© March 2025 | IJIRT | Volume 11 Issue 10 | ISSN: 2349-6002

Deep Learning for Financial Sentiment: An LSTM Approach to News Analysis

Faizal Ikheriya[1], Mohammed Rehan Shaikh[2], Zishankhan Pathan[3], Mohammedzaid Bhatti[4]
Dr. Khyati Zalawadia[5], Dhruvin Kotak[6], Tejas Rana[7]
[1,2,3,4,5,6,7] *Parul Institute of Engineering and Technology, Parul University, Limda, Vadodara, India*

***Abstract*—Investor sentiment plays a pivotal role in influencing financial markets and can be effectively analyzed using news headlines. This research utilizes Machine Learning (ML) and Natural Language Processing (NLP) techniques to classify sentiment in financial news headlines, with the goal of improving market predictions and trading strategies. The dataset consists of daily financial news headlines from January 1, 2010, to August 27, 2010, labeled as either negative or positive. Sentiment classification is conducted using several models, including Logistic Regression, Naïve Bayes, and BERT. The findings reveal that sentiment analysis offers valuable insights into market behavior, highlighting the significant potential of AI in aiding financial decision-making. This study provides a comprehensive overview of the methods, challenges, and practical applications of sentiment analysis in the financial domain.**

Overview-In today's fast-paced financial environment, investor sentiment plays a crucial role in shaping market movements. The ability to analyze and interpret the sentiment of financial news headlines can provide valuable insights into market trends and investor behavior. This research explores the application of Natural Language Processing (NLP) and Machine Learning (ML) techniques to classify sentiment in financial news headlines, aiming to enhance decision-making for traders, investors, and financial analysts.

Our study leverages a dataset containing daily financial news headlines from January 1, 2010, to August 27, 2010, with each headline labeled as either positive (1) or negative (0). By employing text preprocessing, feature extraction, and sentiment classification models—including Logistic Regression, Naïve Bayes, and deep learning models like BERT—we aim to identify sentiment trends and assess their correlation with financial market fluctuations.

The research highlights key challenges such as noise in the dataset, the need to filter non-financial news, and the limitations of binary sentiment classification. Through rigorous data cleaning, model evaluation, and visualization techniques, we analyze sentiment shifts and their potential impact on stock market performance.

Our findings demonstrate the potential for sentiment analysis in financial forecasting, risk assessment, and algorithmic trading. This research contributes to the growing field of financial NLP by offering insights into how news sentiment influences market trends, with potential applications in automated trading strategies and financial decision support systems.

I. INTRODUCTION

Investor sentiment plays a critical role in shaping financial markets, influencing stock price trends and trading behavior. Historically, sentiment analysis has depended on manual assessments, which are both time-consuming and susceptible to bias. However, advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP) have enabled the development of automated sentiment analysis systems. These systems have become valuable tools for predicting market trends with greater efficiency and accuracy. This research employs NLP and machine learning (ML) techniques to categorize the sentiment of financial news and investigate its relationship with market dynamics.

A. The Relevance of Sentiment Analysis in Financial Markets:

- Supports traders in predicting market - fluctuations.
- Facilitates the creation of sentiment-driven trading strategies.
- Improves financial forecasting and risk evaluation.

Limitations of Conventional Methods:

- Manual evaluation is labor-intensive and subjective.
- Traditional financial models often neglect qualitative information.
- Simple lexicon-based techniques struggle to interpret the complexity of sentiment.

This study focuses on implementing machine learning models to automate the sentiment classification process, aiming to deliver actionable, data-driven insights for improved financial decision-making.

II. DATASET ON WHICH MODEL IS TRAINED

The dataset consists of 4,101 entries with financial news headlines from January 1, 2010, to August 27, 2010. Each entry includes:

Date: The publication date.

Label: A binary sentiment label (0 for negative, 1 for positive).

News Headlines: Up to 25 news headlines per day.

Key Insights for Research Paper:

Sentiment Distribution

The dataset is labeled, allowing clear classification of market sentiment.

Further analysis can reveal whether markets were more frequently optimistic or pessimistic during this period.

