Innovation in Computer and Data Sciences
1(1), 32–50
DOI: https://doi.org/10.64389/icds.2025.01126
https://sphinxsp.org/journal/index.php/icds/

Research article

Hybrid LSTM-CNN deep learning framework for stock price prediction with google stock and reddit sentiment data

Emmanuel Chibuogu Asogwa[1], Mmesoma P. Nwankwo[2], Emmanuel E. Oguadimma[3], Chinyere P. Okechukwu[4,*], Ahmad Abubakar Suleiman[5]

[1] Department of Computer Science, Faculty of Physical Sciences, Nnamdi Azikiwe University, P.O. Box 5025 Awka, Nigeria; ec.asogwa@unizik.edu.ng

[2] Department of Statistics, Faculty of Physical Sciences, Nnamdi Azikiwe University, P.O. Box 5025 Awka, Nigeria; mp.nwankwno@stu.unizik.edu.ng

[3] Department of Mathematics, Oregon State University, Corvallis, OR 97331, USA; oguadime@oregonstate.edu

[4] Department of Statistics, School of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab, 144411, India; chinyere.12404612@lpu.in

[5] Fundamental and Applied Sciences Department, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia; ahmadabubakar31@gmail.com

* **Correspondence:** chinyere.12404612@lpu.in

ARTICLE INFO

Keywords:
Hybrid Deep Learning
Stock Price Prediction
Sentiment Analysis
LSTM (Long Short-Term Memory)
CNN (Convolutional Neural Network)

Mathematics Subject Classification:
68T07, 91G60, 62M10, 91B84, 68T50, 68T05

Important Dates:
Received: 28 May 2025
Revised: 15 June 2025
Accepted: 22 June 2025
Online: 24 June 2025

ABSTRACT

This study evaluates a hybrid model that integrates Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN) to predict stock prices. The model leverages two datasets: historical Google stock data and sentiment data from Reddit comments. Sentiment analysis was performed using VADER from NLTK, which classified comments as negative, neutral, or positive, while a CNN model was trained to predict sentiment scores. Separately, an LSTM model was built using ten years of Google stock data from Yahoo Finance, with features scaled using MinMax normalization to improve learning and a dropout layer added to prevent overfitting. Model performance was evaluated using Root Mean Squared Error (RMSE) and Mean Squared Error (MSE). The LSTM model performed well on test data but showed lower accuracy on unseen data during forecasting. The hybrid model successfully combined the outputs of both the CNN and LSTM, demonstrating superior performance with lower RMSE and higher classification accuracy compared to the standalone models. This highlights the potential of integrating sentiment analysis with traditional stock prediction. The study acknowledges challenges in classifying neutral sentiments, suggesting that more comprehensive sentiment data is needed for future research.

1. Introduction

The buying and selling of shares takes place electronically, making the stock market a truly volatile and uncertain field. Accurate prediction is a long-standing challenge for investors and financial experts due to the non-linear and dynamic nature of the market, a challenge highlighted by [19]. While early mathematical models, such as linear regression, were used to forecast future price trends, they often struggled to achieve accuracy because stock price changes are influenced by multiple factors and involve complex, nonlinear relationships [2]. The advent of Artificial Intelligence (AI) and deep learning has offered new avenues for more accurate and timely predictions. As noted by [22] and [6], emerging technologies like machine learning play an important role in forecasting stock prices due to their ability to handle large datasets and capture intricate patterns. Deep learning, a subset of machine learning, uses multi-layered neural networks to process data in a way that is inspired by the human brain [11]. This method has significantly advanced fields such as speech and visual recognition. Due to its ability to identify patterns in vast datasets, deep learning is increasingly being applied to the challenging task of stock market prediction, offering a more effective approach than traditional manual analysis [25, 24, 21]. Among the various deep learning algorithms, Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks are prominent. While CNN models have shown high precision rates, as evidenced by a study achieving 98.92% accuracy, they have limitations in handling temporal data [16]. This is where LSTM models excel, as they are specifically designed to retain and leverage temporal relationships and long-term dependencies within data sequences [3]. The superior ability of LSTMs to capture complex, non-linear patterns has made them a compelling choice for stock price forecasting [4]. A key trend in this field is the development of hybrid models that combine the strengths of different architectures. The CNN-LSTM framework is particularly popular, integrating the CNN's feature extraction capabilities with the LSTM's proficiency in temporal modeling [15]. Researchers have successfully used these hybrid models with sentiment analysis and technical indicators to improve prediction accuracy [8]. For example, some studies show that integrating financial data with sentiment from social media and news can significantly enhance a model's performance, achieving accuracy rates up to 74.3% [1]. Another approach uses weighted and categorized news to further refine predictions, demonstrating that the relevance of the news to specific stocks is a crucial factor [26].

Despite these advancements, it is important to acknowledge that there is no perfect model for predicting stock prices with complete certainty due to the inherent volatility and non-stationary nature of financial markets [14, 1]. However, the consistent findings from extensive research indicate that hybrid deep learning approaches, especially those that incorporate sentiment analysis, offer the most dependable option for analyzing market trends and forecasting future stock prices [5]. Deep learning, a subfield of artificial intelligence (AI), has revolutionized data processing by teaching computers to analyze information in a way inspired by the human brain [11]. As the stock market gains popularity, experts are increasingly exploring deep learning as a method to improve the accuracy of stock price predictions [25]. Researchers assert that analyzing historical data through these models allows for the discernment of patterns and the acquisition of valuable insights more effectively than manual methods [24, 21]. Despite certain constraints, forecasting techniques offer significant advantages to investors and other market participants [12]. A wide range of machine learning and deep learning algorithms have been applied to this problem. Common network models include the Artificial Neural Network (ANN) and the Convolutional Neural Network (CNN), with a study by [16] showing a CNN model achieving a 98.92% precision rate, slightly outperforming an ANN model at 97.66%. However, while these models are powerful, they have limitations in capturing the temporal char-

acteristics of financial data. This has led to the rise of models like the Long Short-Term Memory (LSTM) network, which is highly effective in dealing with time-series data and addressing long-term dependencies [3]. A comparative study by [20] found that an LSTM model with Principal Component Analysis (PCA) achieved superior performance metrics (MAE of 0.032) compared to a 1D-CNN model (MAE of 0.039), despite the CNN's faster training time. Other research has similarly found LSTMs to be more accurate than conventional models like ARIMA and moving averages due to their ability to handle the complex and non-linear nature of stock market data [18].

