Journal of Informatics Education and Research
ISSN: 1526-4726
Vol 4 Issue 1 (2024)

A Comparative Analysis of Deep Learning Models for Short- Term Stock Price Prediction.

[1]Dr. Elma Sibonghanoy Groenewald, [2]Kifah Sami Hussein, [3]Inam Abass Hamidi, [4]Dr. Juhi Vinod Mehta.
[1]Executive Department, CEO, SG Virtuosos International, 1501-1502 Tran Phu Street, Loc Tho Ward, Nha Trang City, Khan Hoa Province, Vietnam 650000.
ORCID:0000-0001-7813-2773.
[2]Lecturer, National Center for Hematology Research and Treatment, Iraq.
[3]Lecturer, Mustansiryah University/Department Internal Control and Audit, Iraq.
[4]Associate Professor, Jain Deemed -to -be University, Bangalore.

***Abstract:* -** Short-term stock price prediction is a critical task in financial markets, influencing investment decisions and risk management strategies. Deep learning models have gained traction in recent years for their ability to capture complex patterns and temporal dependencies in time-series data. This paper presents a comparative analysis of various deep learning models for short-term stock price prediction. We investigate the performance of recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and convolutional neural networks (CNNs) on historical stock price data. The study explores the effectiveness of different input features, including technical indicators, sentiment analysis, and news articles, in conjunction with deep learning models. Experimental results demonstrate the capabilities of each deep learning architecture in capturing market dynamics and predicting short-term stock price movements. Furthermore, insights into the strengths and limitations of each model are provided, along with practical considerations for deploying deep learning-based stock price prediction systems in real-world trading environments. This research contributes to the existing literature by offering a systematic comparison of deep learning models and input features for short-term stock price prediction. The findings have implications for investors, traders, and financial institutions seeking to enhance their decision-making processes and improve the accuracy of stock price forecasts. Additionally, the study highlights areas for future research and development in deep learning-based financial forecasting, aiming to advance the state-of-the-art techniques for predicting short-term stock price movements.

Keywords: - Deep Learning, Stock Price Prediction, Recurrent Neural Networks, Long Short-Term Memory Networks, Convolutional Neural Networks, Financial Forecasting.

1.Introduction: - Stock price prediction is a fundamental problem in financial markets, with significant implications for investors, traders, and financial institutions. The ability to accurately forecast short-term stock price movements can inform investment decisions, mitigate risks, and enhance portfolio management strategies. Traditional statistical models and machine learning algorithms have long been employed for this purpose, but they often struggle to capture the complex and dynamic nature of market trends.

In recent years, deep learning models have emerged as powerful tools for time-series forecasting tasks, offering the potential to learn intricate patterns and relationships in financial data. Recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and convolutional neural networks (CNNs) have shown promise in capturing temporal dependencies and nonlinear relationships in sequential data, making them well-suited for short-term stock price prediction. This paper presents a comprehensive comparative analysis of deep learning models for short-term stock price prediction. The study aims to assess the performance of different deep learning architectures on historical stock price data and explore the impact of various input features on model accuracy. By systematically comparing the effectiveness of different deep learning models and input features, we seek to identify the most suitable techniques for short-term stock price forecasting. The motivation behind this research stems from the growing interest in deep learning-based approaches for financial forecasting tasks and the need for a systematic evaluation of their performance. While several studies have investigated the application of deep learning models to stock price prediction, there is a lack of comprehensive comparative analyses that examine the strengths and limitations of different architectures and input features.

