MSIN0100: Business Research Project 2022-23

Integrating Deep Learning and Sentiment Analysis in Quantitative Trading: Designing and Assessing ASTRA, an Algorithmic Trading System.

MSc Management

Module Code: MSIN0100

Module Title: Business Research

Date: 24/08/2023

Word Count: 6,992

Abstract

The digitalisation of financial markets has ushered in advanced trading strategies. This research examines the potential of deep learning and sentiment analysis in algorithmic trading. Recognising the value of high-quality datasets, the study evaluates the predictive modelling capabilities across different stocks, revealing disparities in predictability. The analysis indicates the prevalence of neutral sentiments in market dynamics and the complexities of trading amidst market inefficiencies and investor irrationalities. Building on a comprehensive review of previous research and applied methodologies, the study recommends diversifying data sources, refining predictive models, enhancing sentiment analysis and adapting dynamic portfolio management strategies. As the landscape of algorithmic trading evolves, this research underscores continuous refinement for greater sophistication and accuracy in digital trading. The implementation of ASTRA is publicly available at https://github.com/daps05ayoade/ASTRA.

Acknowledgements

I want to extend my gratitude to Alpha Vantage for its comprehensive financial data platform and to Alpaca Trading for its exceptional trading API. Both have been pivotal in advancing this project. Special thanks to Magda David Hercheui, the programme director for the MSc Management, for her guidance and mentorship throughout this journey.

Table of Contents

List of Figures 6

List of Tables 7

1 Introduction 8

1.1 Background 8

1.2 Research Objective & Structure 8

2 Literature Review 9

2.1 Historical Evolution and Impact of Algorithmic Trading 9

2.1.1 Quantitative Trading Strategies 9

2.1.2 Portfolio and Risk Management 10

2.2 Deep Learning in Trading 10

2.3 Sentiment Analysis in Trading 11

2.4 Existing Systems 12

3 Theoretical Framework 14

3.1 Deep Learning Framework 14

3.1.1 Convolutional Neural Network 15

3.1.2 Other Deep Learning Models 17

3.2 Sentiment Framework 17

3.2.1 Behavioural Finance and Investor Sentiment 17

3.2.2 Sentiment Analysis with Natural Language Processing 18

4 Research Design & Methodology 20

4.1 Machine Learning Logic 20

4.1.1 Input Features 20

4.1.2 Data Processing 20

4.1.3 CNN Architecture 21

4.2 Model Evaluation 22

4.3 Trading Strategy 23

4.3.1 Signal Generation 23

4.3.2 Risk Management 25

4.4 Ethical Considerations 26

5 Finding & Analysis 27

5.1 Model Performance 27

5.2 Overall Sentiment Distribution 29
5.3 ASTRA's Performance 30
6 Conclusion & Recommendations 34
6.1 Summary of Key Findings 34
6.2 Limitation and Further Research 34
7 References 36
8 Appendix 38

List of Figures

FIGURE 3.1: BASIC STRUCTURE OF AN ARTIFICIAL NEURON NETWORK..14
FIGURE 3.2: A COMPARISON OF DIFFERENT ACTIVATION FUNCTIONS. 15
FIGURE 3.3: ILLUSTRATION OF THE HIDDEN LAYERS OF A CONVOLUTIONAL NEURAL NETWORK. 16
FIGURE 3.4: THE PROSPECT THEORY CURVE: A GLIMPSE INTO THE HUMAN PSYCHE'S DISTINCT RESPONSES TO GAINS AND LOSSES..18
FIGURE 3.5: AN OVERVIEW OF THE NATURAL LANGUAGE PROCESSING PROCESS...........................19
FIGURE 4.1: STOCK PRICE SPLIT AMONG TRAINING, VALIDATION, AND TEST SETS OF AAPL & INTC. 21
FIGURE 5.1: STOCK PRICE PREDICTION OF CNN, RNN, LSTM AND GRU MODELS.........................28
FIGURE 5.2: BREAKDOWN OF STOCK SENTIMENTS: A GLIMPSE INTO INVESTOR PERSPECTIVES. 29
FIGURE 5.3: UNPACKING PORTFOLIO HEALTH: A DIVE INTO DAILY VARIATIONS AND CUMULATIVE RETURNS RELATIVE TO SPY. 31
FIGURE 5.4: BROADENING HORIZONS: HOW PORTFOLIO PERFORMANCE EVOLVES WITH INCREASED STOCK DIVERSIFICATION. 32

List of Tables

TABLE 5.1: COMPARISON OF MSE AND MAE FOR DIFFERENT MODELS ON AAPL AND INTC STOCK PRICES. 27
TABLE 5.2: DIVERSIFICATION INSIGHTS: HOW INCREASING STOCK NUMBERS INFLUENCE PORTFOLIO PERFORMANCE METRICS. 32

1 Introduction

The financial landscape is undergoing a seismic shift spurred by technological advancements redefining the boundaries of possibility. This research delves into a pioneering intersection of two advancements, deep learning and sentiment analysis and their integrative potential in quantitative trading.

1.1 Background

Over the past decade, there has been a surge in interest in Artificial Intelligence (AI), primarily driven by significant advancements in deep learning techniques. Deep learning has demonstrated capabilities beyond traditional computational methods. Unlike conventional algorithms that follow strictly programmed instructions, deep learning systems learn from experience. Deep Neural Network (DNN), a subset of deep learning, draws inspiration from the human brain's architecture, offering unparalleled capabilities in processing vast unstructured data. Parallelly, sentiment analysis, rooted in Natural Language Processing (NLP), provides a mechanism to harness the emotional undertones embedded within data. Integrating these technologies holds profound implications in finance, where a blend of data and sentiment often influences decisions.

For contemporary businesses, especially in the finance sector, staying ahead of the curve is not just a competitive advantage but a necessity. The pace at which markets move, influenced by global events and news, necessitates tools to analyse, predict and respond in near-real-time. The relevance of integrating deep learning with sentiment analysis becomes evident, offering businesses a lens to view markets with enhanced clarity and foresight.

1.2 Research Objective & Structure

The primary objective is to design and assess ASTRA (Advanced Sentiment and Technical Analysis Real-time Algorithm), an innovative algorithmic trading system that combines the strengths of deep learning and sentiment analysis. Additionally, the research aims to elucidate the broader implications of this integration for the future landscape of algorithmic trading.

The ensuing report is structured to delve into the individual realms of deep learning and sentimental analysis, detailing their evolution and relevance in modern trading. This is followed by exploring ASTRA, its design principles, and its assessment in real-world trading scenarios. The report will discuss the implications of the findings, outlining the potential challenges and future trajectories of integrating these technologies in finance.

2 Literature Review

Much of the research in algorithmic trading emanates from private financial entities, often making advanced findings inaccessible (Li, 2017). This creates a knowledge gap regarding the latest advancements in algorithmic trading. The absence of a universal framework for evaluating trading strategies can lead to biases in self-designed metrics (Théate and Ernst, 2021). Furthermore, high-quality datasets are essential for successful computational models (Whang et al., 2023). This review provides a balanced assessment of the literature on algorithmic trading, mainly focusing on deep learning and sentiment analysis applications, and identifies areas for further exploration.

2.1 Historical Evolution and Impact of Algorithmic Trading

Algorithmic trading blends advanced mathematical models and computer systems to optimise decisions based on market patterns (Monks and Lajoux, 2011). It can be traced back to the early 1970s with the introduction of the "Designated Order Turnaround" system (DOT), which increased efficiency in security order transmissions, circumventing the need for a broker (Grossman, 1988). The emergence of fully electronic markets heralded the advent of program trading. Defined by the NYSE (2012) as a portfolio trading strategy involving purchasing or selling 15 or more stocks, program trades were pre-programmed to automatically enter or exit trades based on various factors. This evolution accelerated into high-frequency trading (HFT), implementing varied trading strategies at millisecond intervals (McGowan, 2010).

2.1.1 Quantitative Trading Strategies

At the core of algorithmic trading reside quantitative trading strategies. They use rigorous mathematical computations and sophisticated statistical techniques to forecast price movements, relying on historical and real-time market data. Fama and French (1992) highlight the pivotal roles of size and book-to-market equity in explaining stock returns from 1963 to 1990. Their findings challenge the conventional Sharpe-Linter-Black model, suggesting a nuanced relationship between stock returns and market beta.

Further enriching the quantitative landscape, Gottwald (2012) exploration into the Price-to-Earnings (P/E) ratio underscores its significance in stock valuation, introducing the PERS measure, a refined representation of the traditional P/E ratio, offering enhanced accuracy in stock valuation. This evolution of fundamental metrics like the P/E ratio is crucial for algorithmic trading, as it bridges traditional financial metrics with modern computational techniques. Such insights underscore the importance of leveraging precise quantitative metrics to inform trading decisions.