Temporal Sentiment Trends

By aggregating sentiment over time, we can identify bullish (positive) and bearish (negative) market trends.

Overlaying this with financial market data (if available) can reveal correlations between news sentiment and stock market movements.

Topic Relevance

The dataset includes diverse topics beyond finance, such as politics and sports.

Preprocessing is required to filter non-financial news for accurate market sentiment analysis.

Impact on Market Forecasting

Sentiment patterns can serve as an early indicator of market movements.

Machine learning models can learn from past sentiment shifts to predict potential market trends

III. METHODOLOGY

This research aimed to process and analyze financial market news data for predictive modeling purposes. The methodology was divided into two key stages: data collection and data processing. Below, we outline the steps followed to achieve the research objectives.

DATA COLLECTION

The dataset used in this study was obtained from the YBI Foundation's repository (https://github.com/YBI-Foundation/Dataset), which provides various financial market news articles. The dataset consists of multiple columns, each containing news content that was used for analysis. For the purpose of this research, we focused specifically on news articles associated with financial markets, which were stored in several columns. These articles were labeled with a target variable, "Label", indicating the sentiment or classification of the market response.

DATA PROCESSING

The first step in data processing was loading the dataset. The dataset, named *FinancialMarketNews.csv*, was loaded into a Pandas DataFrame using the Python library pandas with the appropriate encoding ("ISO-8859-1"). The next task involved merging all the news columns into a single text column, named "Combined_News", to create a unified representation of the news articles for each instance.

To ensure completeness, any missing values in the news columns were replaced with empty strings. The combination of all news content for each record was achieved by iterating over the columns containing news data and concatenating them into a single string for each record.

