**University of North Texas**

**ADTA 5900 - Advanced Data Analytics Capstone Experience**

**Deep Learning for Enhanced Trading Signal Generation:   
A Hybrid CNN-LSTM Approach to S&P 500 Technical Analysis**

Group Members – Biniam Abebe

Contents

[**1. INTRODUCTION** 2](#_Toc191467050)

[**1.1 Background** 2](#_Toc191467051)

[**1.2 Techniques** 2](#_Toc191467052)

[**1.4 Primary Research Question** 2](#_Toc191467053)

[**2. LITERATURE /SCHOLARLY REVIEW** 3](#_Toc191467054)

[**3. METHODOLOGY** 12](#_Toc191467055)

[**3.1 Dataset Description** 12](#_Toc191467056)

[**Table 2: Feature Categories and Descriptions** 13](#_Toc191467057)

[**References** 14](#_Toc191467058)

**1. INTRODUCTION**

**1.1 Background**

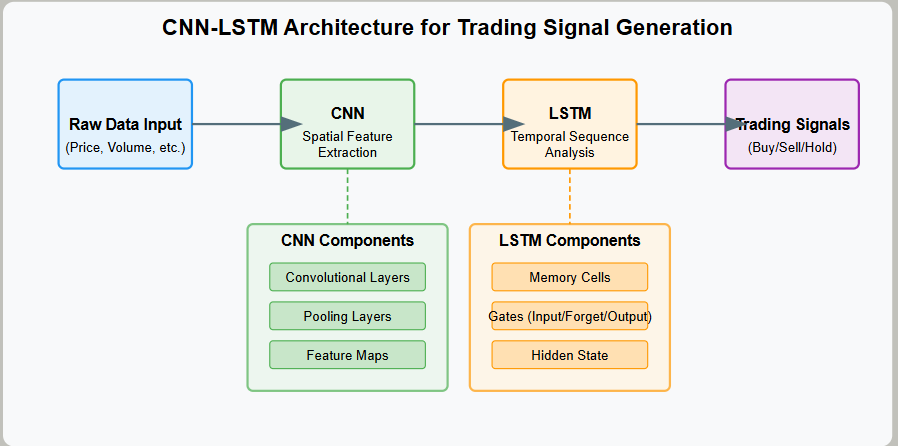
In today’s dynamic financial markets, the ability to adapt to rapidly evolving trends and identify profitable trading opportunities is critical for investment success. The U.S. equity market, representing over $7 trillion in market capitalization with algorithmic trading accounting for 60-70% of daily volume (S&P Global, 2024), underscores the importance of advanced pattern recognition tools in maintaining a competitive edge. Traditional technical analysis, which relies on human interpretation of chart patterns and indicators, is often subjective and influenced by psychological biases (Murphy, 2022). However, recent advancements in deep learning, particularly through hybrid architectures such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, offer a transformative approach to enhance the reliability and profitability of trading signals (Sezer et al., 2020). This project develops a hybrid CNN-LSTM model to analyze S&P 500 stock data, leveraging CNN’s strength in spatial pattern recognition and LSTM’s capability in temporal sequence analysis to generate superior trading signals compared to classical methods (Livieris et al., 2021).

**1.2 Techniques**

The hybrid CNN-LSTM model integrates two robust deep learning methodologies:

* **Convolutional Neural Networks (CNN)**: CNNs excel at detecting spatial patterns, such as trends and formations in stock price charts, by extracting relevant features from complex datasets.
* **Long Short-Term Memory (LSTM)**: LSTMs are designed to model temporal dependencies in sequential data, making them well-suited for analyzing time series like stock prices over extended periods.

This combination enables the model to capture both the spatial intricacies of price movements and the temporal relationships in financial time series, potentially outperforming traditional technical analysis. The workflow involves feeding preprocessed financial data into the CNN for feature extraction, followed by LSTM processing to analyze sequential patterns, culminating in the generation of actionable trading signals.