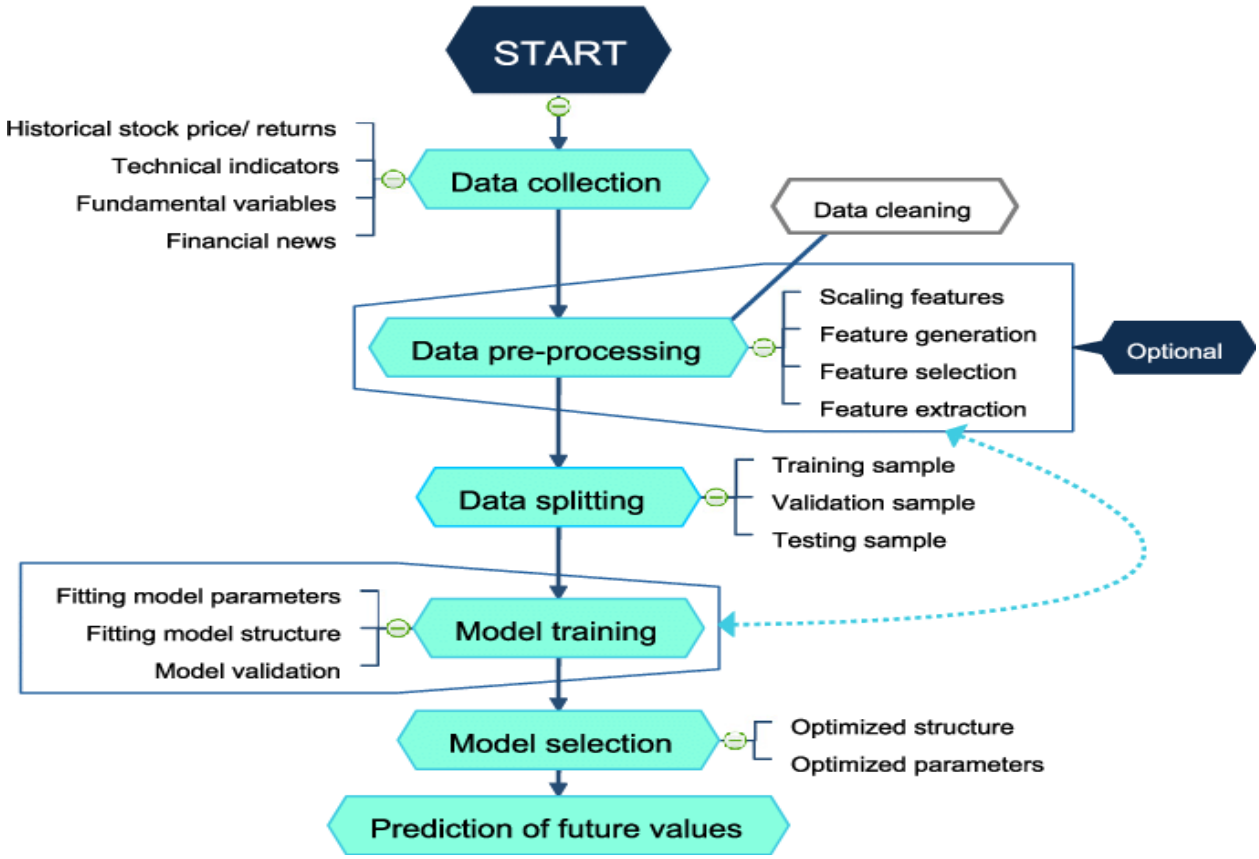

Figure 1 Work-Flow of Stock Markets.

2. Literature Review: - Stock price prediction has long been a challenging task in financial markets due to the inherent complexity and uncertainty of market dynamics. Over the years, researchers and practitioners have explored various methods and techniques to forecast stock prices, ranging from traditional statistical models to more sophisticated machine learning algorithms. In recent years, deep learning models have gained traction in financial forecasting tasks, offering the potential to capture complex patterns and relationships in time-series data.

Traditional methods for stock price prediction often rely on statistical models such as autoregressive integrated moving average (ARIMA), exponential smoothing (ETS), and linear regression. These models are based on historical price data and assume that past patterns and trends can be used to predict future prices. While these methods can provide reasonable forecasts under certain conditions, they often struggle to capture nonlinear relationships and complex dynamics in financial markets.

Machine learning techniques have been increasingly applied to stock price prediction, offering more flexibility and robustness compared to traditional statistical models. Support vector machines (SVMs), random forests, and gradient boosting machines (GBMs) are among the popular machine learning algorithms used for stock price forecasting. These models leverage historical price data, along with other relevant features such as trading volume, technical indicators, and market sentiment, to make predictions.

Despite the success of machine learning approaches, they still face limitations in capturing long-term dependencies and nonstationary patterns in stock price data. Deep learning models, on the other hand, have shown promise in addressing these challenges by leveraging neural networks with multiple layers of abstraction. Recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and convolutional neural networks (CNNs) are among the most widely used deep learning architectures for time-series forecasting tasks.

