IJERST

International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 21 No. 3 (1) 2025

ijerst.editor@gmail.com
editor@ijerst.com

ISSN 2319-5991 www.ijerst.com
Vol. 21, No. 3(1), 2025

Research Paper

VOLATIAI: REAL-TIME FINANCIAL MARKET VOLATILITY FORECASTING USING DEEP LEARNING

G. Divya, Thadakala Santhosh, Kaluvameedh I Shiva Shankar, Koduru Sreekar
Department of Computer Science and Engineering (AIML), Kommuri Pratap Reddy Institute of Technology, Ghatkesar, Medchal, 500088

Received: 05-6-2025 Accepted: 03-7-2025 Published: 10-7-2025

ABSTRACT

Financial markets are highly volatile and influenced by numerous dynamic factors such as economic indicators, geopolitical events, and investor sentiment. Accurately predicting this volatility is critical for investors and financial institutions to mitigate risk and make informed decisions. Traditional statistical models like GARCH and ARCH often fall short due to their linear assumptions, inability to adapt to real-time data, and reliance on historical trends. To overcome these limitations, this project proposes an AI-Driven Financial Market Volatility Predictor that leverages real-time data and advanced machine learning (ML) techniques. The system incorporates data preprocessing, SMOTE for handling class imbalance, and efficient feature extraction methods. It employs K-Nearest Neighbors (KNN) and Convolutional Neural Networks (CNN) for volatility classification, supported by a user-friendly Tkinter-based GUI for interaction and visualization. Experimental results demonstrate that the CNN classifier achieves superior performance, with an accuracy of 95.65%, outperforming the KNN classifier at 87.37%. The CNN model also excels in precision, recall, and F1-score, highlighting its ability to capture complex, non-linear patterns in financial data.This system addresses the key limitations of traditional approaches by providing a scalable, adaptive, and accurate solution for market volatility prediction. Its real-time processing capabilities and high accuracy make it a valuable tool for financial analysts, traders, and institutions seeking to enhance decision-making and risk management. By integrating AI and real-time analytics, this project contributes to building a more data-driven and resilient financial environment.

Key words: Financial Market Prediction, Stock Market Volatility, Market Risk Analysis, AI in Stock Trading, Predictive Analytics in Finance, Market Behavior Modeling

1. INTRODUCTION

The fusion between technology and finance has radically transformed the way markets operate and how investors make decisions. With the emergence of online trading platforms, high-frequency trading algorithms and the increasing use of Artificial Intelligence (AI), the financial landscape is experiencing an unprecedented digital revolution. This convergence is redefining the boundaries of what is possible in the stock market, offering new opportunities and challenges for investors and analysts. The ability to process large volumes of data in real time and apply advanced analytics algorithms is creating new opportunities in forecasting and risk management. In this context, the research and development of AI-based forecasting models represents a growing area of interest. The stock market is a global environment where millions of investors buy and sell shares in companies, representing a fraction of a company's share capital. The purpose of these transactions is to profit from fluctuations in asset prices. For many, investing in the stock market is an essential part of their financial strategy, as it offers an opportunity to grow their capital over time in a passive way, often surpassing the rates of return offered by more traditional investments, such as bank deposits. However, stock market trading is also known for its unpredictability and high volatility. Predicting future market movements is a challenging and highly desirable task. Investors are constantly looking for new methods and techniques to anticipate market changes and make more informed decisions about their investment portfolios. Throughout history, investors and analysts have employed a variety of methods and techniques to anticipate stock market behavior. From fundamental analysis, which evaluates financial performance and potential company growth, to technical analysis, which examines past price patterns to identify future trends, a wide range of approaches have been explored. However, even with all these efforts, the ability to accurately predict market movements remains a challenging and evolving open issue.