2.1.2 Portfolio and Risk Management

A study by Sukrianingrum and Manda (2020) explored the effects of systematic and unsystematic risks on the expected return on a portfolio of companies listed in the LQ45 index from 2015 to 2019. Systematic risk is inherent to the entire market or market segment; unsystematic risk pertains to a particular company or industry (Maginn et al., 2007). Their research revealed that systematic risk negatively correlated with returns, while unsystematic risk positively correlated with returns. Interestingly, when considering both risks simultaneously, the study found that they collectively influenced the expected return. This emphasises the need for a holistic portfolio management approach, where market-wide and company-specific factors are considered.

Building on this understanding of risk, Valeyre et al. (2017) explored the beta-neutral model, highlighting its potential to address biases in market-neutral strategies, especially during market stress periods like the financial crisis, showing pronounced biases in hedging strategies. The beta-neutral model, which accounts for the leverage effects of systematic and unsystematic risks, significantly reduces these biases, offering a more robust approach to risk management in algorithmic trading. These studies highlight the importance of understanding and managing risk for successful portfolio management and algorithmic trading strategies.

2.2 Deep Learning in Trading

AI, specifically machine learning, has reshaped numerous industries, enabling computational models to learn autonomously and identify complex datasets (Alpaydin, 2020). One subset of machine learning, deep learning, employs artificial neural networks (ANNs) to discern complex, non-linear relationships in data, proving them highly effective in predicting the inherently intricate patterns of financial markets. Deep learning, therefore, provides immense potential in developing algorithmic trading systems by aiding in analysing multidimensional financial markets to facilitate better-informed trading decisions.

Various deep learning architectures exist, including Gated Recurrent Units (GRUs), Long Short-Term Memory (LSTM) networks and Recurrent Neural Networks (RNNs), each with unique strengths and challenges. Chen et al. (2017) utilised LSTM to predict stock prices, and while their findings demonstrated superior predictive accuracy, they also illuminated the requirement for high-quality data, a challenge in a field where data can be noisy or incomplete.

Doering et al. (2017), Gudelek et al. (2017) suggest Convolutional Neural Networks (CNNs) as superior in predicting stock market trends, reinforcing their potential in avenues for future research. However, they also cautioned against the threat of overfitting, where models provide overly optimistic results during training but poor performance when applied to new data. This highlights a critical literature gap and a potential exploration area.

A comparative study by Selvin et al. (2017) investigated the prediction capabilities of three deep learning models (RNN, LSTM and CNNs) on stock prices. The result highlighted the superior accuracy of CNN over its counterparts, notably over traditional linear model forecasting, ARMIA. CNN's outstanding performance is due to its independence from previous information for prediction (Selvin et al., 2017). Unlike RNNs and LSTMs, which rely on information from previous lags, CNNs exclusively focus on the current data window. This characteristic enables CNNs to capture dynamic changes and patterns in the current window more effectively, a critical aspect considering the highly dynamic nature of the stock market.

However, challenging this CNN dominion, Bao et al. (2017) argue for the performance of LSTM and stacked autoencoders (SAEs) in financial prediction. The researchers assert that this combined approach outperforms other models in predictive accuracy and profitability, regardless of the examined financial market. Ballings et al. (2015) indicated that traditional machine learning models like Random Forest could outperform deep learning models in predicting stock price direction. The authors suggest these findings are due to the Random Forest algorithm's ability to handle noisy financial data and make robust predictions.

This perspective contrasts with the previously discussed studies, leaving room for further research regarding the optimal deep learning architectures for financial forecasting. This underexplored area forms part of this research's focus, emphasising the study's relevance and timeliness in the current landscape of algorithmic trading.

The literature collectively highlights the vast potential of deep learning in algorithmic trading while acknowledging the barriers to its successful implementation. Notably, the success of these models hinges on access to high-quality data and the threat of overfitting. Furthermore, the workings of deep learning models – often termed for their 'black box' nature – necessitate further research. The tension between retaining knowledge and adapting to new data – the stability-plasticity dilemma – presents a significant challenge, particularly in a dynamic environment of financial markets (Mermillod et al., 2013). Therefore, these findings warrant further exploration and verification, forming the basis for this study to enhance strategies.

2.3 Sentiment Analysis in Trading

Sentimental analysis, a prominent technique in NLP, systematically identifies, extracts and quantifies data to discern its emotional tone, categorising it as positive, negative or neutral, offering insights into the impact of unstructured market news in investor behaviour. Given the multifaceted nature of sentiment data from diverse sources, applying sentiment analysis in trading strategies necessitates a nuanced approach.

Zhang and Skiena (2010) reveal the role of sentiment analysis in modulating stock trading volumes and returns, critical insight for algorithmic trading. They formulated a market-neural trading strategy hinged on media sentiment data. Their findings showed consistently favourable returns with lower volatility. However, this approach relies heavily on the quality and timeliness of news sources. Furthermore, the difference in influence between blog sentiment and news-source sentiment illustrates the multifaceted application of sentiment analysis in trading strategies, drawing attention to the need for a nuanced approach to assimilating and interpreting sentiment data from varied sources.

Yadav et al. (2019) explored sentiment analysis classification in algorithmic trading, highlighting its capacity for trend identification and news classification. However, their study uncovers several limitations, such as mislabelling unrelated news or overlooking critical information within news articles beyond headlines and choosing a time window for news-market trend alignment that might accurately represent news impact on the market. Similarly, Hajek and Barushka (2018) combined sentiment analysis with topic detection for financial news content; while their approach showed improved results over the traditional method, it’s not exempt from data quality and timeliness issues, especially given its reliance on news content.

2.4 Existing Systems

Algorithmic trading has seen innovation like the AZFin Text system by Schumaker and Chen (2009), which employs three models, M1, M2, and M3, for stock prediction and textual representation. Particularly, M2, integrating real-time data with textual analysis, excelled in accuracy, directional, and simulation trading, highlighting the relevance of textual analysis stock forecasts. Similarly, Mittermayer (2004) NewsCATS combines machine learning and text analysis using SVM Light Classifier. Despite success in generating profits, the system showed relatively low precision for categories “Good News” and “Bad News”; it demonstrates promising potential for improvements.

These systems underscore the benefits of merging deep learning and sentiment analysis in trading. However, challenges like news categorisation accuracy persist. This research seeks to build these foundations, aiming for refined models and enhanced machine learning techniques.

To conclude this literature review, AI, particularly deep learning, has transformed algorithmic trading, and the optimal deep learning architecture for financial forecasting remains debated, signalling an area for further study. Sentiment analysis, although promising, challenges with data quality and timeliness persist. Existing systems validate the potential of integrating machine learning with text analysis; however, precision in news categorisation remains a concern. The literature advocates for blending deep learning and sentiment analysis in algorithmic trading but highlights significant research gaps that need addressing.

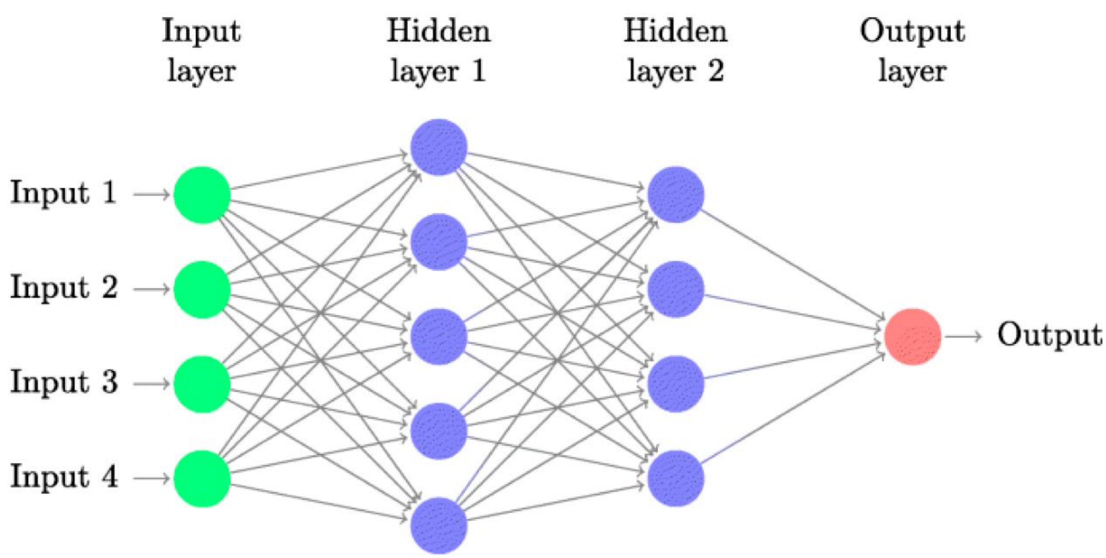
3 Theoretical Framework

The selected theoretical framework is grounded on cognitive science and machine learning, which facilitates the design, implementation, and evaluation of the ASTRA system. Two theoretical perspectives underpin the system: deep learning algorithms and sentiment analysis.

3.1 Deep Learning Framework

Deep learning employs algorithms to construct multi-layered ANNs. These layers extract higher-level features from raw inputs, enabling the model to learn complex patterns within vast datasets. Deep learning algorithms autonomously learn feature representations, eliminating manual feature extraction (Kiranyaz et al., 2019).

