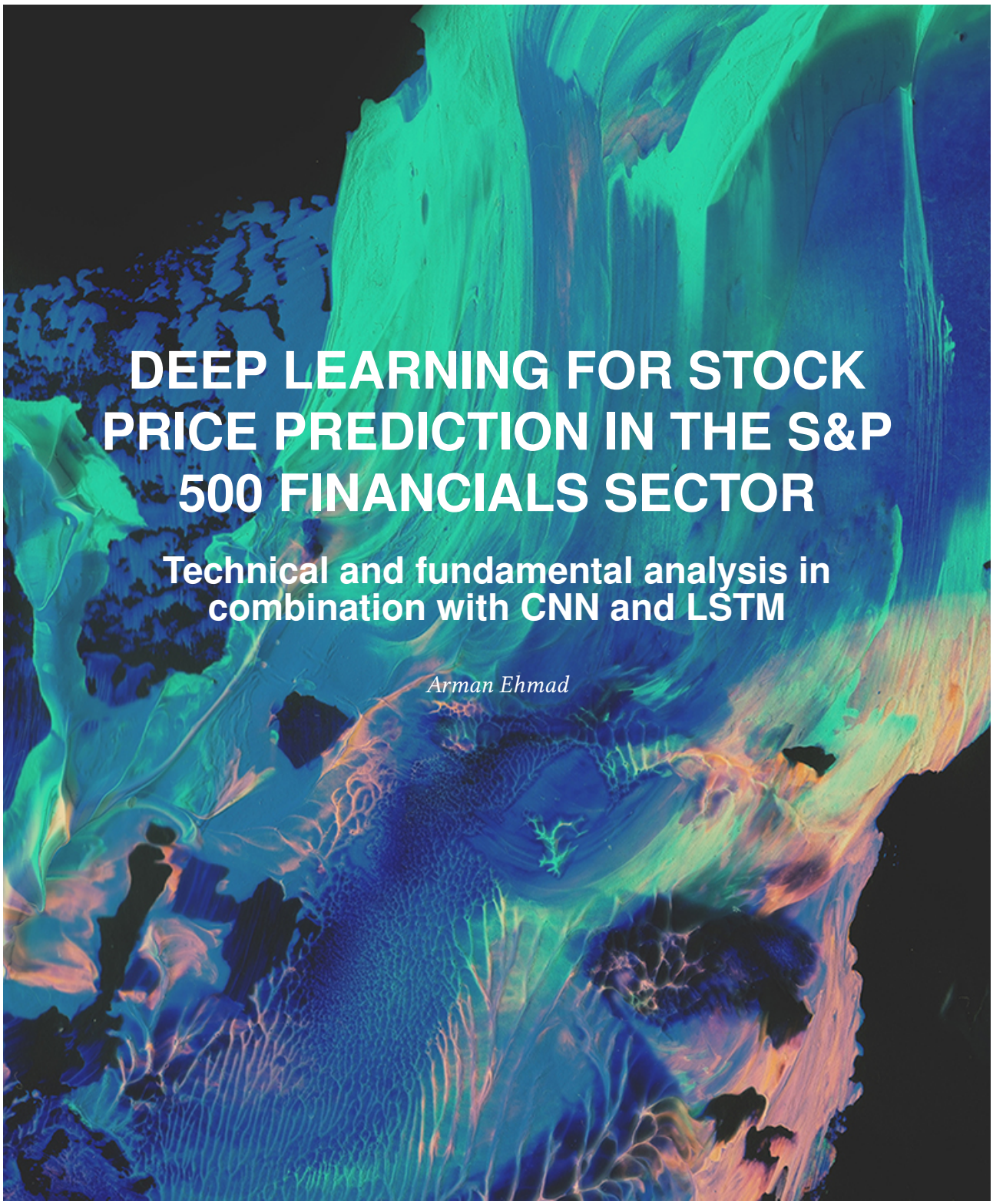

DEEP LEARNING FOR STOCK PRICE PREDICTION IN THE S&P 500 FINANCIALS SECTOR

Technical and fundamental analysis in combination with CNN and LSTM

Arman Ehmad

Bachelor Thesis, 15 hp/credits
Bachelor's degree in computer science

2025

Abstract

Unpredictable and limited information are critical reasons behind the challenges faced in forecasting one-day-ahead stock prices across S&P 500 financial sector. Through this thesis, we propose a structured deep-learning mechanism which combines twenty-days-worth of sliding windows of per-ticker-normalized closing price across nine quarterly financial ratios. The financial ratios include P/B (price-to-book), EPS (earnings per share), P/E (price-to-earnings), EV/EBITDA (Enterprise Value-to-EBITDA), PEG, Beta, ROE (Return on Equity), ROA (Return on Assets), and Debt-to-Equity. The study proposes three models based on data collected via seventy-three stocks. The goal is to to evaluate whether the integration of quarterly fundamental features with technical price windows improves prediction accuracy across the financial sector of the S&P 500, in comparison to relying solely on technical data.

Each model is assessed using standard error metrics along with the share of variance it explains. Hybrid architecture model showcases highly significant performance in comparison to the single LSTM as well as pure CNN primarily due to lowered errors in predictions while it effectively explains a larger number of variability in target sections. The thesis further finds that the effectiveness of the hybrid setup diminishes when small random variations are applied to technical inputs without ensuring adjustments to fundamental data, especially since such disturbances interfere with crucial signals. The research findings further indicate that integrating technical features with fundamental concepts helps generate reliable one-day-ahead predictions in comparison to only focusing on price information.

Contents

1 Introduction

The stock market is a highly volatile industry involving activities where people and companies are engaged in gaining market returns through share trading. Individuals who understand the market well are able to predict its ups and downs, further generating a line of passive income. Analytical strategies for stock forecasting and market activities are critical for profitable investment decisions [14]. The activities surrounding forecasting market future require sequential as well as multidimensional data, wherein the critical challenge exists in finding patterns and using it to improve prediction accuracy.

Technical analysis and fundamental analysis are common approaches toward financial analysis; technical analysis using existing trends and trading information to predict upcoming movements primarily depending on the argument that market history repeats itself; on the other hand, financial analysis uses financial strength of an organization to estimate its true value. In recent years, sentiment analysis has gained traction globally; it uses conceptual methods such as NLP (Natural Language Processing) to evaluate news and social media, further evaluating how public opinions shape investor decisions and stock value [5, 14].

The structure of stock market forecasting has evolved due to the advent of DL (Deep Learning) and ML (Machine Learning) as these measures are effective in identifying complex and non-linear patterns in data which traditional statistics may overlook. Some of the widely used neural networks in the digital age are LSTM (Long Short-Term Memory) and CNNs (Convolutional Neural Networks). On one hand, LSTMs are largely effective for time series information as they are able to remember longer sequences of data, which solves the issues posed by long-term dependencies [8]. On the other hand, CNNs are effective in identifying local patterns in developed data, which can further be integrated into higher-level features [13].