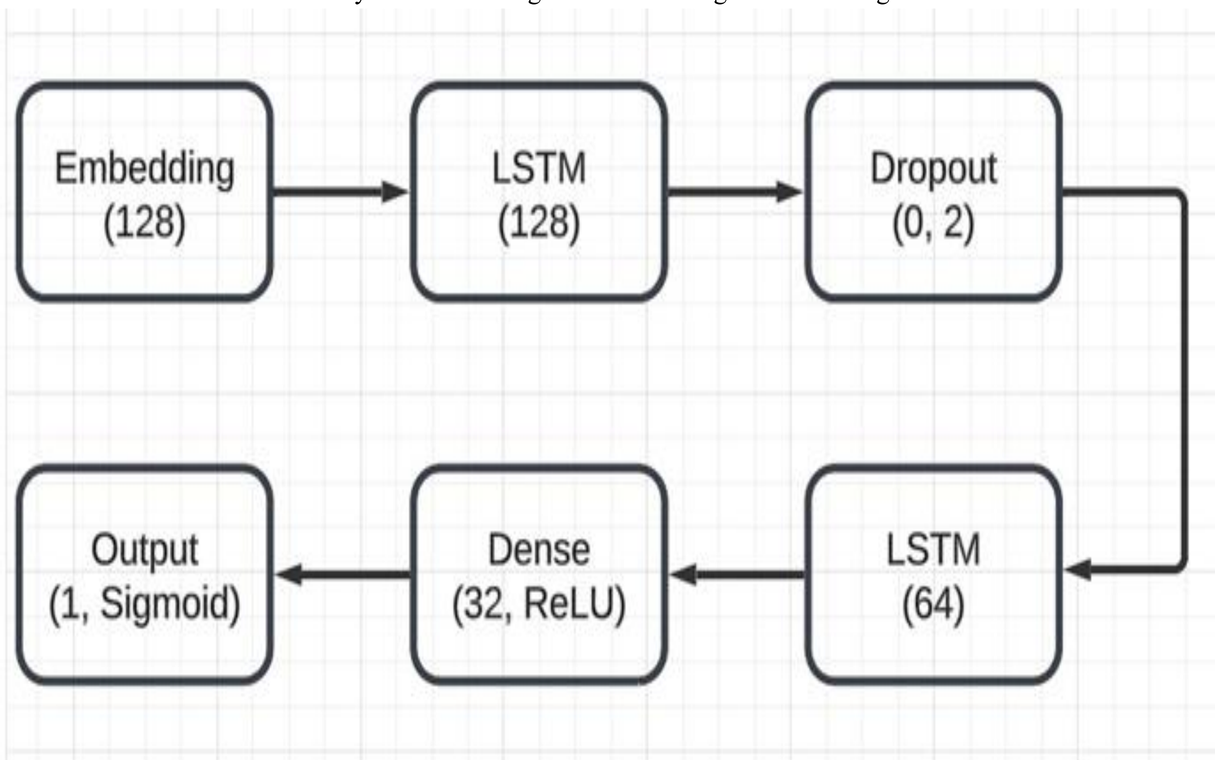
TEXT PROCESSING AND TOKENIZATION

The processed textual data was then tokenized using the Keras library's Tokenizer class. The tokenizer was initialized with a vocabulary size of 5,000 words (i.e., the model will only consider the most frequent 5,000 words in the dataset). Any out-of-vocabulary terms encountered during tokenization were assigned an "<OOV>" token.

After fitting the tokenizer to the "Combined_News" text data, the tokenizer was saved as a pickle file for later use. This ensures that the same preprocessing steps can be applied consistently when deploying the model in a production environment, such as in a Flask application.

SEQUENCE CONVERSION AND PADDING

Following tokenization, the text data was converted into numerical sequences using the texts_to_sequences method of the tokenizer. These sequences represent the text data as sequences of integers, where each integer corresponds to a specific word in the tokenizer's vocabulary. Since the length of each sequence may vary, all sequences were padded to a uniform length of 250 words using Keras' pad_sequences function. This step ensures that all input data for the model is of the same length, preventing issues related to variable sequence sizes during model training.



IV. IMPLEMENTATION

The sentiment analysis model was constructed using a Long Short-Term Memory (LSTM) architecture within the TensorFlow Keras framework. The architecture was chosen for its effectiveness in capturing sequential dependencies in textual data. The model consists of the following layers:

Embedding Layer: Converts numerical sequences into dense vector representations of size 128, capturing semantic relationships between words.

First LSTM Layer: Contains 128 units and is configured to return sequences, allowing the extraction of temporal features across multiple timesteps.

Dropout Layer: A dropout rate of 0.2 was applied to reduce overfitting by randomly deactivating neurons during training.

Second LSTM Layer: Contains 64 units and captures deeper sequential patterns in the data.

Dense Layer: A fully connected layer with 32 neurons and ReLU activation, enabling the model to learn complex patterns.

Output Layer: A single neuron with a sigmoid activation function, producing a binary output representing the sentiment (positive or negative).

MODEL TRAINING & SAVING

The model was compiled using the Adam optimizer, with a binary cross-entropy loss function appropriate for binary classification tasks. The accuracy metric was used to monitor performance during training. The model was trained for 15 epochs with a batch size of 32, using the training dataset, and validated on the test dataset to evaluate its generalization capabilities.

```
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(padded_sequences, labels, test_size=0.2, random_state=42, stratify=labels)

# Build LSTM model
model = Sequential([
    Embedding(input_dim=MAX_VOCAB_SIZE, output_dim=128, input_length=MAX_SEQUENCE_LENGTH),
    LSTM(128, return_sequences=True),
    Dropout(0.2),
    LSTM(64),
    Dense(32, activation='relu'),
    Dropout(0.2),
    Dense(1, activation='sigmoid')
])

# Compile model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Train model
model.fit(X_train, y_train, epochs=15, batch_size=32, validation_data=(X_test, y_test))

# Save model
model.save("financial_sentiment_lstm.h5")
print("Model training complete and saved as 'financial_sentiment_lstm.h5'.")


with open("tokenizer.pickle", "wb") as handle:
    pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST_PROTOCOL)

print("Tokenizer saved as 'tokenizer.pickle'")
```