Figure 3.1: *Basic Structure of an Artificial Neuron Network.*



Source: *(Sahu et al., 2023)*

Input Layer: An ANN's input layer corresponds to a biological organism's sensory system, where information about the world is received. This is the layer where the neural network receives its input in the form of data features. Each neuron in the input layer represents a unique feature present in the data, where it accepts an n-dimensional vector with varying dimensions to describe data features (Dongare et al., 2012).

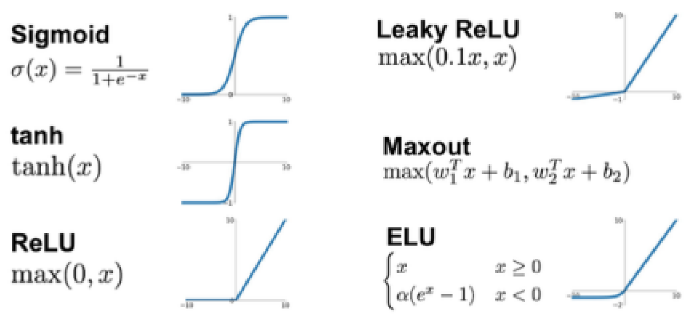
Hidden Layer: An ANN's hidden layer(s) mimic the brain's processing of sensory data. Inputs are processed using systems of weighted connections. The network parameters include hidden layer and node counts (Lecun et al., 2015). The computations start with a linear combination of preceding layer output, weighted by coefficients and a bias term. Each hidden layer neuron is represented as:

$$z = \sum (w\_i * x\_i) + b$$

Where *w_i* are the weights, *x_i* are the inputs, *b* is the bias term, and the sum is all neuron inputs. A non-linear function activates this linear combination (*z*) to learn complex patterns. For example, the Rectified Linear Unit (ReLU) is a commonly used activation function.

Output Layer: An ANN's output layer corresponds to the decision or action based on sensory input and processing. It complies and presents the final output values. The output layer and hidden layer perform similar computations. However, different activation functions are utilised depending on the network's task. For instance:

Figure 3.2: *A Comparison of Different Activation Functions.*



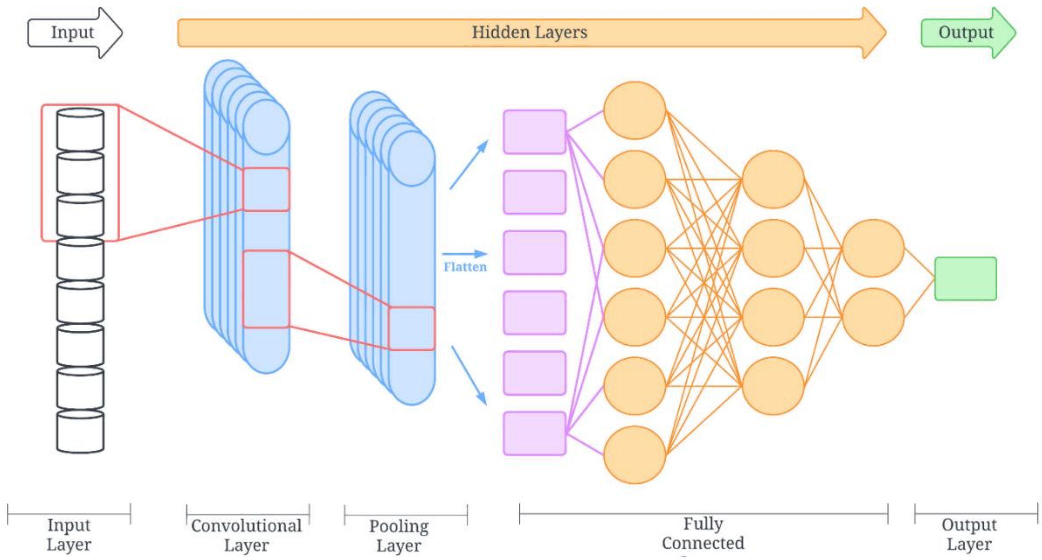
Source: *(Baili, 2020)*

The sigmoid function squashes input between 0 and 1 to provide a probabilistic output, often utilised in binary classification problems (Sharma et al., 2017). In contrast, linear activation functions are often employed in regression problems, as the neuron outputs the same value it receives, accommodating any range of values.

3.1.1 Convolutional Neural Network

CNNs, a class of deep learning models, are primarily used for image processing but can also be applied to one-dimensional data, such as time-series data. A CNN comprises convolutional, pooling, and fully connected layers. Each of these layers contributes to the overall capability of a CNN to effectively learn spatial hierarchies, where higher-level features are derived from lower ones (Sanchez-Reolid et al., 2022).

Figure 3.3: Illustration of the Hidden Layers of a Convolutional Neural Network.



Convolutional Layer: In time-series data, a convolutional layer employs 1D convolution instead of the 2D convolution used for image data. In 1D convolution, a filter slides over the time-series data and performs an elementwise multiplication with the part of the series it currently covers, summing these up to get a single value (Li et al., 2017).

$$(f * g)(t) = \int_0^t f(\tau)g(t-\tau)\, d\tau$$

Where *f* is the input, *g* is the filter, and τ is the convolution window position. The convolution operation captures the local dependencies in the original time series. Different filters can learn to capture different types of information in the series.

Pooling Layer: The pooling layer is a form of non-linear down-sampling. Its main functions are to progressively reduce the spatial or temporal size of the input representation and reduce the output's sensitivity to shifts and distortions (Kiranyaz et al., 2019). Thus, reducing the computational complexity, memory usage, and parameters, preventing overfitting.

The max pooling function takes the maximum value in a particular window (or sub-region). Where *p* is the pooling size, and the index I slides with the stride, *X* as input and *Y* as output, each element of *Y* is computed as:

$$Y[i] = max(X[i : i + p])$$

Fully Connected Layer: The fully connected layer follows several convolutional and pooling layers towards the network's end. The purpose is to produce a final output, such as classification or prediction, using activation functions.

3.1.2 Other Deep Learning Models

Beyond the models discussed, Deep Q-learning Networks (DQN), Deep Reinforcement Learning (DRL) and Autoencoders (AE) hold potential for algorithmic trading. DQNs adapt to market conditions, emphasise vital features and detect anomalies, while DRLs refine strategies through interactions, optimising decisions for long-term reward (Chen et al., 2018).

3.2 Sentiment Framework

This section explores theories that interlink financial markets and investor sentiment, focusing on behavioural finance, text sentiment representation, and the application of NLP in extracting sentiment from text.

3.2.1 Behavioural Finance and Investor Sentiment

Behavioural finance stands at the crossroads of psychology and economics. As defined by Glaser et al. (2003), this field seeks to uncover investment abnormalities that aren't easily explained by traditional frameworks. It questions the Efficient Market Hypothesis (EMH), which asserts that financial markets inherently reflect the true intrinsic value of securities (Ţiţan, 2015). In contrast, behavioural finance proposes that certain market anomalies can be attributed to investors acting irrationally. These irrational systematic errors often lead to price distortions, abnormal returns, and consequent market inefficiencies, prompting the development of several frameworks.

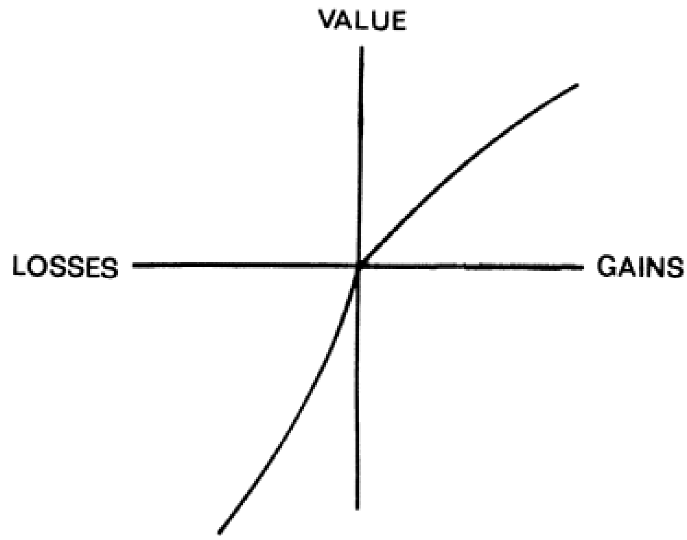
A standout example is the Prospect Theory, proposed by Kahneman and Tversky (1979). It delves into the decision-making process of investors, especially in uncertain, risky situations. At its core, the theory argues that decisions are not always based on actual probabilities. Instead, the perceived likelihood of a potential gain or loss can hold more impact. This concept is embedded in two core components of the Prospect Theory:

Value Function *(V)* quantifies the perceived value for gains/losses *(x)*. Notably, it highlights that losses often resonate more deeply than equivalent gains, a phenomenon known as loss aversion *(λ)*. Mathematically represented as:

$$V(x) = \begin{cases} x^{\alpha} \; if \; x \geq 0 \\ -\lambda(-x)^{\beta} \; if \; x < 0 \end{cases}$$

Where α and β represent the diminishing sensitivity to gains and losses, respectively.