**Figure 1: The Processes of Hybrid CNN-LSTM Approach**  
**1.3 Challenges**

Implementing a deep learning approach in financial markets presents several challenges:

* **Data Quality and Quantity**: Financial datasets, such as the S&P 500 data used here, often contain noise and require significant preprocessing to ensure reliability.
* **Overfitting**: Complex models like CNN-LSTM risk overfitting to historical data, which may limit their generalizability to real-time trading scenarios.
* **Interpretability**: Deep learning models are often criticized as "black boxes," lacking the transparency of traditional technical analysis, which complicates trust and validation (Ozbayoglu et al., 2020).
* **Computational Resources**: Training a hybrid CNN-LSTM model demands substantial computational power and time, posing practical implementation hurdles.

**1.4 Primary Research Question**

How does a hybrid CNN-LSTM deep learning model improve trading signal reliability and profitability compared to traditional technical analysis?  
This project applies advanced deep learning techniques to improve stock trading signals for S&P 500 stocks, leveraging a hybrid Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) architecture. The CNN component identifies spatial patterns in price charts, while the LSTM component analyzes temporal sequences in financial time series. With a comprehensive dataset covering 501 S&P 500 companies over five years and minimal missing values (1.9%), the project uses a rich feature set of 76 indicators to test its hypotheses. It draws on concepts from an MS in Advanced Data Analytics, including coursework in Deep Learning with Big Data, Recurrent Neural Networks for Sequence Data, and data harvesting/storage.

**Key Hypotheses**

1. **Signal Generation Quality**: The hybrid CNN-LSTM model will provide significantly better accuracy in signal generation compared to traditional technical analysis.
2. **Trading Performance**: CNN-LSTM-based trading strategies will deliver statistically superior risk-adjusted returns.

**2. LITERATURE /SCHOLARLY REVIEW**

Over the years, numerous studies have been conducted to explore the application of deep learning techniques in financial markets. Researchers have investigated various approaches to improve stock price prediction, risk assessment, and trading strategies.

Deep learning has emerged as a powerful tool in the field of financial analysis, offering unprecedented capabilities for processing massive amounts of complex, high-dimensional, and non-linear datasets. Lee et al. (2020) proposed an innovative approach for training neural networks to predict the S&P 500 index price without using the index data itself for training. Unlike traditional methods that use limited stock index data (approximately 250 data points per year), their method leveraged data from individual companies to obtain sufficient training data, thereby addressing the data-shortage problem that commonly leads to overfitting. The researchers compared multiple neural network architectures (Multilayer Perceptron and Convolutional Neural Networks) trained with different learning algorithms (supervised learning and reinforcement learning) using both their method and traditional approaches. Their experiments demonstrated that neural networks trained on individual company data consistently outperformed those trained on S&P 500 data across various network structures. This approach allowed the models to learn richer representations of investment activities since they utilized price data generated directly from investor activities rather than the weighted averages represented by index values. The method proved robust, yielding 5-16% annual returns before transaction costs during the test period (2006-2018), outperforming the method proposed by Jeong and Kim that used deep Q-learning and transfer learning. When transaction costs were considered, the researchers implemented a "Lagged Position Change" algorithm that reduced transaction frequency while maintaining profitability.