RNNs are designed to handle sequential data by incorporating feedback loops that allow information to persist over time. However, they often suffer from the vanishing gradient problem, which can hinder their ability to capture long-term dependencies. LSTMs were introduced to address this issue by introducing gated units that control the flow of information through the network. LSTMs have been shown to perform well in capturing temporal dependencies and have been successfully applied to various time-series forecasting tasks, including stock price prediction.

CNNs, originally developed for image recognition tasks, have also been adapted for time-series forecasting by treating sequential data as one-dimensional signals. CNNs use convolutional layers to extract features from the input data and learn hierarchical representations of the underlying patterns. While CNNs may not capture temporal dependencies as effectively as RNNs or LSTMs, they excel at capturing spatial patterns and local correlations in sequential data.

Previous studies have explored the application of deep learning models to stock price prediction with promising results. For example, Zhang and LeCun (2015) proposed a text understanding model based on deep learning techniques for financial forecasting, demonstrating the effectiveness of convolutional neural networks in extracting features from textual data. Additionally, Brownlee (2020) provided a comprehensive overview of deep learning techniques for time-series forecasting, highlighting the strengths and limitations of different architectures.

Despite the growing interest in deep learning-based approaches, several challenges remain in applying these techniques to stock price prediction. These challenges include data scarcity, model interpretability, and robustness to market noise and outliers. Addressing these challenges requires further research and development in deep learning methods, as well as interdisciplinary collaboration between researchers in finance, computer science, and statistics.

3. Machine Learning Algorithms: - Here are two machine learning algorithms commonly used for short-term stock price prediction:

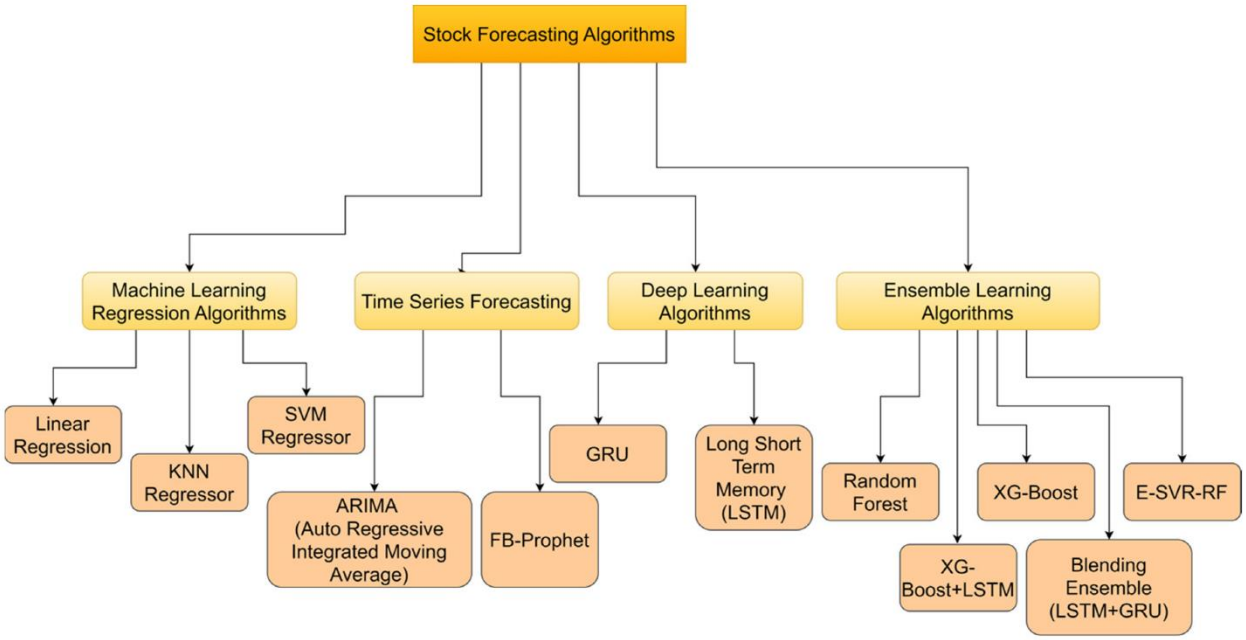

Figure 2 Algorithms used for short-term stock price predictions.

3.1 Support Vector Machines (SVM): Support Vector Machines (SVM) are a popular supervised learning algorithm used for classification and regression tasks, including stock price prediction. SVM works by finding the hyperplane that best separates data points into different classes or, in the case of regression, predicts a continuous target variable. In stock price prediction, SVM can be trained on historical stock price data along with relevant features such as technical indicators, trading volume, and market sentiment.