Figure 3.4: *The Prospect Theory Curve: A Glimpse into the Human Psyche's Distinct Responses to Gains and Losses.*



Source: *(Kahneman and Tversky, 1979)*

Weighting Function represents how individuals subjectively weigh probabilities. Mathematically represented as:

$$w(p) = \frac{p^{\gamma}}{(p^{\gamma} + (1-p)^{\gamma})^{\frac{1}{\gamma}}}, for\ all\ 0 \leq p \leq 1$$

Where γ represents the extent of probability distortion. When $\gamma < 1$, there is a tendency to place undue emphasis on small probabilities, often leading to irrational decision-making in risk situations.

The Prospect Theory explains why investors' decisions might deviate from traditional rational models by accounting for these behavioural biases. It underscores that emotional and cognitive biases can offer deeper insights into shifts in investor sentiment and, by extension, the consequent market impacts, be they bullish or bearish.

3.2.2 Sentiment Analysis with Natural Language Processing

Sentiment analysis, a subfield of NLP, deciphers written text to understand the inherent sentiment –positive, negative, or neutral. This sentiment has the potential to significantly influence readers' perceptions and their subsequent actions, thereby impacting financial markets and investor decisions (Tetlock, 2007).

Figure 3.5: *An Overview of the Natural Language Processing Process.*

The sentiment analysis methodology initially takes raw text data collected from news sources. Given the noise in raw data, including irrelevant symbols, punctuation, and stop words (such as ‘the’, ‘is’, ‘and’), it undergoes text preprocessing. This process includes tokenisation, where the data is split into words or phases; stop words removal; lemmatisation to reduce words to their root form; and occasionally, handling of n-grams, or combination of adjacent words.

Once cleaned, the text is used to train an ML model on a labelled dataset, enabling it to predict the sentiment of previously unseen text. Techniques like Term Frequency-Inverse Document Frequency (TF-IDF) or Word2Vec can convert text into numerical vectors, which can subsequently be utilised to interpret market trends.

4 Research Design & Methodology

Primary questions include:

1. How does the performance of various deep learning models (RNN, LSTM, GRU, CNN) vary when applied to the stock prices, e.g. AAPL and INTC, and what factors might influence these discrepancies in model efficiency?
2. How does the investor sentiment across a selection of 200 stocks reflect broader market dynamics, and what implications does this sentiment have for understanding potential trading cues and market fluctuations?
3. How does integrating deep learning and sentiment analysis within the ASTRA trading system influence its behavioural dynamics and risk management in the context of market inefficiencies?
4. How does increasing the diversity of a portfolio, from 50 stocks to 200, impact the ASTRA trading system's financial performance and risk profile over time?

To answer these research questions, an experimental approach will be adopted, combining software development, quantitative data analysis and forward performance testing of the system.

4.1 Machine Learning Logic

4.1.1 Input Features

Deep learning methodologies are known for effectively predicting the intricate patterns of financial markets, as they can discern complex, non-linear relationships in data (Alpaydin, 2020). Thus, our input features consist of three sets of variables. The first set is the historical daily adjusted trading data of each securities. These features provide basic information about each asset, forming the foundation for trend analysis. The second set is the technical indicators, acting as secondary metrics to understand intricate security behaviour patterns. The third set is the market indexes, providing market context.

4.1.2 Data Processing

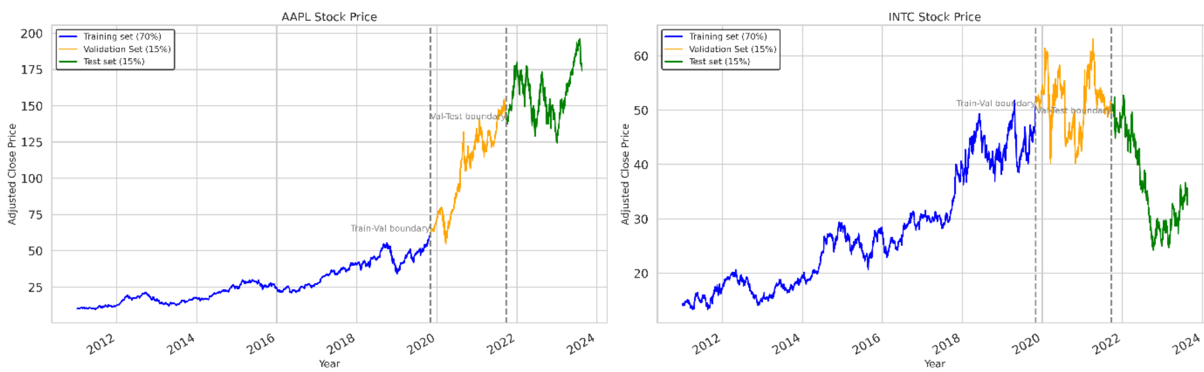
For the system to function optimally, the raw time-series data undergoes a series of transformations:

Time-Window Slicing: Data (*D*) is transformed into overlapping windows (*W*) of length (*w*) (defaulted to 60 days), where each window predicts the next day's outcome.

$$W_i = [D_{i-w+1}, D_{i-w+2}, \dots, D_{i-1}, D_i]$$

Data Splitting: The dataset is partitioned into training (70%), validation (15%), and testing (15%) subsets. These partitions ensure robust training and unbiased evaluation.

Figure 4.1*: Stock Price Split Among Training, Validation, and Test Sets of AAPL & INTC.*



Data Scaling: Each feature *f* in the dataset is scaled between 0 and 1 using the Min-Max scaling:

$$f' = \frac{f - min(f)}{max(f) - min(f)}$$

Here, f' represents the scaled feature. The scaling parameters are derived from the training data, ensuring no data leakage.

Window Data Creation: Each window utilises the first $w - 1$ data points as features, denoted as *X* and the last data point as the target *y*.

$$X_i = [D_{i-w+1}, D_{i-w+2}, \dots, D_{i-1}]$$

$$y_i = D_i$$

This ensures that the model is learning from the temporal data pattern within the data.

Output: The processed data comprises features-label pairs (*X, y*) tailored for the training, validation, and testing sets. The structure of these sequences is optimal for feeding into a 1D CNN, allowing the model to discern and learn from the temporal patterns within the defined windows.

4.1.3 CNN Architecture

The system will use a 1D CNN to analyse the time-series data on a range of 50-200 largest capitalisation US stocks. The architecture consists of the following layers:

First Convolutional Layer: For each element *i* of the input sequence *x*, with a kernel size of 2:

$$y_i = ReLU\left(\sum_{j=1}^{2} x_{i+j-1} * k_j\right)$$

This generates feature maps $y = [y_1, y_2, \dots, y_{n-1}]$ with a length of $n - 1$.

First Max-Pooling Layer: For each pair of values $(y_{2i}, y_{2i} + 1)$

$$pool(y_{2i}, y_{2i+1}) = max(y_{2i}\ , y_{2i+1})$$

This halves the size of *y,* producing: $y' = \left[y'_1, y', \dots, y'_{\frac{n-1}{2}}\right]$.

Second Convolutional Layer and Pooling Layer: Further manipulates the feature maps, resulting in $y''' = \left[y'''_1, y''', \dots, y'''_{\frac{n-3}{4}}\right]$.

Flattening and Dense Layer: Transforms the sequence into a single vector and processes through 50 neurons:

$$z = ReLU(W_1 * y'' + b_1)$$

Where *z* is a vector of size 50, W_1 is a matrix of size $50 * \frac{n-3}{4}$, and b_1 is a bias vector of size 50.

Output Layer: Computes the predicted value:

$$\hat{y} = W_2 * z + b_2$$

Here, $\hat{y}$ is the predicted value, W_2 is a matrix of size 50, and b_2 is a bias scaler.

The architecture is designed to capture the latent temporal patterns in the time-series data. The architecture integrates several layers to capture the underlying patterns and trends in the historical stock data.

4.2 Model Evaluation

To assess the model’s performance, two essential metrics will be used:

Mean Squared Error (MSE): Calculates the average squared differences between predicted and actual values, expressed as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y_i} - y_i)^2$$

Mean Absolute Error (MAE): Calculates the average absolute differences between predicted and actual values, represented as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\widehat{y_i} - y_i|$$

During the training phase, the model will employ the MSE loss function, given that a lower MSE or MAE indicates better accuracy, with MSE being particularly advantageous due to its larger forecasting error that can be more impactful in sequential data trends (Huang et al., 2020). The Adam (Adaptive Moment Estimation) optimisation algorithm will be used for refinement, given its ability to dynamically adjust learning rates based on the gradient's first and second moments, thereby boosting prediction validity (Singarimbun et al., 2019). To mitigate overfitting, checkpoints will be established to capture iterations demonstrating the lowest validation MAE, ensuring the most performant iteration of the system will be retained for optimal reliability.