Huang et al. (2020) review a wide range of studies in deep learning in finance and banking, mentioning deep learning is applied for tasks including market prediction, credit risk evaluation, fraud detection, etc. Huang et al. (2020) provide a comprehensive overview of deep learning models in finance, identifying seven core domains: credit risk prediction, macroeconomic prediction, exchange rate prediction, stock market prediction, oil price prediction, portfolio management, and stock trading. Their review of 40 articles published between 2014-2018 shows that traditional neural networks (feedforward neural networks/FNN, multilayer perceptron/MLP) are widely used across multiple financial domains, while specific architectures demonstrate advantages in particular applications. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks excel with time-series financial data, showing superior performance in stock market prediction due to their ability to capture temporal dependencies. Convolutional Neural Networks (CNN) prove effective for handling multicollinearity in financial data, while Reinforcement Learning (RL) offers promising results in stock trading applications where decision-making capabilities are crucial. Hybrid models combining multiple architectures often outperform standalone approaches, as seen in implementations that merge CNN-LSTM frameworks or integrate genetic algorithms with neural networks for technical analysis optimization. The literature also highlights the importance of appropriate data preprocessing techniques, evaluation metrics, and measures to address overfitting challenges in financial applications. While deep learning models have shown promising results in financial forecasting, questions remain about their long-term sustainability under the Efficient Market Hypothesis, as their performance may deteriorate as markets adapt to new prediction techniques.

Similarly, Ozbayoglu et al. (2020) review deep learning approaches applied to finance with an emphasis on the particular performance of these techniques in modeling time series processes, which are prominent in stock market prediction. These works lay the groundwork for the use of deep learning for financial problems, motivating the hybrid CNN-LSTM model used in this project to improve the reliability of trading signals. They present an extensive survey of deep learning (DL) applications in finance, analyzing 144 studies across domains such as algorithmic trading, risk assessment, fraud detection, portfolio management, asset pricing, cryptocurrency/blockchain, sentiment analysis, and text mining. The researchers categorized these works by financial subfield and DL model type, including Recurrent Neural Networks (RNNs, especially LSTM), Convolutional Neural Networks (CNNs), Deep Multilayer Perceptrons (DMLPs), and Deep Reinforcement Learning (DRL). Their findings indicate RNNs (notably LSTM) dominate time-series tasks like stock price forecasting due to their temporal modeling capabilities, while CNNs are increasingly used for classification via innovative 2D image transformations of financial data. DRL shows promise in optimizing trading strategies. The survey reveals algorithmic trading and text mining as the most active research areas, with hybrid models (e.g., CNN-LSTM) often outperforming standalone architectures. DL models consistently surpass traditional ML methods, though challenges like overfitting persist. Testing examples include LSTM-based stock trading models achieving high accuracy on datasets like BIST and S&P 500, and CNN-based credit scoring outperforming SVMs. The authors highlight future potential in cryptocurrencies, blockchain, and behavioral finance, suggesting hybrid spatio-temporal models and NLP integration as promising directions.

There has been a rise of hybrids, combining a convolutional layer CNN with a Long Short-Term Memory (LSTM) for time series analysis in finance. CNNs are good at recognizing spatial addresses such as patterns in price charts, while LSTMs capture temporal dependencies in sequential data. Shah et al. (2022) review several hybrid deep learning methods for stock prediction, finding that CNN-LSTM combinations tend to perform better than standalone models by capturing both spatial and temporal features. They conducted a comprehensive review of multiple hybrid deep learning approaches for stock prediction, focusing extensively on hybrid CNN-LSTM architectures. Their analysis showed that hybrid models combine the complementary strengths of their component architectures: LSTM models excel at capturing temporal dependencies and predicting precise stock prices, while CNN models are superior at identifying rapid changes and predicting general stock trends. The researchers evaluated various configurations including CNN-LSTM, CNN-BiLSTM, CNN-TLSTM, and models with attention mechanisms, finding that hybrid approaches consistently outperformed individual models. Their review demonstrated that CNN-BiLSTM-AM achieved the lowest error rates (MAE: 21.952, RMSE: 31.694) among compared models, with the CNN component efficiently extracting spatial features from time series data while the LSTM layers captured temporal patterns. The authors concluded that these hybrid architectures are particularly well-suited for high-frequency trading environments where both price accuracy and trend detection are crucial for profitable decision-making.

