INSTITUTO
UNIVERSITÁRIO
DE LISBOA

How Small Neural Networks can Improve Strategies in Financial Markets

Diogo Filipe da Conceição Rolo

Master in Finance

Supervisor:
PhD in Finance, José Carlos Gonçalves Dias,
Full Professor,
ISCTE Business School

September, 2024

Department of Finance

How Small Neural Networks can Improve Strategies in Financial Markets

Diogo Filipe da Conceição Rolo

Master in Finance

Supervisor:
PhD in Finance, José Carlos Gonçalves Dias,
Full Professor,
ISCTE Business School

September, 2024

Acknowledgments

I would like to express deep gratitude to all who have assisted me throughout my academic journey.

First, I express my gratitude to my supervisor and previous professor for their invaluable advice. His capacity to provide assistance when necessary, while also allowing me the autonomy to cultivate my own ideas has been important in my development as a researcher.

I am thankful to my girlfriend and father for their constant encouragement and patience, especially throughout difficult phases of this journey. Their comprehension helped me as a source of strength throughout my academic pursuits.

I would like to express my gratitude to my former professor and collaborator from the Undergraduate Research Opportunity Program, Dr. João Leite, and Dr. Manuel de Sousa Ribeiro. Their research on "Aligning Artificial Neural Networks and Ontologies towards Explainable AI" has profoundly impacted my academic trajectory. The experience acquired under their guidance was important in developing my research interests and finally resulted in the formulation of my thesis.

I extend my heartfelt gratitude to all those acknowledged and those who have aided my academic pursuits. Your assistance has been essential to the completion of this work.

Resumo

Esta dissertação examina o uso de técnicas de *machine learning* para melhorar estratégias de negociação nos mercados financeiros. O estudo avalia a eficácia de *deep learning*, *reinforcement learning* e métodos tradicionais de *machine learning* para o melhoramento de negociação usando vários indicadores técnicos.

Este estudo avaliou várias técnicas de negociação, incluindo *Moving Average Convergence Divergence* (MACD), Média Móvel Simples, *Relative Strength Index* (RSI), *Bollinger Bands* e *Stochastic Oscillator*. A metodologia inclui pré-processamento de dados, *feature engineering* e avaliação de modelos.

Foram realizadas avaliações comparativas de *Support Vector Classifiers,* Árvores de Decisão, *Random Forests*, redes neuronais profundas e agentes de aprendizagem por reforço. A pesquisa examina várias representações de entrada e transformações de saída, avaliando a sua influência na eficácia do modelo.

Os resultados deste estudo demonstram que modelos de aprendizado profundo superou frequentemente métodos convencionais de *machine learning*. A aprendizagem por reforço é promissora, especialmente ao integrar indicadores técnicos, embora tipicamente não exceda o desempenho da aprendizagem profunda. O estudo indica que variações de preço fornecem geralmente melhores resultados que preços em vários designs de modelo.

Esta pesquisa avança na área de finanças computacionais ao ilustrar a eficácia das redes neuronais na melhoria de estratégias de negociação, destacando a importância do pré-processamento de dados na modelagem financeira e oferecendo *insights* sobre as vantagens relativas de várias estruturas de aprendizado de máquina na negociação.

Os resultados desafiam paradigmas atuais em negociação algorítmica e fornecem base para futuras investigações sobre algoritmos de negociação adaptativos usando métodos avançados de *machine learning*.

Palavras-chave: Redes Neuronais Pequenas, Aprendizagem Profunda, Aprendizagem por Reforço, Previsão de Mercado, Indicadores Técnicos, Negociação algorítmica

Sistema de Classificação JEL: G14, G17

Abstract

This dissertation examines the use of machine learning techniques to improve algorithmic trading strategies inside financial markets. The study evaluates the efficacy of deep learning, reinforcement learning, and traditional machine learning methods in enhancing trading performance using various technical indicators.

This study used four years of minute-by-minute Volatility Index (VIX) data to assess several trading techniques, including Moving Average Convergence Divergence (MACD), Simple Moving Average, Relative Strength Index (RSI), Bollinger Bands, and Stochastic Oscillator. The methodology includes data preprocessing, feature engineering, and extensive model evaluation.

Comparative assessments of Support Vector Classifiers, Decision Trees, Random Forests, deep neural networks, and reinforcement learning agents were performed. The research examines several input representations and output transformations, evaluating their influence on model efficacy.

The findings of this study demonstrate that deep learning models frequently surpass conventional machine learning methods. Reinforcement learning is promise, especially when integrating technical indications, although typically does not exceed the performance of deep learning. The study indicates that relative price inputs often provide better outcomes than absolute prices across various model designs.

This research advances computational finance by illustrating the efficacy of neural networks in improving trading strategies, highlighting the significance of data preprocessing in financial modeling, and offering insights into the relative advantages of various machine learning frameworks in algorithmic trading.

The results challenge current paradigms in algorithmic trading and provide the groundwork for future investigations into adaptive trading algorithms using sophisticated machine learning methods. This research has significance for both professionals and academics in finance and artificial intelligence.

Keywords: Tiny Neural Networks, Deep Learning, Reinforcement Learning, Market Forecast, Technical Indicators, Algorithmic Trading

JEL Classification System: G14, G17

Index

ACKNOWLEDGMENTS .... **I**

RESUMO .... **III**

ABSTRACT .... **V**

INDEX .... **VII**

1. INTRODUCTION **1**

2. LITERATURE REVIEW **3**

2.1 TRADITIONAL MACHINE LEARNING METHODS 3
2.1.1 Linear Support Vector Classifier (LinearSVC) *3*
2.1.2 Decision Tree Classifier *3*
2.1.3 Random Forest Classifier *3*
2.2. NEURAL NETWORKS BASICS 4
2.2.1. The Mathematical Neuron *4*
2.2.2. Advancements with the Perceptron *4*
2.2.3. Backpropagation *5*
2.2.4. Reinforcement Learning Integration *5*
2.3. NEURAL NETWORKS IN FINANCE 5
2.3.1 Application in Finance *5*
2.3.2. The Role of Pattern Recognition in Financial Markets *6*
2.4 CHALLENGES AND LIMITATIONS 6
2.4.1 Emerging Challenges in Financial Machine Learning *6*
2.5 FUTURE DIRECTIONS 7
2.5.1 Small Neural Networks for Improved Accessibility *7*
2.5.2 Adaptability of Small Neural Networks *7*

3. DATA **9**

3.1 DATA TRANSFORMATION 9

4. METHODOLOGY **11**

4.1. TRADING TECHNIQUES APPROACHES 11
4.1.1. Moving Average Convergence Divergence (MACD) *11*
4.1.2. Bollinger Bands *11*
4.1.3. Relative Strength Index (RSI) *11*
4.1.4. Moving Average *11*
4.1.5. Stochastic Oscillator *11*
4.2. NEURAL NETWORK AND MACHINE LEARNING INTEGRATION 11
4.2.1 Data Split and Model Evaluation *12*
4.3 TRADE DISTRIBUTION ACROSS STRATEGIES 13

5. RESULTS **15**

5.1 MACHINE LEARNING RESULTS 15
5.1.1 Baseline Model: LinearSVC Performance *15*
5.1.2 Decision Tree Classifier Performance *16*
5.1.3 Random Forest Classifier Performance *18*
5.1.4 Overview *20*
5.2 DEEP LEARNING RESULTS 21
5.2.1 Binary Profits Aproach *22*
5.2.2 Normalized Profits Aproach *24*
5.2.3 Raw Profits Aproach *25*
5.2.4 Overview *27*

5.3 Reinforcement Learning Results 28
5.3.1 Preprocecing Prices Aproach 29
5.3.2 Moving Averages Aproach 31
5.3.3 Technical Indicators Aproach 31
5.3.4 Overview 33
5.4 Technical Imputs on Deep Learning 34
6. CONCLUSIONS 37
6.1. Comparative Analysis of Performance Among ML, DL, and RL Methodologies 37
6.2. Influence of Data Preprocessing Techniques: 37
6.3. Efficacy of Technical Indicators 37
6.4. Initial Hypothesis 38
7. FUTURE WORK 39
7.1 Advanced Data Preprocessing Methods 39
7.2 Hybrid Models 39
7.3 Robustness and Generalization 39
7.4 Interpretability and Explainability 39
7.5 Integration of Risk Management 40
7.6 Applications of High-Frequency Trading 40
BIBLIOGRAPHY 41

1. Introduction

The emergence of artificial intelligence (AI) and machine learning technologies has significantly altered the financial markets environment. These breakthroughs have initiated a new epoch of algorithmic trading, whereby intricate mathematical models and computer algorithms influence investing choices with unparalleled velocity and magnitude. This dissertation investigates the vanguard of this transition, focusing on the use of advanced machine learning techniques to enhance and optimize algorithmic trading strategies.

The intersection of artificial intelligence and finance has become a pivotal focus in academic research and industry progress. Heaton et al. (2017) contend that deep learning techniques are transforming several facets of financial analysis and decision-making. Their research highlights the efficacy of neural networks in tackling intricate financial issues, such as asset pricing and risk management, providing benefits over conventional statistical techniques in managing non-linear connections and extensive datasets.

This study assesses the efficacy of deep learning, reinforcement learning, and conventional machine learning techniques in enhancing trading performance via the use of diverse technical indicators. Our objective is to enhance the current comprehension of AI-driven financial decision-making by concentrating on the incorporation of these sophisticated approaches into established trading platforms.

This study is motivated by a core inquiry: Can accessible neural networks improve the efficacy of technical trading methods without requiring substantial processing resources? This study challenges the prevailing notion that AI applications in finance often need significant computing resources, which may restrict wider involvement and innovation in the sector.

Our inquiry starts with a clear objective: to assess the efficacy of compact neural networks in enhancing technical trading techniques. In contrast to traditional methods that often demand substantial computing resources, we want to ascertain if more accessible neural network models may improve strategy performance. This technique aims to improve trading outcomes while also broadening access to advanced trading tools.

Our study analyzes many trading methodologies, including Moving Average Convergence Divergence (MACD), Simple Moving Average, Relative Strength Index (RSI), Bollinger Bands, and Stochastic Oscillator. By utilizing various machine learning approaches on these established strategies, we aim to discover potential market inefficiencies and provide innovative insights for both scholars and practitioners in computational finance.

Our study methodologically employs four years of minute-by-minute Volatility Index (VIX) data as a foundation. This high-frequency data enables us to evaluate our models across diverse market circumstances and time periods, establishing a solid basis for our research. Our methodology encompasses meticulous data pretreatment, feature engineering, and model assessment approaches to guarantee the validity and trustworthiness of our results.

Throughout this dissertation, we will go from fundamental notions to intricate subjects. We will examine the progression of neural networks, from the basic neuron to sophisticated deep learning and reinforcement learning models. Subsequently, we will use these principles in financial markets, analyzing the efficacy of various machine learning paradigms inside algorithmic trading.

Our aspirations go beyond mere improvements in efficiency. Our objective is to find market inefficiencies that might disturb equilibrium pricing, posing a challenge for both scholars and practitioners. We want to improve understanding of the interaction between neural networks and financial markets by examining known paradigms and exploring new possibilities.

The next chapters will include a comprehensive literature review, delineate our approach, elucidate our empirical results, and provide suggestions aimed at fostering more research and innovation in this rapidly evolving field. This work aims to link theoretical advancements in machine learning with practical applications in financial trading, perhaps revolutionizing our methodology for algorithmic trading in the future.

2. Literature Review

2.1 Traditional Machine Learning Methods

It is crucial to have a basic understanding of conventional machine learning techniques before delving into the nuances of neural networks. Such algorithms are fundamental to model prediction and serve as vital benchmarks in many fields. This section investigates four critical machine learning algorithms: the Linear Support Vector Classifier (LinearSVC), the Support Vector Classifier (SVC), the Decision Tree Classifier, and the Random Forest Classifier.

2.1.1 Linear Support Vector Classifier (LinearSVC)

The Linear Support Vector Classifier is a variant of Support Vector Machines (SVM), a class of algorithms that were first presented by Cortes and Vapnik in 1995. The LinearSVC algorithm operates by finding the most effective hyperplane that divide distinct classes within a feature space. LinearSVC's primary objective is to maximize the margin, which refers to the space between the hyperplane and the closest data point from each class. This approach is particularly effective for binary classification tasks in high-dimensional spaces. The optimization problem in LinearSVC can be expressed as:

$$min(w,b)\ (1/2)||w||^2 + C\ \Sigma\ max(0, 1 - yi(w^T xi + b))$$

where w is the weight vector, b is the bias term, C is the regularization parameter, and (xi, yi) are the training samples and their corresponding labels.

LinearSVC has numerous advantages, such as its effectiveness in high-dimensional spaces and computational efficiency when dealing with huge datasets. Nevertheless, its main drawback is its reliance on the assumption of linear separability across classes, which may not be valid in complex, real-world datasets.

2.1.2 Decision Tree Classifier

Decision Tree Classifiers, based on Quinlan's work in 1986 using the ID3 method, present a distinct approach in the field of machine learning. The ID3 algorithm constructs decision trees using a top-down, greedy approach. The procedure begins with the whole dataset at the root node, which recursively determines the ideal attribute for data splitting, therefore creating child nodes for each possible value of that attribute. The "optimal" feature is chosen based on the highest information gain, which measures the efficacy of each attribute in differentiating training cases according to their target categorization.