4.3 Trading Strategy

Inspired by the intricate relationship between stock returns and market beta highlighted by Fama and French (1992), the proposed trading strategy integrates quantitative metrics and sentiment scores to inform buying and selling decisions. This synergy of traditional financial indicators and sentiment analysis provides a holistic approach to stock evaluation.

4.3.1 Signal Generation

For each stock (*t*), a trained model (*m*) anticipates its future price. The percentage change in the predicted stock price from one day to the next is calculated as follows:

$$\Delta P = \left(\frac{P_{NextDay} - P_{PreviousDay}}{P_{previousDay}}\right)$$

This metric serves as an indicator of the expected price movement direction.

Gottwald (2012) exploration into the P/E ratio's significance in stock valuation shows the importance of quantitative indicators in trading decisions. Following this insight, a Quantitative Value (QV) score is derived from a dataset of financial metrics. These metrics include:

- Price-to-Earning ratio
- Price-to-Book ratio
- Price-to-Sales ratio
- Enterprise Value to EBITDA ratio

- Enterprise Value-to-revenue ratio

Each metric is converted into its respective percentile within the data universe. For each metric (*M*), its corresponding percentile P_M is computed as:

$$P_M = \frac{Rank(M)}{TotalNumber}$$

The QV score for a given ticker is then determined by taking the average of the percentiles of all its metrics:

$$QV_{score} = \frac{1}{n}\sum_{i=1}^{n} P_{Mi}$$

The QV score quantifies a stock's financial strength and attractiveness, providing a comprehensive view of its potential.

News articles' sentiment labels are quantified by leveraging the Alpha Vantage API's NEWS-SENTIMENT function. For instance, "Very Bullish" is assigned $S = 5$, while "Very Bearish" corresponds to $S = -1$. Each article has a sentiment score s_i and relevance score r_i. The weighted average sentiment score *W* for a ticker is:

$$W = \frac{\sum_i s_i * r_i}{\sum_i r_i}$$

This approach aligns with sentiment analysis techniques highlighted by Tetlock (2007), aiming to gauge the impact of news sentiment on financial markets accurately. Integrating sentiment and relevance scores offers a nuanced representation of market sentiment, a vital component in algorithmic trading, as Zhang and Skiena (2010) and Yadav et al. (2019) noted.

Upon acquiring the percentage change, sentiment and QV score for a stock, they are all normalised to the [0, 1] range using the Min-Max scaling technique, assuring magnitude homogeneity and model reliability. Subsequently, a weighted composite score for each stock is derived based on equally distributed weights:

$$\begin{aligned} CompositeScore &= w_s * SentimentScore + w_p * PercentChange \\ &+ \frac{w_q * QuantitativeValueScore}{QuantitativeValueScore + 0.0001} \end{aligned}$$

w_s, w_p, w_q are weights assigned to the percentage change, sentiment and QV score, respectively. To classify the stocks based on their potential desirability, two thresholds are determined: below the 20th percentile of the composite scores indicates a 'Sell' signal, while one above the 80th percentile indicates a 'Buy' signal.

4.3.2 Risk Management

Sukrianingrum and Manda (2020) accentuate the influence of systematic and unsystematic risks on portfolio returns. The strategy limits a single asset's concentration in the portfolio to address company-specific risk. Specifically, no asset exceeds 10% of the trading capital:

$$Position\ Size = \frac{Trading\ Capital}{Number\ of\ Buy/Sell\ Assets}$$

$$Max\ Position\ Size = 0.10 * Trading\ Capital$$

Additionally, risks not captured by standard financial metrics are considered. Assets with negative sentiment are deemed riskier; adjusting investment downwards:

$$Adjusted\ Shares = 0.50 * Original\ Shares$$

A trailing stop-loss will be used to protect profits and limit losses. The mechanism adjusts as follows:

$$Trail\ Stop \begin{cases} If\ Long: & Sell\ Price = Highest\ Price - (Highest\ Price * Trail\ Percent) \\ If\ Short: & Buy\ Price = Lowest\ Price - (Lowest\ Price * Trail\ Percent) \end{cases}$$

Understanding an asset's impact on the portfolio is vital, especially in market shifts. This is gauged through the Beta value:

$$Portfolio\ Beta = \sum_{i=1}^{n} \beta_i * w_i$$

Where β_i is the Beta of the i^{th} asset and w_i is its weight in the portfolio.

Valeyre et al. (2017)'s exploration of the beta-neutral model resonates with this strategy's risk assessment. This Beta value provides insights into the portfolio's sensitivity to market movements. To hedge this sensitivity, the system strategically buys or sells SPY (S&P 500 index ETF), maintaining a balanced beta value for the portfolio:

$$Quantity\ of\ SPY = \frac{Portfolio\ Beta * Portfolio\ Value}{Current\ SPY\ Price}$$

4.4 Ethical Considerations

The research utilises publicly accessible historical stock data, ensuring no violation of proprietary data. Methodological transparency is maintained by openly disclosing all algorithms, transformations, and evaluations, thus ensuring the replicability of the research. The research minimises potential human biases by anchoring the study in quantitative metrics and model-driven outputs, emphasising its commitment to objectivity and data-driven insights.

5 Finding & Analysis

The research aims to comprehend the implications of integrating deep learning and sentiment analysis into quantitative trading. ASTRA, our proprietary system, was developed using Python and TensorFlow. Its real-world effectiveness was tested on a simulated trading platform facilitated by Alpaca Trading. All quantitative assessments and statistical analyses were conducted using Python and Excel. This study draws from contemporary literature and established frameworks, contributing meaningful insights to algorithmic trading advancements.

5.1 Model Performance

The CNN model was predominantly favoured in exploring predictive modelling across a portfolio ranging from 50 to 200 stocks. This decision emerged from a combination of empirical evidence from academic literature, computational efficiency, and the unique attributes of financial time series data. When applied to all 200 stocks, the ASTRA system registered an average test loss (MSE) of 917.46, with an accompanying average MAE score of 12.77.

To lay the groundwork for a comprehensive understanding of performance, four deep learning algorithms, RNN, LSTM, GRU, and CNN, were applied to two distinct stocks: AAPL and INTC. The subsequent data, encapsulated in Table 6.1, presents a comparative view of MSE and MAE outcomes on these stocks.

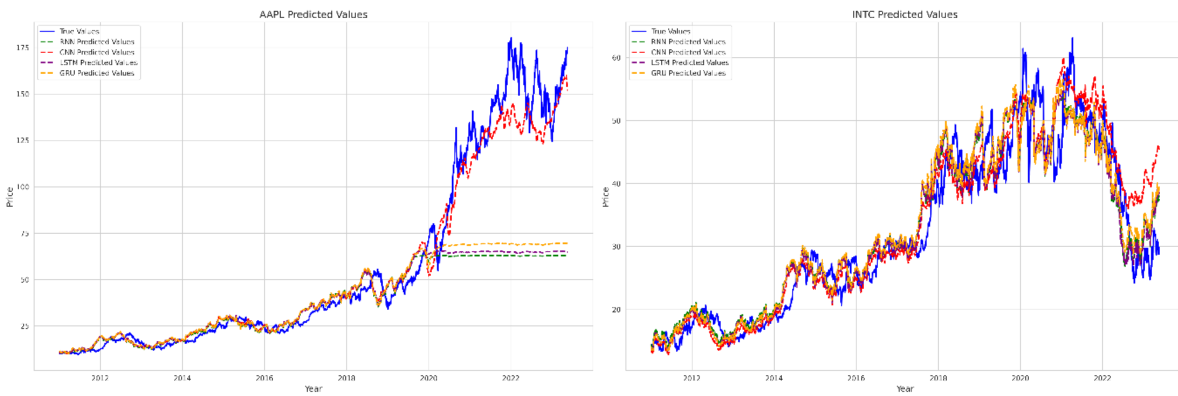
Table 5.1*: Comparison of MSE and MAE for Different Models on AAPL and INTC Stock Prices.*

	AAPL		**INTC**	
	MSE	MAE	MSE	MAE
RNN	1545.77	21.13	15.98	3.06
LSTM	1468.47	20.63	16.78	3.14
GRU	1326.07	19.83	18.69	3.31
CNN	101.57	6.32	19.97	3.11

For AAPL, there was a notable disparity in performance across the models. While RNN, LSTM, and GRU models registered considerable MSE, the CNN model exhibited a dramatically reduced loss and markedly lower MAE. As depicted in Table 5.1, CNN’s performance stands out, particularly in capturing intricate patterns in the stock price movements. The observed performance by the CNN model aligns with previous findings from Doering et al. (2017), Gudelek et al. (2017), and Selvin et al. (2017), highlighting the capabilities of the CNN, especially in capturing short-term trends embedded within the data, an observation consistent with our results.

In contrast, for INTC, the model's performances converged more closely. While RNN and LSTM had close metrics, GRU reported slightly higher values. Interestingly, CNN, which demonstrated exceptional proficiency with AAPL, presented the highest MSE for INTC, although its MAE remained competitive MAE. As illustrated in Figure 6.1, while the models showed a more uniform performance on the INTC stock, there were still subtle differences in their ability to the actual stock prices. Although closely aligned with the stock prices, CNN's predictions showed occasional deviations, hinting at its reduced dominance in this dataset.