Preliminary investigation suggests that previous studies have investigated if combining technical and fundamental data improves stock price prediction for financial companies in the S&P 500 [1]. The research findings suggest that the models which used both types of data significantly outperformed models which were only technical or included only fundamental inputs. This finding further suggests that for models to capture multiple dimensions of market behavior, combining diverse data-sources is needed which creates stronger feature sets. Researchers have further suggested testing the method on more organizations in future research studies.

In this thesis, we further use these findings by applying DL models using both technical and fundamental features across financial sector of the S&P 500 rather than three organizations. The main research question is: to what extent does including quarterly fundamental-analysis features improve prediction accuracy? To answer the research question, we design a comprehensive pipeline, processing 20-day sliding windows of stock closing prices and nine quarterly financial ratios. We further test three models: an LSTM with only price windows, a CNN with same data input, and a hybrid model (combined Conv1D–Dense model) including CNN-based price qualities with basic information.

The research hypothesis for this thesis is that a combination of fundamental ratios and technical data will allow reduced forecasting mistakes in comparison to only using the latter.

The key idea supporting the hypothesis is that fundamentals are a representation of long-term financial strength of an organisation and short-term is reflected in current market actions, combing them provides a comprehensive understanding of stock behaviour. To further test its effectiveness data augmentation is implemented to only price sequences while not on fundamentals, enabling each sequence to remain connected to financial profile of the organization.

Performance is measured with the help of multiple metrics; MSE (Mean Squared Error) represents the average squared difference between predictions and actual values; coefficient of determination (R^2) points toward the value of price variations the model details; MAE (Mean Absolute Error) measures difference between prediction and actual data; and, RAE (Relative Absolute Error) compares results with baseline data. Using multiple metrics creates balanced evaluation.

Initial results showcase that LTSM largely performs better than CNN as per an evaluation of close to 35,000 sliding window data samples from 73 companies in financial sector. Despite this, it is further gathered that hybrid model achieves most optimum comprehensive results. Moreover, a key finding is that implementing augmentation reduces performance across hybrid models, representing the challenge of balancing different input types during augmentation process.

The entirety of the thesis is further divided into chapters as follows: second chapter provides background data, third provides review of literature, fourth chapter details the methodology, fifth chapter provides results, sixth conducts discussion, and seventh chapter concludes the research.

2 Background

2.1 Technical analysis

Technical analysis refers to the method of predicting the future of how stocks and stock market will perform and move in the future by evaluating their movement in the past. Differentiated from other methods of analysis, technical analysis does not analyse the financials of companies or global news, instead it is entirely focused on data gathered through such as stock prices, volumes, and trends in charts. The process of conducting technical analysis may involve activities such as tracing how a stock goes up after falling to a certain level, or may even focus on evaluating if the stock tends to follow patterns of market trends. Technical analysis can be conducted with the help of tools such as Moving Averages (MA), support as well as Resistance Levels (RL), Relative Strength Index (RSI), and it may also be conducted through MACD (Moving Average Convergence Divergence). MA calculates the average price over, such as 50 or 200 days[1]. On the other hand, RL may showcase the turning point of the stock price [2]. Dissimilar to MA and RL, RSI and MACD shows if a stock is overbought or oversold [11]. In conclusion, market prices are not completely arbitrary; wherein the primary idea is that individuals often take similar actions in similar situations, causing certain patterns to take shape, indicating that past price information provide key information about future.

2.2 Fundamental analysis

The process of fundamental analysis involves discovering the real value of a stock, based on key information such as financial data of the organization, future goals, and global market conditions. In essence, it does not focus on price graphs, rather annual reports, key data points, and economic market conditions. There are three levels of fundamental analysis:

1. Global Analysis: Examines the overall condition of the world economy, including factors such as interest rates, inflation levels, and general economic performance.
2. Industry Analysis: Focuses on the company's sector, assessing whether the industry is expanding or facing intense competition.
3. Company Analysis: Evaluates the specific firm's performance by reviewing its financial statements, debt obligations, profitability, and cash flow position.

In our evaluation, we use key ratios to find valuation, profit data, leverage, as well as market risks. The nine ratios include:

[1] E8 Markets, *Understanding Support and Resistance Levels in Forex Trading* (2025) `https://blog.e8markets.com/articles/understanding-support-and-resistance-levels-in-forex-trading`

[2] CME Group, *Support and Resistance* (2025) `https://www.cmegroup.com/education/courses/technical-analysis/support-and-resistance.html`

1. Price-to-Book (P/B): This ratio compares a company's market price per share with its book value per share (shareholders' equity divided by total shares)[3].

2. Earnings Per Share (EPS): Calculated by dividing net income by the total number of outstanding shares, and shows how much profit is earned for each share, serving as a key measure of profitability[4].

3. Price-to-Earnings (P/E): The market price per share divided by EPS, reflecting how much investors are willing to pay for each dollar of a company's earnings[5].

4. Enterprise Value-to-EBITDA (EV/EBITDA): This ratio compares enterprise value with earnings before interest, taxes, depreciation, and amortization, adjusts for capital structure, and is often used to compare companies regardless of how they are financed[6].

5. PEG Ratio (Price/Earnings-to-Growth): The P/E ratio divided by the earnings growth rate. By including growth, this ratio helps judge whether a high P/E is reasonable given expected future expansion[7].

6. Beta: A beta above 1 indicates higher volatility than the market, while a beta below 1 shows lower volatility. It is a measure of a stock's volatility compared to the overall market, often the S&P 500[8].

7. Return on Equity (ROE): Net income divided by average shareholders' equity, wherein the ROE assesses how well a company uses its equity capital to generate profits[9].

8. Return on Assets (ROA): ROA is conducted through the calculation of net income divided by average total assets, which shows how effectively a company uses its assets to create earnings[10].

9. Debt-to-Equity (D/E): It is calculated when total liabilities divided by shareholders' equity. This ratio indicates how much debt a company uses compared to equity to finance its operations[11].

It is further assessed that fundamental analysis generally assumes that the market often exaggerates values, however, in the long run the data usually catches up. Based on this, it is argued that if an organization generates revenue and steadily improves its market positioning, while lowering debt, it will be rewarded for the same in terms of share prices. Recent studies have also found that there is a direct connection between fundamental elements and returns generated in the future.