To further improve accuracy, researchers are turning to hybrid models that combine multiple approaches. The CNN-LSTM framework has been deployed to forecast the stock market by integrating both sentiment analysis and technical indicators [15]. The model's ability to process both types of data led to a mean squared error of just 0.0011, outperforming other models. The synergy of these approaches is also demonstrated in a hybrid CNN-LSTM model for big data, which achieved over 85% for the POCID metric [8], and a Feature Fusion LSTM-CNN model that combines time series data and stock chart images to reduce prediction error [9]. Other hybrid models, such as the Recurrent Convolutional Neural Kernel (RCNK), have been developed to learn complementary features from historical prices and text data from message boards [13]. The incorporation of sentiment analysis is a major trend, as financial data alone may not be sufficient for accurate predictions [1]. Research shows that integrating sentiment from social media posts and news can significantly enhance a model's performance. For example, a study using Twitter sentiment and financial data achieved an accuracy rate of 74.3% [1], and a deep neural network framework that processed stock-related comments improved prediction accuracy for Chinese stocks by 1.25% [23]. The impact of news is further underscored by the LSTM-based Weighted and Categorized News Stock prediction model (WCN-LSTM), which improved accuracy by integrating news categories with learned weights [26]. The analysis of investor sentiment has also been applied to specific markets, showing that it can enhance predictive accuracy for Chinese A-share stocks [17]. While different machine learning and deep learning techniques have been proposed to tackle the complexities of stock market forecasting [10, 27, 7], a comparative analysis by [5] confirmed that incorporating financial news data into LSTM and GRU models consistently improves prediction accuracy over using stock features alone. In summary, a diverse array of models exist, but the most effective solutions tend to be hybrid approaches that leverage the strengths of models like LSTM and CNN while incorporating external factors like sentiment to better capture the market's dynamic nature.

This study proposes and evaluates a hybrid deep learning framework that combines the strengths of Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) models for stock trend analysis and future price prediction. The framework integrates historical Google stock data with sentiment analysis derived from Reddit discussions. By uniting quantitative time-series data with qualitative sentiment, this research aims to create a more robust predictive model that captures the full picture of market dynamics. This approach addresses the limitations of standalone models and seeks to enhance prediction accuracy, thereby providing a novel, integrative methodology for the field of financial forecasting.

2. Research Design

This section provides a concise overview of the methodology, model design, and evaluation metrics used for stock price prediction. This research employs a data-driven, machine learning-based approach to predict stock prices using a hybrid LSTM-CNN model. This model was chosen for its ability to simultaneously process sequential time-series data and text-based sentiment data. The study utilizes two distinct

datasets. Stock data for Google was sourced from Yahoo Finance, covering August 25, 2014, to August 22, 2024, (https://finance.yahoo.com/quote/GOOG/history/), and was preprocessed using Min-Max normalization. Sentiment data was collected by scraping Reddit posts, with text preprocessed and sentiment scores assigned using tools like VADER. The two datasets were then integrated on a daily basis. The hybrid model combines two parallel paths. An LSTM model analyzes the time-series stock data, while a CNN model processes the text-based sentiment data. The outputs of both models are concatenated and fed into a final set of fully connected layers for the stock price prediction. The models were trained using the Adam optimizer with a learning rate of 0.001, a batch size of 64, and for 200 epochs. Regularization techniques such as dropout were used to prevent overfitting, and hyperparameters were tuned using Grid Search. Model performance is evaluated using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The research will conduct two key comparisons: the performance of the hybrid model against standalone LSTM and CNN models, and an ablation study to quantify the impact of including sentiment data on the predictions.

3. Implementation and Testing

In this section, we provide an overview of the hybrid LSTM-CNN framework used for stock price prediction and sentiment analysis. The system architecture is designed to process Google stock prices and sentiment data separately, combining them for a final hybrid model. The choice of using LSTM for time-series stock prediction and CNN for feature extraction from textual sentiment data is discussed. The Google stock prices were collected from Yahoo Finance API, spanning a timeframe from [start date] to [end date]. Data preprocessing steps included handling missing values, scaling using Min-Max normalization, and removing any outliers. The sentiment data was sourced from Reddit, using textual information to gauge market sentiment. The preprocessing involved tokenization, stopword removal, stemming, and converting the text into vectors using Word2Vec embeddings. Sentiment labels (positive, neutral, negative) were assigned using a [sentiment lexicon/model, VADER]. The LSTM model architecture consisted of 2 layers with 100 units, optimized using the Adam optimizer. The stock price data was fed into the LSTM and for the sliding window size time steps size of 50 for sequence prediction. Hyperparameter tuning was done using Random Search to find the optimal learning rate, batch size, and number of epochs. The CNN architecture for sentiment analysis involved [number] convolutional layers with 64 filters and kernel sizes of 3. The embedding layer converted words into dense vector representations using [Word2Vec, GloVe, or FastText]. The final layer used a Softmax activation function for sentiment classification. The outputs of the LSTM and CNN models were concatenated and passed through additional fully connected layers for final prediction. The hybrid architecture is illustrated in Figure 1, showing the flow from the LSTM and CNN models into the combined output layer.

3.1. Testing Strategy

The dataset was split into training, validation, and test sets, using a [70-15-15] split. Cross-validation with [k] folds was used to ensure robustness. Model training and evaluation were performed on [hardware specifications], using [TensorFlow/Keras] as the software framework.

The return column is introduced to this data to show the volatility of Alphabet Inc stock. the return column is the percent change in the closing price of the stock.