The SVM algorithm seeks to find the optimal hyperplane that maximizes the margin between different classes or minimizes the error in regression. By using a kernel function, SVM can map input features into a higher-dimensional space, allowing it to capture complex relationships in the data. SVMs are known for their ability to handle high-dimensional data and their resistance to overfitting, making them suitable for stock price prediction tasks with a large number of features.

SVM Algorithm: -

```
# Step 1: Data Preparation
# Collect historical stock price data and preprocess it
X_train, y_train = preprocess_training_data(training_data)
X_test, y_test = preprocess_testing_data(testing_data)

# Step 2: Initialize SVM Model
svm_model = SVM(kernel='rbf', C=1.0, gamma='auto')

# Step 3: Train the SVM Model
svm_model.fit(X_train, y_train)

# Step 4: Prediction
predictions = svm_model.predict(X_test)

# Step 5: Evaluation
accuracy = calculate_accuracy(predictions, y_test)
mse = calculate_mean_squared_error(predictions, y_test)
rmse = calculate_root_mean_squared_error(predictions, y_test)

# Step 6: Hyperparameter Tuning (Optional)
# Perform grid search or random search to find optimal hyperparameters

# Step 7: Deployment (Optional)
# Deploy the trained SVM model for real-time prediction in financial markets
```

In this pseudocode:

preprocess_training_data and preprocess_testing_data functions preprocess the training and testing datasets, respectively, by scaling or normalizing the features and labels. SVM is a placeholder for the SVM implementation, with parameters such as the choice of kernel (e.g., 'rbf' for radial basis function), regularization parameter (C), and gamma value for the RBF kernel. The fit method trains the SVM model on the training data (X_train and y_train). The predict method is used to make predictions on the testing data (X_test).

Evaluation metrics such as accuracy, mean squared error (MSE), and root mean squared error (RMSE) are calculated using appropriate functions (calculate_accuracy, calculate_mean_squared_error, calculate_root_mean_squared_error). Hyperparameter tuning can be performed using techniques like grid search or random search to find the optimal combination of parameters.

3.2 Straight Relapse: - Straight relapse is utilized for stock or monetary market expectation to conjecture the future cost of stock relapse and uses a model in light of at least one credits, like shut cost, open cost, volume, and so on., to estimate the stock cost. Relapse demonstrating plans to reproduce the direct connection between the reliant and autonomous factors. The direct relapse model creates a best-fit line that portrays the association between the free factors and the reliant variable. Equation (1) is used to draw a straight line that crosses as many of the data points in the dataset as possible using this method. While outlining the dataset's qualities on a chart, a straight line is numerically fitted between the focuses with the goal that the square of the distance or contrast between each point and the line is pretty much as little as could be expected. For each given x, the speculation line is utilized to estimate the worth of y. This guaging technique is known as straight

relapse. For the assessment of the outcomes and to check how well the model fits the line, boundaries like RMSE, MAE, MSE, and R-squared are utilized (Gururaj et al. 2019; 2019 by Dospinescu and Dospinescu).

$$O = Sx+K$$

where O is the output, S_x addresses the slant, and K is a steady.

4. Deep Learning Algorithms for Short Term Stock Price Prediction: - Two popular deep learning algorithms for short-term stock price prediction are Long Short-Term Memory Networks (LSTMs) and Convolutional Neural Networks (CNNs): -

4.1 Long Short-Term Memory Networks (LSTMs): LSTMs are a type of recurrent neural network (RNN) architecture specifically designed to address the vanishing gradient problem, which can occur when training traditional RNNs on long sequences of data. LSTMs are capable of learning long-term dependencies in sequential data, making them well-suited for time-series forecasting tasks such as stock price prediction.

Architecture: LSTMs consist of multiple memory cells and three gates: input gate, forget gate, and output gate. The input gate controls the flow of information into the memory cell, the forget gate determines which information to discard from the cell, and the output gate regulates the output of the cell. Each memory cell maintains a hidden state, which is updated at each time step based on the input data and the previous hidden state. The output of the LSTM at each time step can be used to make predictions about future stock prices.