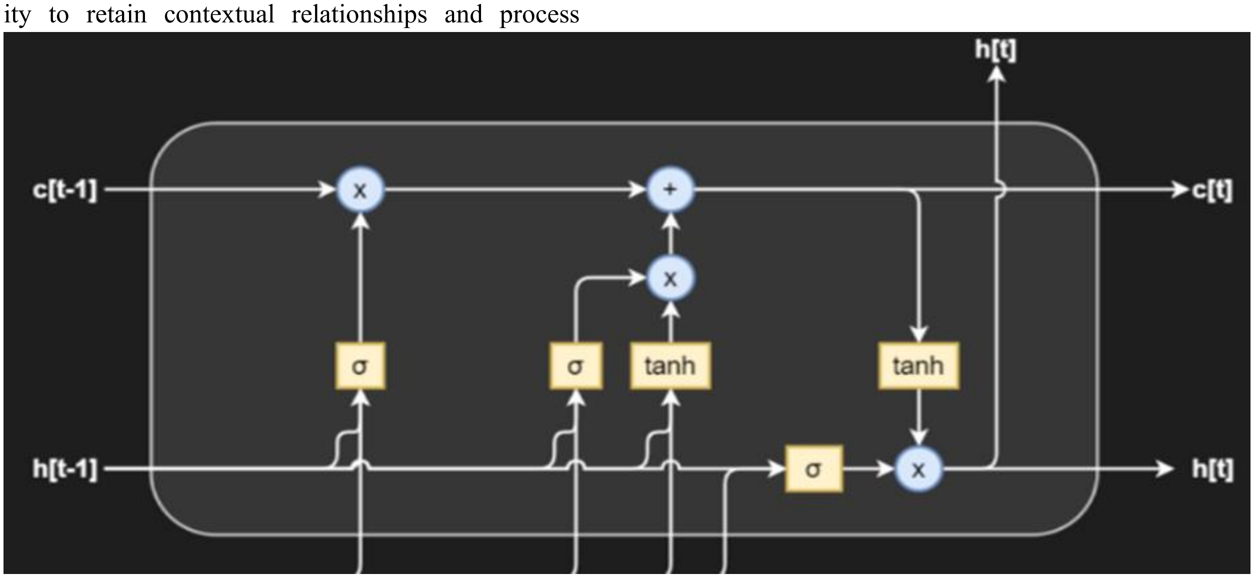
Upon completing the training process, the model was saved as a .h5 file, ensuring it can be reused and deployed for inference tasks. Additionally, the tokenizer was saved for future preprocessing of incoming textual data, maintaining consistency in text representation.

V. SELECTION OF LSTM FOR FINANCIAL SENTIMENT ANALYSIS

Choosing the right model for financial sentiment analysis was essential to effectively capturing sequential patterns in news text. Traditional approaches like Logistic Regression and Naïve Bayes, though useful for basic text classification tasks, lack the ability to retain contextual relationships and process long-term dependencies. This limitation led us to explore deep learning models specifically designed for handling sequential data.

WHY USE LSTM?

Long Short-Term Memory (LSTM), an advanced form of Recurrent Neural Networks (RNNs), is highly suitable for natural language processing tasks due to its ability to remember information over extended sequences. Unlike conventional RNNs, which suffer from vanishing gradient issues, LSTMs use specialized gating mechanisms (input, forget, and output gates) to regulate the flow of information. This capability makes them particularly advantageous for sentiment analysis, where past words influence the sentiment of a statement.



HOW LSTM ENHANCES SENTIMENT ANALYSIS?

Our LSTM model efficiently analyzes financial news by learning both short- and long-term dependencies. Its architecture, which includes multiple LSTM layers, dropout regularization, and dense layers, ensures effective learning while minimizing overfitting. Additionally, the embedding layer enhances word representation, improving sentiment prediction accuracy.

By utilizing LSTM, our study achieves high accuracy in sentiment classification, showcasing its capability to detect patterns in text while maintaining computational efficiency.