The opening and closing stock price of Alphabet inc are plotted against time and displayed in figures 3

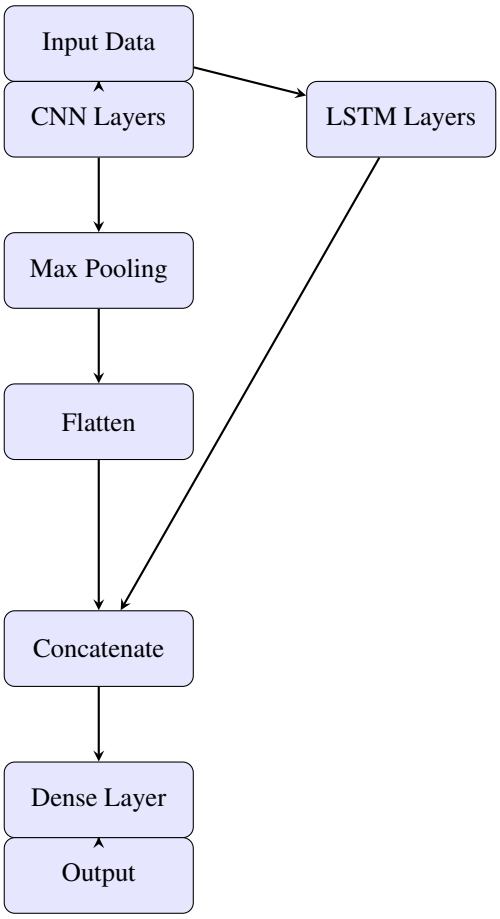

Figure 1. Hybrid CNN-LSTM Architecture for Time-Series and Sentiment Analysis

Date	Open	High	Low	Close	Adj Close	Volume	Return
2014-08-26	28.983425	29.010351	28.750067	28.813890	28.781139	32793789	-0.403314
2014-08-27	28.784472	28.845304	28.427103	28.471830	28.439468	34067276	-1.187136
2014-08-28	28.400028	28.584023	28.277365	28.382076	28.349815	25858801	-0.315238
2014-08-29	28.488285	28.523687	28.275820	28.501747	28.469351	21675347	0.421643
2014-09-02	28.514214	28.812395	28.481304	28.787464	28.754742	31568434	1.002454
...	...	...	...	...	...	...	...
2024-08-16	163.410004	166.949997	163.080002	164.740005	164.740005	16853100	0.962191
2024-08-19	167.000000	168.470001	166.089996	168.399994	168.399994	13100800	2.221676
2024-08-20	168.740005	170.410004	168.660004	168.960007	168.960007	12622500	0.332549
2024-08-21	166.990005	168.639999	166.570007	167.630005	167.630005	15269600	-0.787170
2024-08-22	169.039993	169.419998	165.029999	165.490005	165.490005	19111700	-1.276621

2515 rows × 7 columns

Figure 2. Google Stock data

and 4 the volatility of the stock is also shown in figure 5

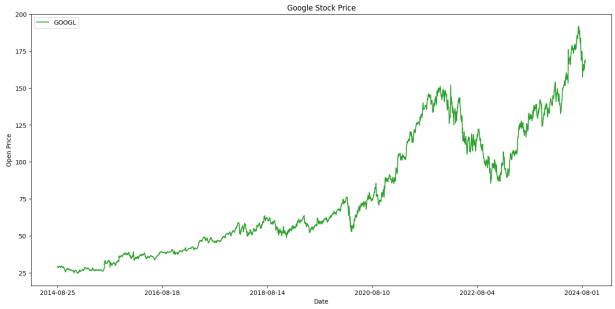

Figure 3. Opening price of Alphabet inc

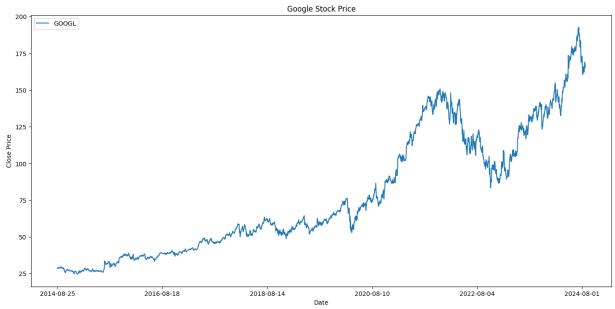

Figure 4. Closing price of Alphabet inc over time

The return plot shows the volatility (day to day fluctuation) of Google stock price how it changes over time, it depicts the spread of their closing price, their big loss and their big gains.

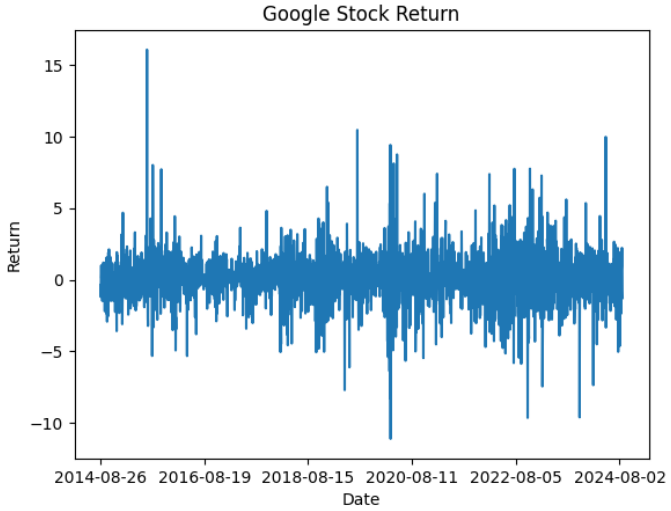

Figure 5. Daily return of Alphabet inc

The summary statistics for the financial indicators is given

The stock data was split into train-test split by using loc method in pandas, we cannot employ the train-test split method from sklearn to avoid randomness of data.The train data contains 80% of the observation

	Open	High	Low	Close	Adj Close	Volume	Return
count	2515.000000	2515.000000	2515.000000	2515.000000	2515.000000	2.515000e+03	2515.000000
mean	77.587469	78.420466	76.827244	77.641761	77.557656	3.144183e+07	0.085166
std	42.130326	42.616580	41.711345	42.165504	42.127355	1.544465e+07	1.780164
min	24.664783	24.730902	24.311253	24.560070	24.532154	6.936000e+06	-11.100819
25%	41.189251	41.387425	41.023825	41.200750	41.153920	2.181235e+07	-0.716122
50%	60.500000	61.250000	60.179001	60.750000	60.680946	2.763000e+07	0.107725
75%	113.902253	115.629273	112.853252	114.574253	114.444023	3.628500e+07	0.939828
max	191.750000	193.309998	190.619995	192.660004	192.660004	2.232980e+08	16.052427

Figure 6. Summary Statistics

and the test split the remaining 20%. The MinMaxScalar module from sklearn library was used to scale the target and feature variables.