[3] Investopedia, *Price-to-Book (P/B) Ratio* (2025) https://www.investopedia.com/terms/p/price-to-bookratio.asp

[4] QuarterChart, *Earnings Per Share (EPS)* (2025) https://www.quarterchart.com/financial-insight/eps

[5] QuarterChart, *Price-to-Earnings (P/E) Ratio* (2025) https://www.quarterchart.com/financial-insight/price-to-earnings-ratio

[6] Keystocks, *The Retail Investor's Guide to Financial Analysis* (2025) https://keystocks.com/market-commentary/the-retail-investors-guide-to-financial-analysis

[7] HDFC Sky, *PEG Ratio* (2025) https://hdfcsky.com/sky-learn/share-trading/peg-ratio

[8] Investopedia, *Beta: Know the Risk* (2025) https://www.investopedia.com/investing/beta-know-risk

[9] Investopedia, *Return on Equity (ROE)* (2025) https://www.investopedia.com/terms/r/returnonequity.asp

[10] Investopedia, *Return on Assets (ROA)* (2025) https://www.investopedia.com/articles/fundamental/04/012804.asp

[11] Alliance Advisers, *Understanding Balance Sheets: Assets, Liabilities, and Equity* (2025) https://www.allianceadvisers.com/blog/36/understanding-balance-sheets-assets-liabilities-and-equity

2.3 Convolutional neural networks

CNN was originally developed for image recognition activities, however, it has increasingly benefitted stock market predictions [12]. As the one-dimensional filters are implemented to time series activities such as stock prices to understand local patterns, these help in gathering short-term trends and sudden fluctuations [9]. These filters play the role of feature identifiers, gathering key signals to be used for anticipating future price changes.

Although CNNS are not designed to hold long-term information, they are largely effective in identifying short-term dependencies. Through application of filters of different sizes, CNNs can effectively identify structures at diverse levels, such as patterns in 5-days or even 10-days. There are various studies which have found that CNNs can be largely more effective than traditional statistical models, and in various cases they have also provided better results than recurrent models when short-term behavior is most important for prediction [12]. Despite its strengths, CNN is often limited when focusing on long-term dependencies, apart from deeper networks being built or combining CNN with other effective models such as the LSTMs [2].

2.4 Long short-term memory neural networks

The LSTMs are categorized as an advanced version of RNN (Recurrent Neural Network) [8], which is mostly designed for managing sequential data such as stock prices as they capture and learn from past data inputs, ensuring that they are transparent for predicting time series data. The basic RNNs often face challenges when dealing with long-term dependencies as it is obstructed by gradient issues [8], wherein their influence vanishes considering how the sequence grows.

LSTM is the key differentiator as it addresses the problem by designing personalized architecture including memory cells and gating system and the gates under the mechanism structurally regulates flow of data by making decisions such as which data to retain, update, or remove in the next step. The LSTM design enables the network to decide which dataset to retain in the long-term, further removing unrequired information. Considering this, the LSTMs gain information from both short-term market changes and longer-term financial cycles, offering flexible memory retention system.

Research further indicates that LSTMs have constantly performed significantly better than traditional prediction methods including ARIMA (Autoregressive Integrated Moving Average) model and normal RNNs across stock prediction activities. LSTMs have been positively applied to large datasets such as S&P 500, as well as adapted effectively to both unpredictable and cyclical market patterns [7]. Despite their need for high computational resources, it remains a largely effective method of gathering time-regulated financial information.

3 Literature review

According to [16], a combination of engineered technical indicators along with non-linear ML models can be used to improve the precision of next-day stock closing price prediction. In their study, the authors directly address the gap faced in traditional methods such as ARIMA which implemented engineered methods as well as intelligent infrastructure to evaluate and understand complexities of financial time series in an improved manner. The study uses multiple daily data values including Open, High, Low, Close, Adj Close, and Volume from five NYSE (New York Stock Exchange) listed organizations. In their research, the authors developed these six key indicators and tested three-layered feed-forward ANN (Artificial Neural Network) against RF (Random Forest regressor), and findings from the study suggest that ANNs were far better in performance than RF across most organizations. Minor changes were found in the datasets, however, it was gathered that ANN's non-linear learning ability was better. The study validated the role of ML in forecasting stock values as feature-engineered ANNs were significantly better than RF across short-form price evaluation.

Following research [3], conducted a systematic review of literature for neural network based stock forecasting techniques between the years 2015-2023, tracing the evolution in architecture (RNNs, CNNs, Transformer Models, GNNs Graph Neural Networks, GANs (Generative Adversarial Networks), as well as LLMs (Large Language Models)). The study further identified gaps such as the under-evaluation of emerging models including LLMs as well as detailed insights. Researchers conducted a thorough survey of peer-reviewed studies, ensuring they are categorized as per core neural families and data modalities, such as price, news, social media, earnings calls, and knowledge graphs. The key argument of the study is that the choice of design must be aligned with characteristics of data, for example the RNNs are best aligned with temporal dynamics, CNNs perform significantly with local features, transformers are for the long dependency data, GNNs for relational data, GANs for ensuring augmentation, and best use of LLMs is for texts. It brings together earlier research by connecting what models can do with the requirements of forecasting, and highlights the growing use of advanced, multimodal approaches.

It is gathered that researchers [7] conducted a large-scale time-series prediction primary study focusing on S&P 500 constituents, and further conducted a comparison and evaluation of their results with other models including RF, basic deep neural network, and based on a logistical regression, they gathered that LSTM achieves a higher result every day. Through their research, the authors found various characteristics and qualities which makes stocks favourites to be selected for trading. Some of these features include stocks which are unpredictable, which are often the ones that have moved significantly up and down and are expected to be reversed in some time. Based on their findings, the researchers developed a strategy wherein short-time losers are bought and winners are sold.

The research conducted by [15] cultivates a hybrid DL model including every day headlines which CNN model processes along with technical indicators analyzed through LTSM to predict Chevron's stock price next day. The researchers assessed four models: two LSTM variants using different sets of indicators (7 features and 17 features), and two SI-RCNN models that integrate CNN for extracting sentiment from financial news with LSTM for modeling technical indicators. The process of training used 1,385 days of training for model optimiza-

tion with the help of SDG (Stochastic Gradient Descent), focusing in momentum as it was applied to Google News word2vec embeddings and temporal CNN filters and LSTM units.

It is further gathered that the SI-RCNN model achieves directional accuracy of 56.8%, which is 2.8% of an improvement than LSTM model. Using the hybrid strategy achieved a 13.94% return during 139-day back-test period, including transaction costs, while the buy-and-hold strategy produced 3.22% returns. The study further gathers that two different hybrid models implemented through diverse tech indicator was able to produce resembling result, confirming the impact of news sentiments in system stabilization. Moreover, researchers [15] found that combining the CNN/LSTM models improves performance more than individual models in terms of both prediction precision and financial results, offering a basis for more advanced integration of news and technical data.