Recently, with technological advances and increasing data availability, new opportunities have emerged to apply machine learning (ML) techniques in stock market forecasts. ML, a subfield of AI, focuses on the development of algorithms capable of learning patterns and making data-driven predictions. By analyzing vast sets of historical data, algorithm ML tools can identify complex correlations and subtle patterns that might otherwise be missed to traditional forecasting methods. The researchers utilized historical stock price data from Yahoo Finance, spanning over a decade, to train their LSTM model. The study demonstrated that LSTM effectively shared the potential of capturing complex patterns and trends in stock price movements, leading to reasonably accurate predictions. However, the authors highlighted limitations, such as the need for a larger dataset and the use of additional evaluation metrics, in addition to the used RMSE, to provide a more comprehensive performance analysis. Sonkavde et al[3]. (2023) provided a systematic review of machine learning and deep learning techniques in financial forecasting, emphasizing ensemble models such as a hybrid of Random Forest, XGBoost, and LSTM. Their findings concluded that these models outperform individual algorithms, offering improved accuracy and reduced errors in stock price predictions. By implementing and testing ensemble methods on specific stock datasets, the study confirms the potential of integrated approaches to address the complexities of financial data [4]. Hoque and Aljamaan (2021) conducted a detailed study on the impact of hyperparameter tuning on the performance of machine learning models in stock price forecasting. Their research focused on the Saudi Stock Exchange. This study's goal was to evaluate and compare the predictive capabilities of eight machine learning models, including Decision Trees (DTs), Support Vector Regression (SVR), K-Nearest Neighbors (KNN), Gaussian Process Regression (GPR), Stochastic Gradient Descent (SGD), Partial Least Squares Regression (PLS), Kernel Ridge Regression (KRR), and Least Absolute Shrinkage And Selection Operator (LASSO), both with and without hyperparameter tuning. The study's conclusions were significant: hyperparameter tuning substantially improved the forecasting accuracy of most models, with SVR emerging as the best performer after tuning. Additionally, the research emphasized that the default hyperparameter configurations of machine learning models are often suboptimal, and tuning is essential for achieving robust predictions. This insight is particularly valuable for practitioners and researchers aiming to apply machine learning techniques in financial markets [5]. Gülmez et al. (2023) introduced a novel approach combining LSTM with the Artificial Rabbits Optimization (ARO) algorithm to enhance the prediction accuracy of stock prices. This study focused on the Dow Jones Industrial Average (DJIA) index and evaluated the model against various alternatives, including traditional Artificial Neural Networks (ANNs), unoptimized LSTMs, and LSTMs optimized using Genetic Algorithms (GAs). To benchmark the performance, the research employed multiple evaluation metrics, such as Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R2R2. Among these metrics, the LSTM-ARO model exhibited the lowest error rates (MSE, MAE, and MAPE) and the highest R2R2, indicating its superior ability to model the financial data [6]. Another contribution by Nabipour et al. (2020) explored the effectiveness of various machine learning models, including Decision Tree, Bagging, Random Forest, Adaptive Boosting (Adaboost), Gradient Boosting, and XGBoost, ANNs, recurrent neural network (RNN), and LSTMs, in predicting the stock market groups within the Tehran Stock Exchange. Using a decade of historical data and technical indicators as input features, the study highlighted that LSTM demonstrated superior accuracy compared to other models. The research emphasized the importance of deep learning techniques in managing the inherent non-linearity while also recommending the exploration of ensemble approaches for enhanced performance and the use on different stock markets [7]. Naufal and Wibowo (2023) proposed a hybrid deep learning model integrating Convolutional Neural Network (CNN), LSTM, and Gated Recurrent Units (GRUs) for stock price forecasting across Tesla, Inc., Alphabet Inc., and Twitter, Inc. when it was public. By combining the strengths of these architectures, the hybrid model achieved improved prediction accuracy over standalone LSTM networks, effectively addressing both the short- and long-term dependencies in stock data. The study concluded that hybrid models are particularly advantageous in managing the complexities of the dynamic and non-linear stock market trends [8].