3.2. Sentiment Analysis

Top comments related to stock on Reddit was scraped using the scraper PRAW (Python Reddit API Wrapper). The NLTK library was used to process the comment from the Reddit, other libraries were employed to remove strings and emojis if available in the data. The Vader module from NLTK library group these data into positive, neutral or negative which are called sentiment score and we then group them by the date they were posted. The label column of the sentiment was gotten by aggregating the values of the compound variable by assigning 1 to values greater than 0.1 and -1 to values less than 0.1 and zero otherwise.

	neg	neu	pos	compound	headlines	date	label
0	0.000	1.000	0.000	0.0000	Anyone trim down their positions	02-07-24	0
1	0.000	1.000	0.000	0.0000	SoftBank stock hits its first record high in 2...	02-07-24	0
2	0.000	1.000	0.000	0.0000	What's up with SOFI	02-07-24	0
3	0.000	0.548	0.452	0.5106	Is short selling indices ethical	02-07-24	1
4	0.143	0.744	0.114	0.2640	What is the best way to diversify an aggressiv...	02-07-24	1
...	...	...	...	...	...	...	...
912	0.000	1.000	0.000	0.0000	Chip Gear Giant Applied Materials Delivers Bea...	02-07-24	0
913	0.000	1.000	0.000	0.0000	Whats the next ASTS	02-07-24	0
914	0.123	0.738	0.139	0.0516	How does this make sense in the spirit of cost...	02-07-24	0
915	0.182	0.341	0.477	0.5574	Best resources for aggressive investing	02-07-24	1
916	0.204	0.796	0.000	-0.6249	Investing in Corsair Gaming, Inc. (CRSR) slowl...	02-07-24	-1

917 rows × 7 columns

Figure 7. Sentiment data concerning stock from Reddit

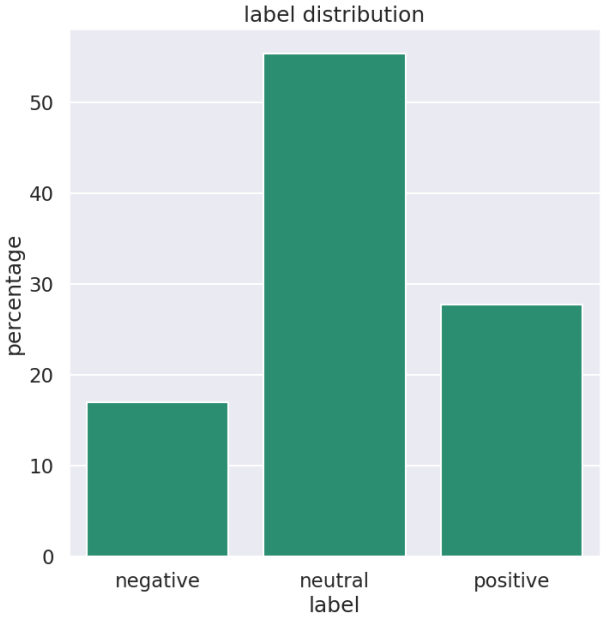

Figure 8. Sentiment value

We also deployed stop words from the world cloud library to see some of the stop word available in our data.

Figure 9. 150 stop words in our text

3.3. LSTM model training

The keras library is imported from the tensorflow module. For the LSTM, some parameters to be used for the prediction are defined. First, the time steps is defined as as the number of days the model will look back and then make the next prediction. In the model, the timesteps is taken as 50 days so the model will

look at the previous 50 days to make the next prediction. The stacked model is built of two layers of 100 units each, so the LSTM model has 100 neurons each in each layer. The dropout is also assigned as 0.2. The drop out is to ensure that the model is not over-fitting the data. It simply means fraction of the unit to drop for the linear transformation of the inputs. The dropout is a float between 0 and 1. Similarly, the dense layer is defined to be 25 and 1 these dense layer are also known as fully connected layer, which means how the layers are interconnected with each other. For the target vector, the closing price is utilized, and all other columns are used as the features except the return column. During the feature selection, other columns are used to as independent variables to predict the target vector since the return column added little or nothing in the prediction. The data is scaled by using MinMax scaler this is to ensure uniformity between the data and no huge outlier. This also makes the model to train faster. The model is fitted by using an epoch size of 200. One epoch is when the entire dataset is passed forward and backward through the neural network only once. Multiple epochs help the model generalize better. The dataset might contain a lot of example, passing it at once might be difficult so the dataset are divided into batches. In the model, a batch size of 64 is chosen and they are fed into the neural network. Iteration is the number to complete one epoch and 35 iterations are required. Activation function essentially introduce non-linearity to the neural network. For the activation function, the default activation function is used in LSTM, that is the hyberbolic tangent(tanh. There are other popular activation functions in keras that can be used which include RELU, Leaky RELU, sigmoid and Linear. The Mean squared error is utilized as the loss function to calculate the loss. For the optimizer, the ADAM (Adaptive Moment Estimation) is chosen. They are other popular optimizer like stochastic gradient descent, Adagrad, and RMSprop. The aim of these optimizers is to minimize the loss function. After the prediction, the RMSE is used to find the difference between our actual value and the predicted value. After the prediction, we forecast the value for the next day.