A thorough review of existing literature was done by researchers [4], they investigated the use of single-layer LSTM model, which uses an amalgamation of fundamental, macroeconomic and technical indicators to project the closing values for next day S&P 500. The study further addressed the gap of existing LSTM studies which either use few feature sets or multilayer design which did not systematically select the indicators. The study used data from daily S&P 500 as well as nine predictors, overall utilizing 3,900 days of trade. Researchers compared single-layer and multi-layer LSTMs, tested both shallow and deep designs, and optimized training settings through stepwise grid search method and early stopping technique. Each setup developed by the researchers was then thoroughly evaluated via multiple differentiated trials to find most consistent next-day forecasting combination. Thus, the finding from the study reveals that implementing a rich and detailed combination of data inputs allows a largely simple LSTM, with only one layer of 150 units, to outperform complex and deeper LSTM models by a significant statistic margin as well as remains reliable during major market changes, showcasing how design effectiveness and diversity features hold a significant power. Thus, based on these findings, the authors argue that when a carefully selected set of indicators are implemented in a simple LSTM, it generalizes better than deeper networks which may be challenged with overfitting. The findings are aligned with earlier research on RNN [7] and differentiated from authors advocating multilayer LSTM [10].

The research conducted by [9] examine CNNs can effectively integrate diverse market as well as economic indicators to project the next day direction of critical US stocks. To conduct a critical examination, the authors design a specialized filter mechanism which can be used for both understanding variable relationship on the day as well as identify key trends over a small period of time. The study further illustrates that authors designed 2 convolutional predictors, 2D-CNNpred and 3D-CNNpred, which use 60-days of data from the market, yet is able to process data through differentiated mechanism. Both of the models provide a distinguished advantage in terms of variable relations and data use. It is further gathered that the 2D-CNNpred model converts daily market data into vectors, which are then combined to form daily features and implementing in identifying short-term patterns over several days. The 3D-CNNpred model builds on this by including data from multiple market segments, merging their daily information, and then applying convolutional learning to capture both the relationships between markets and how these relationships change over time. The networks include stacked convolutional layers along with ReLU activations. These models also include pooling as well as dropout mechanism, reducing overfitting through a reduction of neurons as the training stage is ongoing, before the fishing phase with sigmoid output which is able to convert values into effective probabilities across two digits, 0 and 1. The researchers use Adam optimizer to train both models using data from 2010-2017 for five significant US indices, considering that 2D-CNNpred performs better than basic ANN-based benchmarks and 3D-CNNpred reflects higher accuracy in prediction considering its ability of using market as

well as temporal data.

In the following study [17] researchers develop a hybrid model combining of CNN-LSTM model refers to SACLSTM which uses futures (which is a standard agreement requiring the buyer to purchase or the seller to complete a sale of asset at a price which was determined before based on a future specified date) and options (which is an agreement granting the holder the option, however not a requirement, to buy or sell asset at an established price, on or before a specific date), leading derivative indicators to be formed together, based on established prices to further improve next-day predictions. The study further indicates that the authors tackle the absence of unified methods by introducing "stock sequence arrays", which capture both cross-instrument links and temporal patterns from price data and derivative signals. They test their hybrid SACLSTM model against a standard CNN by applying convolution and pooling layers, with batch normalization and dropout. Results show that SACLSTM outperforms CNN-only and LSTM-only models in next-day predictions and continues to perform better for multi-day forecasts using both price and derivative data. By combining both approaches, CNN and LSTM, the model uncovers price-derivative joint signals that sequential or price-only models miss, highlighting the value of diverse multimodal inputs. By unique and differentiated integration of graph-focused filters, LSTM layers, as well as SACLSTM leading indicators, the researchers significantly improved short-term forecasting precision.

Following study [1] investigated the training effectiveness of MLP (Multilayer Perceptron) while integrating fundamental indicators along with technical signals and market sentiment information, with the goal of enhancing next-day predictions of stock movement for the selected S&P 500 firms. It is further gathered that their study addresses key gaps in research by developing integrated models combining financial data and numbers with behavioral variables, and tailor the inputs to match the sector. Based on the large number of data integrated into the study, and a thorough research and analysis, the results showed that sentiment-based models outperformed those using only price, volume, fundamentals, or technical indicators in most sectors. It is further gathered that despite the network achieved effective exploratory powers and minimal mistakes, combining three data types did not achieve better results than using sentiment and technical indicators together, further suggesting that adding too many features may weaken model effectiveness. A key finding of the study states that in the financial sector, combining technical and fundamental data achieved strong predictive performance. Overall, the research states that data combinations are highly effective for diverse sectors, and it is further assessed by the researchers that sentiment data positively contributes to forecasting accuracy of short-term stock.

Even though existing research clearly outlines the effectiveness of ML and DL in stock projections, there are several critical gaps within current literature. Research conducted by [16] , [7], and [9] largely focus on technical indicators as well as short-term dynamics in pricing, and others such as [4] focus on a combination of macroeconomic and technical variables, these largely ignore sectoral fundamental data. Recent research papers such as the findings gathered through [1] effectively highlights the benefits of combining technical and fundamental analysis within the financial sector, and despite this, there is a critical gap in terms of scope and scale limitations. This thesis carefully addresses the gaps by cultivating an approach to entire S&P 500 financial sector, as well as systematically quantifies the additional predictive data of quarterly fundamental indicators, along with technical indicators.

4 Methodology

This thesis focuses on examining if integrating fundamental analysis has a positive impact on the accuracy of one-day ahead stock projections, and this experimental methodology follows multiple stages. The foremost stage is when fundamental and technical data is gathered; closing prices of the stocks are normalize as well as converted into overlapping input windows of twenty-day each, with each of the values being paired with next-day closing price being considered as the prediction target. During this, the nine financial ratios selected are aggregated into five quarters, with the key goal of presenting organisational fundamentals into structured format. To further enhance the validity, the dataset is then divided into training as well as tests through a division of 80/20 split. Further, three neural networks are evaluated: an LSTM model trained on rolling time windows, a CNN built to capture short-term temporal patterns, and a hybrid approach that merges CNN-based price sequence features with embedded fundamental data. Effectiveness is further tested by considering hybrid model with data augmentation, wherein synthetic variations enlarge the tech window and fundamental features are same. The models are further tested on distinguished dataset through four evaluation measures: MSE (Mean Squared Error) is implemented in an effort to understand the squared differentiations across predictions and actual values. MAE (Mean Absolute Error) showcases the basic error size. The coefficient of determination (R^2) indicates how much of the variance in the data is explained by the model. RAE (Relative Absolute Error) further compares models and baseline value performance. Together, these metrics give a well-rounded view of accuracy, clarity, and how the models perform relative to basic benchmarks.