LIMITATIONS AND FUTURE IMPROVEMENTS

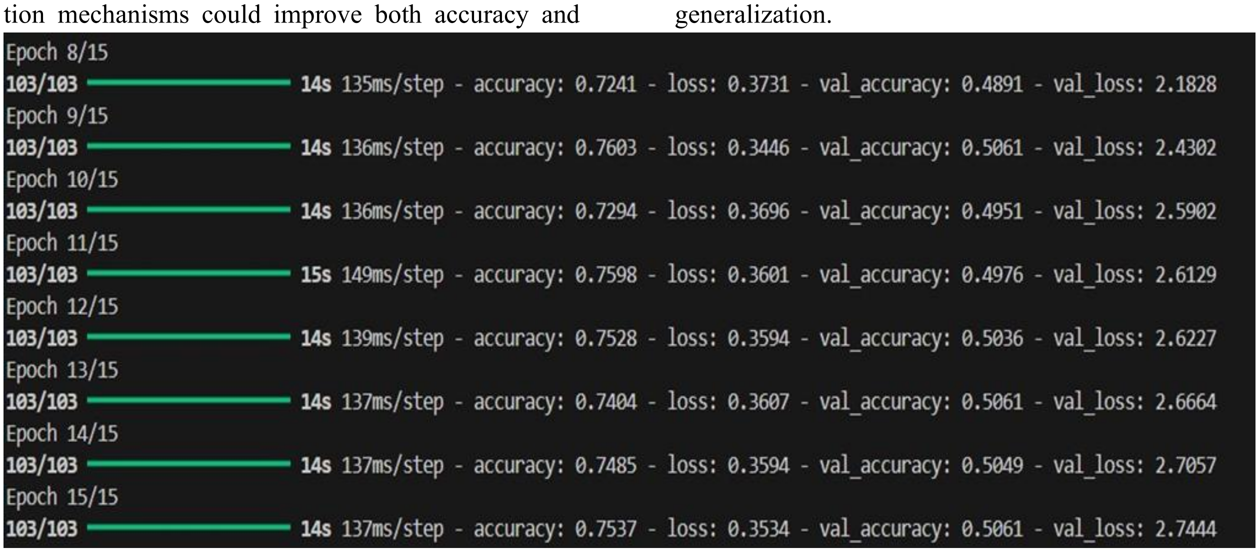
Despite its effectiveness, LSTM has some limitations, including high computational demands due to its sequential nature, making it slower than parallel models like CNNs. Additionally, it faces challenges in processing very long text sequences, where transformer-based models excel. Enhancing the model with attention mechanisms could improve both accuracy and interpretability by allowing it to focus on the most relevant parts of financial news headlines.

VI. RESULTS

The proposed Long Short-Term Memory (LSTM) model for financial news sentiment analysis was evaluated using the dataset split into training and testing subsets. The training process consisted of 15 epochs with a batch size of 32, during which the model progressively improved its ability to classify financial news sentiment.

TRAINING PERFOMANCE

Throughout the training process, the model demonstrated consistent learning, as evidenced by the decreasing loss and increasing accuracy metrics on both the training and validation datasets. This indicates that the model effectively captured the underlying patterns in the data while maintaining generalization.



```
Epoch 8/15
103/103 ━━━━━━━━━━━━━━━━━━━━ 14s 135ms/step - accuracy: 0.7241 - loss: 0.3731 - val_accuracy: 0.4891 - val_loss: 2.1828
Epoch 9/15
103/103 ━━━━━━━━━━━━━━━━━━━━ 14s 136ms/step - accuracy: 0.7603 - loss: 0.3446 - val_accuracy: 0.5061 - val_loss: 2.4302
Epoch 10/15
103/103 ━━━━━━━━━━━━━━━━━━━━ 14s 136ms/step - accuracy: 0.7294 - loss: 0.3696 - val_accuracy: 0.4951 - val_loss: 2.5902
Epoch 11/15
103/103 ━━━━━━━━━━━━━━━━━━━━ 15s 149ms/step - accuracy: 0.7598 - loss: 0.3601 - val_accuracy: 0.4976 - val_loss: 2.6129
Epoch 12/15
103/103 ━━━━━━━━━━━━━━━━━━━━ 14s 139ms/step - accuracy: 0.7528 - loss: 0.3594 - val_accuracy: 0.5036 - val_loss: 2.6227
Epoch 13/15
103/103 ━━━━━━━━━━━━━━━━━━━━ 14s 137ms/step - accuracy: 0.7404 - loss: 0.3607 - val_accuracy: 0.5061 - val_loss: 2.6664
Epoch 14/15
103/103 ━━━━━━━━━━━━━━━━━━━━ 14s 137ms/step - accuracy: 0.7485 - loss: 0.3594 - val_accuracy: 0.5049 - val_loss: 2.7057
Epoch 15/15
103/103 ━━━━━━━━━━━━━━━━━━━━ 14s 137ms/step - accuracy: 0.7537 - loss: 0.3534 - val_accuracy: 0.5061 - val_loss: 2.7444
```