Figure 5.1*: Stock Price Prediction of CNN, RNN, LSTM and GRU Models.*



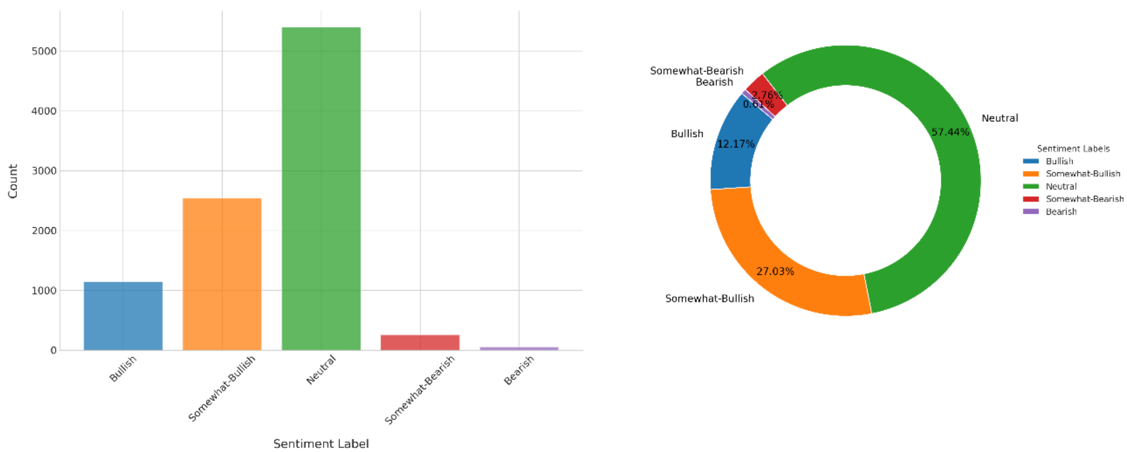
These findings underscore the significance of understanding stock-specific behaviour when selecting predictive models, as the efficiency of a particular model can diverge based on the inherent volatility and historical data patterns of the stock in question. While INTC stock seemed more visually volatile, the nature of volatility might be different. APPL could have short-term, rapid fluctuations that resemble spatial patterns, which CNNs excel at capturing. INTC's volatility, on the other hand, might be more sequential, which RNN-based models like LSTM and GRU are better equipped to handle.

The contrasting performance between the CNN model and the RNN, LSTM, and GRU models can be attributed to a series of challenges intrinsic to deep learning architectures. While LSTM and GRU were designed to handle the challenges of RNNs, they appeared susceptible to issues like the vanishing gradient problem, potentially leading to static predictions. Overfitting could also have hindered their performance on new data sequences.

5.2 Overall Sentiment Distribution

The sentiment distribution of stocks provides insights into the overall market sentiment and can be used to identify potential investment opportunities and manage risk. In our study encompassing a selection of 200 stocks, we observed a predominant trend towards the 'Neutral' sentiment, indicating a lack of a strong positive or negative bias among most stocks.

Figure 5.2: *Breakdown of Stock Sentiments: A Glimpse into Investor Perspectives.*



The dominant neutral sentiment observed in the findings suggests that the market might either be in an anticipatory mode, awaiting significant events or cues to shape its sentiment, or the stocks within this category may lack impactful news or updates, leading to a stagnation in sentiments.

Drawing on the insights of Zhang and Skiena (2010), a predominant neutral sentiment could highlight an environment where media sentiment data is not providing strong trading cues. Given the importance of the quality and timeliness of the news sources they mentioned, the neutral state could indicate a period where recent news lacks strong sentiment cues or directional insights for trading. Furthermore, Yadav et al. (2019) approach underlying the complexity of using sentiment analysis for market trends, given the inherent challenges. The neutral dominance could also imply a potential overlap of unrelated news stories, adding noise and diluting strong sentiments.

Although not as prevalent as the neutral sentiment, the bullish sentiment sheds light on an optimistic market segment. A potential catalyst could range from promising financial projections and broader economic indicators to specific positive news events related to these stocks. Conversely, the bearish sentiment's minor representation, possible reasons could be industry-specific challenges, adverse news affecting these stocks, or perhaps underwhelming financial outcomes for these entities. However, drawing from Hajek and Barushka (2018), advancements like combining sentiment analysis and topic detection could further refine findings and present a clearer picture of the underlying causes.

Sentiment analysis's significance in understanding market dynamics is well documented. Our dominant neutral sentiment discovery mirrors market trends observed in foundational research. Periods of neutral sentiments often act as precursors to significant market fluctuations, affirming the insights from Zhang and Skiena (2010) and Yadav et al. (2019).

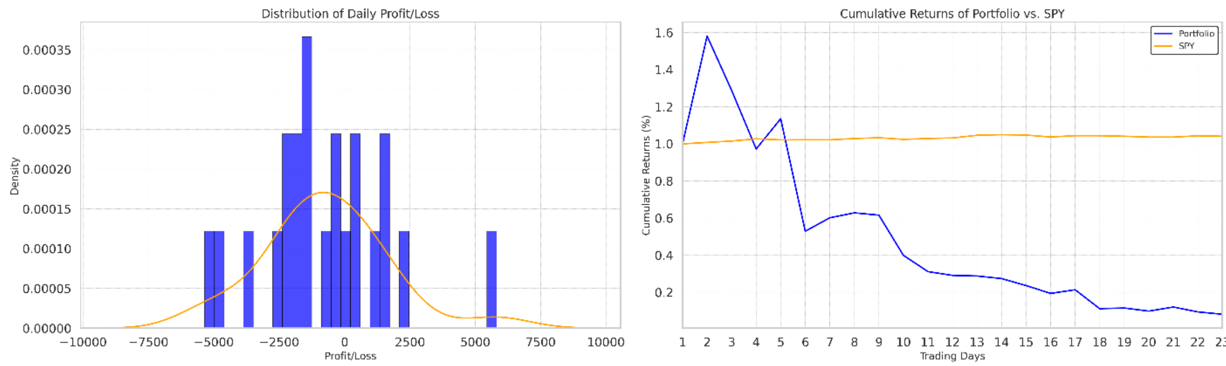
5.3 ASTRA's Performance

The ASTRA trading system, designed to optimise trading strategies, has exhibited distinct performance characteristics. This analytical dive into its performance provides insights into its efficiency, risk management, and return capabilities.

In a recent evaluation, the ASTRA trading system concluded with a net loss of $16,738.63, marking a decline of 1.68% from its initial equity of $1,000,000.00 as of August 18th. This performance was characterised by a volatility of returns with a standard deviation of 0.23%, hinting at its risk exposure, especially considering the maximum daily drawdown of 2.24%. A particularly telling metric is the Sharpe ratio of -0.32, which, when negative, indicates the system's underperformance relative to the risk-free rate of return, highlighting the inherent challenges and risks it encountered during the period.

Delving into ASTRA's trading dynamics revealed significant volatility in its day-to-day returns. The range of this volatility was extensive; on the positive side, the system registered a remarkable single-day peak gain of $5808.24. Contrarily, it also witnessed significant drawdowns, with the largest daily setback pegged at $5355. These fluctuations showcase the system's dynamic trading approach, potentially capitalising on short-lived market movements. In comparison, profit days showcase the system's adeptness in capitalising on rewarding opportunities.

Figure 5.3: *Unpacking Portfolio Health: A Dive into Daily Variations and Cumulative Returns Relative to SPY.*



The cumulative returns trajectory is a comprehensive view of ASTRA's temporal performance. Starting with a base cumulative return coefficient of 1, the system peaked early at 1.58%. However, as trading progressed, the portfolio experienced phases of both growth and decline, eventually settling at a modest cumulative yield of 0.08% by the end of the observation period.

Figure 5.3 highlights an intriguing facet: ASTRA's daily profit/loss distribution deviates from a typical normal curve in its returns, challenging traditional finance theories like the EMH. We observe a disproportionate number of negative data points compared to positive, leading to a right-skewed representation. Additionally, the pronounced tails of the distribution denote a higher occurrence of extreme values than usually anticipated with normal distribution. EMH posits that markets inherently reflect the true intrinsic value of securities (Ţiţan, 2015). However, ASTRA's behaviour aligns more with behavioural finance theories, suggesting that markets might not always be efficient, potentially due to investor irrationalities. This finding provides a coherent link between ASTRA's trading dynamics and observed market inefficiencies.

In response to our research question regarding the rationale behind the non-normal distribution, our analysis suggests that the significant gains and drawdowns may manifest the market's reaction to investor decisions that deviate from the rational model. The Prospect Theory by Kahneman and Tversky (1979) reveals that such decisions can be driven by perceived probabilities rather than actual ones. The erratic performance of ASTRA may result from the system attempting to capitalise on or mitigate the impacts of these deviations.