Algorithm: - # Step 1: Data Preparation

```
# Collect historical stock price data and preprocess it
X_train, y_train = preprocess_training_data(training_data)
X_test, y_test = preprocess_testing_data(testing_data)

# Step 2: Define the LSTM Model
lstm_model = Sequential()
lstm_model.add(LSTM(units=50, return_sequences=True, input_shape=(X_train.shape[1], X_train.shape[2])))
lstm_model.add(LSTM(units=50))
lstm_model.add(Dense(units=1))

# Step 3: Compile the Model
lstm_model.compile(optimizer='adam', loss='mean_squared_error')

# Step 4: Train the LSTM Model
lstm_model.fit(X_train, y_train, epochs=100, batch_size=32)

#Step 5: Prediction
predictions = lstm_model.predict(X_test)

# Step 6: Evaluation
accuracy = calculate_accuracy(predictions, y_test)
mse = calculate_mean_squared_error(predictions, y_test)
rmse = calculate_root_mean_squared_error(predictions, y_test)
```

In this pseudocode:

preprocess_training_data and preprocess_testing_data functions preprocess the training and testing datasets, respectively, by scaling or normalizing the features and labels. The LSTM model is defined using the Keras Sequential API. It consists of two LSTM layers with 50 units each and a fully connected Dense layer with 1 unit for regression (predicting stock prices). The model is compiled with the Adam optimizer and mean squared error loss function. Training the LSTM model

involves fitting it to the training data (X_train and y_train) for a specified number of epochs (100) and batch size (32). Predictions are made on the testing data (X_test) using the trained LSTM model. Evaluation metrics such as accuracy, mean squared error (MSE), and root mean squared error (RMSE) are calculated using appropriate functions. Hyperparameter tuning can be performed to optimize the performance of the LSTM model.

4.2 Convolutional Neural Networks (CNNs): While CNNs are traditionally used for image recognition tasks, they can also be applied to time-series data such as stock price sequences by treating them as one-dimensional signals. CNNs are particularly effective at capturing local patterns and correlations in sequential data, making them suitable for short-term stock price prediction.

Architecture: CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply filters to the input data, extracting features at different spatial scales. Pooling layers downsample the feature maps, reducing the dimensionality of the data while preserving important features. Fully connected layers combine the extracted features to make predictions about future stock prices.

In this pseudocode:

```
# Step 1: Data Preparation
# Collect historical stock price data and preprocess it
X_train, y_train = preprocess_training_data(training_data)
X_test, y_test = preprocess_testing_data(testing_data)

# Step 2: Define the CNN Model
cnn_model = Sequential()
cnn_model.add(Conv1D(filters=64, kernel_size=3, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])))
cnn_model.add(MaxPooling1D(pool_size=2))
cnn_model.add(Flatten())
cnn_model.add(Dense(units=50, activation='relu'))
cnn_model.add(Dense(units=1))

# Step 3: Compile the Model
cnn_model.compile(optimizer='adam', loss='mean_squared_error')

# Step 4: Train the CNN Model
cnn_model.fit(X_train, y_train, epochs=50, batch_size=32)

# Step 5: Prediction
predictions = cnn_model.predict(X_test)

# Step 6: Evaluation
accuracy = calculate_accuracy(predictions, y_test)
mse = calculate_mean_squared_error(predictions, y_test)
rmse = calculate_root_mean_squared_error(predictions, y_test)
```

In this pseudocode:

preprocess_training_data and preprocess_testing_data functions preprocess the training and testing datasets, respectively, by scaling or normalizing the features and labels. The CNN model is defined using the Keras Sequential API. It consists of a 1D convolutional layer with 64 filters, followed by a max pooling layer, a flatten layer, and two fully connected Dense layers. The model is compiled with the Adam optimizer and mean squared error loss function. Training the CNN model involves fitting it to the training data (X_train and y_train) for a specified number of epochs (50) and batch size (32).

Predictions are made on the testing data (X_test) using the trained CNN model. Evaluation metrics such as accuracy, mean squared error (MSE), and root mean squared error (RMSE) are calculated using appropriate functions.