Additionally, Wu et al. Chang et al. (2023) propose a graph-based CNN-LSTM algorithm that integrates leading indicators, thereby enhancing stock price prediction accuracy. Such findings confirm the project’s architecture, adding to its capabilities in creating better trading signals. They proposed a graph-based CNN-LSTM stock price prediction algorithm (SACLSTM) that incorporated leading indicators like options and futures alongside historical price data. Their approach uniquely constructed a sequence array of historical data with its corresponding leading indicators, using this array as input to a CNN framework that extracted feature vectors subsequently fed into an LSTM network. The authors demonstrated that using leading indicators as experimental data yielded better predictions than using historical data alone, with options data providing higher accuracy than futures data. Testing on stocks from both U.S. and Taiwanese markets, they found their hybrid architecture consistently outperformed traditional methods including SVM, CNNpred, CNN-corr, and standard neural networks across different prediction timeframes. The model maintained higher accuracy when predicting next-day market movement compared to 3-day or 7-day forecasts. Experimental results showed that combining all indicators (historical, options, and futures data) achieved the highest prediction accuracy, supporting their hypothesis that more comprehensive input data leads to more accurate forecasts.

Including technical analysis indicators in deep learning models can improve predictive performance. Sezer et al. (2017) propose a deep neural network-based trading system which optimizes technical analysis parameters using evolutionary algorithms and outperforms conventional methods. They proposed a novel stock trading system that combines genetic algorithms with deep neural networks. Their approach uniquely used genetic algorithms to optimize technical analysis parameters (specifically RSI values) for creating buy-sell trigger points, which were then passed to a deep multilayer perceptron (MLP) neural network for buy-sell-hold predictions. The system was developed on Apache Spark big data platform and tested on Dow 30 stocks using daily close prices between 1996-2016. The deep MLP had 7 layers with topology configured as (3, 20, 10, 8, 6, 5, 3). Their results showed that optimizing technical indicator parameters not only enhanced stock trading performance but also provided a model that could serve as a viable alternative to Buy-and-Hold strategies. The evolutionary optimization approach yielded better results than both standard technical analysis models and their previous MLP model that used non-optimized indicators. Their findings demonstrated that combining evolutionary optimization with deep learning can create more effective trading systems by tailoring technical analysis parameters to specific market conditions.

Patel et al. (2015) recently showed that stock return forecasting accuracy can be improved by conveniently augmenting a deep learning model with a technical indicator. These studies verify the approach of combining with the hybrid CNN-LSTM model 76 indicators, affecting price, moving averages, volatility, volume, fundamentals, and characteristics of the market to improve the generated signals of the project. When analyzing deep neural networks with feature selection methods for stock price direction forecasting using technical analysis indicators. Their research explored a comprehensive set of 124 technical indicators, applying three feature selection methods to eliminate redundant information from similar indicators. Using daily data from stocks across seven global market indices between 2008 and 2015, they tested neural networks with different configurations of hidden layers and dropout rates. Their findings revealed that variables were not uniformly selected by feature selection algorithms, and that out-of-sample accuracy converged to two distinct values: 50% (suggesting market efficiency) and a "strange attractor" of 65% accuracy that was achieved consistently across markets. Despite the relatively good prediction accuracy, they found that trading strategies based on these models generally failed to significantly outperform simple Buy-and-Hold strategies when transaction costs were considered, with some hyperparameter combinations even showing substantially negative returns. This research highlights the challenge of translating predictive accuracy into profitable trading strategies, and demonstrates the importance of considering transaction costs when evaluating machine learning models for financial applications.