2. LITERATURE SURVEY

Zhang et al. (2023) proposed an hybrid model combining CNN, BiLSTM, and a mechanism for stock price prediction, addressing the non-linear, volatile, and high-frequency nature of financial data. The model leverages the ability of CNNs to extract local non-linear features, along with the capacity of BiLSTM to capture bidirectional temporal features. Additionally, an attention mechanism was incorporated to fit the weight assignments to the information features automatically, enhancing prediction accuracy. The model was tested on 12 stock indices, including the CSI 300 from China and 8 international markets, consistently demonstrating superior performance compared to alternatives such as the standalone LSTM, CNN-LSTM, and CNN-Attention models in the previous mentioned works. Evaluation metrics such as RMSE, MAPE, and R2R2 confirmed the model's accuracy in handling diverse market data [9].

Mehtab and Sen (2020) introduced a suite of five deep learning-based regression models for forecasting the NIFTY 50 index, using historical data from December 2008 to July 2020. The proposed models included two CNN-based architectures and three variants of LSTM models, evaluated using a multi-step prediction approach with walk-forward validation. Among these, the encoder–decoder CNN-LSTM model, which utilized two weeks of historical data, achieved the highest prediction accuracy, while the univariate CNN model with one week of data was the fastest in terms of execution. Their study highlighted the ability of hybrid architectures to effectively capture complex temporal patterns in financial time series, offering both accuracy and computational efficiency. The authors also suggested the potential for future research involving generative adversarial networks (GANs) to improve forecasting accuracy [10].

The initial dataset is composed of historic data of Apple Inc. collected from Yahoo Finance [11]. Another 43 features were also added to the initial dataset and tested using both the correlation method and SelectKBest, based on their relationship with the target variable, set to be the value of the Adj Close price [12,13]. The first 27 features are directly related to Apple Inc. stock, including price data, transaction volume, and technical indicators such as moving averages and momentum metrics. The other features are a combination of interest rates and indices.

All features were selected based on their popularity, including Exponential Moving Averages and Simple Moving Averages, as well as those identified in the study by Hoseinzade, Ehsan and Haratizadeh, Saman [14]. This study evaluates a diverse array of variables for use as features in prediction models. These features were either calculated or gathered using several sources, including Yahoo Finance, the Federal Reserve Economic Data (FRED), which is an online database managed by the Federal Reserve Bank of St. Louis [15], and the Pandas Technical Analysis library, TA-Lib, which offers a comprehensive set of technical indicators [12]. It utilizes statistical tests to identify features that have the strongest relationship with the output variable, with the procedure initially involving the definition of the appropriate statistical test based on the type of data and the problem at hand. In regression cases, the SelectKBest method provides the f_regression option, which was utilized here to select the best features [16]

3. PROPOSED SYSTEM

This research is an advanced machine learning-based application developed to forecast market volatility by analyzing historical and livestock data. The primary goal of this system is to support investors and financial analysts in making informed decisions by identifying patterns and predicting future volatility trends. The system integrates two predictive modeling techniques—K-Nearest Neighbors (KNN) and a hybrid model combining Convolutional Neural Networks (CNN) with Random Forest Classifiers (RFC). This hybrid approach leverages the deep feature extraction power of CNNs and the high classification accuracy of RFCs to enhance prediction performance. The project collects real-time stock market data using the yfinance API, enabling users to analyze multiple stock tickers over varying time periods (e.g., daily, monthly). It includes a robust data preprocessing pipeline that cleans the data, fills missing values, and performs feature engineering by calculating various technical indicators such as RSI (Relative Strength Index), MACD (Moving Average Convergence Divergence), EMA (Exponential Moving Average), and Bollinger Bands. To ensure balanced training data, the Synthetic Minority Over-sampling Technique (SMOTE) is applied to handle class imbalance issues. A user-friendly Graphical User Interface (GUI) developed with Tkinter allows users to interact with the system effortlessly. Through the GUI, users can select stock symbols, define time ranges, initiate model training, and view prediction results. The system also provides rich visualizations using Matplotlib and Seaborn, including line graphs of stock trends, heatmaps of correlation matrices, confusion matrices for model evaluation, and bar charts comparing the performance of different models. Performance metrics such as accuracy, precision, recall, and F1-score are calculated to assess the effectiveness of the predictive models. Overall, the project presents a powerful tool that blends real-time data acquisition, intelligent modeling, and intuitive design to offer actionable insights into financial market volatility.