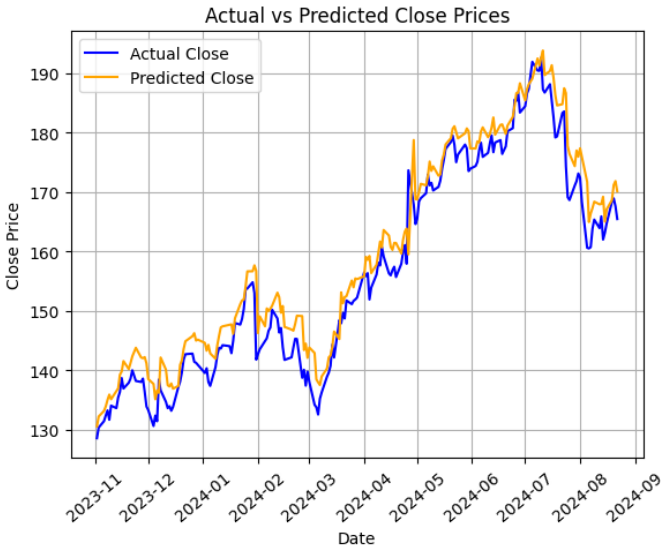

Figure 10. Actual Data vs Predicted Data

The hyperparameter values need to be carefully selected to ensure the model does not overfit or underfit the predictions. As seen in Figure 12, the predicted value follows the actual value and performs well in prediction; however, it does not perform as well when forecasting. It performed effectively on the test data but less so on unseen data during forecasting.

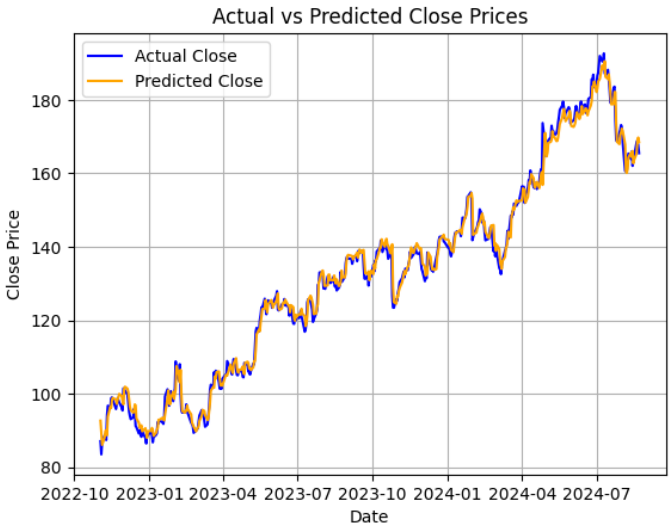

Figure 11. An example of over-fitted model

Date	Predicted_Close	Actual_Close
2023-11-02	130.565018	128.580002
2023-11-03	132.235062	130.369995
2023-11-06	133.170303	131.449997
2023-11-07	133.938538	132.399994
2023-11-08	135.030014	133.259995
...	...	...
2024-08-16	167.118225	164.740005
2024-08-19	168.679810	168.399994
2024-08-20	171.218048	168.960007
2024-08-21	171.871857	167.630005
2024-08-22	170.145004	165.490005

202 rows × 2 columns

Figure 12. Predicted close and actual close price

Calculating the RMSE yielded a value of 4.2971, indicating that the prediction might deviate by this amount. Figure 13 presents the forecast for the next day. While using this forecast, it is important to consider that the RMSE suggests a potential deviation of 4.2971. Since only the closing price was predicted, the NAN values represent variables that were not forecasted. To predict other variables, a Multistep LSTM can be utilised.

	Date	Open	High	Low	Close	Adj Close	Volume	Return	Predicted_Close
2515	2024-08-23 00:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	169.951620

Figure 13. Forecast value for the next day

For the sentiment stock, sentiment data is concatenated, involving both stock and investment aspects. Initially, a tokenizer is employed on the text, facilitating the preprocessing of the text and converting it into a format that machines can better comprehend. Subsequently, the data is split into training and test sets, with 80% allocated for training and 20% for testing. As evidenced in Figure 8, the data is imbalanced; thus, class weights are utilised to address this imbalance. Alternative methods for handling imbalanced data include using random sampling, which allows for either over-sampling or under-sampling to resample the available data. The model is defined as sequential. The embedding layer transforms text into continuous dense vectors, with the input dimension representing the number of different embeddings to be learned and the output dimension set to 128, corresponding to the number of embedding vectors. An additional layer is incorporated into the model: a one-dimensional convolutional layer with 64 filters, representing the number of features to be learned. The kernel size is set to 3 for the first layer and 5 for the second layer, indicating the number of features being analysed. The activation function employed is the ReLU activation function, introducing non-linearity into the model. A max pooling layer, with a pool size of 2, is utilised to reduce the dimensionality of the data, making it less susceptible to overfitting. L2 regularisation (Ridge regulariser) is implemented to mitigate over-fitting, ensuring that the influence of correlated features is evenly distributed among the coefficients and preventing any single feature from dominating the model's predictions. During model compilation, the ADAM optimizer is applied with a learning rate of 0.001, and categorical cross-entropy is designated as the loss function, while accuracy serves as the performance metric. The model is fitted with an epoch size of 10 and a batch size of 32, resulting in 46 iterations for each epoch. Additionally, a classification report is utilised as a performance metric, with F1-score, precision, and recall serving as key metrics for evaluating the performance of a classification model, particularly in the context of imbalanced data.

A trade-off exists when choosing between precision and recall. Recall measures the proportion of actual positive instances correctly identified by the model, while precision indicates the proportion of positive predictions made by the model that were accurate, reflecting the model's ability to correctly identify observations belonging to the positive class without incurring false positives.

The predicted sentiment is compared alongside the actual sentiment and the actual text where 0 is neutral 1 is positive and -1 is negative. With accuracy of classification of 59%, the CNN sentiment model perform fairly good.