Tree-structured models employ a series of feature-based divisions to make decisions, producing a prediction model that is highly comprehensible.

The construction of a decision tree involves repeatedly dividing the data based on feature values to maximize a chosen criterion, such as information gain or Gini impurity. To calculate the information gain for a binary split on feature j at threshold t, apply the following formula:

$$IG(Dp, j, t) = H(Dp) - (|DL|/|Dp|)H(DL) + (|DR|/|Dp|)H(DR)$$

where H is the entropy, Dp is the parent node, and DL and DR are the left and right child nodes, respectively.

Decision Trees have several advantages, including their ability to handle both numerical and categorical data and their high interpretability. However, as emphasized by Breiman et al. (1984), individual Decision Trees are vulnerable to overfitting, especially when they are allowed to grow extensively which can limit their ability to generalize.

2.1.3 Random Forest Classifier

Random Forest is an ensemble learning technique that was presented by Breiman in 2001. It aims to overcome certain limitations of individual Decision Trees. Random Forests enhance prediction stability and robustness by creating several decision trees and aggregating their outputs. The fundamental tenets of Random Forests encompass:

1. Bootstrap Aggregating (Bagging): each tree is trained on a randomly selected subset of the training data.
2. Random Feature Selection: During each split, only a randomly selected subset of features is taken into consideration.

The ultimate prediction is usually determined by employing majority vote (for classification) or averaging (for regression) of the various tree outputs.

Random Forests have a tendency to reduce the problems caused by overfitting and offer rankings of the significance of features. Nevertheless, the enhanced efficiency is accompanied by a decrease in comprehensibility when compared to individual Decision Trees, resulting in a compromise between predictive accuracy and model clarity.

2.2. Neural Networks Basics

Neural networks are vital for artificial intelligence and machine learning as they model the brain workings and architecture. Understanding the various applications of these applications, particularly for finance, requires examining their basics.

2.2.1. The Mathematical Neuron

McCulloch & Pitts (1943) defined the mathematical neuron as a crucial starting point for investigating neural networks. In their analysis, the two scientists examined how human nervous system nerve cells receive and exchange information, aiming to learn the mechanism behind this exchange.

McCulloch and Pitts consider a neuron as being a component with a binary threshold. The mathematical concept discussed in this article is similar in complexity to the mathematical nature of biological neurons. It illustrates how a neuron reacts to signals from other interconnected neurons, leading to either a firing state (represented as "1") or a non-firing state (represented as "0"), depending on the inputs received, with different weights assigned to each input.

The specific binary threshold model has had a notable impact in offering a comprehensive understanding of neural interactions. Establishing a foundation for a wider range of neural network structures opened up opportunities for further improvement and progress. Currently we design neural networks under the assumption that a neuron operates in the same way to a binary outcome, where feedback accumulation surpassing a specific threshold determines its functionality.

2.2.2. Advancements with the Perceptron

Rosenblatt created the perceptron model, marking a noteworthy advancement in the field of neural network development (Rosenblatt, 1958). It enhanced the characteristics of neural networks by refining the weighted connections, based on his study of the Pitts and McCulloch mathematical neuron.

In the structure of the perceptron, the perceptron assigns a specific weight to each input it receives. The weights have an impact on the intensity of the input signals, making it possible for the perceptron to assign importance to inputs of different levels of significance. An activation function is applied to the sum of inputs, each multiplied by specific weights, to determine the perceptron's output.

Fixed perceptron model supported neural networks learning and making choices with weighted inputs, though it had limitations. The perceptron struggled with data featuring nonlinear patterns and was limited to learning functions that could be separated by a straight line. Rosenblatt's perceptron lay the groundwork for future progress in the design of neural network structures.

2.2.3. Backpropagation

Rumelhart, Hinton, and Williams proposed a novel approach for neural network training in 1986. Backpropagation is a technique used for training neural networks (Rumelhart et al., 1986). The algorithm included a systematic approach to aligning connection weights with the discrepancy between the actual output and desired output in order to improve the domain.

Iterative learning is the primary benefit of backpropagation. The neural network's first step involves comparing predictions on the input data to the real outputs to determine the level of error. Then, this specific mistake spreads in a reverse manner across the network, taking responsibility for each connection based on its contribution to the overall error. Adding weights in the opposite direction and repeating the process until it reaches convergence can reduce this specific mistake.

The distinctive aspect of backpropagation rests in training the network using supervised learning with labeled data. This approach enhances the network's ability to capture complex relationships in data, playing a crucial role in modern techniques for training neural networks.

2.2.4. Reinforcement Learning Integration

Incorporating reinforcement learning broadens the range of learning capabilities in neural networks. Commonly linked with other machine learning approaches, reinforcement learning introduces a dynamic element to the process of learning.

Reinforcement learning is a novel approach in the world of neural networks. Mnih et al. (2013) propose a new approach known as deep reinforcement learning emerged, which combines the principles of reinforcement learning with neural networks to facilitate making decisions in a sequential manner.

Deep reinforcement learning, also known as deep Q-networks (DQN), employs neural networks as function approximators. They develop the skill to link actions with states in a given setting and receive either financial rewards or penalties for their efforts. Agents trained with deep reinforcement learning principles can make sequential decisions and adapt their behavior over time by utilizing neural network representations.

2.3. Neural Networks in Finance

2.3.1 Application in Finance

Neural network application in finance varies from stock price prediction to algorithmic trading and risk management. These networks have demonstrated superb performance modeling the nonlinear and often erratic dynamics of financial markets. As evidenced in Dixon et al. (2017), the authors apply deep learning techniques to historical financial data to predict upcoming market behaviors with clear advantages over conventional statistical techniques.

Atsalakis and Valavanis (2009) provide an extensive review of soft computing techniques in stock market forecasting, highlighting the increasing significance of neural networks in financial applications. Their research emphasizes the adaptability of neural networks in handling several financial problems, ranging from price forecasting to risk evaluation. This report highlights the growing use of neural network methodologies in many sectors of finance, emphasizing their importance in contemporary financial analysis and decision-making processes.

Neural networks are applied in the stock price prediction domain to forecast the future price of historical data, considering market trends, company performances, and economic indicators. Their processing and learning abilities on big datasets permit them to discover patterns and correlations that other analytical approaches would miss.

Another promising application of neural networks can be algorithmic trading. These networks can help in creating trading strategies based on new information and changing market conditions by looking at market data. They can process huge quantities of data in real time that is essential in high-frequency trading where milliseconds count.

The generality and learning potential of neural networks makes them excellent for financial applications. Training the networks to adapt to new patterns and conditions as financial markets change preserves their effectiveness over time.

2.3.2. The Role of Pattern Recognition in Financial Markets

In financial trading approaches, pattern recognition is often the aim to locate repeating patterns in market data, which can signal possible lucrative trades. Neural networks are good at this area since they learn and detect patterns in data.

Kim & Lee (2004) 6mphasize the role of pattern recognition in trading strategies. Their analysis demonstrates the possibility of neural networks to identify and exploit trending in stock prices for prediction. Likewise, Guresen et al. (2011) demonstrates further the ability of neural networks to recognize patterns that aren't immediately obvious with conventional analysis.

According to Fischer and Krauss (2018), the efficacy of more sophisticated neural network architectures in financial market prediction is demonstrated by building on these discoveries. Their study use Long Short-Term Memory (LSTM) networks, a kind of recurrent neural network, to detect complex patterns in financial data. The authors demonstrate that LSTM networks may exceed traditional machine learning methods in predicting out-of-sample directional changes in S&P 500 components, highlighting the power of sophisticated neural network models to capture intricate temporal connections in financial time series.

These experiments demonstrate that neural networks, which can handle huge amounts of data and detect complex nonlinear relationships, are optimal for pattern detection in financial markets. This ability is essential for producing efficient trading strategies since it permits anticipating market movements according to identified patterns.

2.4 Challenges and Limitations

The heavy computational needs of neural network implementation in finance are one of the major obstacles (Lawrence, 1997). This issue is particularly applicable for large deep neural networks processing massive datasets as is common for financial applications. The quantity of data needed influences the computation demands, which oftentimes becomes obsolescent, thus threatening both expense, and the operation viability due to the lack of updated data.

The work of Hawkins (2004) addresses another critical difficulty: model complexity versus overfitting risks. Overfitting happens when a model is way too sensitive to the training data and loses generalizability to new unseen data. This creates a huge issue in finance where models must work effectively in dynamic and often unpredictable market environments.

Hawkins' work also offers perspectives on overfitting mitigation methods (cross-validation and regularization). Maintaining the robustness of neural networks to genuine financial data requires these kinds of methods.

2.4.1 Emerging Challenges in Financial Machine Learning

As the field of financial machine learning progresses, more difficulties and concerns emerge. De Prado (2018) conducts a comprehensive analysis of advanced machine learning applications in banking, emphasizing current limitations and potential developments. His research highlights many critical issues that scholars and practitioners must consider:

- The significance of high-performance computers in financial machine learning: As models get more intricate and data quantities expand, computing resources emerge as a constraining factor in model development and implementation.
- The need for appropriate data management and feature engineering: Financial data often necessitates sophisticated preprocessing and feature engineering for optimal use in machine learning models.

- The difficulty of formulating resilient backtesting methodologies: Ensuring optimal out-of-sample model performance and preventing overfitting necessitates advanced validation methodologies.
- The promise of other data sources: Non-traditional data offers significant insights while also presenting novel obstacles in data integration and interpretation.

De Prado's research underscores that the exponential increase of complexity in financial machine learning necessitates significant computing resources and specialized knowledge. The inclination towards more intricate models and expensive datasets is a substantial obstacle for several academics and smaller entities.

These constraints provide an opportunity to investigate alternative methods that preserve the advantages of sophisticated machine learning techniques while diminishing computing complexity and enhancing accessibility. The following sections will examine possible solutions to these issues, including the implementation of smaller, more specialized neural networks and strategy-specific methodologies.

2.5 Future Directions

2.5.1 Small Neural Networks for Improved Accessibility

Having explored the hurdles, let us pivot to potential solutions. We propose utilizing small neural networks for resolving issues related to computational resources in finance. By separating the whole market from particular trading strategies, we keep comparable outcomes without adding extra computational burden. This strategy-orientated approach simplifies data processing and also enhances small neural networks' sensitivity to individual trading strategies.

2.5.2 Adaptability of Small Neural Networks

The question whether tiny neural networks can adapt to dynamic financial contexts is crucial. We intend to investigate how to train these networks to adapt and meet changing market conditions, ensuring effectiveness in the long run.

This literature review prepares the groundwork for this dissertation by offering a nuanced perspective on the evolution of neural networks, their financial applications and the future paths suggested consistent with our hypothesis. The following chapters discuss the methodology, empirical outcomes and conclusions that support the overall objective of improving technical trading strategies in financial markets.

3. Data

The study conducted in this paper includes VIX (Volatility Index) data from a period of almost four years. The VIX, also referred to as the "fear index," is a crucial gauge of market expectations for near-term volatility, as reflected by the valuation of S&P 500 stock index options. The VIX dataset was collected from the Refinitiv database, covering the period from January 2, 2020, to August 31, 2023, ensuring a dependable and academic foundation for our research.

The data continuously tracks VIX values on a minute-by-minute basis, providing a full and thorough perspective on the dynamics of market volatility during the four-year duration. The VIX is highly beneficial for our research since it reflects the market's anticipation of volatility over the next 30 days. This makes it a critical indicator for comprehending market sentiment and probable future fluctuations.

The primary factors considered in this analysis are price and time. Time determines the sequence of events, and VIX prices accurately represent the prevailing volatility expectations at any given point. Our focus is on detecting and documenting dynamic changes in market volatility at the smallest level, as reflected by fluctuations in the VIX.

3.1 Data Transformation

A vital aspect of the research revolves around the transformation of the raw price data. Instead of utilizing standard pricing values, we chose to employ variations in prices. More precisely, we computed the five price changes that occurred right before each trade signal produced by our trading techniques, whether they were generated by individual methods or a combination of strategies.

This transformation was a critical decision in our methodology, driven by several key factors:

1. Normalization
2. Comparability
3. Feature Relevance
4. Model Performance

By utilizing VIX fluctuations rather than absolute numbers, all of these problems can be addressed concurrently. Variations inherently standardize the data, improving the ability to compare across different time periods and market conditions. This approach captures the volatile changes in market sentiment, which might offer more important insights for making trading decisions in comparison to fixed VIX levels.

Our algorithms primarily concentrate on analyzing oscillations in the VIX, rather than its exact numerical value. This approach enables them to get valuable understanding of patterns that may be used in many market volatility conditions. It is essential to capture significant market dynamics and trends that impact trading decisions. The relative nature of changes in the VIX enables more accurate comparisons between different trading instances, irrespective of the particular levels of the VIX at any given moment.

Furthermore, this conversion was crucial in order to generate significant outcomes from our models. The models might concentrate on the patterns of fluctuation in market volatility, which are frequently more indicative of future market trends than the actual levels themselves. This method improves the models' capacity to adjust to various market regimes and conditions, perhaps resulting in more resilient and flexible trading strategies.

In some scenarios, we explored additional techniques for normalizing, but concluded that the price variations alone were sufficient to achieve the appropriate level of normalization.

4. Methodology

Our approach integrates well-established technical trading strategies with advanced machine learning techniques, including deep learning and reinforcement learning neural networks. The combination of both methodologies seeks to use the benefits of both conventional market insights and sophisticated machine learning techniques.