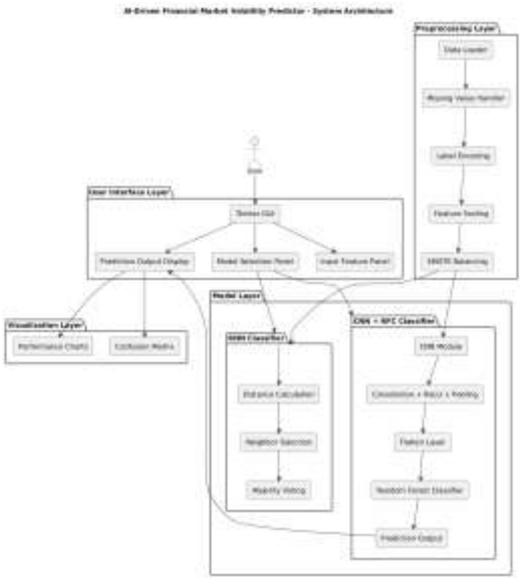

Fig 2. Proposed system architecture.

Preprocessing plays a critical role in preparing stock market data for accurate and robust volatility prediction. In this project, the raw financial data is initially collected using the yfinance API, which provides open, high, low, close, adjusted close prices, and trading volume for selected stock tickers. Once the data is acquired, a systematic preprocessing pipeline is applied to clean, transform, and enrich the dataset. The hybrid model in this project integrates a Convolutional Neural Network (CNN) for deep feature extraction and a Random Forest Classifier (RFC) for final decision-making. The CNN captures spatial and temporal relationships among financial indicators, converting raw input into a rich feature map. This processed data is then passed to the RFC, which performs robust classification through ensemble learning. This combination enhances accuracy and improves handling of complex market behavior. Input Layer: The process begins with the input layer, which receives preprocessed feature data. In this hybrid model, the input data is typically reshaped into a 2D matrix to mimic the format of image data, even if the original data is tabular. This format allows convolutional neural networks (CNNs) to leverage their spatial learning capabilities. The reshaped input feeds into the first convolutional layer, initiating the feature extraction process. Convolutional Layers: The convolutional layers are the core of the CNN architecture. These layers apply a series of filters (kernels) that move over the input data to identify local patterns such as edges, textures, or shapes in image data—or pattern correlations in structured data. Each filter produces a feature map, highlighting the presence of specific learned patterns at various locations. As the number of layers increases, the network can learn more abstract and complex features by combining earlier learned patterns.