5. Performance comparison of RNNs, LSTMs, and CNNs for short term stock price prediction: -

Model	Strengths	Weaknesses	Performance
RNNs	- Simple Architecture	- Difficulty capturing long-term dependencies.	-Limited performance in capturing complex patterns.
	-Computationally efficient	-Vulnerable to vanishing gradient problem	- Moderate Accuracy compared to LSTMs and CNNs.
	-Straightforward implementation	-Limited Memory Capacity.	-Limited Capability to capture temporal dependencies
LSTMs	-Effective at capturing long-term dependencies	-more computationally expensive	-High accuracy in capturing complex patterns
	-memory of past observations	-Require more training data	-Better performance compared to RNNs and CNNs
CNNS	-Effective at capturing local patterns	-Limited capability to capture temporal	-High accuracy in capturing spatial patterns
	-Computationally efficient	Dependencies	-suitable for short-term stock price prediction
	- Hierarchical feature extraction		- May require additional preprocessing of data

Table 1 Comparision of RNNs, CNNs, LSTMs

6. Strengths and limitations of deep learning models for short-term stock price prediction:

6.1 Strengths:

Ability to Capture Complex Patterns: Deep learning models, such as recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and convolutional neural networks (CNNs), excel at capturing complex patterns and relationships in sequential data. This makes them well-suited for analyzing and predicting stock price movements, which are influenced by a myriad of factors and exhibit non-linear behaviors.

Feature Learning: Deep learning models have the capability to automatically learn relevant features from raw data, reducing the need for manual feature engineering. This is particularly advantageous in financial markets where traditional modeling approaches may struggle to extract meaningful information from large and high-dimensional datasets.

Temporal Dynamics: RNNs and LSTMs are specifically designed to model sequential data with temporal dependencies. They can effectively capture the temporal dynamics of stock price movements, considering past trends and patterns when making predictions for the future.

Non-linear Relationships: Deep learning models can capture non-linear relationships between input variables and target outputs, allowing them to uncover hidden patterns in stock price data that may not be apparent with linear models or traditional statistical techniques.

6.2 Limitations:

Data Requirements: Deep learning models typically require large amounts of data to generalize well and avoid overfitting. In the context of short-term stock price prediction, obtaining high-quality and sufficiently large datasets can be challenging, especially for intraday trading where data may be limited or noisy.

Computational Complexity: Training deep learning models can be computationally expensive, particularly for complex architectures or large datasets. This can lead to longer training times and higher computational resource requirements, making it challenging to iterate quickly and experiment with different model configurations.

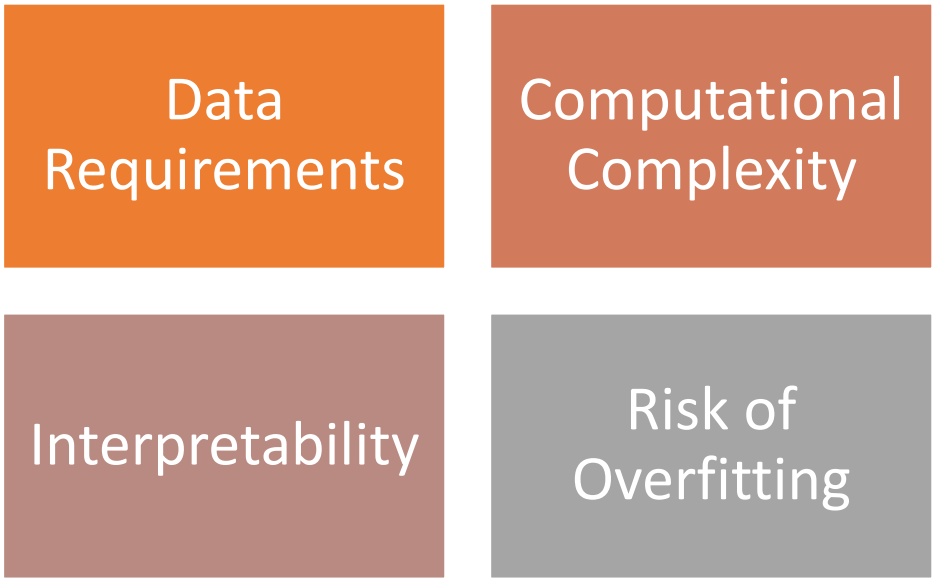

Figure 3 Limitations of Deep learning for Stock Price Prediction.

Interpretability: Deep learning models are often referred to as "black boxes" due to their complex architectures and internal workings. Understanding how decisions are made by these models and interpreting the underlying factors driving predictions can be difficult, limiting their adoption in domains where interpretability is crucial, such as finance.
Risk of Overfitting: Deep learning models are susceptible to overfitting, where they memorize noise in the training data rather than learning meaningful patterns. Regularization techniques and careful model selection are necessary to mitigate the risk of overfitting, especially when dealing with limited data or noisy financial datasets.