4.1. Trading Techniques Approaches

4.1.1. Moving Average Convergence Divergence (MACD)

Utilizing long and short price windows (12 and 26) and a signal price window of nine, MACD recognizes buying opportunities whenever the MACD line crosses above the signal line. Exit signals show up whenever the MACD line dips beneath the signal line.

4.1.2. Bollinger Bands

The closing price is in or bellow the lower Bollinger Band that has a price average of 20 prices and four standard deviations – a buy signal. The exit conditions occur once the closing price crosses the upper band.

4.1.3. Relative Strength Index (RSI)

RSI measures a security's strength using a price window of 14 and upper/lower thresholds of 80 and 20. Whenever RSI crosses the lower bound, it identifies buy signals. Once RSI gets to the higher limit from bellow, then it meets exit requirements.

4.1.4. Moving Average

Simplifying market trends with a 50 prices window and a 200 prices window, the Moving average strategy triggers buy signals when the short term average crosses above the long term average. Exit signals take place whenever the short term average reaches the long term average again.

4.1.5. Stochastic Oscillator

Using a K period of 14 prices along with a D period of three, the Stochastic Oscillator locates buying opportunities when %K line is above %D. Exit signals will occur whenever the %K line is below the %D line.

4.2. Neural Network and Machine Learning Integration

Our methodology integrates both deep learning neural networks and traditional machine learning algorithms, augmented by reinforcement learning techniques. The inputs to our models consist of the five price variations preceding each entry point identified by our trading strategies. This applies to both individual strategies and all possible combinations of strategies.

Both deep learning models and typical machine learning algorithms use binary profit labels as the intended result.

- 0 indicates a negative profit (loss)
- 1 indicates a positive profit (gain)

Our models may effectively learn patterns in price changes that are linked to winning trades, irrespective of the particular method used, using this binary classification technique. By using a uniform target variable across diverse model types, we may facilitate the comparison and assessment of the effectiveness of different methodologies.

Utilizing binary profit labels provides several benefits:

1. The issue space is simplified, enabling models to concentrate on the essential subject of profit vs loss.
2. Mitigation of the influence of extreme values that may arise with continuous profit values.
3. Enhanced comprehension of model results, particularly beneficial for investors with little familiarity

4.2.1 Data Split and Model Evaluation

To guarantee robust model evaluation and simulate live trading scenarios, we performed a careful train/test split of our dataset. The data was categorized into two groups:

- Training set: 75% of the data
- Test set: 25% of the data

We implemented methods to preserve the same ratio of wins and losses in both the test and the training sets for every strategy. This approach ensures that the probability of incurring a loss is almost uniform across each group, therefore mitigating biases in our model evaluation. The primary group (75%) serves as our training dataset, used for model training and improvement. The smaller fraction (25%) is classified as unsee data, used for model classification.

Our deep learning models use this binary classification problem to allow the neural networks to acquire knowledge of intricate, non-linear connections within the data that may not be easily discernible using conventional analytic methods.

The binary classification technique offers advantages to our conventional machine learning algorithms, such as decision trees, random forests, or support vector machines. These algorithms frequently produce results that are more comprehensible, offering significant insights into the primary components, namely the five price variations, that have the most influence on predicting successful transactions.

Furthermore, we use reinforcement learning methods in addition to the aforementioned supervised learning methodologies. In reinforcement learning, the real profit serves as the reward signal, enabling the model to acquire optimum trading behavior by means of trial and error. This technique has the ability to capture more intricate trading strategies that adjust to dynamic market circumstances.

By using price fluctuations instead of absolute prices throughout our whole range of models, we allow users to get insights from relative changes in the market rather than exact price levels. This strategy improves the models' capacity to generalize across many market circumstances and time periods.

This comprehensive solution utilizes the predictive capabilities of several machine learning algorithms to validate and enhance trading performance, relying on the signals produced by our technical indicators. Using binary profit labels consistently across several model types facilitates comparison and enables the use of ensemble approaches, which may enhance the overall reliability of forecasts.

Our objective is to rethink technical trading tactics in financial markets by combining existing indicators with new modeling approaches and concentrating on price fluctuations to identify potential synergies. Our technique focuses on comparing price trends rather than looking at specific prices. We also employ binary profit labels, which is a significant novelty. This strategy has the potential to create more reliable and widely applicable trading models.

4.3 Trade Distribution Across Strategies

Prior to analyzing the performance of the models, it is essential to comprehend the distribution of trades among different strategies. Figure 1 shows the transaction count for each strategy dataset.

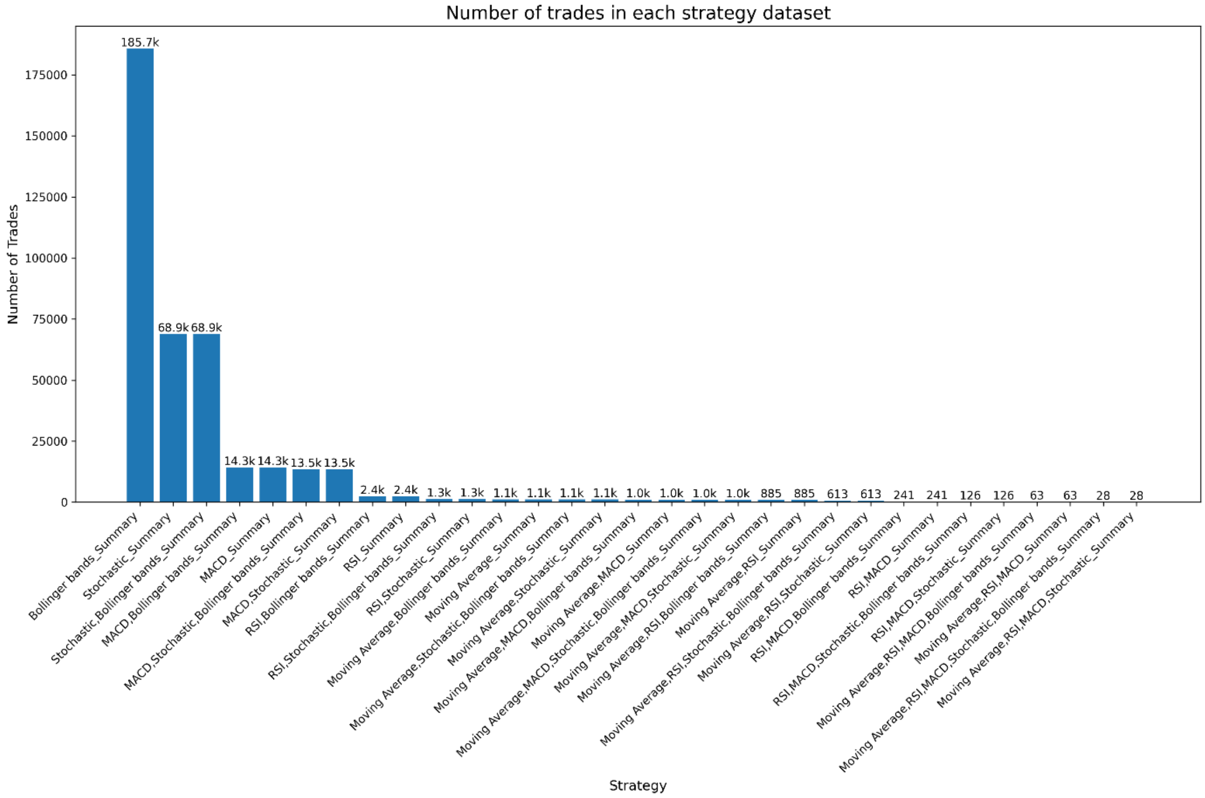

Figure 1: Number of trades in each strategy dataset

Key observations about the distribution of trades:

1. High Volume strategies: The Bollinger Bands strategy exhibits the largest trading volume, totaling 185.7k transactions, which is notably more than other techniques. Following this strategy are Stochastic (68.9k) and MACD Bollinger bands (68.9k) strategies.
2. Low Volume Strategies: Various combination strategies, including Moving Average RSI MACD Stochastic and RSI MACD Stochastic, exhibit very low levels of trading activity, with just 28 transactions executed for each strategy.
3. The distribution exhibits a significant skew, where a few number of strategies dominate in terms of trading volume, while a large number of combined strategies have very little trade activity.

The trade distribution has significant ramifications for the model's performance, as we will see in the following research.

5. Results

5.1 Machine Learning Results

5.1.1 Baseline Model: LinearSVC Performance

We designated a Linear Support Vector Classifier (LinearSVC) as the foundational model for our study. This establishes a foundation for comparison with sophisticated deep learning and reinforcement learning models, which are central to the thesis. Two datasets were analyzed: one using absolute prices and the other utilizing relative prices, as seen in Figures 2 and 3, respectively.

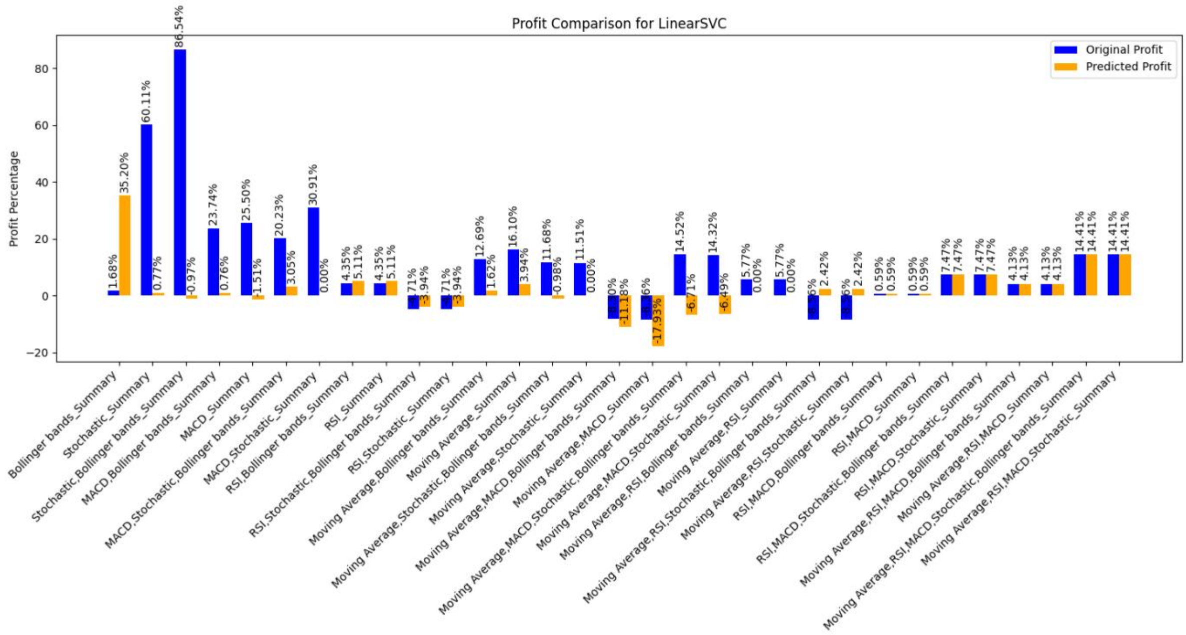

Figure 2: Results LinearSVC (Absolute Prices)

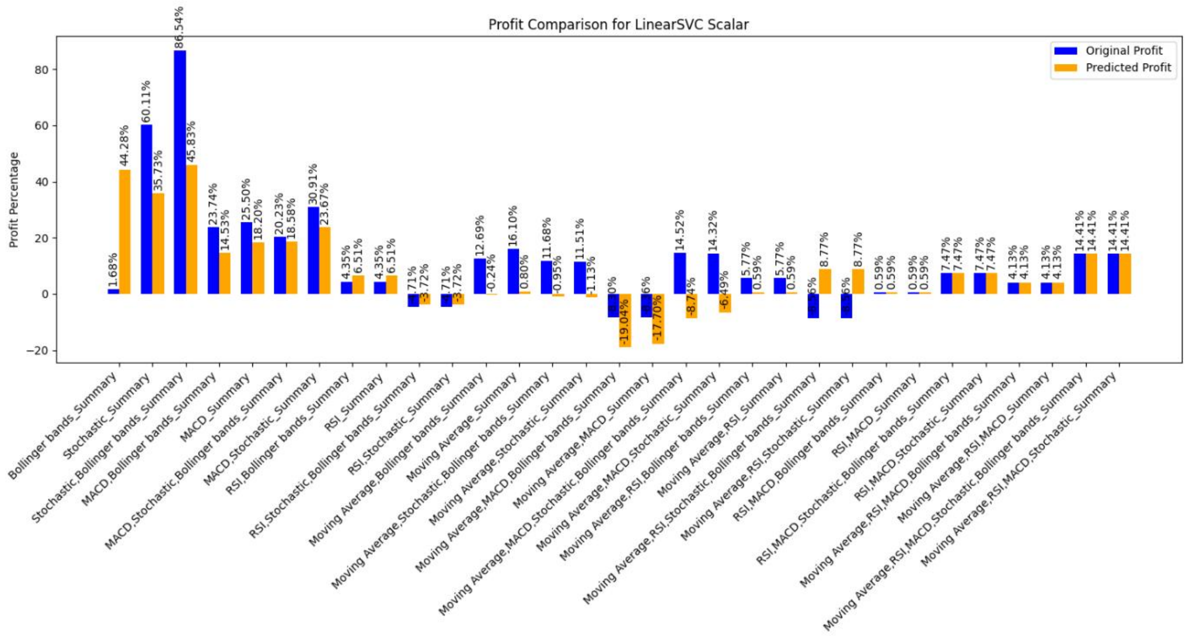

Figure 3: Results LinearSVC (Relative Prices)

The LinearSVC model has variable performance across different trading strategies, underscoring its advantages and limitations as a predictive tool in financial markets. The absolute and relative pricing variations of the model exhibit substantial discrepancies in expected earnings relative to the original profits, suggesting that the model aims to discern patterns rather than just mimic the original methodologies.