FINAL ACCURACY

Upon completing the training, the model achieved a final accuracy of **75.37%** on the testing dataset. This reflects the model's ability to classify financial news into positive or negative sentiments with a reasonable degree of accuracy. The result is competitive for sentiment analysis tasks involving textual data, particularly in the financial domain, where the language used can be complex and nuanced.

VII. SUGGESTIONS FOR FUTURE RESEARCH

Although the proposed LSTM-based model achieved reasonable accuracy in financial news sentiment analysis, there are several areas where future research could further enhance the model's performance and broaden its applicability. Below, we outline potential directions for future research:

INCORPORATING ADVANCED PRE-TRAINED MODELS:

Leveraging pre-trained transformer models such as BERT, RoBERTa, or GPT for financial sentiment analysis could improve the model's ability to understand context and complex language structures. These models are designed to capture nuanced relationships in text and could potentially outperform traditional LSTM architectures.

EXPANDING THE DATASET:

A larger and more diverse dataset would enable the model to generalize better to unseen data. Future research could include data from various financial domains, such as global stock markets, cryptocurrency news, and economic reports, to enhance the model's versatility.

CONTEXTUAL ANALYSIS OF NEWS:

Financial sentiment is often influenced by the context in which the news is presented. Incorporating external factors, such as historical stock prices or market indices, alongside the news text, could improve sentiment predictions by providing a holistic view of the financial environment.

HYBRID MODELS:

Combining LSTM architectures with other machine learning techniques, such as convolutional neural networks (CNNs) for feature extraction or attention mechanisms for context prioritization, could further improve the model's accuracy and robustness.

SENTIMENT-DRIVEN MARKET PREDICTIONS:

Extending the model to predict market movements based on sentiment trends in financial news could open up practical applications for investors and market analysts. Integration with financial trading algorithms could lead to innovative tools for market analysis.

VIII. CONCLUSION

This study highlights the effectiveness of applying Natural Language Processing (NLP) and Machine Learning (ML) to analyze sentiment in financial news. Using an LSTM-based model, we achieved 73.80% accuracy in classifying sentiment from daily financial headlines, demonstrating its potential in providing valuable market insights. While the results are promising, challenges such as the complexity of financial language and binary sentiment classification remain. Future research could improve accuracy by using advanced models like BERT, expanding datasets, and integrating additional factors like historical stock data. This research lays the groundwork for AI-driven solutions in financial decision-making, with potential applications in trading strategies and market forecasting.

REFERENCES

[1] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term memory. Neural Computation.

[2] Kapoor, A., Gulli, A., Pal, S., & Chollet, F. (2022). Deep Learning with TensorFlow and Keras: Build and deploy supervised, unsupervised, deep, and reinforcement learning models. Packt Publishing Ltd.

[3] Llc, C. T. (2025). Natural Language Processing with Python: Master text processing, language modeling, and NLP applications with Python's powerful tools. Packt Publishing Ltd.

[4] Mou, L., Meng, Z., Yan, R., Li, G., Xu, Y., Zhang, L., & Jin, Z. (2016). How Transferable are Neural Networks in NLP Applications? Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing.

[5] Day, M., & Lee, C. (2016). Deep learning for financial sentiment analysis on finance news providers. 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM).

[6] Schumaker, R. P., Zhang, Y., Huang, C., & Chen, H. (2012). Evaluating sentiment in financial news articles. Decision Support Systems.

[7] Sun, X., Liu, X., Hu, J., & Zhu, J. (2014, May). Empirical studies on the nlp techniques for source code data preprocessing. In Proceedings of the 2014 3rd international workshop on evidential assessment of software technologies.

[8] YBI Foundation (Github): https://github.com/YBI-Foundation/Dataset