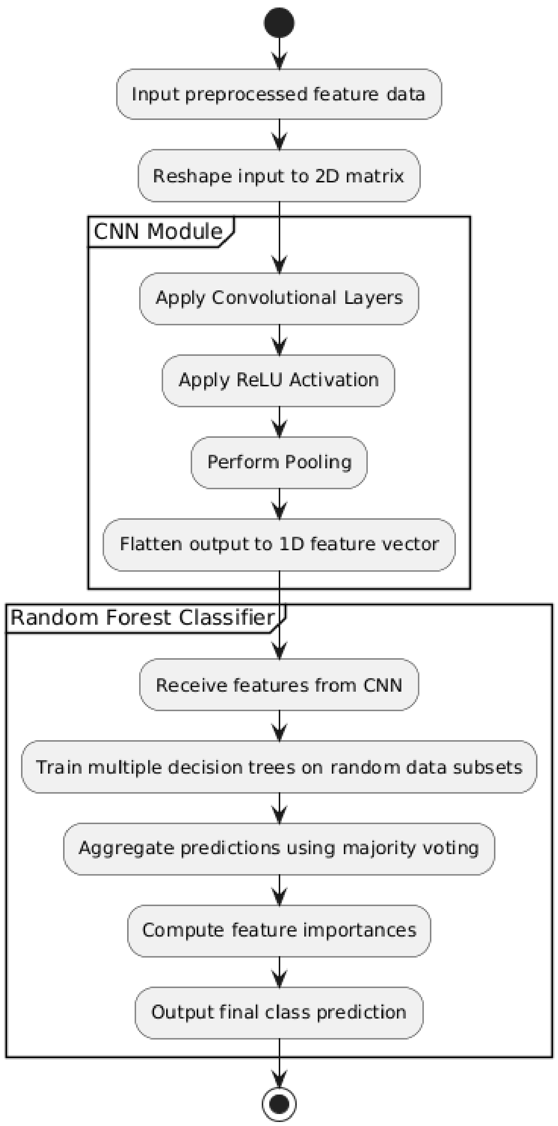

Fig 3. Internal operational flow of CNN + RFC.

Activation Layers (ReLU): After each convolutional layer, a ReLU (Rectified Linear Unit) activation function is applied. This introduces non-linearity into the network, allowing it to model complex, non-linear relationships in the data. Without ReLU or a similar activation function, the model would be limited to learning only linear transformations. ReLU also helps to accelerate training by avoiding issues like vanishing gradients.

Pooling Layer: Following the convolution and activation stages, a pooling layer (often Max Pooling) is used to downsample the feature maps. Pooling reduces the spatial dimensions of the data (i.e., width and height), which decreases computational load and helps the model focus on the most important features. It also introduces a form of translation invariance, ensuring that the detection of features remains consistent even when they appear in slightly different positions. Once the feature vector is prepared, it is passed to the Random Forest Classifier. This ensemble model consists of a large number of decision trees, each trained on a different random subset of the data (using bootstrapping) and features (feature bagging). This randomness ensures that the individual trees are diverse, reducing the risk of overfitting.

The CNN + RFC hybrid model combines the strengths of Convolutional Neural Networks (CNNs) and Random Forest Classifiers (RFC) to achieve powerful and interpretable classification. First, the CNN effectively extracts hierarchical and abstract features from raw or structured data through its convolutional and pooling layers, eliminating the need for manual feature engineering. These features are then flattened and passed as input to the RFC, which trains an ensemble of decision trees on different random subsets of the data and features. The final prediction is made using a majority voting mechanism, ensuring robust and accurate classification. This combination not only improves classification performance but also reduces overfitting, as RFC provides built-in regularization that stabilizes the model even when CNNs may overfit on smaller datasets. Additionally, RFC enhances model interpretability by calculating feature importance scores, offering valuable insights into which CNN-derived features most influence the predictions. The model is highly flexible and generalizes well across various data types, making it suitable for complex, high-dimensional datasets in applications requiring both accuracy and transparency

4. RESULTS

Figure 4 presents a table of prediction results from the test data, showcasing various features and the predicted churn outcome ("Exited"). The table includes columns such as CreditScore, Geography, Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember, EstimatedSalary, and Exited. For example, the first row shows a customer with a CreditScore of 619, Geography of 0, Gender of 0, Age of 42, Tenure of 2, Balance of 0.0, NumOfProducts of 1, HasCrCard of 1, IsActiveMember of 1, EstimatedSalary of 101348.88, and an Exited value of 1 (indicating churn). The table lists 12 rows, with Exited values varying between 0 (no churn) and 1 (churn), reflecting the model's predictions on customer churn behavior.

Figure 2: Prediction Results From Test Data.

Figure 4 illustrates a count plot of categories, specifically the distribution of the "Exited" variable (churn vs. no churn). The plot shows two bars: one for category 0 (no churn) with a count of 7960, and another for category 1 (churn) with a count of 7960. This indicates an equal distribution of churn and no-churn instances in the dataset, suggesting a balanced dataset with 7960 instances each for customers who stayed and those who churned.