4.1 Dataset Description

The dataset implemented in the thesis was developed through Yahoo Finance, with the help of the `yfinance` Python Library of the platform, and consists of two primary components, technical and fundamental data. The segment of organizations used for this study comprise of firms which are in Financial sector of the S&P 500 index, wherein broadly 73 total organizations were assumed with validated historical price information. It is further assessed that the daily adjusted OHLV (Open, High, Low, Close, Volume) data was gathered during 31 May 2023 to 31 May 2025. Since the primary purpose was model training, the researchers only used closing prices. For each ticker (firms), for the 2023-2025 period, the everyday closing time series was further developed into a collection of fixed-length input sequences. We developed a sliding window of twenty days to develop the sequences, and each sequence presents more or less a trading month of regular closing prices. The defined projection target is the closing price on the next day after the 20-day window, that is, day 21. Normalization was further applied across each ticker price history to make sure each company was comparable and the raw prices were further converted into standard values through `StandardScaler`. The transformation is further showcased as z-scores, which enables the models to focus on the relative fluctuations other than the complete price levels.

Table 1 Example structure for one ticker (WFC).

Date	Open	High	Low	Close	Volume
2023-06-01	39.51	40.02	39.12	39.87	19,823,400
2023-06-02	39.80	40.25	39.60	40.10	18,456,700
...	...	...	...	...	...
2023-06-28	40.80	41.25	39.60	40.10	18,456,700

A key example is drawn from the case of Wells Fargo & Company (WFC) wherein a single training sequence starts from 1 June 2023 and ends on 28 June 2023, which is twenty training days, with the target as 29 June 2023.

Fundamental dataset: Along with the price history, we collected quarterly fundamental data for each of the 73 firms to further integrate into the combined model. We used yfinance to collect up to five recent quarters of statements, including income statement and balance sheet, for every organization at the end of the data period. Nine key financial ratios were collected for each organization:

1. Price-to-Book (P/B)
2. Earnings per Share (EPS)
3. Price-to-Earnings (P/E)
4. Enterprise Value to EBITDA (EV/EBITDA)
5. PEG ratio
6. Beta
7. Return on Equity (ROE)
8. Return on Assets (ROA)
9. Debt-to-Equity

We condensed the fundamental data into short features rather than feeding raw data of quarterly values directly to the model, which would further create time-series aspect for the fundamentals of each organization. The following six statistics across available quarters, up to five quarters per ticker, were developed:

1. Mean: The average value of the ratio across the observed quarters.
2. Median: The median value over the quarters.
3. Standard Deviation: The volatility or variability of the ratio over the quarters.
4. First Quarter Value: The oldest quarter's value (chronologically first in the period).
5. Last Quarter Value: The most recent quarter's value (latest available).
6. Count: The number of quarters available.

Due to this, the result appears in six per ratio, and since there are nine ratios each, each of the firm is showcased as a 54-dimensional fundamental feature vector, thus 9 ratios × 6. We implemented this method to gather the fundamentals, and most importantly, the recent historical trend as well as fundamental stability, all while ensuring that input vector size remains manageable. To develop the inputs for our models, we combined the technical and fundamental data. Each 20-day price sequence, used as the technical input, is linked to a specific ticker and to enrich the model, we include its 54-dimensional fundamental vector as additional features.

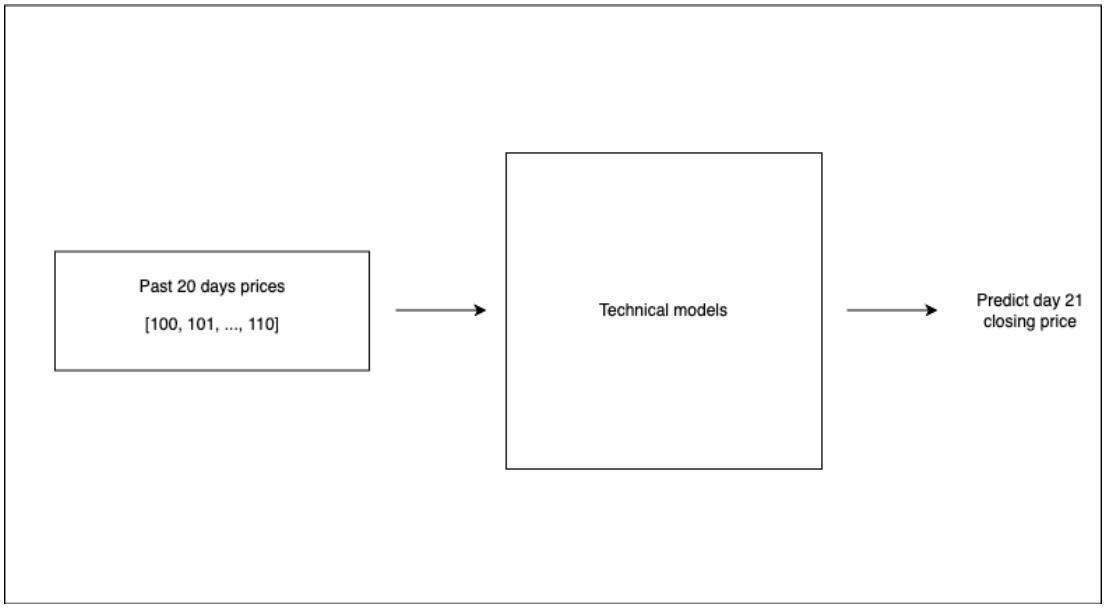

Figure 1: Architecture of the technical stock prediction model. The model uses only historical stock prices (past 20 days) as input to predict the closing price on day 21.

For models that use only technical data (such as the stand-alone LSTM and CNN models), the input used is 20-day normalized price sequence. The model learns exclusively through price trends as well as patterns, as illustrated in Figure 1.

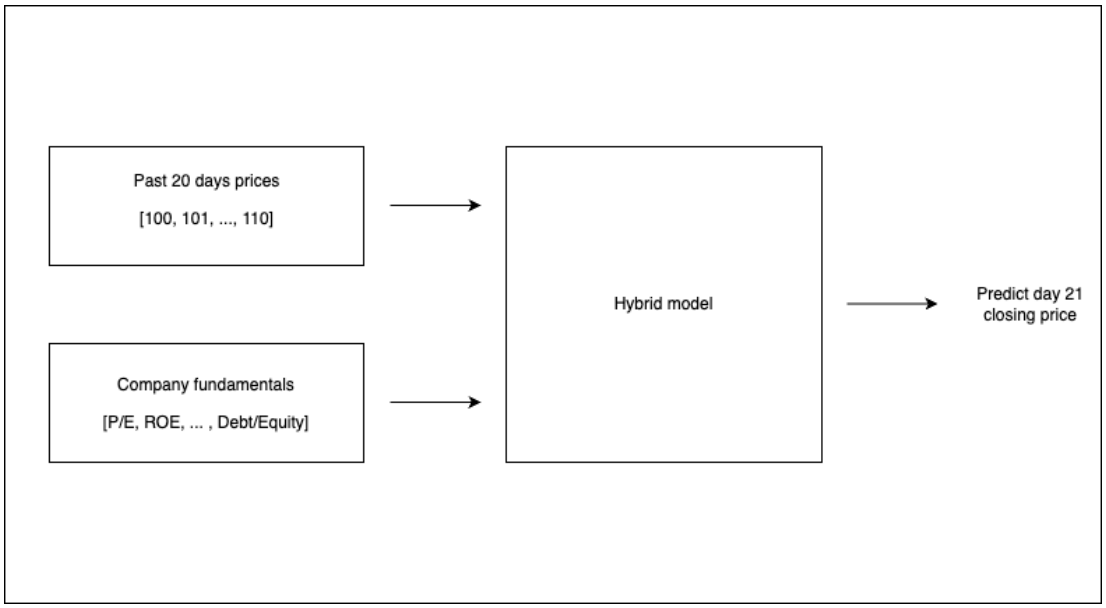

Figure 2: Architecture of the hybrid stock prediction model. The model takes as input both historical stock prices (past 20 days) and company fundamentals (e.g., P/E ratio, ROE, Debt-to-Equity). These heterogeneous features are combined within the hybrid model to predict the closing price on day 21.