For the hybrid model, data preprocessing involves the use of a tokenizer on the sentiment data and a scaler on the stock data. The dataset is split using a train-test split for the sentiment data and a slicing method for the stock data. A CNN model is employed for the sentiment data and an LSTM model for the stock data. The CNN model is defined as a sequential model, comprising an embedding layer, a one-dimensional convolutional layer, a max pooling layer, a flatten layer, and a dense layer with 64 units. The

	precision	recall	f1-score	support
1	0.26	0.56	0.36	48
0	0.79	0.65	0.71	226
-1	0.55	0.46	0.50	93
accuracy			0.59	367
macro avg	0.53	0.56	0.52	367
weighted avg	0.66	0.59	0.61	367

Figure 14. Classification Report

	Text	Actual Sentiment	Predicted Sentiment
1468	Anyone else concerned their portfolio is going...	0	Neutral
1469	Tips on better financial planning	1	Neutral
1470	If I have 40k saved up, is there any reason wh...	1	Neutral
1471	"Strike it big stocks for the next decade", ye...	-1	Neutral
1472	Name of this stock that tanked in last week an...	1	Negative
...	...	...	...
1830	Taxes! What do you do Advice plz.	1	Positive
1831	Vanguard Bond ETF or Robinhood Cash	0	Positive
1832	Investment account for godchild	0	Neutral
1833	Why are there so many S&P 500 funds	0	Positive
1834	Diversify large concentrated stock position; o...	0	Positive

Figure 15. Comparison of actual sentiment and predicted sentiment on our test data

Predicted Sentiment	count
Positive	185
Neutral	104
Negative	78

Figure 16. Value count on the three classes based on the test data

dense layer indicates the degree of connectivity within the network, with the ReLU activation function applied. The LSTM model is defined with 150 units, including a dropout layer and a dense layer. The outputs from the CNN and LSTM models are concatenated, forming a parallel architecture where both models are trained independently before building the hybrid model. This hybrid model utilises inputs from both the CNN and LSTM, and is compiled using the ADAM optimizer with mean squared error (MSE) as the loss function. The model is then fitted on the input sentiment and stock data, with the target being the closing price of the stock. The dataset is trimmed to ensure that the time-series data aligns with the length of the sentiment data, preventing the potential issue of jumpy data. The hybrid model is then employed to predict the target. The predicted and actual data are plotted alongside each other, illustrating the relationship between sentiment data and stock prices, as news events shape investor expectations. The hybrid model facilitates the network's learning from both stock trends (LSTM) and sentiment data (CNN). The graph suggests that while the predicted values did not capture minor fluctuations in the actual data, the overall trend was identified. This may indicate that the sentiment data was not sufficiently comprehensive, as only top comments can be extracted from Reddit, omitting the full spectrum of user comments.

The performance of these models are compared using the performance metric MSE (Mean Squared Error). From the MSE, it is observed that the LSTM has a smaller value when compared to the hybrid model containing the time Series data and sentiment data.

Further, another hybrid model is built where CNN and LSTM takes in the time series data as the input using the same timesteps 50 and dataset split in the same ratio as the LSTM model, that is 90:10 ratio. It is observed that the hybrid model has less MSE when compared to the three different model. One can decide to tweak the parameters for better result but also be careful not to under-fit or over-fit the model.

The performance of the three models is compared using the mean squared error, as illustrated in Figure 22 and it is observed that the hybrid model from the concatenation of CNN-LSTM has the least MSE followed by the standalone LSTM model but the Hybrid sentiment has the highest MSE. These indicate that the CNN-LSTM model is the best.

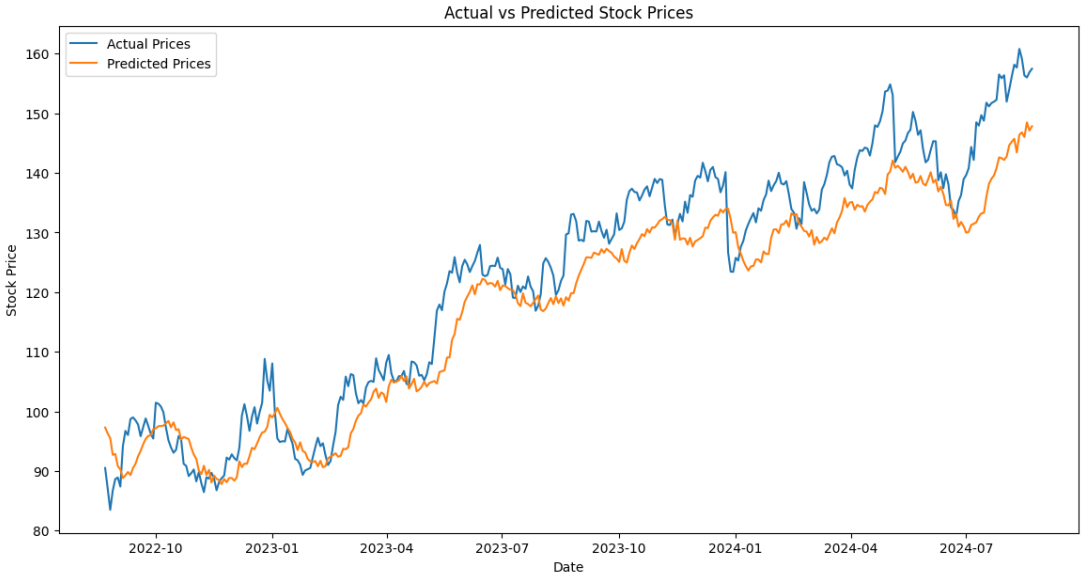

Figure 17. Actual vs predicted

Date	Actual test Price	Predicted test price
2022-08-22	90.500000	97.293114
2022-08-23	87.070000	96.305031
2022-08-25	83.489998	95.513313
2022-08-27	86.699997	92.709732
2022-08-29	88.650002	92.865814
...	...	...
2024-08-14	159.190002	146.794388
2024-08-16	156.330002	146.008118
2024-08-18	156.000000	148.470383
2024-08-20	156.880005	147.089905
2024-08-22	157.460007	147.814835