The influence of transaction volume on the model's efficacy is critical. The model demonstrates exceptional outperformance for high-volume strategies, shown by the Bollinger Bands approach with 185.7k transactions. The absolute price model forecasts a profit of 35.20%, in contrast to the initial 1.68%, however the relative price model anticipates an even greater profit of 44.28%. This significant improvement indicates that enough data enables the model to learn efficiently and maybe recognize more lucrative trading opportunities than the initial approach alone.

In contrast, for low-volume strategies like the Moving Average RSI MACD Stochastic, which comprises just 28 trades, the model's projections precisely align with the original profits, yielding 7.47% for both iterations. This indicates that with little data, the model struggles to identify meaningful patterns and defaults to always anticipating a buy signal, resulting in an identical reward to the original strategy. This finding highlights a notable deficiency of the LinearSVC model in managing sparse datasets.

The combination of absolute and relative pricing models reveals compelling patterns. The relative pricing model often produces more profitable predictions than the absolute price model, suggesting that this change may improve certain trends.

Instances in which the model predicts favorable outcomes for previously unprofitable methods are especially significant. A significant instance is seen in the center of the graph, when a method demonstrates an initial profit of -8.56%. The LinearSVC model enhances this to a 2.42% profit in the absolute price variant and a more substantial 8.77% profit in the relative price variant. This scenario illustrates the model's capacity to discern lucrative patterns despite the initial plan's failure, with the relative pricing variant being especially adept at converting a losing approach into a potentially profitable one.

The findings from our baseline LinearSVC model[1] provide significant insights into the correlation between data volume, data preparation (relative price transformation), and predictive accuracy. The model's ability to significantly exceed original strategies in many cases, while facing challenges in others, highlights both the potential and limitations of this approach.

The discoveries lay the path for exploring more advanced machine learning algorithms. These sophisticated models may be more proficient in recognizing complex patterns in high-volume approaches. Furthermore, they may provide enhanced reliability across various trading strategies and market conditions, alleviating some constraints seen in the LinearSVC model.

5.1.2 Decision Tree Classifier Performance

After analyzing the LinearSVC model, we developed a Decision Tree Classifier to evaluate its efficacy in forecasting successful trades across different strategies. This methodology seeks to elucidate non-linear interactions and provide more comprehensible outcomes than the prior paradigm.

We examined two iterations of the Decision Tree model: one using absolute prices and the other utilizing relative pricing. Figures 4 and 5 represent the profit comparisons for these approaches, respectively.

[1] We further conducted experiments using the conventional Support Vector Classification (SVC) model. The findings achieved were virtually identical to those of Linear SVC. To eliminate repetition in our study and presentation, we have concentrated on providing the Linear SVC findings as indicative of the performance of both models.

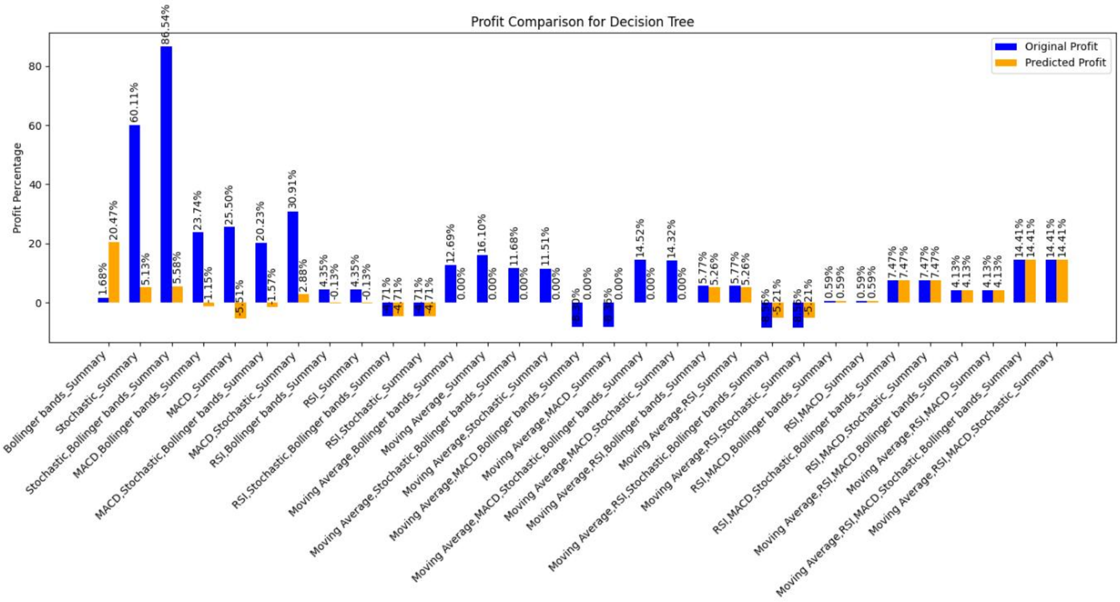

Figure 4: Results Decision Tree (Absolute Prices)

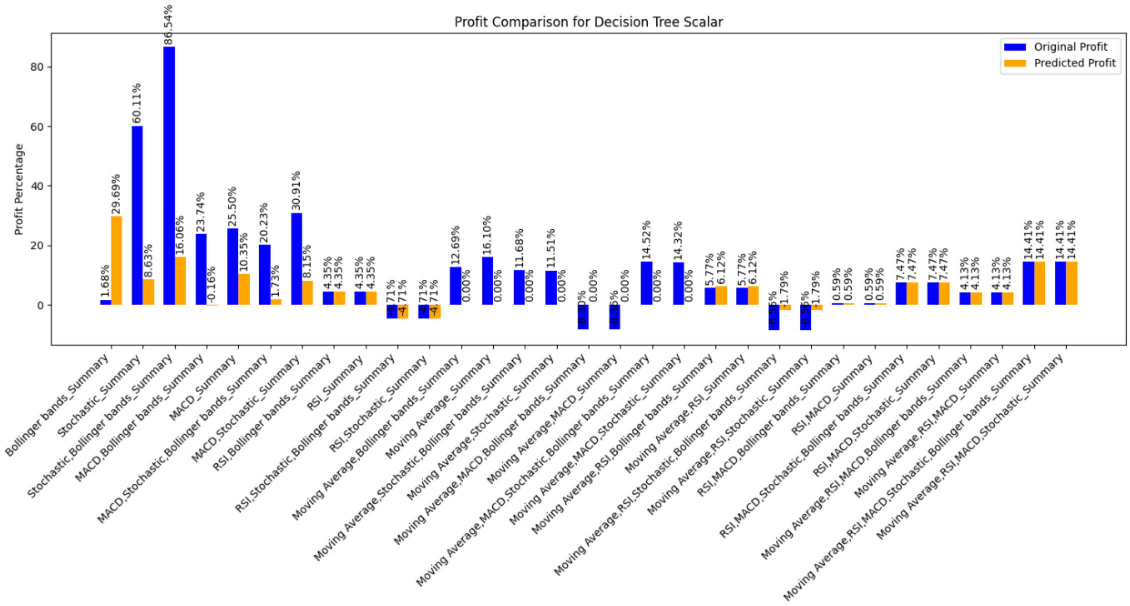

Figure 5: Results Decision Tree (Relative Prices)

The Decision Tree model exhibits diverse performance across various trading methods, highlighting both possible benefits and drawbacks in comparison to the LinearSVC model. The most notable finding is the inconsistency in its forecasts, as the model surpasses the original approach in some instances while underperforming in others.

The Decision Tree demonstrates favorable outcomes for high-volume strategies, namely the Bollinger Bands method, which encompasses 185.7k transactions. The absolute price version forecasts a profit of 20.47%, in contrast to the initial 1.68%, whilst the relative price version anticipates 29.69%. This significant enhancement of the original technique indicates that the model has identified potentially lucrative patterns within this data-intensive approach. It is important to observe that these forecasts are inferior than those of the LinearSVC model (35.20% for absolute prices and 44.28% for relative prices), suggesting that the Decision Tree may choose a more cautious approach in its estimations for this high-volume strategy.

The model's performance on strategies with modest trade volumes (ranging from 1,000 to 100,000 deals) is inconsistent. The MACD approach, including 68.9k transactions, forecasts a profit of 2.88% in the absolute price variant and 8.15% in the relative price variant, comparing to the initial 30.91%. This underperformance indicates that the Decision Tree fails to encapsulate the whole intricacy of this strategy's patterns.

For low-volume strategies (fewer than 1,000 transactions), the Decision Tree model often forecasts either full market entry for each trade (aligning with the original strategy's profit) or complete non-entry (resulting in 0% profit). This trend is especially apparent here, as several low-volume techniques have a projected profit of 100%. This indicates that the model is unable to extract significant patterns from little data and resorts to extreme judgments instead.

The relative pricing variant of the Decision Tree model demonstrates distinct learning skills for certain strategies, similar to previous observations. In the Moving Average Bollinger Bands method, a profit of 10.33% is anticipated, compared to the initial 12.69%, whilst the absolute price version forecasts 0%. This indicates that the Decision Tree may more effectively identify certain patterns in the processed data for certain strategies, a proficiency also seen in the LinearSVC model.

The findings from the Decision Tree model provide significant insights into its advantages and constraints in forecasting winning trades across several strategies. Although it demonstrates potential in specific domains, particularly with processed data for targeted strategies and enhancing certain unprofitable strategies, its erratic performance and difficulties with low-volume and highly profitable strategies suggest that additional refinement or alternative methodologies may be required.

The findings indicate that while Decision Trees provide interpretability and can describe non-linear interactions, they may not consistently surpass simpler models such as LinearSVC in this financial context. This establishes a foundation for investigating more sophisticated ensemble methods or machine learning techniques that might integrate the advantages of several approaches to get more dependable and uniform forecasts across all trading contexts.

5.1.3 Random Forest Classifier Performance

After analyzing the Decision Tree model, we used a Random Forest Classifier to assess its efficacy in forecasting successful trades across different strategies. Random Forests, as an ensemble approach, may mitigate some constraints of individual Decision Trees by consolidating the forecasts of numerous trees.

We examined two iterations of the Random Forest model: one using absolute pricing and the other utilizing relative prices. Figures 6 and 7 depict the profit comparisons for these methodologies, respectively.

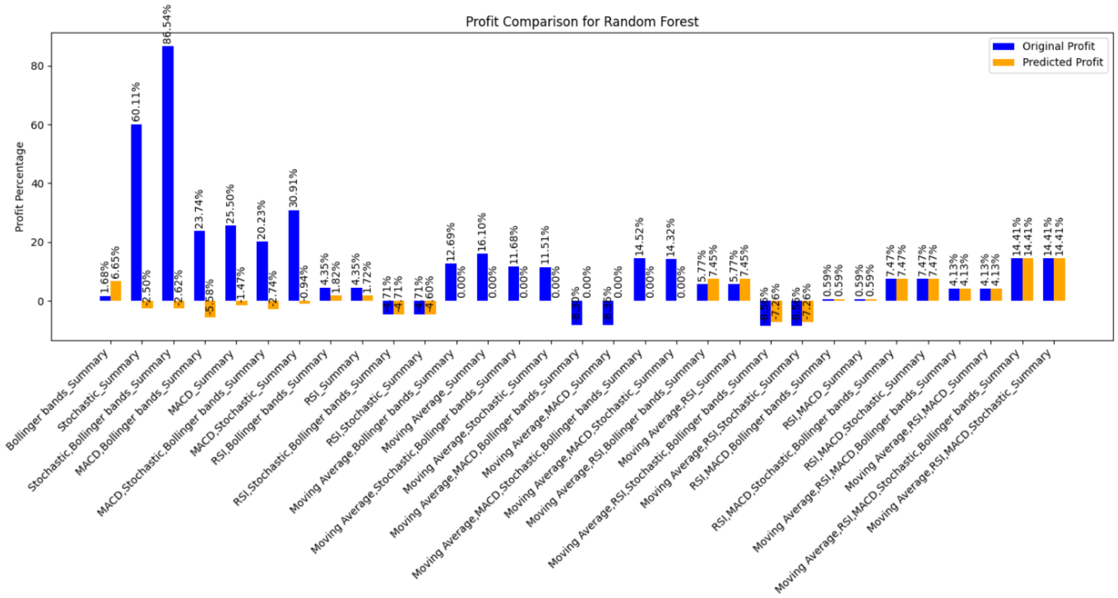

Figure 6: Results Random Forest (Absolute Prices)

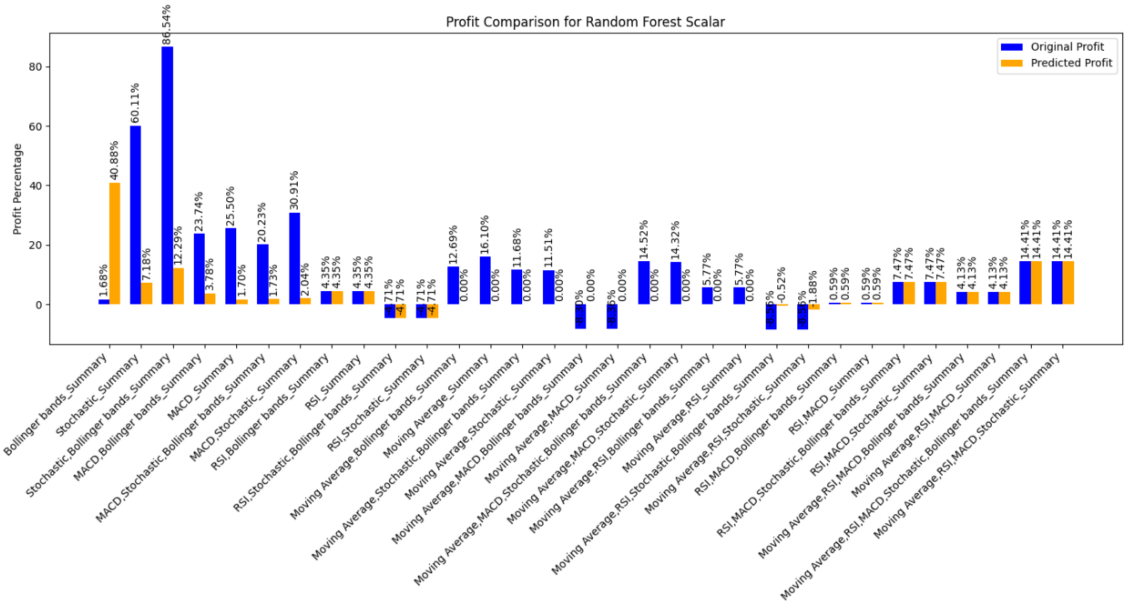

Figure 7: Results Random Forest (Relative Prices)

The Random Forest model has heterogeneous efficacy across various trading techniques, indicating both advancements and ongoing difficulties relative to prior models.