For the combined model, the input is two-fold: (1) the 20-day price sequence (same as above), and (2) the 54-dimensional fundamental feature vector based on each ticker. From a practical sense, this suggests that a particular firm is further paired with same fundamental values, especially because fundamental do not provide variety every day, as presented in Figure 2.

Thus, it is gathered that the data showcases combined data of two years including price patterns and financial fundamentals, for 73 financial sector companies in the S&P 500, further converted into standardized sequences, which is effective for time-series stock projection model.

4.2 Data preparation

Each ticker's daily closing prices were organized into overlapping 20-day windows. For a ticker with N trading days, this method produced $(N - 20)$ sequences, since the first 20 days were used to form the input window and the 21st day served as the prediction target.We further combined all the sequences through tickers into single dataset, denoted `X_tech`, with dimensions $35{,}186 \times 20$. On the next day, the closing targets were set in target array `y_tech`, including 35,186 valuations. Each sequence also carried an identifier for its original ticker, which allowed later alignment with company-level financial data.

For model evaluation, the dataset was further divided into training and test-sets, utilizing scikit-learn's `train_test_split` function. It is further assessed that a fixed seed, 42, is implemented to ensure that the split can be further reproduced. Considering this, close to 28,148 sequences were implemented to the training set and 7,038 sequences were set to be tested. Considering this, the split was then used after window generation, further introducing a random segmentation of sequences throughout the tickers, rather than dividing by organization.

Note: We did not use a strict time-based split, thus, some sequences from same company appear across both train and test.

Further, we implemented three core structures: an LSTM-based model, a one-dimensional CNN, and a hybrid model.

The LSTM model consists of two stacked LSTM layers. The first layer has 64 units and returns full sequences, followed by a dropout layer for regularization.

The one-dimensional CNN model uses stacked `Conv1D` layers, followed by flattening and dense layers. Like the LSTM, it processes input tensors of (20×1) to capture local temporal patterns.

The hybrid design, combines technical and fundamental data. One branch processes the 20-day closing-price sequence through a CNN, while another branch processes a 9-dimensional vector of financial ratios.

All models were implemented in Python and trained using the Adam optimizer with its default learning rate. The LSTM and CNN models were trained for 30 epochs, with a batch size of 32, and without early stopping. The hybrid CNN models were trained for up to 50 epochs. The standard version used a batch size of 32, while the augmented version used 64.

To improve generalization, each 20-day window of normalized prices was augmented using three simple methods: (i) random Scaling: Multiplying prices by a factor (for example between 0.98 and 1.02); (ii) noise Injection: Adding Gaussian noise to the normalized prices; (iii) temporal Shifting: Shifting the sequence forward or backward by a few days, clipped to valid ranges. Each of the augmented sequence was then attached to the same next-day closing price, which remained in the normal form, even when the temporal evolution developed the calendar alignment. This process expanded the training set by introducing variations in amplitude and alignment without requiring additional market data. Fundamental features were kept unchanged and reused for all augmented samples. For the evaluation, one-day-ahead predictions were developed on the global test set, and model performance was further evaluated using four metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), Coefficient of Determination (R2), and Relative Absolute Error (RAE). Even though the metrics were computed on a global scale, if any model produced NaN predictions, they were replaced by the global mean of valid forecasts.

The data retrieval process is performed through the Yahoo Finance API using the `yfinance` library and processing is managed with `pandas` and `numpy`, normalization and evaluation with `scikit learn`, and model training with `TensorFlow 2.x (keras)`. It is further assessed that the dependencies are effectively managed through virtual environment in Python 3.9. A short snapshot of the collected data is available online[1]; along with this, all the sources are online , public, and do not include any form of personal or sensitive information.

[1]Dataset `https://github.com/armanehmad/Datasets`

5 Results

Based on Table 2, the LSTM model achieved an MSE 0.015900 of and MAE of 0.081380 on technical data, with a R^2 of 0.983561 and a RAE of 0.098370, further indicating that it details over 98% variance in one-day-ahead on normalized closing price target. On the other hand, the CNN trained on the same tech data achieves a relatively higher MSE 0.016379 and MAE 0.083791, with R^2 of 0.983064 and RAE of 0.101285, which indicates that LSTM on data largely outperformed the CNN. As we added the baseline fundamental ratios to the technical ratios, without the augmentation, the hybrid model (technical + fundamental), with the error being reduced further: MSE = 0.015363, MAE = 0.079781, R^2 = 0.984116, and RAE = 0.096438. Thus, it is further gathered that upending the fundamental values to the raw values of closing window helped achieve the best performance, on the unaugmented data. However, as augmentation was applied solely to the technical inputs (that is, the 20-day normalized closing-price windows, modified through random time-shifting, noise addition, or scaling) and the augmented hybrid model was trained on these augmented sequences along with their corresponding fundamental vectors, performance showed a slight decline: MSE = 0.019909, MAE = 0.091714, R^2 = 0.979415, and RAE = 0.110862. In other words, augmentation of the normalized closing prices across hybrid setting is slightly limited, especially one-day-ahead projections, especially compared to the unaugmented hybrid model. Overall, this thesis indicates that the combined CNN of technical and fundamental inputs achieved the lowest test set error. It is further gathered that the pure LSTM was right after that, followed by largely technical CNN. Lastly, it was gathered that the combined CNN with the augmented tech inputs was the least effective in performance. The complete results of the test are illustrated below in Table 2.

Table 2 Summarizes model performance across four metrics. In an ideal scenario, both Mean Squared Error (MSE) and Mean Absolute Error (MAE) equal zero, indicating perfect agreement between predictions and targets, while the coefficient of determination (R^2) reaches one, reflecting complete explanation of the data's variance, and the Relative Absolute Error (RAE) equals zero, indicating zero total error relative to a constant-mean predictor. In the heat-map, green denotes strong performance, yellow indicates moderate performance, and red highlights the weakest results.