367 rows × 2 columns

Figure 18. Actual test price vs predicted test price

LSTM MSE	Hybrid MSE
18.4651	49.0342

Figure 19. MSE of the two different model

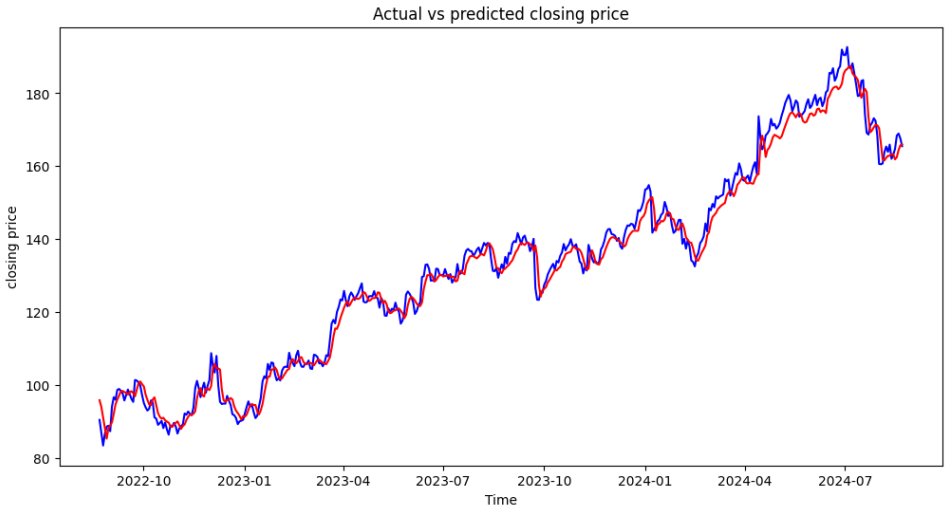

Figure 20. Actual vs predicted

Date	Actual	Predicted
2022 -08-22	90.500000	95.908699
2022 -08-23	87.070000	94.072499
2022 -08-25	83.489998	91.159434
2022 -08-26	86.699997	88.082859
2022 -08-28	88.650002	85.376955
...	...	...
2024 -08-15	164.740005	161.878010
2024 -08-17	168.399994	162.494604
2024 -08-18	168.960007	164.455004
2024 -08-20	167.630005	165.717665
2024 -08-22	165.490005	166.025465

454 rows × 2 columns

Figure 21. Actual closing price vs predicted closing price

	CNN-LSTM MSE	LSTM MST	Hybrid Sent MSE
0	10.962336	18.4651	49.0342

Figure 22. Mean Squared Error of the three models

4. Evaluation and Findings

This chapter evaluates several machine learning and deep learning models for predicting stock prices and market sentiment. The sentiment analysis model, a Convolutional Neural Network (CNN), processed Reddit comments to classify sentiment as negative, neutral, or positive, achieving a 59% accuracy. Due to a class imbalance, class weights were applied to improve predictions for the minority classes. For stock price prediction, a Long Short-Term Memory (LSTM) model was trained on ten years of Google stock data from Yahoo Finance. The data was scaled using MinMax normalization to prevent features like trading volume from dominating the model's learning. The LSTM model had two stacked layers and a dropout layer to prevent overfitting. It performed well on test data, with an RMSE of 4.2971, but was less effective at forecasting unseen data. Two hybrid models were also developed. The first concatenated a CNN's sentiment data with an LSTM's stock data, but this model had a higher Mean Squared Error (MSE) than the standalone LSTM, possibly due to incomplete sentiment data. The second hybrid model, a CNN-LSTM that processed stock data through both layers, showed the best performance, with the lowest MSE among all models, demonstrating its superior ability to predict stock price trends.

This study successfully utilized a hybrid CNN-LSTM model to predict Google stock prices by integrating historical price data with sentiment analysis from Reddit. The CNN model, while accurately classifying sentiment, did not perfectly match the rule-based Vader system, suggesting that machine learning models can outperform rule-based methods in this context. A weighting system was also applied to address class imbalances and improve the model's performance on minority classes. For stock price prediction, the LSTM model proved robust, accurately predicting closing prices even after redundant features were dropped. Data scaling was crucial to prevent features like trade volume from causing overfitting. The hybrid model, combining both sentiment and stock data, consistently outperformed the standalone models in prediction accuracy. This research aligns with several studies on hybrid models for stock market prediction. The use of LSTM for time-series data and CNN for sentiment analysis is consistent with the work of [15] and [9], who also used hybrid models to combine different data types. However, this study's specific focus on integrating real-time sentiment data from Reddit distinguishes it from approaches that rely solely on technical indicators or historical prices, such as those by [24] and [8]. The evaluation metrics (RMSE, MSE) are standard, mirroring those used by [20] and [15]. The finding that a hybrid model outperforms standalone models is also a common conclusion in the literature, as noted by [14] and [9]. The inclusion of sentiment analysis proved beneficial, which contrasts with [23]'s finding that sentiment data had a negative effect on predictions for some markets. The model's reliance on LSTM for handling time-series dependencies is well-supported by studies from [10] and [18].

The hybrid LSTM-CNN approach is a key strength, leveraging the best of both models to capture complex patterns. However, a major weakness is the model's dependence on the quality and potential ambiguity of sentiment data, which can lead to misclassifications. Future research could explore more advanced models like Transformers (e.g., BERT) for more nuanced sentiment analysis. The approach could also be expanded to predict a broader range of financial assets, like cryptocurrencies or commodities, and incorporate additional data sources, such as news articles, to further enrich the sentiment analysis framework.

5. Conclusion

This study investigated stock price prediction and sentiment analysis using two datasets: Google stock prices and Reddit comments. We developed a hybrid model that integrates an LSTM for time-series stock data and a CNN for sentiment data, and compared its performance to that of standalone models. Our findings show that the hybrid CNN-LSTM model outperformed individual models, highlighting the benefit of combining diverse datasets for improved forecasting. The results confirm that incorporating sentiment analysis enhances stock market predictions, as the hybrid model demonstrated superior performance (lower RMSE and MSE) compared to standalone models. This suggests that sentiment data provides valuable context for predicting market movements.