For high-volume strategies, particularly the Bollinger Bands strategy with 185.7k trades, the Random Forest shows promising results. In the absolute price version, it predicts a profit of 6.65% compared to the original 1.68%, while the relative price version predicts a remarkable 40.88%. The significant enhancement compared to the original approach and the Decision Tree indicates that the Random Forest has discerned possibly more lucrative patterns inside this data-intensive technique. The remarkable outperformance of the relative pricing variant is especially notable, suggesting that data normalization may be essential to the model's efficacy for this approach.

For low-volume strategies (fewer than 1,000 trades), the Random Forest model often predicts either 0% profit or matches the original profit. This behavior, consistent with previous models, suggests that the Random Forest also struggles to learn meaningful patterns from limited data, essentially defaulting to always entering or never entering the market for these strategies.

A significant finding is the continual superiority of the relative price version compared to the absolute price version across almost all strategies. We have never observed such a distinct advantage of data processing before.

The insights derived from the Random Forest model elucidate its strengths and limits in forecasting winning transactions. Its enhanced performance on particular strategies, especially with processed data, indicates it may be an important benefit to a model ensemble. Nonetheless, its variable effectiveness across different strategies and ongoing difficulties with low-volume strategies suggest that more refining or other methods may be required.

5.1.4 Overview

In the realm of high-frequency trading methods, namely the Bollinger Bands approach including 185.7k deals, all algorithms exhibited favorable outcomes. The LinearSVC exhibited superior performance for this particular strategy and consistently beat other algorithms over an expanded variety of strategies.

Figure 8 demonstrates that all models, which did not exclusively predict all buy or never buy, exhibited substantial improvements when used with preprocessed data. These findings underscore the essential function of data preparation in improving model efficacy.

These results provide a foundation for exploring more sophisticated machine learning approaches, including deep learning models, which may possess enhanced capabilities to identify intricate patterns across diverse trading strategies and data quantities.

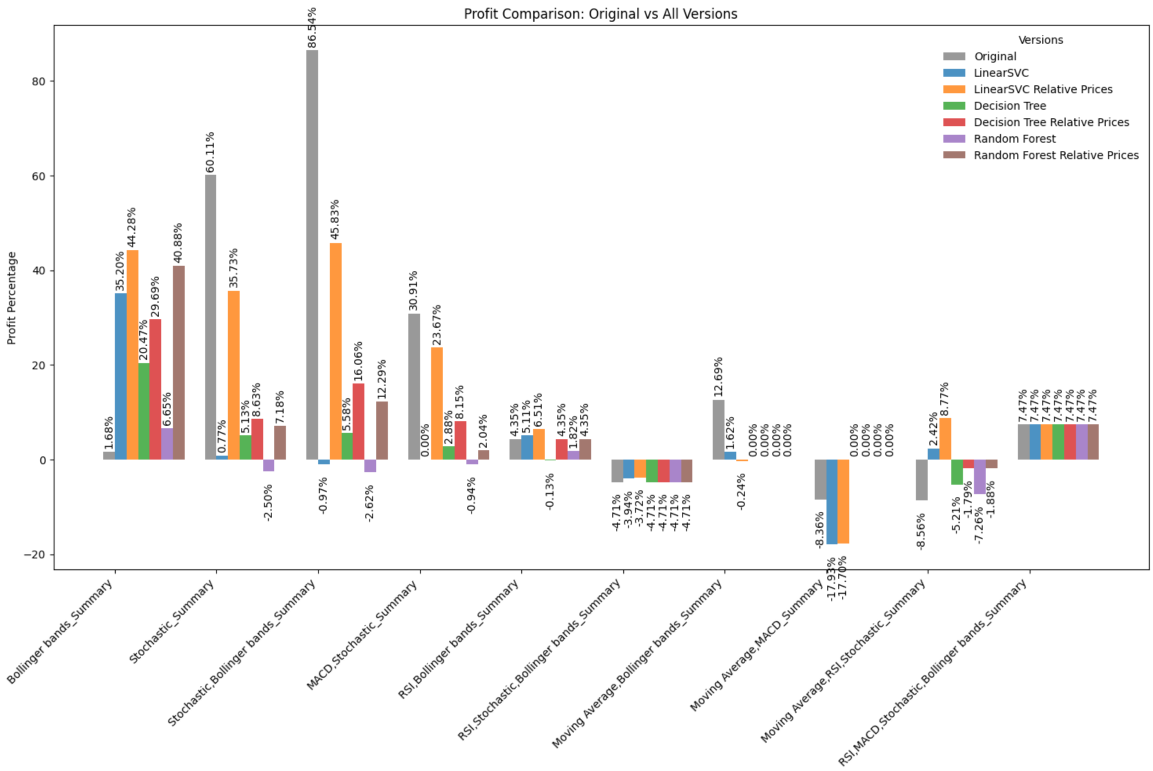

Figure 8: Machine Learning Results Comparison

5.2 Deep Learning Results

This study investigated three distinct output modifications to improve the effectiveness of our deep learning models in predicting profitable trades. The model outputs underwent modifications to enhance decision-making accuracy and the effectiveness of the trading strategy. We refined our process by including these output adjustments, based on our previous work using absolute and relative price data. For each approach, we employed four different architectures: a small one, a medium one, a larger one, and the biggest one. This variety in model sizes allows us to examine the impact of model complexity on performance across different output modifications. It is essential to recognize that all these networks are quite small inside the larger framework of deep learning; even our "biggest" architecture would be considered as little in several other applications. The size descriptors are relative in our work, enabling the assessment of the effects of incremental changes in model complexity on performance across various output variations, while ensuring computational efficiency.

The first transformation used was a binary profit categorization. This method included discretizing the model's output into two values: 1 for successful transactions and 0 for unsuccessful ones. This binary categorization streamlines decision-making, enabling the model to concentrate on differentiating between potentially lucrative and unprofitable interactions, regardless of the extent of profit or loss.

The second transformation normalized all returns to a range of 0 to 1. The normalization procedure was executed using a customized program intended to standardize the raw profit data. The result signifies a confidence score, where higher numbers indicate a greater probability of a lucrative deal. An output of 0.8 indicates that the network forecasts an 80% likelihood of the transaction generating a profit. This method offers a more refined perspective on the model's predictions, facilitating more precise decision-making grounded on the confidence level about the profitability of each transaction.

The third strategy used unrefined profit outputs from the neural network. This technique entails the model clearly predicting the expected profit or loss for each potential transaction. The decision-making process depends on the sign of the expected value: positive forecasts trigger a buy signal, while negative predictions result in action. This approach preserves extensive data on expected profitability, allowing more sophisticated trading strategies that include both the direction and magnitude of predicted returns.

Every output alteration has distinct advantages in the domain of algorithmic trading. The binary profit transformation provides clear and practical insights but may excessively simplify complex market dynamics. The normalized returns provide a blend of simplicity and complexity, aiding in risk-adjusted decision-making. The raw profit outputs preserve the maximum information but may be more susceptible to noise and outliers in the training data.

It is crucial to highlight that these output alterations were implemented on models trained with both absolute and relative price data, as previously stated. This combination of input data transformations (absolute vs relative pricing) and output transformations enables a thorough study of various modeling methodologies.

We will do a comparative analysis of these three approaches, evaluating their effectiveness in various market conditions and trading strategies. This research will clarify the ideal output transformation for various trading algorithms and market conditions, while evaluating the effects of using absolute vs relative price data. This thorough study seeks to determine the optimal mix of input and output adjustments to improve the effectiveness of our deep learning models in algorithmic trading.

5.2.1 Binary Profits Aproach

This section examines the first output transformation analyzed in our study: binary profit categorization. This method involves teaching the model to forecast a binary result, with 1 denoting lucrative transactions and 0 indicating non-profitable trades. We implemented this transformation on models trained with both absolute and relative price data, extending our prior research with conventional machine learning methods, including Support Vector Classifiers, Decision Trees, and Random Forests. The outcomes of this binary classification method demonstrated notable variations in model efficacy based on the kind of input data used.

The model's performance was significantly inadequate while using absolute prices as input (Fig. 9). The prediction pattern demonstrated a uniform binary behavior, with the model consistently predicting either a purchase or a non-purchase across all techniques. This consistent forecast pattern suggests that the algorithm did not acquire any significant insights from the absolute price data. Despite rare minor discrepancies between the estimated profit and the original strategy's profit, these variations were not statistically significant. This conclusion indicates that the binary classification job based on absolute prices was too difficult for the algorithm to identify actionable patterns, irrespective of the trading strategy used.

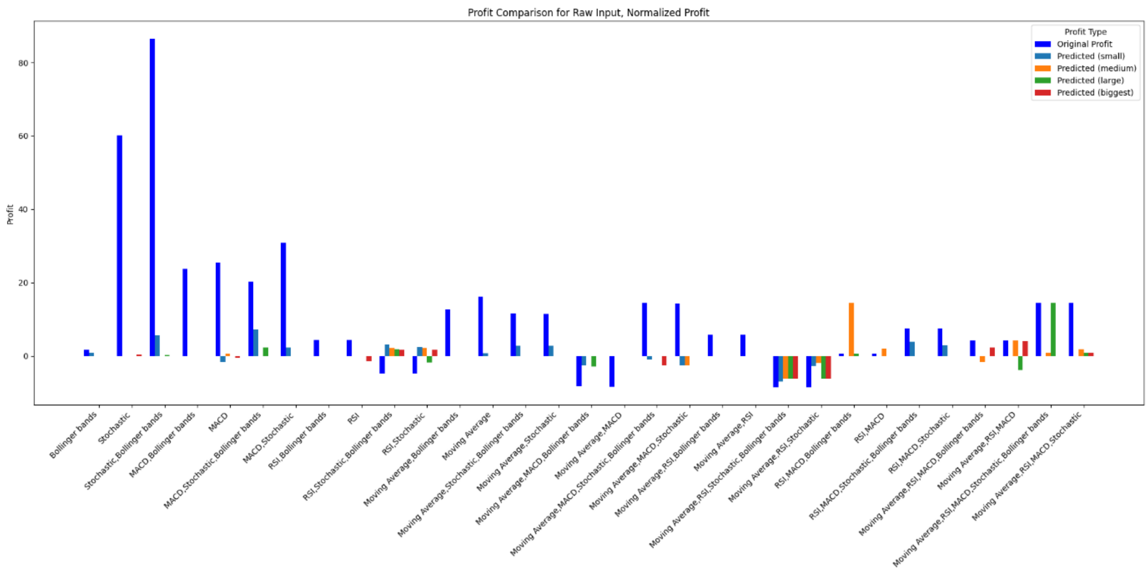

Figure 9: Results Deep Learning with Binary Profit (Absolute Prices)

Conversely, using relative pricing as an input yielded more favorable and varied outcomes (Fig.10). The model had the capacity to discern patterns and provide predictions that deviated from the actual data in a significant way. This enhancement signifies that relative pricing data offers more meaningful attributes for the binary classification problem, enabling the model to discern pertinent market dynamics. Nonetheless, it is crucial to acknowledge that despite this enhancement, the performance of the model with the greatest number of trades (presumably the Bollinger Bands strategy, according to our prior analysis) still did not meet the outcomes attained by the conventional machine learning algorithms examined earlier in our research.

