancing accessibility for users and investors. Investors, traders, and businesses stand to benefit from the project's robust predictive models and user-friendly interface. The MS-SSA-LSTM model and its extensions offer valuable insights, reducing investment risks, and enhancing decision-making in the dynamic landscape of the Chinese financial market.

6. FUTURE SCOPE

Expanding the model's capabilities to handle real-time data feeds can enable investors to make even more timely decisions. Integrating data sources that provide up-to-the-minute information could be a valuable addition. [34] Further refining the sentiment analysis component by incorporating NLP techniques and sentiment-specific ML models can provide a more nuanced understanding of market sentiment. Exploring and integrating data from diverse sources, such as social media, news feeds, and macroeconomic indicators, can offer a comprehensive view of the market and potentially improve predictive accuracy. Developing tools or features that offer explanations for the model's predictions can make it more transparent and user-friendly. Investors may benefit from understanding the reasons behind specific forecasts. Extending the model's capabilities to include risk assessment and portfolio optimization can provide investors with a holistic approach to managing their investments. This could involve considering the diversification of assets and risk-adjusted returns.

REFERENCES

[1] "Effects of using principal component analysis and artificial neural network models on stock price prediction on the Tehran Stock Exchange," by J. Zahedi and M. M. Rounaghi Publication date: November 2015, volume 438, pages 178–187, DOI: 10.1016/j.physa.2015.06.033..

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

[2] Gold price predictions using the ARIMA model, G. Bandyopadhyay Journal of Advanced Management Science, 2016, 4, 117–121, doi: 10.12720/joams.4.2.117–121.

[3] "Analysis and prediction of Shanghai composite index by ARIMA model based on wavelet analysis," published by H. Shi, Z. You, and Z. Chen. Mathematical Practice Theory, volume 44, issue 23, pages 66-72, 2014.

[4] "An empirical assessment of stock return prediction under GARCH," by H. Herwartz "International Journal of Forecasting," volume 33, issue 3, pages 569–580, July 2017, doi: 10.1016/j.ijforecast.2017.01.002.

[5] "Global evidence on the dynamics of crude oil prices: Using ARIMA-GARCH models," by H. Mohammadi and L. Su Publication date: September 2010, volume 32, issue 5, pages 1001–1008, doi: 10.1016/j.eneco.2010.04.009.

[6] An approach to predict the unpredictability of financial returns using recurrent support and relevance vector machines, by A. Hossain and M. Nasser Journal of Intelligent Learning Systems and Applications, volume 3, issue 4, pages 230–241, 2011, doi: 10.4236/jilsa.2011.34026.

[7] "Stock market index prediction using artificial neural network," by A. H. Moghaddam, M. H. Moghaddam, and M. Esfandyari (2013) Article published in December 2016 in the Journal of Economics, Finance, and Administrative Science with the DOI: 10.1016/j.jefas.2016.07.002.

[8] "Forecasting market price of stock using artificial neural network," by A. Murkute and T. Sarode, published in Aug. 2015 in the International Journal of Computer Applications, volume 124, issue 12, pages 11–15, doi: 10.5120/ijca2015905681.

[9] "Artificial neural network for Bitcoin price forecasting," D. Banjade December 2019, DOI: 10.2139/ssrn.3515702.

Author's profile

1. U D S V Prasad, an accomplished educator in the field of Computer Science and Engineering, holds a distinguished M.Tech degree from KL Deemed to be University Vaddeswaram, ,located in the Guntur District, A.P, INDIA. With an illustrious career spanning over 5 years, he currently serves as an Assistant Professor at the esteemed SASI Institute of Technology & engineering in Tadepalligudem.

2.AMARAPALLI DEVI SAI PRASANNA a Btech student specializing in Computer Science and Engineering at Sasi Institute of Technology and Engineering in Tadepalligudem . I had completed certifications in Java Full Stack from Wipro TalentNext, AIML Virtual Internship ,Cloud computing from Eduskills , Machine learning from SkillDzire.

3.KUDUPUDI AKHIL is a B. Tech student specializing in Computer Science and Engineering at Sasi Institute of Technology and Engineering. I completed certifications in Java Full Stack from Wipro TalentNext and Java basic at Stepping Stone solutions .I am good at Python, Machine

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

Learning , and problem-solving and is committed to continuous learning.

4. GEDDAM GOWRIDHAR a B. Tech student specializing inComputer Science and Engineering at Sasi Institute of technology and Engineering . Iam passionate about exploring emerging technologies . I completed internships in Cloud Computing and AIML, gaining practical experience. Additionally, I earned Java Full Stack certification from Wipro (TalentNext).I am good at Python Programming, Web-Development and Iam passionate about enhancing his technical abilities.

5. LOKESH KURMAPU is a B. Tech student specializing in Computer Science and Engineering at Sasi Institute of Technology and Engineering. Passionate about Learning cybersecurity as a EH.Completed an internship in palo alto cyber security.Learned networking in cisco gained certificate.Done some internships in cloud computing by AWS.Passoinate about on cybersecurity