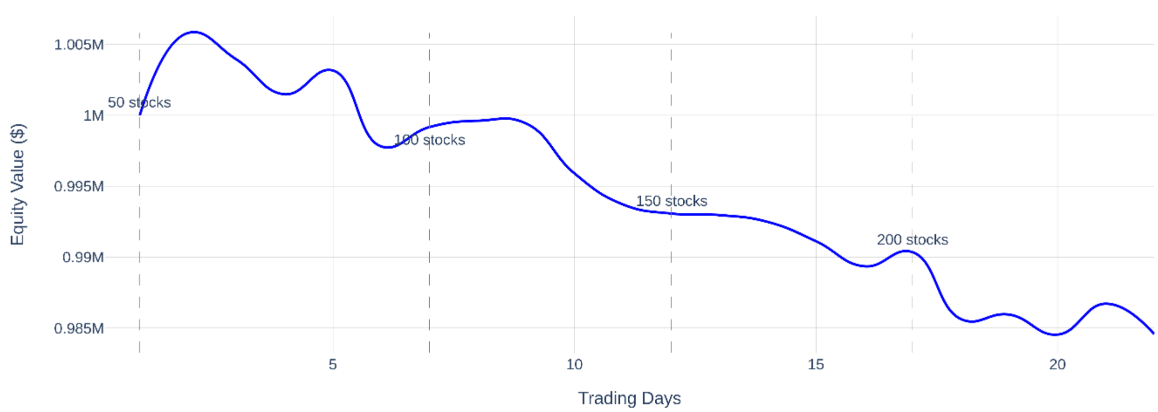
Additionally, the significant drawdowns in ASTRA's performance could be symptomatic of a loss aversion tendency. According to Prospect Theory, losses are felt more deeply than equivalent gains. ASTRA was designed with this principle in mind; its risk management strategies were skewed conservatively, leading to scenarios where it exits positions prematurely, especially after brief market dips.

Table 5.2: *Diversification Insights: How Increasing Stock Numbers Influence Portfolio Performance Metrics.*

Number of Stocks	Equity Value	Volatility	Returns
50	$999,174.65	0.33%	-0.08%
100	$993,075.11	0.14%	-0.61%
150	$990,350.78	0.10%	-0.27%
200	$983,261.37	0.21%	-0.72%

Table 5.2 illustrates a decline in the equity value of the portfolio as the number of stocks increases. This indicates that, within the parameters of the ASTRA trading system, portfolios with higher levels of diversification tend to have lower overall value. This contradicts a common belief in finance, where diversification is often seen as a method to enhance portfolio value by capturing a broader market segment (Fama and French, 1992).

Figure 5.4: *Broadening Horizons: How Portfolio Performance Evolves with Increased Stock Diversification.*



The volatility also provides insight that aligns with the findings of Sukrianingrum and Manda (2020), who noted the influence of systematic and unsystematic risks on expected returns. The noticeable decreases in volatility as the selection expands from 50 to 150 suggest a reduction in overall portfolio risk, supporting the traditional finance perspective of risk reduction through diversification. However, the slight reversal in risk reduction at the 200-stock threshold indicates the need for a holistic approach in portfolio management, considering both market-wide and company-specific factors (Mittermayer, 2004, Schumaker and Chen, 2009).

Moreover, the returns across the portfolios offer another dimension to the analysis. Returns are consistently negative, yet they tend to become more negative with increased diversification. This suggests that while diversification might reduce risk, it seems to do so at the cost of returns, indicating an optimal point of diversification beyond which the benefits diminish. This suggests that an optimal level of diversification should be aimed for; veering off this optimum might lead to over-diversification and potentially erode the advantages sought.

6 Conclusion & Recommendations

Despite the backdrop of the digitalisation of financial markets, quantitative trading has flourished over the past decades. This growth, as noted by Grossman (1988) and Monks and Lajoux (2011), underscores the evolution from basic electronic systems to sophisticated algorithmic trading methods. The advancements in deep learning and sentiment analysis, as detailed by Alpaydin (2020), have catalysed this evolution. Our study aimed to dissect the potential of these technologies when applied to algorithmic trading.

6.1 Summary of Key Findings

The literature often emphasises the necessity of high-quality datasets for deep learning models (Chen et al., 2017, Whang et al., 2023). In line with this, our evaluation through ASTRA provided nuanced insights into predictive modelling and sentiment analysis. As Selvin et al. (2017) did, we found distinctions in predictability between stocks. For instance, while AAPL mirrors the pattern recognition strengths of CNN models, as Doering et al. (2017) suggested, INTC resonates more with the temporal capabilities of RNNs. This variance supports Théate and Ernst (2021) assertion that individual stock characteristics, like their volatility and data lineage, play an integral role in predictive modelling.

Sentiment analysis, as explored by Zhang and Skiena (2010) and Yadav et al. (2019), also plays a pivotal role in stock market dynamics. In our analysis, a prevailing trend of neutral sentiments emerged, highlighting the intricacies of market events and their influence on stocks. This is consistent with the challenges highlighted in the literature regarding quality and timeliness of sentiment data.

ASTRA's varied performance accentuates the delicate balance of algorithmic trading amidst market inefficiencies, echoing findings by Kahneman and Tversky (1979) regarding investors' irrational decision-making processes. Our exploration into portfolio diversification further ties back to the findings by Sukrianingrum and Manda (2020), emphasising the balance of risk and return. While diversification is generally acknowledged to mitigate volatility, our results point to a threshold beyond which benefits diminish.

6.2 Limitation and Further Research

In our exploration of algorithmic trading, sentiment analysis, and market anomalies, we found a pronounced dependence on the Alpha Vantage API as our primary data source. Echoing the concerns raised by Whang et al. (2023) about the necessity of high-quality, diverse datasets, our reliance on a single data source could have influenced the accuracy and comprehensiveness of our results. For future studies, diversifying data sources, as indicated in the literature, can enhance the robustness of the analysis.

While our methodology provided valuable insights, it also highlighted the absence of an optimal weighting system in merging sentiment, prediction and quantitative scores. This resonates with Théate and Ernst (2021) concerns about potential biases in self-designed metrics. Our composite score might not offer the most holistic reflection of market potential, and further research could refine this aspect, perhaps employing advanced methods like machine learning or genetic algorithms, which have proven effective in other contexts.

Further, our system's static nature in portfolio management posed another significant limitation. Valeyre et al. (2017) emphasise the effectiveness of the beta-neutral model, especially during market stress periods. Our inability to dynamically adjust a beta-neutral portfolio according to market triggers, like trailing stops, is a gap that future research needs to address to enhance real-world application.

Lastly, while our research offers a valuable contribution to the domain, future endeavours must be guided by past literature lessons, particularly data quality, methodology biases, dynamic portfolio management, and real-world considerations like transaction costs. With such iterative refinement, the domain of algorithmic trading will undoubtedly advance to greater sophistication and accuracy.

7 References

ALPAYDIN, E. 2020. *Introduction to machine learning*, MIT press.

BAILI, N. 2020. Automatic target recognition with convolutional neural networks.

BALLINGS, M., VAN DEN POEL, D., HESPEELS, N. & GRYP, R. 2015. Evaluating multiple classifiers for stock price direction prediction. *Expert systems with Applications,* 42**,** 7046-7056.

BAO, W., YUE, J. & RAO, Y. 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLOS ONE,* 12**,** e0180944.

CHEN, G., CHEN, Y. & FUSHIMI, T. 2017. Application of deep learning to algorithmic trading. *Tech. Rep, Tech. Rep.*

CHEN, Y.-Y., CHEN, W.-L. & HUANG, S.-H. Developing arbitrage strategy in high-frequency pairs trading with filterbank CNN algorithm. 2018 IEEE International Conference on Agents (ICA), 2018. IEEE, 113-116.

DOERING, J., FAIRBANK, M. & MARKOSE, S. Convolutional neural networks applied to high-frequency market microstructure forecasting. 2017 9th computer science and electronic engineering (ceec), 2017. IEEE, 31-36.

DONGARE, A., KHARDE, R. & KACHARE, A. D. 2012. Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT),* 2**,** 189-194.

FAMA, E. F. & FRENCH, K. R. 1992. The Cross-Section of Expected Stock Returns. *The Journal of Finance,* 47**,** 427-465.

GLASER, M., NÖTH, M. & WEBER, M. 2003. Behavioral finance. *Rationalitätskonzepte, Entscheidungsverhalten und ökonomische Modellierung,* 3.

GOTTWALD, R. 2012. The use of the P/E ratio to stock valuation. *European Grants Projects Journals,* 2012**,** 21-24.

GROSSMAN, S. J. 1988. Program trading and market volatility: A report on interday relationships. *Financial Analysts Journal,* 44**,** 18-28.

GUDELEK, M. U., BOLUK, S. A. & OZBAYOGLU, A. M. A deep learning based stock trading model with 2-D CNN trend detection. 2017 2017. IEEE.

HAJEK, P. & BARUSHKA, A. Integrating sentiment analysis and topic detection in financial news for stock movement prediction. Proceedings of the 2nd International Conference on Business and Information Management, 2018. 158-162.