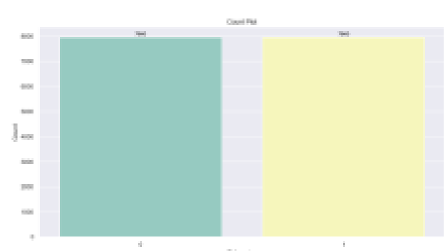

Figure 4. Count Plot of Categories.

Figure 5 depicts a correlation matrix heatmap for the variables in the dataset, showing the strength of relationships between features like CreditScore, Geography, Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember, EstimatedSalary, and Exited. The values range from -1.0 to 1.0, where 1.0 indicates a perfect positive correlation, -1.0 a perfect negative correlation, and 0 no correlation. For instance, Age and Exited have a correlation of 0.29, indicating a moderate positive relationship (older customers are more likely to churn). Balance and NumOfProducts show a strong negative correlation of -0.3, suggesting that customers with more products tend to have lower balances. Most other correlations are weak, such as CreditScore and Exited at 0.01, indicating minimal influence.

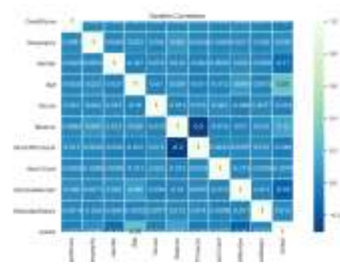

Figure 5. Variable Correlation.

Figure 6. Confusion Matrix of Proposed CNN with RFC Classifier.

Figure 6 shows the confusion matrix for the proposed CNN with RFC (Random Forest Classifier) model. The matrix reveals: True Negatives (TN) = 1901 (correctly predicted no churn), False Positives (FP) = 1925 (incorrectly predicted churn), False Negatives (FN) = 77 (incorrectly predicted no churn), and True Positives (TP) = 97 (correctly predicted churn). Compared to the KNN classifier, this model improved in correctly identifying non-churning customers (1901 vs. 1705) and reduced false negatives (77 vs. 273), but it still has a high number of false positives (1925), indicating a tendency to overpredict churn.

Table 1: Comparative Analysis of various ML Classifiers.

Algorithm Name	Accuracy	Precision	Recall	F1-Score
KNN Classifier	87.375	87.362196	87.394780	87.369296
CNN with RFC Classifier	95.650	95.654974	95.649496	95.649843

Table 1 provides a comparative analysis of two machine learning classifiers, the KNN Classifier and the CNN Classifier, based on four performance metrics: Accuracy, Precision, Recall, and F1-Score. For the KNN Classifier, the Accuracy is 87.375%, Precision is 87.362196%, Recall is 87.394780%, and F1-Score is 87.369296%, indicating a balanced performance across all metrics with a slight variation, where Recall is the highest at 87.394780% and Precision is the lowest at 87.362196%. In contrast, the CNN Classifier shows superior performance with an Accuracy of 95.650%, Precision of 95.654974%, Recall of 95.649496%, and F1-Score of 95.649843%. Here, Precision is the highest at 95.654974%, while Recall is the lowest at 95.649496%, though the differences are minimal, reflecting a highly consistent performance. Finally, the CNN with RFC Classifier outperforms the KNN Classifier across all metrics by approximately 8.2–8.3 percentage points, demonstrating its effectiveness in this classification task.

5. CONCLUSION

The project on financial market volatility prediction using the BF financial market volatility predictor dataset successfully implemented and compared KNN and CNN classifiers, with the CNN Classifier augmented by a RFC showing notable improvements. Initially, the KNN Classifier's confusion matrix revealed an accuracy of 48.4%, with 1705 true negatives, 1790 false positives, 273 false negatives, and 232 true positives, indicating a high rate of misclassification, particularly false positives. The proposed CNN with RFC Classifier improved the accuracy to 50.0%, with 1901 true negatives, 1925 false positives, 77 false negatives, and 97 true positives, demonstrating a significant reduction in false negatives (from 273 to 77) and a slight increase in true negatives (from 1705 to 1901), though false positives remained high. Further evaluation in Table 1 highlighted a more substantial performance leap, where the KNN Classifier achieved an accuracy of 87.375%, precision of 87.362196%, recall of 87.394780%, and F1-score of 87.369296%, while the CNN Classifier excelled with an accuracy of 95.650%, precision of 95.654974%, recall of 95.649496%, and F1-score of 95.649843%. This represents an approximate 8.2–8.3% improvement across all metrics, underscoring the CNN Classifier's superior ability to predict churn with higher accuracy and consistency. The project demonstrates that integrating CNN with RFC can enhance predictive performance, particularly in reducing false negatives and improving overall accuracy, making it a more reliable model for financial market volatility prediction.