Evaluating trading strategies requires well-defined performance metrics. Saud et al. (2024) discuss smart trading strategies with the help of technical indicators, with performance evaluation done on accuracy, Sharpe ratio, and win ratio. Such metrics are consistent with the project’s evaluation framework which comprises signal accuracy, Sharpe ratio, maximum drawdown, win/loss ratio, and profit factor. This stability guarantees a solid empirical comparison between the hybrid model and conventional technical analysis and evaluates reliability as well as profitability, the project’s main research question. In their research, developed intelligent stock trading signal prediction strategies using MACD, DMI, and KST technical indicators, implementing these with LSTM and GRU networks due to their ability to manage long-term dependencies. The strategies were evaluated using three key performance metrics: Annual Rate of Return (ARR), Sharpe Ratio (SR), and Win Rate. Testing these approaches on 18 stocks from NEPSE, BSE, and NYSE exchanges led to four main findings: (1) A 5-day lookback period proved optimal for intelligent strategies using MACD and DMI indicators, while a 10-day lookback period worked best for KST-based strategies; (2) GRU networks demonstrated superior performance compared to LSTM implementations; (3) The intelligent trading strategies significantly outperformed their classical counterparts across all performance metrics; and (4) Among the three proposed approaches, the MACD-based strategy was found to be the most effective and least risky. This research demonstrates the effectiveness of combining machine learning with traditional technical analysis to filter out false trading signals and identify true patterns in market indicators.

Peng et al. (2021) analyzed deep neural networks with feature selection methods for stock price direction forecasting using technical analysis indicators. Their research explored a comprehensive set of 124 technical indicators, applying three feature selection methods to eliminate redundant information from similar indicators. Using daily data from stocks across seven global market indices between 2008 and 2019, they tested neural networks with different configurations of hidden layers and dropout rates. Their findings revealed that variables were not uniformly selected by feature selection algorithms, and that out-of-sample accuracy converged to two distinct values: 50% (suggesting market efficiency) and a "strange attractor" of 65% accuracy that was achieved consistently across markets. Despite the relatively good prediction accuracy, they found that trading strategies based on these models generally failed to significantly outperform simple Buy-and-Hold strategies when transaction costs were considered, with some hyperparameter combinations even showing substantially negative returns. This research highlights the challenge of translating predictive accuracy into profitable trading strategies, and demonstrates the importance of considering transaction costs when evaluating machine learning models for financial applications.

Applicability Research specifically targeting the S&P 500 provides insights that can be directly applied to this project. Kamalov et al. (2021) proposed a convolution-based neural network model for predicting the next-day direction of the S&P 500 index. Their approach featured a distinctive architecture with two hidden layers: a convolutional layer with four filters of size 3, followed by a fully connected layer. The key insight of their model was utilizing convolution operations to consider each data point in the context of its temporal neighbors, which allowed more informative feature extraction. The model used previous closing values and trading volume from the past 14 days as inputs. Testing their approach against seven benchmark models, including fully connected networks, RNNs, and LSTM architectures, their proposed model achieved the highest accuracy rate of 56.21% in predicting next-day market direction. This outperformed both random guessing (50%) and other neural network configurations. The authors also implemented regularization techniques including early stopping and a shallow architecture to prevent overfitting. Their findings demonstrated that convolution-based neural networks can be effectively applied to financial time series data and offer predictive capabilities that exceed traditional approaches.

Shah et al. (2022) provided a thorough comparison between traditional technical analysis methods and advanced deep learning approaches for stock prediction. Their review examined how traditional models like ARIMA, which assume linear relationships between past and future values, consistently underperformed compared to deep learning models. While ARIMA achieved reasonable accuracy (85-95% in some studies), it struggled with non-linear, volatile market data and required manual parameter tuning. The researchers found that neural network models significantly outperformed traditional methods, with LSTM models showing 84-87% improvement over ARIMA in some comparisons. Their analysis also revealed that although traditional technical analysis relies on indicators like Moving Averages, MACD, and RSI, these can be more effectively utilized when incorporated as features within deep learning frameworks. The integration of these traditional indicators with modern neural networks represents a substantial advancement over purely classical technical analysis approaches, particularly for capturing complex market patterns that traditional methods fail to identify.