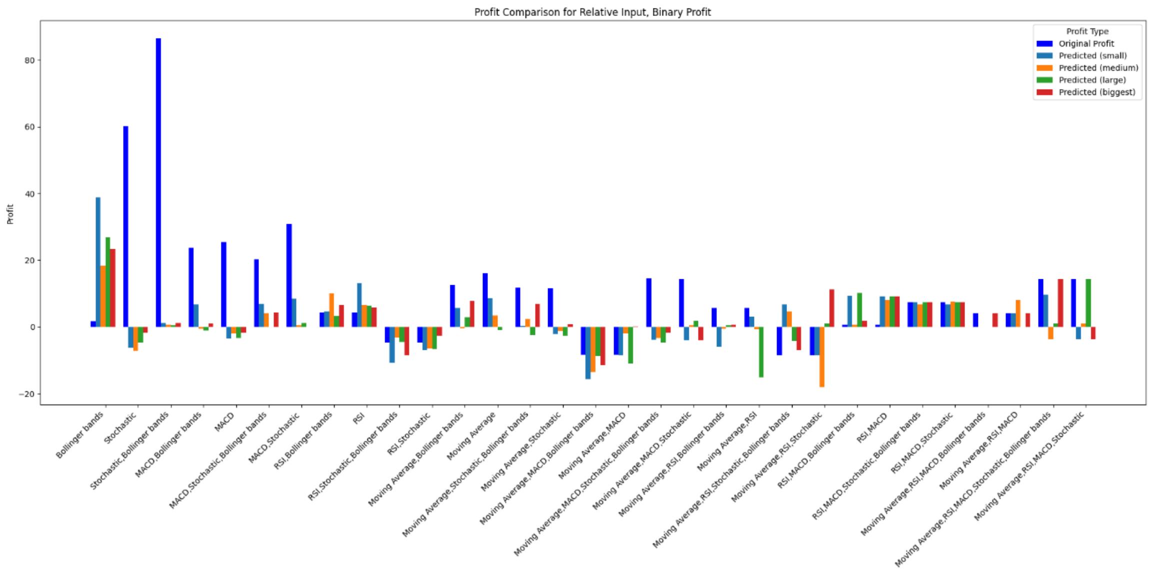

Figure 10: Results Deep Learning with Binary Profit (Relative Prices)

The performance of the model varied significantly across different trading strategies when using relative price inputs. For the strategy with the highest number of trades, the model achieved relatively good results across all tested architectures, although still not matching the performance of our previous machine learning models. This relative performance in high-volume contexts indicates that the model gains from an extensive dataset, enhancing its ability to learn and generalize patterns. The plethora of data points in high-frequency trading techniques seems to provide the neural network with enhanced possibilities to discern repeating patterns and generate more knowledgeable forecasts.

The following six strategies regarding transaction volume exhibited inconsistent and often unsatisfactory outcomes. The model found it challenging to continuously exceed or even replicate the performance of the original strategy for these medium-volume trading methods. This discrepancy may suggest that these methods function in a more intricate or noise-affected segment of the market, complicating the model's ability to identify dependable patterns. The reduced frequency of transactions in these strategies may constrain the model's capacity to learn successfully, since it has fewer instances for training relative to the high-volume approach.

These results underscore the critical importance of data representation in machine learning models for financial forecasting applications. The significant disparity between the performance of absolute and relative price inputs highlights the essential importance of feature engineering in financial modeling. Relative prices seem to provide more relevant information for the binary classification job, facilitating the model's ability to discern significant patterns, particularly in high-volume trading contexts. This finding corresponds with financial theory, which often emphasizes the significance of price fluctuations and returns above absolute price levels in forecasting market dynamics.

The findings from this binary classification method provide vital insights, although they constitute just one facet of our comprehensive examination of neural network-based trading methods. In the next section of this research, we will examine another output transformation and perform further tests to get a more thorough knowledge of the model's capabilities and limits.

The outcomes of this binary output transformation highlight the intricacy of financial forecasting jobs and the need for meticulous evaluation of input data representation and model architecture in formulating efficient algorithmic trading strategies. We will use these insights to enhance the development of more robust and adaptable neural network models for financial prediction and trading.

5.2.2 Normalized Profits Aproach

Following our analysis of binary output modifications, we further evaluated the efficacy of a normalized profit methodology. This strategy entails normalizing profit numbers to a range of 0 to 1, so enabling the model to discern more intricate correlations between market circumstances and trading results. Consistent with our prior work, we used this methodology on both absolute and relative price inputs throughout our spectrum of trading methods.

Unexpectedly, the outcomes of the normalized profit strategy were markedly inferior to our previous results obtained by the binary classification method. When using absolute price data as input, the model's performance remained subpar, exhibiting no improvement relative to the binary classification method (Fig. 11). The anticipated earnings were in line with the performance of the original approach, exhibiting no variation or indication of pattern recognition. The persistent underperformance across several input representations underscores the difficulty of deriving significant prediction characteristics from absolute price data inside our neural network models.

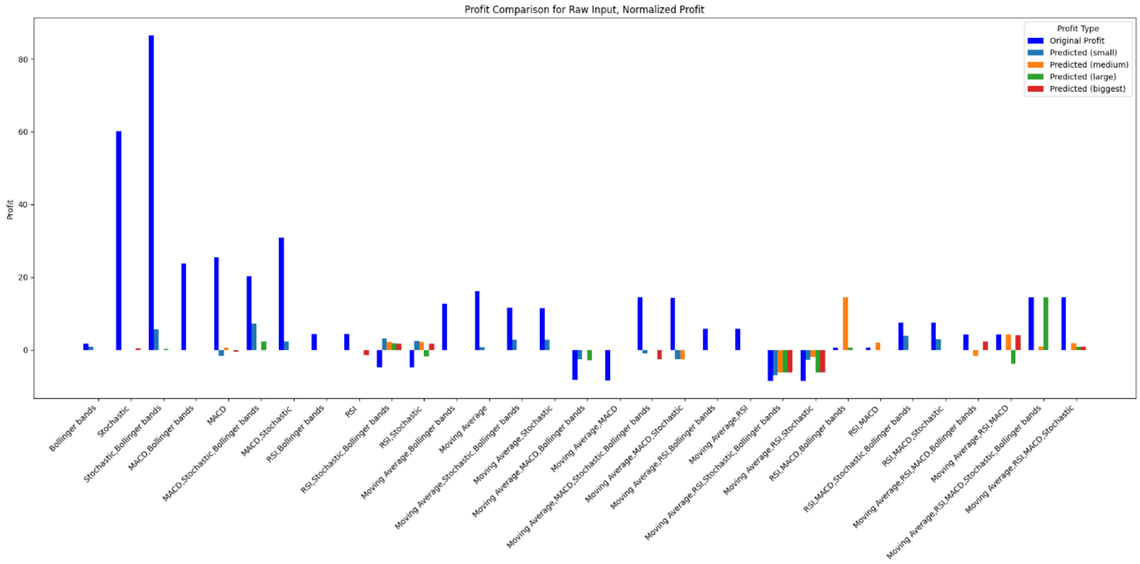

Figure 11: Results Deep Learning with Normalized Profit (Absolute Prices)

Unexpectedly, the use of relative price inputs inside the normalized profit framework did not enhance outcomes compared to the absolute price model and significantly underperformed comparing to the binary classification technique (Fig.12). The model exhibited minor fluctuations in its predictions relative to the absolute price input; however, these discrepancies were minor and did not reliably identify lucrative trading opportunities. The relative price model sometimes sought to deviate from just mirroring the original strategy's performance or remaining inactive; nevertheless, these efforts did not provide significant enhancements in trading results.

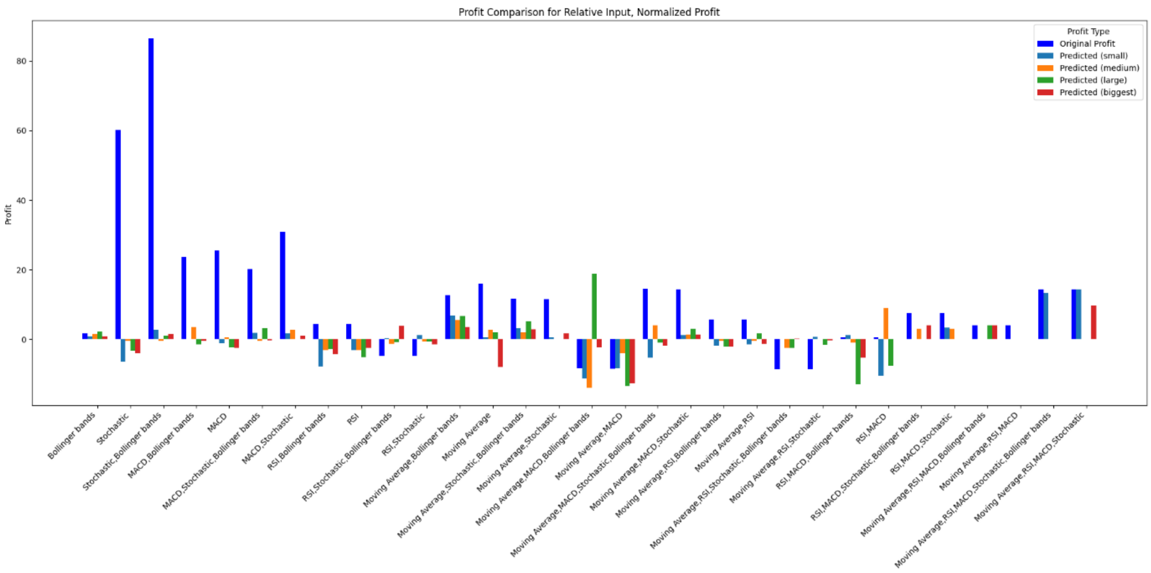

Figure 12: Results Deep Learning with Normalized Profit (Relative Prices)

This unexpected outcome challenges our preliminary premise that normalizing profits would provide the model with a more elucidative goal variable, particularly when integrated with relative price inputs. The analogous poor performance of absolute and relative price inputs indicates that the normalization step may have unintentionally concealed some important characteristics inherent in the relative price data that were advantageous for the binary classification method.

Multiple variables may account for this considerable underperformance. An essential realization, which we first disregarded, is that the average of the gains is likely not zero in the majority of trading situations. Thus, when adjusted to a scale of 0 to 1, the average of the normalized earnings is not centered at 0.5. This skewed distribution presents a considerable problem for the model in discerning the boundary between lucrative and bad deals. In the absence of a distinct, centralized reference point, the model has difficulties in establishing a coherent decision boundary for trade execution.

The normalization approach may have constrained the range of profit values, impeding the model's capacity to distinguish between marginally profitable and highly successful transactions. The persisting feature of the normalized output may have introduced complexity to the learning process, hindering the model's capacity to delineate clear choice boundaries.

These results highlight the intricacy of feature engineering and output representation in financial forecasting jobs. Although normalization is frequently advantageous in machine learning applications, its implementation in profit prediction within this context seems to have significantly constrained the model's capacity to identify valuable patterns, despite the use of relative price inputs that demonstrated potential in our binary classification method.

Despite the encountered obstacles these results provide significant insights into the behavior of neural networks in financial prediction tasks and underscore the need for meticulous analysis of both input and output representations. The subsequent part of our study will concentrate on raw profit outputs, perhaps providing an alternative viewpoint on the model's capacity to identify and use market inefficiencies, devoid of the complexities brought by normalization.

5.2.3 Raw Profits Aproach

The concluding phase of our Deep Learning investigation included using raw earnings as the output for our neural network models. This approach, which avoids the transformations used in our prior tests, produced the most encouraging results to yet, strongly corroborating our original hypothesis and giving significant insights into the capabilities of neural networks in algorithmic trading.

Utilizing absolute prices as input revealed a significant deviation from the performance trends reported in our prior methodologies (Fig. 13). In our first deep learning trials, the model exhibited a distinct effort to identify and use patterns within the data. The anticipated profit distribution was centered around the original profit, demonstrating that the model effectively captured the overarching pattern of the trading methods. This significant improvement indicates that the raw profit methodology fosters a more favorable setting for the model to acquire knowledge and provide substantive predictions, even when using absolute price data that had previously produced unsatisfactory outcomes.

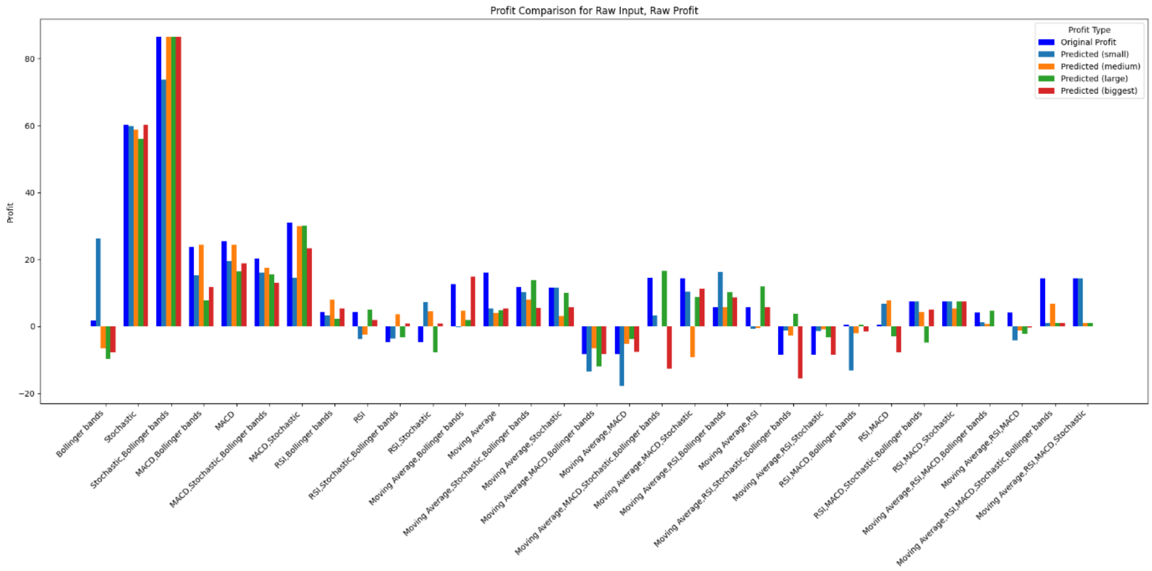

Figure 13: Results Deep Learning with Raw Profit (Absolute Prices)

The relative price input, together with the raw profit output, demonstrated the most consistent and promising performance throughout our studies. In this arrangement, we noted that for almost every trading strategy, at least one network design surpassed the original strategy in profitability (Fig. 14). This substantial enhancement evidences a marked progression in corroboration of our theory about the capacity of neural networks to augment trading strategies.