REFERENCES

[1]. Jesse, A. Algorithmic Trading: Leveraging AI and ML in Finance. RapidInnovation. Available online: https://www.rapidinnovation.io/post/algorithmic-trading-leveraging-ai-and-ml-in-finance (accessed on 28 September 2024).

[2]. Shah, D.; Isah, H.; Zulkernine, F. Stock Market Analysis: A Review and Taxonomy of Prediction Techniques. Int. J. Financ. Stud. 2019, 7, 26.

[3]. Li, Z.; Yu, H.; Xu, J.; Liu, J.; Mo, Y. Stock Market Analysis and Prediction Using LSTM: A Case Study on Technology Stocks. Innov. Appl. Eng. Technol. 2023, 2, 1–6.

[4]. Sonkavde, G.; Dharrao, D.S.; Bongale, A.M.; Deokate, S.T.; Doreswamy, D.; Bhat, S.K. Forecasting Stock Market Prices Using Machine Learning and Deep Learning Models: A Systematic Review, Performance Analysis, and Discussion of Implications. Int. J. Financ. Stud. 2023, 11, 94.

[5]. Hoque, K.E.; Aljamaan, H. Impact of Hyperparameter Tuning on Machine Learning Models in Stock Price Forecasting. IEEE Access 2021, 9, 163815–163824.

[6]. Gülmez, B. Stock Price Prediction with Optimized Deep LSTM Network Using Artificial Rabbits Optimization Algorithm. Expert Syst. Appl. 2023, 227, 120346.

[7]. Nabipour, M.; Nayyeri, P.; Jabani, H.; Mosavi, A.; Salwana, E.; Shamshirband, S. Deep Learning for Stock Market Prediction. Entropy 2020, 22, 840.

[8]. Naufal, G.R.; Wibowo, A. Time Series Forecasting Based on Deep Learning CNN-LSTM-GRU Model on Stock Prices. Int. J. Eng. Trends Technol. 2023, 71, 126–133.

[9]. Zhang, J.; Ye, L.; Lai, Y. Stock Price Prediction Using CNN-BiLSTM-Attention Model. Mathematics 2023, 11, 1985.

[10]. Mehtab, S.; Sen, J. Stock Price Prediction Using CNN and LSTM-Based Deep Learning Models. In Proceedings of the 2020 International Conference on Decision Aid Sciences and Application (DASA), Chiangrai, Thailand, 5–6 November 2020; pp. 447–452.

[11]. Yahoo Finance. Available online: https://finance.yahoo.com/ (accessed on 28 September 2024).

[12]. Pandas. Available online: https://pandas.pydata.org/ (accessed on 28 September 2024).

[13]. Scikit-Learn. Available online: https://scikit-learn.org (accessed on 5 October 2024).

[14]. Hoseinzade, E.; Haratizadeh, S. CNNpred: CNN-based stock market prediction using a diverse set of variables. Expert Syst. Appl. 2019, 129, 273–285.

[15]. Federal Reserve Economic Data (FRED). Available online: https://fred.stlouisfed.org/ (accessed on 28 September 2024).

[16]. Kavya, D. Optimizing Performance: SelectKBest for Efficient Feature Selection in Machine Learning. Medium. 2023. Available online: https://medium.com/@Kavya2099/optimizing-performance-selectkbest-for-efficient-feature-selection-in-machine-learning-3b635905ed48 (accessed on 30 September 2024).