Model	MSE	MAE	R^2	RAE
Technical Models				
LSTM	0.015900	0.081380	0.983561	0.098370
CNN	0.016379	0.083791	0.983064	0.101285
Hybrid Model (Technical + Fundamental)				
Combined CNN	0.015363	0.079781	0.984116	0.096438
Hybrid Model (Augmented Technical + Fundamental)				
Combined CNN	0.019909	0.091714	0.979415	0.110862

6 Discussion

The thesis aims to examine if blending the technical and fundamental data positively contributes to stock projections of the next day within the S&P 500 financial-sector firms. In addition, we implemented whether data augmentation of technical windows could limit overfitting and improve the overall performance of the model. Specifically, we tested the hypothesis that hybrid models integrating quarterly financial ratios with 20-day price windows would achieve higher predictive accuracy than models relying on technical data alone. The technical-only models used the last 20-day of closing values as the input and the hybrid model used the same values of price data, however, also included information about the organizations. Table 2 represents the understanding that the hybrid model achieved significantly accurate prediction and low rates of error in comparison to the technical focused models. The finding suggests that adding fundamental values achieved useful information that price trends alone could not provide. From a practical lens, the hybrid model was able to understand short-term price changes as well as deep financial strengths of the organization, ensuring its reliability; and even though the positive impact was not significant, it was consistent. The hybrid model in particular had the lowest error and highest R^2 score, proving that even quarterly financial data can develop short-term predictions when mixed with technical data. Overall, this means that the stock market does not instantly adjust to fundamental information. The hybrid model achieved better performance than technical only model due to multiple reasons, such as the fundamental features add useful context. For example, an organization's significant financial strength makes it less prone to sudden downfalls, which is something that models using only past prices cannot easily detect. By including measures of financial strength and real value, the hybrid model can separate random short-term noise from real changes caused by company performance or news. On other hand, the technical-only model rely on the past-pattern. It further gathers that they can capture momentum, however may easily be misled by sudden rises. The hybrid model balances this by combining short-term trends with long-term fundamentals and helps prevent overreaction to sudden changes and highlights signals where price movements match improving fundamentals. This illustrates precisely why the hybrid model had significantly lower rates of error in comparison to other models in Table 2. In comparison to other models, the LSTM achieved the most effective result, considering that it has the lowest errors and highest explained variance among technical models and augmented hybrid model. Moreover, the CNN model was only slightly worse, however the difference was not very significant. The slight difference is a result of the fact that LSTMs are better at learning sequences, so they learned patterns such as momentum and mean reversion in the 20-day price windows. However, it is gathered that even though most effective model, LSTM, was still less accurate than the hybrid model. This shows that important drivers of prices were missing when only past prices were used. Key strengths of the hybrid model is the combination of quick responsiveness of technical indicators with the stability of fundamentals. A key finding of the study is directly attached to using augmentation as the training period progressed. Based on the detailed section of Methodology, Section 4.2, the 20-day price windows were modified with small changes such as scaling, adding noise, and shifting in time. Due to this, extra synthetic sequences were created based on the original data. Despite this, details in Table 2 showcases that using augmentation along with the hybrid model did not contribute

positively to the results. In fact, the augmented hybrid performed worse than all other models, showing higher errors and lower explained variance, and a primary reason for this is that the augmentation was applied only to the technical inputs.

6.1 Preprocessing Pipeline Design

The data preprocessing pipeline is built taking into consideration both error-free and efficiency factors. We transform the raw S&P 500 series into labeled training data based on a sliding-window technique [1]: the input is a fixed-length history of past returns/features, and the label is the next future return. In practice, we generate $N - W$ windows with a Python loop, which cost $O(NW)$. Our dataset sizes are on the scale where this is acceptable, though more sophisticated techniques (e.g. NumPy's `sliding_window_view`) can potentially reduce overheads, as window creating can be vectorized[2]. These techniques do not involve Python loops and produce a view without copying data, so if the resulting windows are consumed directly by vectorized operations, the overall cost can become $O(N)$ rather than $O(NW)$. After windowing we apply feature scaling: each feature within each window will be standard-normalized (z-score: subtract mean, divide by std), on the training set. Normalization of data is already known to enhance training stability and performance in several areas[3]. These normalization operations are $O(N)$ (adding and multiplying each element), which is insignificant compared to model training. The pipeline ensures that normalization parameters (means/variances) are computed only once and reused. In general, our preprocessing is more modular by the use of dedicated functions, scales and cleanings on NumPy and Pandas data. Whereas the construction of windows is implemented using Python loops ($O(NW)$), the normalization and the feature preparation can make use of fast bulk array operations.

6.2 Parallelization and Trade-offs

Our models highlight complementary computational trade-offs between CNNs and LSTMs. In a CNN layer, the same convolution kernel is applied independently at each position of the input sequence, enabling simultaneous (parallel) computation on GPUs. In contrast, an LSTM's gating at time t depends on the state from $t - 1$, enforcing sequential evaluation along the time axis. Community reports note that "CNNs are faster by design, since the computations in CNNs can happen in parallel while RNNs need to be processed sequentially"[4]. Empirical benchmarks similarly find CNNs often train several times faster than comparable LSTM/GRU models (e.g., $\sim 5\times$)[5].

From a rough cost perspective, a 1D CNN over sequence length N with kernel size K and C filters performs $O(N \cdot K \cdot C_{\text{in}} \cdot C_{\text{out}})$ operations, while an LSTM over length N with hidden size H is about $O(N \cdot H^2)$ (plus input projections). Thus, for large H or long N, LSTMs can

[1] Machine Learning Mastery, How to Transform Time Series Data for Supervised Learning (2020). https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/

[2] NumPy, numpy.lib.stride_tricks.sliding_window_view (2025). https://numpy.org/devdocs/reference/generated/numpy.lib.stride_tricks.sliding_window_view.html

[3] Ioffe & Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, arXiv:1502.03167 (2015). https://arxiv.org/abs/1502.03167

[4] StackExchange, Is convolutional neural network (CNN) faster than recurrent neural network (RNN)? (2017). https://stats.stackexchange.com/questions/262220-is-convolutional-neural-network-cnn-faster-than-recurrent-neural-network-rnn

[5] StackExchange, Is convolutional neural network (CNN) faster than recurrent neural network (RNN)? (2017).

be substantially more expensive. In short, CNN layers exploit data-level parallelism, whereas LSTMs are inherently step-by-step [6].