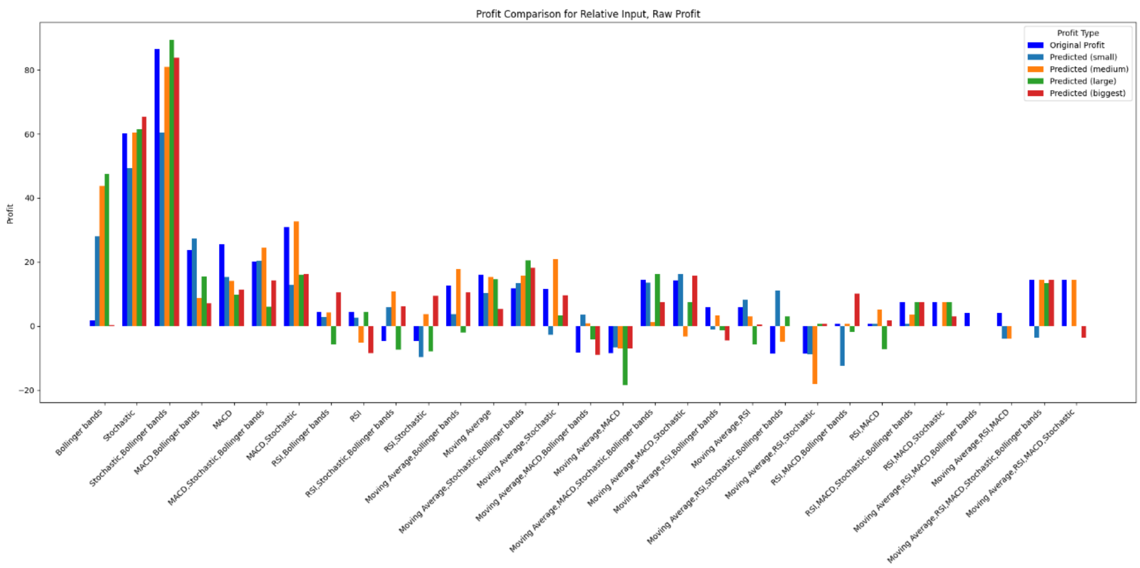

Figure 14: Results Deep Learning with Raw Profit (Relative Prices)

The effectiveness of the relative pricing input in conjunction with raw profit output may be attributed to many aspects. Initially, using relative pricing probably provides the model with more pertinent characteristics for forecasting price fluctuations, as it captures the dynamism of price alterations rather than fixed values. Secondly, the raw profit output retains comprehensive information on trade outcomes, enabling the model to discern more intricate linkages between market circumstances and prospective earnings.

It is important to note that for strategies with a limited number of transactions, we did not detect substantial improvements. This is a recognized constraint of deep learning models, which often need significant quantities of data for successful learning. The absence of performance enhancement in low-volume methods is a natural outcome of the data prerequisites of these models, rather than a shortcoming of the methodology itself.

The efficacy of the raw profit methodology, especially with relative pricing inputs, has significant consequences for our study.

- It confirms our investigation into various input and output representations, illustrating that the selection of data format may significantly affect model performance.
- It indicates that, when well constructed, neural networks may effectively identify and use market inefficiencies, perhaps surpassing conventional trading strategies.
- It underscores the significance of maintaining the whole spectrum of profit information in the output, rather than reducing it via binary categorization or normalization.
- It highlights the possibility of relative price data as a more relevant input for financial forecasting tasks than absolute prices.

The efficacy of this strategy in high-volume tactics, contrasted with its constraints in low-volume contexts, indicates an essential data volume threshold for successful learning. Establishing this threshold and examining strategies to enhance performance in low-volume situations may represent significant avenues for research.

In conclusion, the raw profit methodology, particularly when integrated with relative price inputs, signifies a substantial advancement in our investigation of deep learning for algorithmic trading. Despite ongoing obstacles especially in low-volume strategies, these findings provide compelling evidence for the capacity of neural networks to improve trading performance and establish a robust basis for more study and improvement of these models.

5.2.4 Overview

This section provides a comparative examination of the performance of deep learning models relative to traditional trading strategies and machine learning methods, concentrating on the seven most commonly traded strategies shown in Figure 15. The research indicates that in six of the seven methods evaluated, at least one deep learning architecture exhibited enhanced performance relative to both the original approach and the best machine learning model. This constant outperformance indicates an improved ability of deep learning algorithms to identify intricate financial patterns. Performance variability was discovered across several deep learning architectures (small, medium, large, and biggest), highlighting the significance of architecture selection in strategy optimization. High-volume strategies, especially those using Bollinger Bands and Stochastic indicators, shown significant improvements when applying deep learning methodologies. This discovery corresponds with the theoretical notion that bigger datasets provide more advantageous training circumstances for these models. Moreover, deep learning models exhibited constant performance improvement across many trading methods, suggesting strong application to different market analysis techniques. Certain methods, such MACD and its combinations, demonstrated enhancement but indicated possibility for additional optimization, offering opportunities for further study.

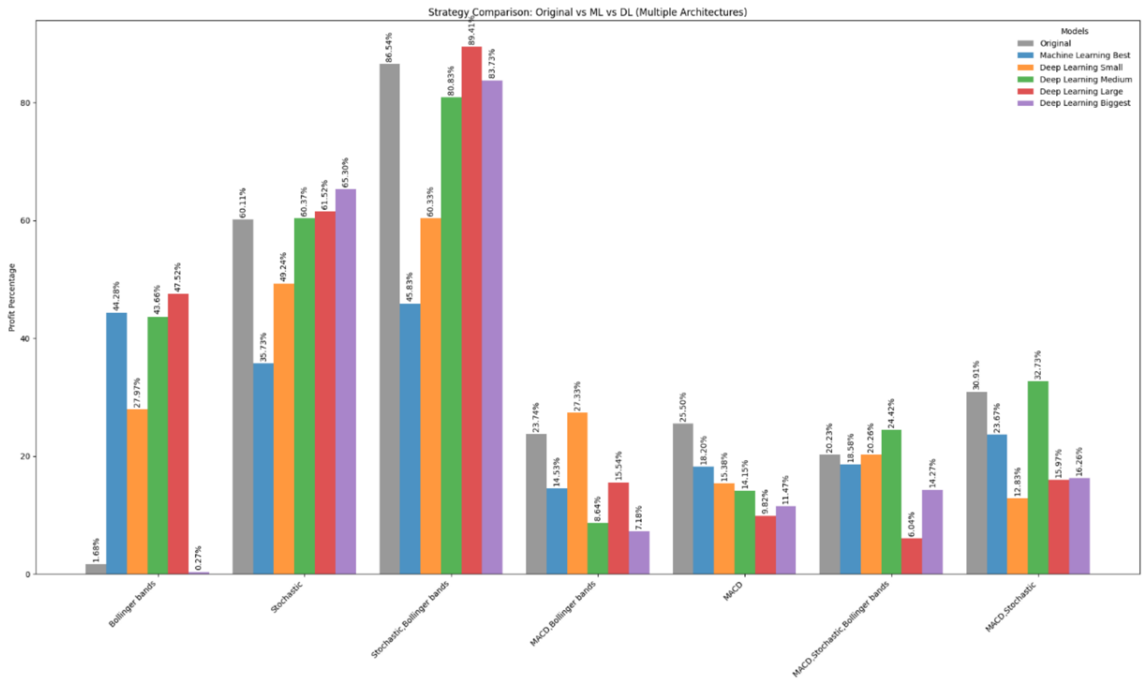

Figure 15: Deep Learning Results Comparison

Following these encouraging findings, the subsequent phase of this project will investigate the possibility of reinforcement learning to significantly improve trading performance. Reinforcement learning enables the creation of adaptive trading methods that can change dynamically to changing market circumstances. We want to formulate the trading issue as a Markov Decision Process to develop agents that can learn optimum trading strategies by engaging with simulated market settings. This method may surpass existing deep learning models by integrating sequential decision-making and optimizing long-term rewards. A comparison examination of reinforcement learning and top-performing deep learning models will provide useful insights into the relative advantages of both methodologies in algorithmic trading.

5.3 Reinforcement Learning Results

Following our prior analysis of output transformations in deep learning models for trade prediction, we now focus on the potential of reinforcement learning (RL) in this field. Reinforcement learning presents a distinctive framework that effectively corresponds with the sequential decision-making elements of trading, possibly addressing the shortcomings seen in conventional supervised learning methods.

Reinforcement learning, a branch of machine learning, investigates how agents tend to behave in an environment to optimize cumulative reward. This method facilitates the creation of adaptive strategies capable of understanding intricate market dynamics and improving trading choices in real-time inside algorithmic trading. The agent, specifically a trading algorithm, acquires knowledge by engaging with a simulated market environment, getting feedback in the form of gains or losses, and then modifying its approach.

Our prior studies on output transformations emphasized the need of suitable data representation and the difficulties inherent in forecasting raw profit figures. These results inform our methods in reinforcement learning, particularly with the structuring of the state space and reward function in our RL models. We want to use the benefits of reinforcement learning to address limitations encountered in our prior supervised learning research, such as difficulties in capturing temporal correlations and the potential for overfitting to historical data.

This research examines the effectiveness of reinforcement learning in formulating resilient trading strategies under diverse market situations. Our technique includes many essential components:

- We investigate several methods for preparing pricing data, based on our previous conclusions on the superiority of relative price information compared to absolute prices.
- We use a model that incorporates moving averages as input characteristics, integrating conventional technical analysis with contemporary machine learning methodologies.
- We evaluate the efficacy of integrating recognized technical indicators, namely the Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD), as state variables into our reinforcement learning model.
- We evaluate the efficacy of these RL-based techniques in comparison to our prior deep learning models and conventional algorithmic trading methods.

This analysis seeks to assess the efficacy of reinforcement learning in improving algorithmic trading strategies, identify critical factors for successful RL applications in financial markets, and elucidate the comparative advantages and disadvantages of RL relative to other machine learning approaches in this field.

The subsequent parts delineate our experimental configuration, exhibit the outcomes of our diverse RL implementations, and analyze the ramifications of our discoveries for the future of algorithmic trading.

5.3.1 Preprocecing Prices Aproach

In our examination of RL for algorithmic trading, we originally concentrated on several techniques for preparing price data. We examined five different methodologies: absolute prices, relative prices, normalized prices, logarithmic returns, and percentage change (Fig. 16). The aim was to find out if any of these preprocessing techniques may provide the RL algorithms with a more informative state representation, potentially improving trading strategies. Nonetheless, in opposition to our anticipations and prior results with other machine learning techniques, none of the preprocessing methods produced consistently favorable outcomes across the several trading strategies evaluated. This result was especially unexpected for the relative pricing strategy, which had previously excelled compared to other strategies in our prior tests with supervised learning models.