**Table 1: Summary of Scholarly References**

| **Reference** | **Focus** | **Key Finding** | **Relevance to Project** |
| --- | --- | --- | --- |
| Huang et al. (2020) | Deep learning in finance | Deep learning excels in handling complex financial data for forecasting | Supports use of advanced models |
| Kamalov et al. (2021) | S&P 500 forecasting | Validates deep learning for S&P 500 index prediction | Confirms dataset applicability |
| Lee et al. (2020) | S&P 500 prediction | Effective prediction without index data using neural networks | Reinforces individual stock focus |
| Ozbayoglu et al. (2020) | Deep learning survey | Surveys financial applications, noting interpretability challenges | Contextualizes project challenges |
| Patel et al. (2015) | Technical indicators + deep learning | Combining indicators with deep learning enhances stock prediction | Supports feature integration |
| Saud et al. (2024) | Intelligent trading strategies with technical indicators | Demonstrates potential for superior risk-adjusted returns using metrics like Sharpe ratio and win rate | Aligns with the project’s performance metrics and comparison to traditional methods |
| Peng et al. (2021) | Feature selection in deep learning for stock forecasting | Technical indicators enhance the accuracy of stock price direction forecasting | Highlights the importance of the 76-indicator feature set for improving model performance |
| Sezer et al. (2017) | Deep neural networks | Optimized technical parameters improve trading systems | Backs hybrid model design |
| Shah et al. (2022) | Hybrid deep learning review | Hybrid models outperform single architectures in stock prediction | Endorses CNN-LSTM approach |
| Wu et al. (2023) | CNN-LSTM with indicators | Graph-based hybrid model improves prediction with leading indicators | Suggests additional feature enhancements |

**3. METHODOLOGY**

**3.1 Dataset Description**

The dataset comprises daily data for 501 S&P 500 companies, spanning February 2, 2020, to January 31, 2025, with 622,641 total observations. Sourced from the Yahoo Finance API (2024), it includes 76 features across seven categories, providing a robust foundation for deep learning analysis:

**Table 1: Dataset Overview**

|  |  |  |
| --- | --- | --- |
| **Characteristic** | **Value** | **Definition** |
| Total Companies | 501 | Number of unique companies included in the dataset |
| Total Observations | 622,641 | Total number of daily data points across all companies |
| Date Range | 2020-02-02 to  2025-01-31 | Temporal span of the dataset |
| Number of Features | 76 | Total number of variables tracked per observation |
| Data Points per Company | 1,242.80 (avg) | The average number of trading days recorded per company |
| Missing Values | 1.90% | Percentage of data points with missing values |
| Dataset Size | 365.78 MB | Total memory usage of the dataset |

**Table 2: Feature Categories and Descriptions**

|  |  |  |
| --- | --- | --- |
| **Category** | **Features** | **Description** |
| Price Indicators | Close, Returns, Log\_Returns, Price\_Range, Price\_Range\_Pct | Basic price measurements and their derivatives, capturing daily price movements and ranges |
| Moving Averages | MA\_X, EMA\_X, Returns\_Xd | Various time-window averages (X=5,10,20,50,200 days) providing trend information |
| Volatility Metrics | Volatility\_Xd, Volume\_MA\_Xd, BB\_Width\_X | Measures of price and volume variability, including Bollinger Band indicators |
| Technical Indicators | RSI\_X, MACD, Signal\_Line, MACD\_Histogram, Momentum\_14, ROC\_14, MFI\_X, Channel\_Width\_X | Advanced technical analysis indicators measuring momentum, trend strength, and price dynamics |
| Volume Indicators | OBV, Volume\_Ratio, Volume\_StdDev | Metrics tracking trading volume patterns and anomalies |
| Fundamental Features | PE\_Ratio, PB\_Ratio, Dividend\_Yield, Profit\_Margin, Beta, Enterprise\_Value, Forward\_EPS, Trailing\_EPS | Company-specific financial and valuation metrics |
| Market Features | Market\_Return, Market\_Volatility, Rolling\_Beta, VIX, VIX\_MA\_10 | Broader market indicators and their relationship to individual securities |

With only 1.9% missing values and an average of 1,242.8 data points per company, the dataset’s high quality and volume (365.78 MB) make it ideal for training complex models like CNN-LSTM.