6.3 Impact of Data Augmentation

Lastly, we explain why our data augmentation degraded performance. In our implementation, we applied three types of synthetic perturbations to the technical windows: (1) random cyclic shifts of the sequence (`time_shift`), (2) additive Gaussian noise (`add_noise`), and (3) random rescaling (`scale_series`). Such augmentations were supposed to lead to expansion of the training data, but they probably led to a distorting effect on the actual distribution of financial returns. From a computer-science perspective, inappropriate augmentation might mismatch the inputs with the actual data distribution and cause confusion in the model. Data augmentation must not alter the underlying structure of the input; otherwise, this introduces a distortion of vital features or irrelevant noise[6]. In the example of financial time series, trivial fluctuations can easily overpower tiny patterns of returns or break the assumptions of autocorrelation and continuity of the trend. Our augmentations, especially random shifting and added noise, formed unrealistic progressions whereby the model treated them as no less valid than the real data. This difference resulted in the model learning spurious features that did not generalize and incurred more error in the augmented model in comparison to the baseline. This outcome also resonates with broader themes in the literature. The reviewed studies ([4], [1]) do not specifically discuss augmentation but point out how improper model architecture or overcomplicated input can impair generalization, which similarly explains our results that naive perturbations resulted in degraded performance. More generally, external guidelines[7] warn that augmentations in time-series forecasting need to maintain time-structure, and overdoing noise or mixing can introduce distortions instead of a beneficial signal.

6.4 Limitations

It is further gathered that despite promising results being gathered, there are multiple limitations to the study. First, it is assessed that the forecasting horizon was minimal to one-day ahead, and even though it delivers detection of current predictive capabilities, it is short term in its perspective. Fundamental indicators usually capture long-term value and may not be well suited for daily price changes[8]. Second, the technical data was significantly limited to normal closing prices during the twenty-day period. Various common indicators, such as moving averages, RSI, volume trends, or volatility measures, were also not included, similar to other OHLCV features, beyond closing prices. This may have removed potentially predictive signals, even though it was an effective and minimal approach. Third, all models were trained globally across the entire set of financial-sector stocks, without developing stock-specific models or separating between firm types (banks versus insurance firms). As a result, the reported metrics may be sensitive to the particular split used. Lastly, even though early stopping and data augmentation were used to reduce overfitting, the dataset was relatively small in comparison to the complex nature of the neural networks. The fact that augmentation reduced performance suggests that the models were already capturing most of the available

[6] Milvus, Can Data Augmentation Degrade Model Performance? (2024). `https://milvus.io/ai-quick-reference/can-data-augmentation-degrade-model-performance`

[7] Milvus, Time Series Forecasting Best Practices (2025). `https://milvus.io`

[8] AdroFX, *Exploring the Limitations of Fundamental Analysis: A Comprehensive Perspective* (2025) `https://adrofx.com/blog/exploring-limitations-fundamental-analysis-comprehensive-perspective`

signal, and the added distortions only reduced accuracy. Even though the study provides useful insights, the results should be further used with caution, considering the limitations of the study.

7 Conclusion

This chapter helps draw the thesis conclusion by directly addressing the research question that we adopted for this study, to what extent does including quarterly fundamental features improve prediction accuracy. To answer this, we tested whether the accuracy gains found in an earlier three-company study could be reproduced at the sector level, and then measured the added value of fundamentals when applied to all financial firms in the S&P 500. Results in Table 2 illustrate adding quarterly fundamental indicators along with technical data, did improve one-day-ahead forecasts and the accuracy gains from [1] could be reproduced. The hybrid CNN achieved the best overall performance, however, compared to the technical-only models, the results were only slightly better. By contrast, data augmentation reduced accuracy rather than improving it, suggesting it may not be suitable when combining technical and fundamental inputs. Thus, the augmented hybrid CNN performed worse across all metrics, showing that augmentation did not improve generalization. Key results of the performance evaluation of the models are as follows: the hybrid model consistently outperformed all models, it reduced prediction error by about 2–3% compared to technical models; among the technical-only models, the LSTM slightly outperformed the CNN; data augmentation consistently failed to improve results. For the hybrid model, it actually worsened performance, increasing error rates by around 30% compared to the non-augmented version Table 2. It is further concluded that future research could build on the findings of this study and move in multiple directions. A key direction may be longer forecasting directions rather than the adopted next-day model, and may focus on one-week or one month, which may better capture the contribution of fundamentals. On the other hand, expanding the technical indicators used may also improve how short-term signals interact with quarterly fundamentals. Future researchers could develop stock specific models to account for differences between organizations, rather than relying only on global models. Finally, methodological improvements such as cross-validation, per-stock training, or transformer-based architectures could help reduce overfitting and further enhance prediction accuracy.

References

[1] Mónica Andrea Arauco Ballesteros and Elio Agustín Martínez Miranda. Stock market forecasting using a neural network through fundamental indicators, technical indicators and market sentiment analysis. *Computational Economics*, 1:1–10, 2024.

[2] Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLOS ONE*, 12:e0180944, 2017.

[3] Wuzhida Bao, Yuting Cao, Yin Yang, Hangjun Che, Junjian Huang, and Shiping Wen. Data-driven stock forecasting models based on neural networks: A review. *Information Fusion*, 99:1–24, 2023.

[4] Hum Nath Bhandari, Binod Rimal, Nawa Raj Pokhrel, Ramchandra Rimal, Keshab R. Dahal, and Rajendra K.C. Khatri. Predicting stock market index using LSTM. *Machine Learning with Applications*, 8:100241, 2022.

[5] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2:1–8, 2011.

[6] Jiyang Chen, Xiaohong Ma, Shikuan Li, Sile Ma, Zhizheng Zhang, and Xiaojing Ma. A hybrid parallel computing architecture based on CNN and transformer for music genre classification. *Electronics*, 13:3313, 2024.

[7] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270:654–669, 2018.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[9] Ehsan Hoseinzade and Saeed Haratizadeh. CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, 129:273–285, 2019.

[10] David M. Q. Nelson, Adriano César Machado Pereira, and Renato A. de Oliveira. Stock market's price movement prediction with LSTM neural networks. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 1419–1426, 2017.

[11] Isaac Kwame Nti, Adebayo Felix Adekoya, and Benjamin Abayie Weyori. A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, 53:3007–3057, 2020.

[12] Sreelekshmy Selvin, Ravi Vinayakumar, E. A. Gopalakrishnan, Vijay Krishna Menon, and K. P. Soman. Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1643–1647, 2017.

[13] Omer Berat Sezer, Muhammed Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, 90:106181, 2020.

[14] Dev Shah, Haruna Isah, and Farhana Zulkernine. Stock market analysis: A review and taxonomy of prediction techniques. *Algorithms*, 12:1–33, 2019.

[15] Manuel Ramon Vargas, Carlos Eduardo Menezes dos Anjos, Gustavo L. G. Bichara, and Alexandre Gonçalves Evsukoff. Deep learning for stock market prediction using technical indicators and financial news articles. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2018.

[16] Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, and Arun Kumar. Stock closing price prediction using machine learning techniques. *Procedia Computer Science*, 167:599–606, 2020.

[17] Jimmy Ming-Tai Wu, Zhongcui Li, Norbert Herencsar, Bay Vo, and Jerry Chun-Wei Lin. A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. *Financial Innovation*, 7:1–20, 2021.