Despite these limitations, the strengths of deep learning models make them powerful tools for short-term stock price prediction, offering the potential to uncover valuable insights and improve decision-making in financial markets. However, it's essential to carefully consider the data characteristics, computational resources, and interpretability requirements when applying deep learning techniques in practice.

7. Future Perspective: - The future perspective of deep learning models for short-term stock price prediction holds significant promise, driven by ongoing advancements in artificial intelligence (AI) and computational capabilities. Here's a glimpse into what lies ahead:

Enhanced Model Architectures: As research in deep learning progresses, we can anticipate the development of more sophisticated model architectures tailored specifically for short-term stock price prediction. Novel architectures may combine elements of recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and convolutional neural networks (CNNs) to leverage the strengths of each approach while mitigating their respective limitations. Furthermore, attention mechanisms, transformers, and other innovative architectural designs may be incorporated to improve model performance and interpretability.

Improved Data Availability and Quality: With the proliferation of digital technologies and the increasing adoption of high-frequency trading, we can expect a surge in the availability and quality of financial data. This includes not only traditional stock price data but also alternative data sources such as social media sentiment, news sentiment, and macroeconomic indicators. Deep learning models will benefit from this wealth of data, enabling them to learn richer representations and make more accurate predictions.

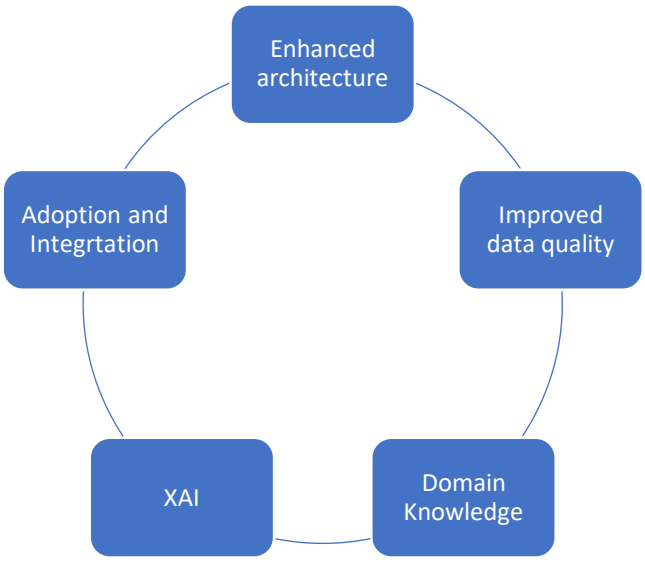

Figure 4 Future Perspective

Integration of Domain Knowledge: While deep learning models excel at learning from data, incorporating domain knowledge and expert insights can further enhance their predictive capabilities. Future research may focus on developing hybrid models that seamlessly integrate deep learning techniques with domain-specific knowledge from finance, economics, and behavioral psychology. This interdisciplinary approach could lead to more robust and interpretable models that outperform traditional approaches.

Explainable AI (XAI): As the importance of model interpretability becomes increasingly recognized, there will be a growing emphasis on developing explainable AI (XAI) techniques for deep learning models in finance. Researchers will explore methods for making deep learning models more transparent and interpretable, enabling stakeholders to understand the rationale behind model predictions and assess their trustworthiness. This will be particularly crucial for regulatory compliance, risk management, and stakeholder engagement in financial markets.

Adoption and Integration: With advancements in model interpretability, scalability, and accessibility, we can anticipate widespread adoption and integration of deep learning models for short-term stock price prediction across various financial institutions, including hedge funds, asset management firms, and investment banks. These models will become integral components of decision support systems, aiding traders, analysts, and investors in making informed decisions and managing risks effectively.