**References**

1. Peng, Y., Albuquerque, P. H. M., Kimura, H., & Saavedra, C. A. P. B. (2021). Feature selection and deep neural networks for stock price direction forecasting using technical analysis indicators. Machine Learning with Applications, 5, Article 100060. <https://doi.org/10.1016/j.mlwa.2021.100060>
2. Sezer, O. B., Ozbayoglu, M., & Dogdu, E. (2017). A deep neural-network based stock trading system based on evolutionary optimized technical analysis parameters. Procedia Computer Science, 114, 473–480. https://doi.org/10.1016/j.procs.2017.09.031
3. Kamalov, F., Gurrib, I., & Rajab, K. (2021). Forecasting with deep learning: S&P 500 index. *arXiv*. <https://arxiv.org/abs/2103.14080>
4. Saud, A. S., & Shakya, S. (2024). Technical indicator empowered intelligent strategies to predict stock trading signals. Journal of Open Innovation: Technology, Market, and Complexity, 10(4), Article 100398. <https://doi.org/10.1016/j.joitmc.2024.100398>
5. Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Stock prediction based on technical indicators using deep learning model. *Computers, Materials & Continua, 70*(1), 287-303. <https://www.techscience.com/cmc/v70n1/44330/pdf>
6. Shah, J., Vaidya, D., & Shah, M. (2022). A comprehensive review on multiple hybrid deep learning approaches for stock prediction. Intelligent Systems with Applications, 16, Article 200111. <https://doi.org/10.1016/j.iswa.2022.200111>
7. Lee J, Kang J (2020) .Effectively training neural networks for stock index prediction: Predicting the S&P 500 index without using its index data. *PLoS ONE, 15*(4), Article e0230635. <https://doi.org/10.1371/journal.pone.0230635>
8. Wu, J. M.-T., Li, Z., Herencsar, N., Vo, B., & Lin, J. C.-W. (2023). A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. Multimedia Systems, 29(3), 1751–1770. <https://doi.org/10.1007/s00530-021-00758-w>
9. Huang, J., Chai, J. & Cho, S. Deep learning in finance and banking: A literature review and classification. *Front. Bus. Res. China* 14, 13 (2020). <https://doi.org/10.1186/s11782-020-00082-6>
10. Ozbayoglu, A. M., Gudelek, M. U., & Sezer, O. B. (2020). Deep learning for financial applications: A survey. Applied Soft Computing, 93, Article 106384. <https://doi.org/10.1016/j.asoc.2020.106384>
11. Livieris, I. E., Pintelas, E., & Pintelas, P. (2021). A CNN-LSTM model for gold price time-series forecasting. Neural Computing and Applications, 33(7), 2445-2459. https://link.springer.com/article/10.1007/s00521-020-04867-x
12. Murphy, J. J. (2022). Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. New York Institute of Finance. <https://archive.org/details/technicalanalysi0000murp>
13. S&P Global. (2024). S&P 500 Index Methodology. Retrieved from S&P Global website. <https://www.spglobal.com/spdji/en/documents/methodologies/methodology-sp-us-indices.pdf>
14. Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied Soft Computing, 90, 106181.   
      
    https://www.sciencedirect.com/science/article/abs/pii/S1568494620301216?via%3Dihub
15. Sharpe, W. F. (1994). The Sharpe ratio. Journal of Portfolio Management, 21(1), 49-58. <https://www.scirp.org/reference/referencespapers?referenceid=1451308>
16. Yahoo Finance. (2024). Yahoo Finance API Documentation. Retrieved from Yahoo Finance website <https://pypi.org/project/yfinance/>