However, the study faced several limitations. The sentiment data was imbalanced, with a disproportionately high number of neutral comments, which required class weighting to improve the CNN model's performance on minority classes. The sentiment analysis was also based on a narrow scope of data (only top Reddit comments), which may not be comprehensive. Additionally, the LSTM model had difficulty capturing minor fluctuations in stock prices, and the large scale of trade volume data posed a challenge, though this was addressed with proper scaling. The hybrid CNN-LSTM model showed significant promise by outperforming standalone models in both stock prediction and sentiment classification. This research provides a strong case for integrating AI in financial analysis, specifically by combining diverse data streams. Future work should focus on expanding the sentiment dataset to include more comprehensive sources and fine-tuning model parameters to further enhance prediction accuracy.

Conflict of Interest

The authors declare there is no existing conflict of interest.

References

1.Al Ridhawi, M. (2021). *Stock Market Prediction Through Sentiment Analysis of Social-Media and Financial Stock Data Using Machine Learning*. PhD thesis, Université d'Ottawa/University of Ottawa.

2.Bojer, C. S. and Meldgaard, J. P. (2021). Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting*, 37(2):587–603.

3.Celik, A. and Eltawil, A. M. (2024). At the dawn of generative ai era: A tutorial-cum-survey on new frontiers in 6g wireless intelligence. *IEEE Open Journal of the Communications Society*.

4.Choi, K., Yi, J., Park, C., and Yoon, S. (2021). Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines. *IEEE access*, 9:120043–120065.

5.Dahal, K. R., Pokhrel, N. R., Gaire, S., Mahatara, S., Joshi, R. P., Gupta, A., Banjade, H. R., and Joshi, J. (2023). A comparative study on effect of news sentiment on stock price prediction with deep learning architecture. *Plos one*, 18(4):e0284695.

6.Dwivedi, Y. K., Kshetri, N., Hughes, L., Slade, E. L., Jeyaraj, A., Kar, A. K., Baabdullah, A. M., Koohang, A., Raghavan, V., Ahuja, M., et al. (2023). Opinion paper:"so what if chatgpt wrote it?" multidisciplinary perspectives on opportunities, challenges and implications of generative conversational ai for research, practice and policy. *International Journal of Information Management*, 71:102642.

7.Himeur, Y., Elnour, M., Fadli, F., Meskin, N., Petri, I., Rezgui, Y., Bensaali, F., and Amira, A. (2023). Ai-big data analytics for building automation and management systems: a survey, actual challenges and future perspectives. *Artificial Intelligence Review*, 56(6):4929–5021.

8.Ishwarappa and Anuradha, J. (2021). Big data based stock trend prediction using deep cnn with reinforcement-lstm model. *International Journal of System Assurance Engineering and Management*, pages 1–11.

9.Kim, T. and Kim, H. Y. (2019). Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data. *PloS one*, 14(2):e0212320.

10.Kumar, S., Gopi, T., Harikeerthana, N., Gupta, M. K., Gaur, V., Krolczyk, G. M., and Wu, C. (2023). Machine learning techniques in additive manufacturing: a state of the art review on design, processes and production control. *Journal of Intelligent Manufacturing*, 34(1):21–55.

11.LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

12.Li, A. W. and Bastos, G. S. (2020). Stock market forecasting using deep learning and technical analysis: a systematic review. *IEEE access*, 8:185232–185242.

13.Liu, S., Zhang, X., Wang, Y., and Feng, G. (2020). Recurrent convolutional neural kernel model for stock price movement prediction. *Plos one*, 15(6):e0234206.

14.Mehtab, S., Sen, J., and Dasgupta, S. (2020). Robust analysis of stock price time series using cnn and lstm-based deep learning models. In *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 1481–1486. IEEE.

15.Moodi, F., Jahangard-Rafsanjani, A., and Zarifzadeh, S. (2024). A cnn-lstm deep neural network with technical indicators and sentiment analysis for stock price forecastings. In *2024 20th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)*, pages 1–6. IEEE.

16.Mukherjee, S., Sadhukhan, B., Sarkar, N., Roy, D., and De, S. (2023). Stock market prediction using deep learning algorithms. *CAAI Transactions on Intelligence Technology*, 8(1):82–94.

17.Niu, H., Pan, Q., and Xu, K. (2023). Hybrid deep learning models with multi-classification investor sentiment to forecast the prices of china's leading stocks. *Plos one*, 18(11):e0294460.

18.Pagliaro, A. (2023). Forecasting significant stock market price changes using machine learning: extra trees classifier leads. *Electronics*, 12(21):4551.

19.Patel, A. (2019). Fifth-generation warfare and the definitions of peace. *The Journal of Intelligence, Conflict, and Warfare*, 2(2):15–28.

20.Rasheed, J., Jamil, A., Hameed, A. A., Ilyas, M., Özyavaş, A., and Ajlouni, N. (2020). Improving stock prediction accuracy using cnn and lstm. In *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, pages 1–5. IEEE.

21.Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, 2(3):160.

22.Schoormann, T., Strobel, G., Möller, F., Petrik, D., and Zschech, P. (2023). Artificial intelligence for sustainability—a systematic review of information systems literature. *Communications of the Association for Information Systems*, 52(1):8.

23.Shi, Y., Zheng, Y., Guo, K., and Ren, X. (2021). Stock movement prediction with sentiment analysis based on deep learning networks. *Concurrency and Computation: Practice and Experience*, 33(6):e6076.

24.Singh, S., Gutta, S., and Hadaegh, A. (2021). *Stock Prediction Using Machine Learning*. PhD thesis, California State University San Marcos.

25.Toorajipour, R., Sohrabpour, V., Nazarpour, A., Oghazi, P., and Fischl, M. (2021). Artificial intelligence in supply chain management: A systematic literature review. *Journal of Business Research*, 122:502–517.

26.Usmani, S. and Shamsi, J. A. (2023). Lstm based stock prediction using weighted and categorized financial news. *PloS one*, 18(3):e0282234.

27.Zhang, X. (2024). Analyzing financial market trends in cryptocurrency and stock prices using cnn-lstm models. *Preprints*.