HUANG, J., CHAI, J. & CHO, S. 2020. Deep learning in finance and banking: A literature review and classification. *Frontiers of Business Research in China,* 14**,** 1-24.

KAHNEMAN, D. & TVERSKY, A. 1979. Prospect theory: An analysis of decision under risk. *Econometrica,* 47**,** 263–292.

KIRANYAZ, S., INCE, T., ABDELJABER, O., AVCI, O. & GABBOUJ, M. 1-D convolutional neural networks for signal processing applications. ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019. IEEE, 8360-8364.

LECUN, Y., BENGIO, Y. & HINTON, G. 2015. Deep learning. *Nature,* 521**,** 436-444.

LI, D., ZHANG, J., ZHANG, Q. & WEI, X. Classification of ECG signals based on 1D convolution neural network. 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), 2017. IEEE, 1-6.

LI, Y. 2017. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*.

MAGINN, J. L., TUTTLE, D. L., MCLEAVEY, D. W. & PINTO, J. E. 2007. *Managing investment portfolios: a dynamic process*, John Wiley & Sons.

MCGOWAN, M. J. 2010. The rise of computerized high frequency trading: use and controversy. *Duke L. & Tech. Rev.,* 9**,** 1.

MERMILLOD, M., BUGAISKA, A. & BONIN, P. 2013. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. Frontiers Media SA.

MITTERMAYER, M.-A. Forecasting intraday stock price trends with text mining techniques. 37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the, 2004. IEEE, 10 pp.
MONKS, R. & LAJOUX, A. 2011. Corporate Valuation for Portfolio Investment. John Wiley & Sons Inc, Hoboken.
NYSE. 2012. *Program Trading Purchases and Sales* [Online]. [Accessed 27/07/2023 2023].
SAHU, S. K., MOKHADE, A. & BOKDE, N. D. 2023. An Overview of Machine Learning, Deep Learning, and Reinforcement Learning-Based Techniques in Quantitative Finance: Recent Progress and Challenges. *Applied Sciences,* 13**,** 1956.
SANCHEZ-REOLID, R., DE LA ROSA, F. L., LOPEZ, M. T. & FERNANDEZ-CABALLERO, A. 2022. One-dimensional convolutional neural networks for low/high arousal classification from electrodermal activity. *Biomedical Signal Processing and Control,* 71**,** 103203.
SCHUMAKER, R. P. & CHEN, H. 2009. Textual analysis of stock market prediction using breaking financial news. *ACM Transactions on Information Systems,* 27**,** 1-19.
SELVIN, S., VINAYAKUMAR, R., GOPALAKRISHNAN, E. A., MENON, V. K. & SOMAN, K. P. Stock price prediction using LSTM, RNN and CNN-sliding window model. 2017 2017. IEEE.
SHARMA, S., SHARMA, S. & ATHAIYA, A. 2017. Activation functions in neural networks. *Towards Data Sci,* 6**,** 310-316.
SINGARIMBUN, R. N., NABABAN, E. B. & SITOMPUL, O. S. Adaptive moment estimation to minimize square error in backpropagation algorithm. 2019 International Conference of Computer Science and Information Technology (ICoSNIKOM), 2019. IEEE, 1-7.
SUKRIANINGRUM, D. R. & MANDA, G. S. 2020. The effect of systematic risk and unsystematic risk on expected return of optimal portfolio. *SAR (Soedirman Accounting Review): Journal of Accounting and Business,* 5**,** 181-195.
TETLOCK, P. C. 2007. Giving Content to Investor Sentiment: The Role of Media in the Stock Market. *The Journal of Finance,* 62**,** 1139-1168.
THÉATE, T. & ERNST, D. 2021. An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications,* 173**,** 114632.
ŢIŢAN, A. G. 2015. The efficient market hypothesis: Review of specialized literature and empirical research. *Procedia Economics and Finance,* 32**,** 442-449.
VALEYRE, S., GREBENKOV, D. & ABOURA, S. 2017. The Beta Neutral Model with Leverage Effect. *Available at SSRN 3026031*.
WHANG, S. E., ROH, Y., SONG, H. & LEE, J.-G. 2023. Data collection and quality challenges in deep learning: a data-centric AI perspective. *The VLDB Journal,* 32**,** 791-813.
YADAV, R., KUMAR, A. V. & KUMAR, A. 2019. News-based supervised sentiment analysis for prediction of futures buying behaviour. *IIMB Management Review,* 31**,** 157-166.
ZHANG, W. & SKIENA, S. Trading strategies to exploit blog and news sentiment. Proceedings of the international AAAI conference on web and social media, 2010. 375-378.

8 Appendix

Appendix A – Input Feature Landscape

Input Features	Definition/Explanation
Category 1. Daily Trading Data	
Open	Opening stock price of the day
High	Highest stock price of the day
Low	Lowest stock price of the day
Close	The stock closing price of the day
Adjusted Close	The stock closing price of the day adjusted for corporate actions
Volume	Number of shares traded during the day
Dividends	Cash payment made to shareholders on that day
Category 2. Technical Indicator	
SMA	**Simple Moving Average:** The average of daily prices over 60 days
RSI	**Relative Strength Index:** A momentum oscillator that measures the speed and change of daily price movements over 60 days
ADX	**Average Directional Movement Index:** Measures the strength of a daily trend over 60 days without considering its direction
CCI	**Commodity Channel Index:** A momentum oscillator used to assess overbought or oversold conditions based on daily prices over 60 days
ATR	**Average True Range:** Measures daily market volatility by calculating the average true range between high and low stock prices over 60 days
EMA	**Exponential Moving Average:** The average of daily prices over 60 days, with more weight given to recent prices
Category 3. General Market Index	
SPY	Daily adjusted close price of the S&P 500 index, reflecting the performance of 500 leading companies in the US stock market
VIX	Daily adjusted close price of the VIX index, a measure of the market's expected volatility over the next 30 days
Additional Feature	
Target	Adjusted close price of the stock for the subsequent day

Appendix B – Forward and Backward Propagation

Neural networks learn by iteratively making predictions (forward propagation) and then updating the model parameters based on the error of those predictions (backwards propagation).

Stage I: Forward propagation.

Signal Transmission: The X_i transmits the signal to all units in the hidden layer.

Network Calculation for Hidden Layer: The value for each hidden layer unit is obtained by summing up the weighted inputs and bias from the input layer. It can be expressed as:

$$h_j = \sum_i w_{i,j} * x\_i + b_j$$

where $w_{i,j}$ is the weight from the i^{th} input to the j^{th} hidden unit, and b_j is the bias.

Hidden Layer Activation: The output of the hidden layer unit is determined using an activation function:

$$f_{hidden}(h_j) = \frac{1}{1 + e^{-hj}}$$

Network Calculation for Output Layer: The value at the layer is computed similarly:

$$O_k = \sum_i w_{j,k} * f_{hidden} + b_k$$

Where $w_{j,k}$ is the weight from j^{th} hidden layer on the k^{th} output unit.

Output Layer Activation: The final output of the neural network is given by another activation function:

$$f_{output}(O_k) = \frac{1}{1 + e^{-ok}}$$

Error Calculation: The prediction error is the difference between the predicted value and the actual target from the dataset.

$$error = t - f_{outputs}(O_k)$$

Stage II: Backward Propagation.

Initialisation: Momentum and adaptive momentums related to weights and biases are initialised to zero. Suppose it's the first epoch; set t to 0.

Output Layer Gradient: The gradient of the error concerning the weights and biases in the output layer is computed as:

$$\delta_{output} = error * f'_{output}(O_k)$$

Here, $f'_{output}(O_k)$ is the derivative of the activation function at the output layer.

Hidden Layer Gradient: The gradient of the error concerning the weights and biases in the hidden layer is derived as:

$$\delta_{hidden} = \delta_{output} * w_{j,k} * f'_{hidden}(h_j)$$

Here, $f'_{hidden}(h_j)$ is the derivative of the activation function at the hidden layer.

First Adaptive Moment Estimation & Update: Calculates the exponential moving average of the gradient and adjusts the model's weights and biases accordingly.

Second Adaptive Moment Estimation & Update: Calculates the exponential moving average of the squared gradient and further updates weights and biases, introducing an adaptive learning rate mechanism.

Appendix C – RNN, LSTM and GRU Architecture

Categories	Choice		
Library	TensorFlow (Keras)		
Model Type	RNN	LSTM	GRU
Number of Layers	2	2	2
Dropout after each Layer	0.0	0.0	0.2
Number of Dense Layers	2	2	2
Neurons in the First Layer	30	30	50
Neurons in the Second Layer	30	30	50
Neurons in First Dense Layer	25	25	25
Neurons in the Output Layer	1	1	1
First Return Sequences	TRUE		
Second Return Sequences	FALSE		
Optimizer	Adam		
Loss Function	Mean Squared Error		
Metrics	Mean Absolute Error		
Number of Epochs	1000		
Batch Size	15		
Early Stop Monitor	Validation Loss		
Early Stop Patience	100		