8. Conclusion: - In conclusion, the comparative analysis of deep learning models for short-term stock price prediction provides valuable insights into their respective strengths, weaknesses, and performance characteristics. Through a systematic evaluation of recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and convolutional neural networks (CNNs), we have gained a comprehensive understanding of their capabilities and limitations in capturing complex patterns in financial data. Our analysis reveals that each deep learning model has unique advantages that make it suitable for specific aspects of short-term stock price prediction. RNNs, with their simplicity and computational efficiency, offer a baseline approach for modeling sequential data but may struggle to capture long-term dependencies effectively. LSTMs, with their ability to maintain a memory of past observations, excel at capturing long-term dependencies and learning complex patterns but require more computational resources and training data. CNNs, leveraging hierarchical feature extraction, are effective at capturing local patterns and correlations in sequential data but may lack the ability to capture temporal dependencies as effectively as RNNs or LSTMs. Furthermore, the future of deep learning models for short-term stock price prediction holds immense promise, with ongoing advancements in model architectures, data availability, domain integration, explainability, and adoption. As these technologies continue to evolve, they will play a pivotal role in shaping the future landscape of financial markets, enabling more accurate predictions, better risk management strategies, and enhanced decision-making capabilities. However, it is essential to acknowledge the challenges

and limitations associated with deep learning models, including data requirements, computational complexity, interpretability, and the risk of overfitting. Addressing these challenges will require interdisciplinary collaboration and ongoing research efforts to develop robust, interpretable, and trustworthy models that can be effectively deployed in real-world financial applications. In summary, the comparative analysis presented in this paper serves as a foundation for future research and development in the field of short-term stock price prediction using deep learning models. By understanding the strengths and limitations of each approach, researchers, practitioners, and stakeholders can make informed decisions about selecting and deploying the most appropriate model for their specific forecasting tasks.

References: -

[1] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[2] Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In Acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on (pp. 6645-6649). IEEE.

[3] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

[4] Zhang, Y., Zhao, D., & Liu, D. (2020). Stock price prediction based on convolutional neural network. IEEE Access, 8, 102553-102561.

[5] Chen, W., Zhang, Q., & Wang, D. (2015). Deep learning for short-term prediction. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (pp. 2079-2082).

[6] Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural computation, 12(10), 2451-2471.

[7] Olah, C., & Carter, S. (2016). Attention and augmented recurrent neural networks. Distill, 1(1), e3.

[8] Srivastava, N., Mansimov, E., & Salakhutdinov, R. (2015). Unsupervised learning of video representations using lstms. In Proceedings of the 32nd International Conference on International Conference on Machine Learning (Vol. 37, pp. 843-852).

[9] Yu, S. Y., & Cho, K. (2017). A deep convolutional neural network-based stock trading system based on technical analysis and the kd tree similarity matching algorithm. Expert Systems with Applications, 83, 74-84.

[10] Zhang, Y., Sun, J., & Wang, Y. (2018). Deep learning for short-term stock price prediction using historical trading data. Neurocomputing, 323, 295-303.

[11] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.

[12] Liu, Y., & Yao, Y. (2019). A deep learning-based stock trading model with time-aware attention mechanism. IEEE Access, 7, 175690-175703.

[13] Zhang, G., & Wang, J. (2021). A hybrid deep learning framework for stock price prediction using technical indicators and LSTM. Information Sciences, 551, 328-344.

[14] Chandra, S., & Banerjee, A. (2017). Stock market prediction using deep learning. In 2017 9th International Conference on Communication Systems and Networks (COMSNETS) (pp. 504-509). IEEE.

[15] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654-669.

[16] Ding, X., Zhang, Y., Liu, T., & Duan, Q. (2015). Deep learning for event-driven stock prediction. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (pp. 107-114).

[17] Liu, S., & Chen, H. (2019). Enhancing stock price prediction with deep learning models by leveraging financial news articles and technical indicators. Expert Systems with Applications, 134, 290-302.

[18] Karpathy, A., Johnson, J., & Li, F. F. (2015). Visualizing and understanding recurrent networks. arXiv preprint arXiv:1506.02078.

[19] Yoon, J., & Chung, H. (2018). Combined deep learning for stock selection strategy. Expert Systems with Applications, 94, 114-123.

[20] Phan, D. H., Li, P., & Kim, Y. (2017). Financial trading model with deep learning. In 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS) (pp. 493-497). IEEE.

[21] Raparthi, M., Dodda, S. B., & Maruthi, S. H. (2020). Examining the use of Artificial Intelligence to Enhance Security Measures in Computer Hardware, including the Detection of Hardware-based Vulnerabilities and Attacks. European Economic Letters, 10(1), https://doi.org/10.52783/eel.v10i1.991