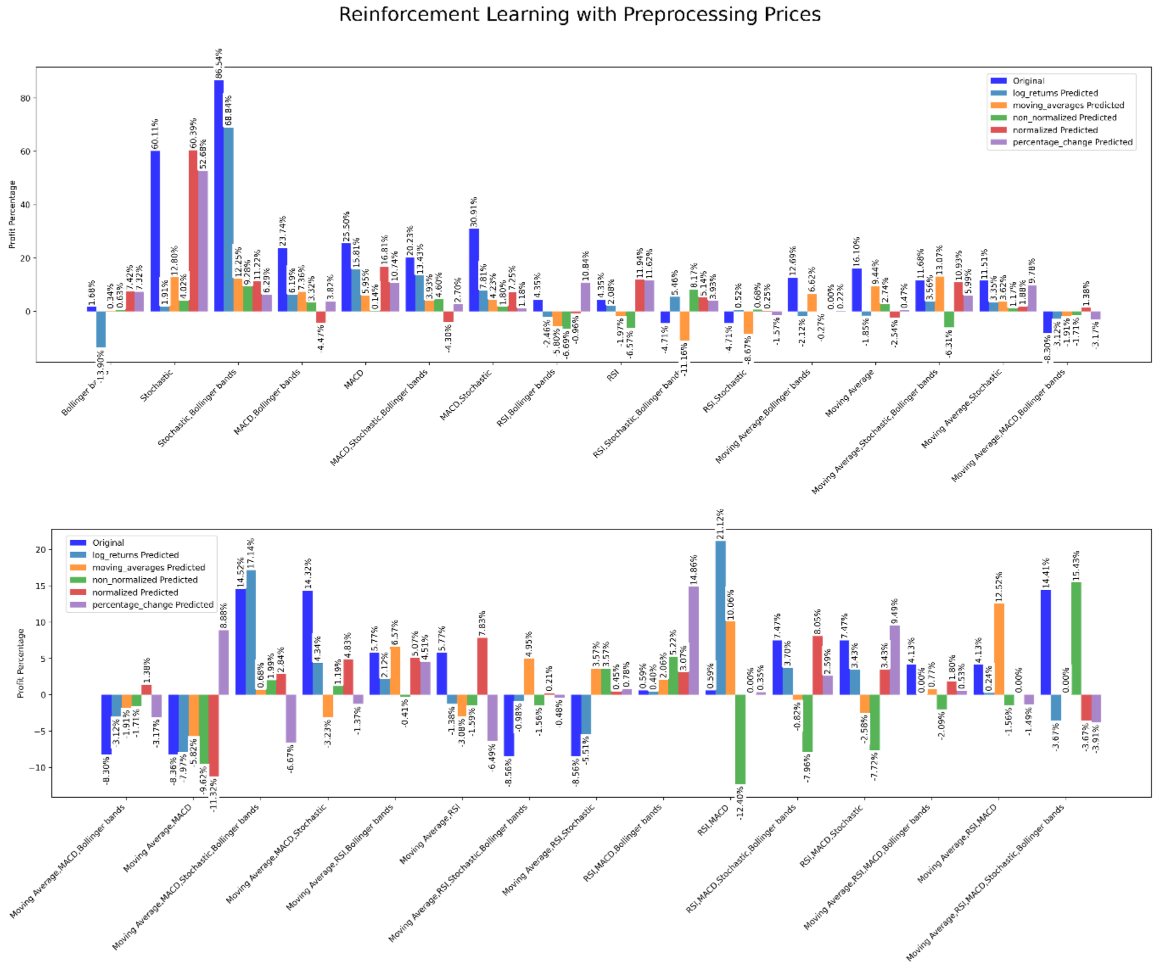

Figure 16: Reinforcemnet Learning with Preprocecing Prices

One positive aspect observed across all preprocessing methods was that the RL algorithms consistently attempted to learn patterns in the data. Unlike some of our previous experiments where models defaulted to predicting uniform outputs, the RL models showed variability in their predictions, indicating an effort to capture market dynamics. Despite this, the anticipated gains often fell short relative to the initial returns for the majority of trading strategies across all preprocessing techniques. None of the preprocessing technique consistently outperformed the others, since each exhibited comparable patterns of suboptimal performance across the different trading strategies.

The poor performance of all preprocessing approaches in our reinforcement learning studies indicates numerous possible concerns. Initially, the preprocessed price data alone may lack enough information for the RL agent to make smart trading choices. This implies that a more comprehensive state representation, maybe integrating more elements or extended historical settings, may be required. Secondly, the suboptimal performance may suggest that our reward function is inadequately directing the RL agent towards lucrative trading tactics. The persistent underachievement across all preprocessing techniques may indicate constraints in our selected reinforcement learning algorithm or its hyperparameters.

Although our preprocessing pricing strategy did not provide the expected good outcomes, it offered significant insights into the difficulties of using reinforcement learning in algorithmic trading. The persistent efforts of the RL algorithms to discern patterns, despite suboptimal performance, indicate possibilities for improvement via additional adjustments to our methodology. These results underscore the need for a more sophisticated approach to reinforcement learning within algorithmic trading, perhaps integrating ideas from conventional financial theory with advanced machine learning techniques.

5.3.2 Moving Averages Aproach

In the wake of the price preprocessing techniques, we examined the potential of moving averages as input features for our reinforcement learning models. This technique, which is frequently employed in traditional technical analysis, was employed to determine its potential to provide more actionable insights for the RL algorithms in the execution of trading decisions.

It is important to note that the calculation of moving averages requires a sufficient amount of historical data. Consequently, we had to exclude the initial trades from our analysis for strategies that lacked the necessary historical context. This adjustment resulted in a slight variation in the number of trades considered and, by extension, in the original profit figures compared to our previous analyses.

Figure 17 clearly demonstrates that the outcomes of the moving averages method closely matched with those of the price preprocessing techniques. The reinforcement learning algorithms consistently exhibited worse performance relative to the original trading methods across many combinations of technical indicators. Prominent instances include the considerable underachievement in methods using Bollinger Bands and MACD, where the RL model's projected earnings markedly lagged behind the initial profits.

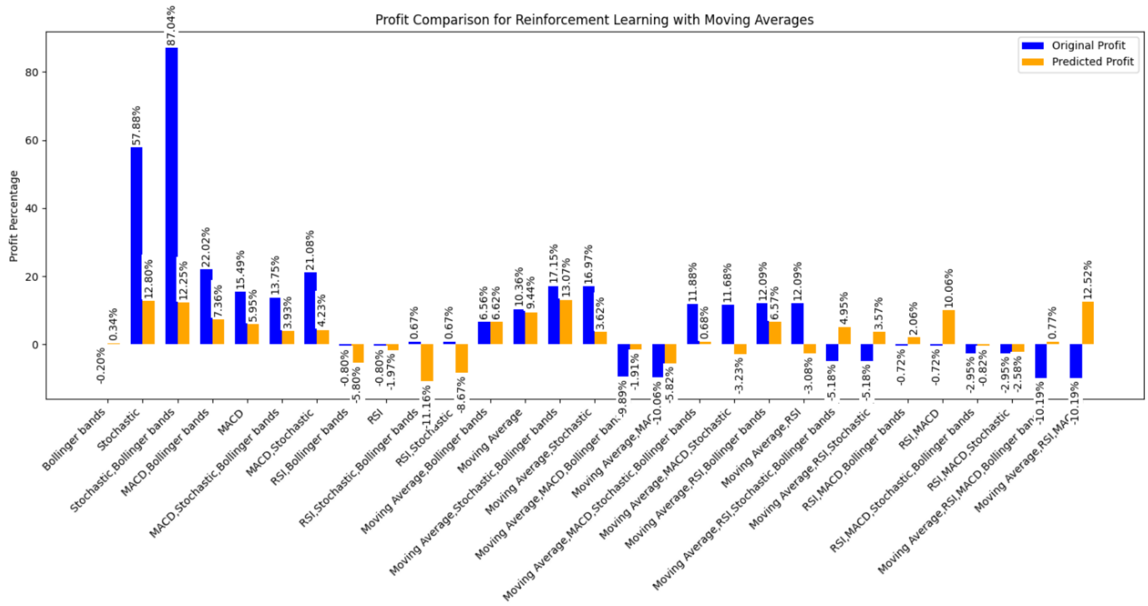

Figure 17: Reinforcement Learning with Moving Averages

The moving averages technique, while projected to enhance the representation of market trends, failed to provide the desired gains in trading performance. The persistent underachievement indicates that, despite the enhanced input feature, the RL models failed to accurately grasp the intricate dynamics of the financial markets.

These findings highlight the difficulties in implementing reinforcement learning in algorithmic trading, suggesting that conventional technical indicators, when used as input features, may lack the necessary information for RL algorithms to formulate improved trading strategies.

5.3.3 Technical Indicators Aproach

The concluding phase of our investigation of reinforcement learning for algorithmic trading included using technical indicators as input features. We specifically included the Moving Average Convergence Divergence (MACD) and Relative Strength Index (RSI), in addition to the three most recent price points. Two variations of this methodology were evaluated: one using absolute prices (Figure 18) and the other utilizing relative prices (Figure 19) for the latest price data.

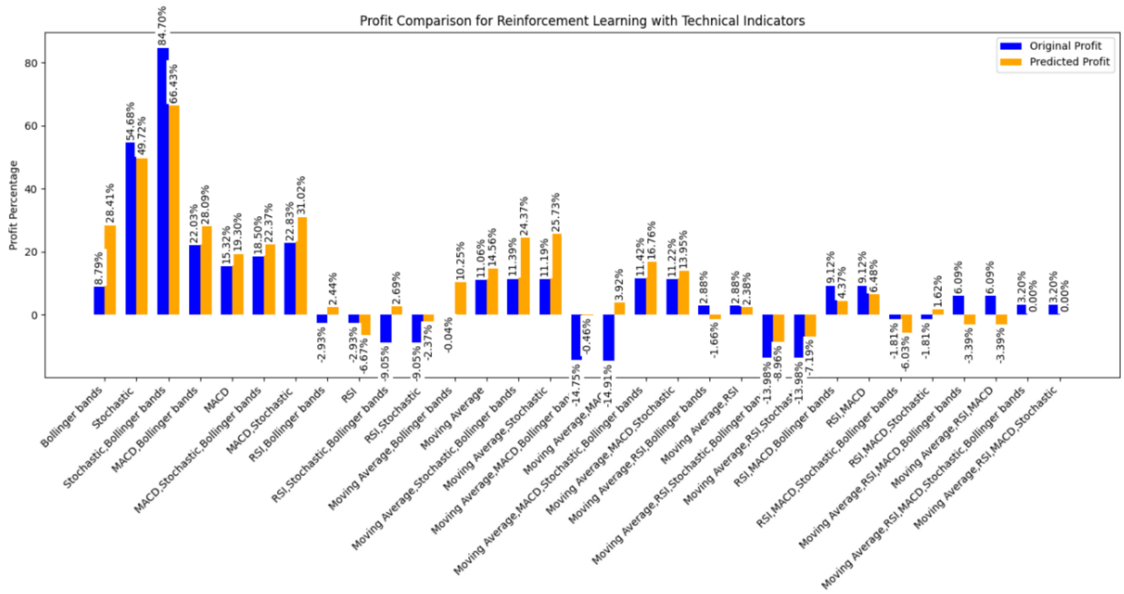

Figure 18: Reinforcement Learning with Tecnical Indicators (Absolute Prices)

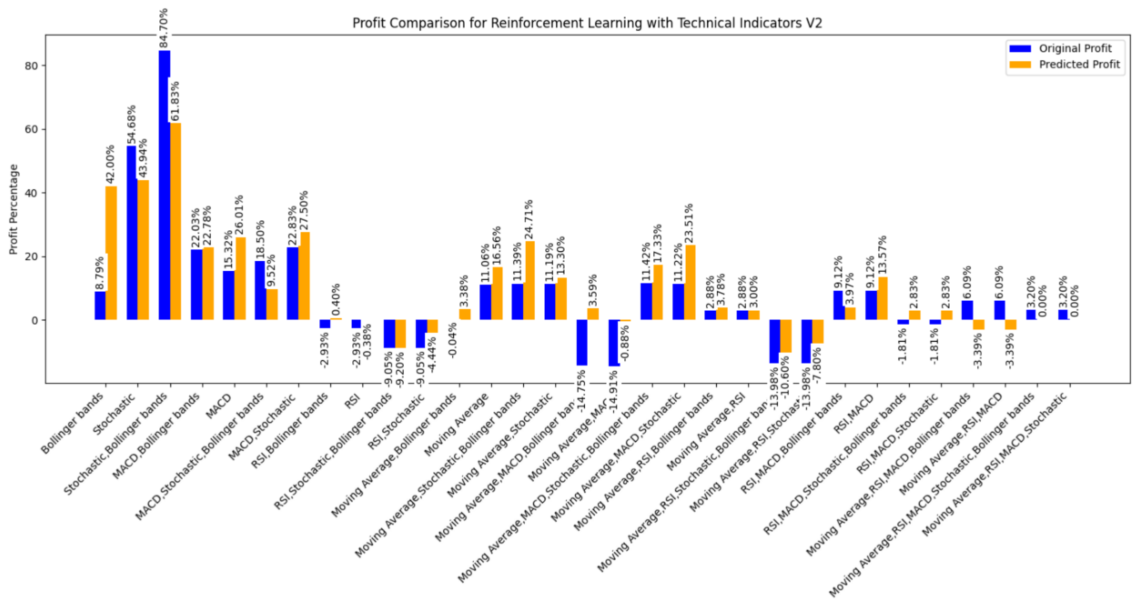

Figure 19: Reinforcement Learning with Tecnical Indicators (Relative Prices)

This strategy produced more favorable and consistent outcomes than our prior tests using reinforcement learning. The model exhibited superior performance compared to the original trading techniques in over fifty percent of instances, representing a notable improvement above previous methodologies.

Contrary to our predictions, the use of relative pricing did not provide a distinct benefit over absolute prices in this scenario. Both variations exhibited comparable performance patterns throughout the many trading techniques, with just negligible discrepancies in certain instances. This discovery questions our prior assumption about the superiority of relative price information in algorithmic trading models.

The uniformity of outcomes across both absolute and relative pricing inputs is especially significant. Unlike previous techniques that shown considerable inconsistency and unpredictability in reinforcement learning models, the technical indicators approach produced more reliable and consistent outcomes. This consistency suggests that the amalgamation of MACD, RSI, and recent price data provides a more thorough representation of market circumstances for the reinforcement learning system's use.

Numerous trading strategies shown significant improvement with this methodology. The Bollinger Bands method, which earlier had subpar results in our reinforcement learning studies, now demonstrates competitive efficacy compared to the original strategy. Likewise, techniques that integrate several indicators, such as MACD with Bollinger Bands or RSI with Stochastic Oscillator, shown improved profitability inside the reinforcement learning framework.

Nonetheless, it is crucial to acknowledge that despite the general increase in performance, there were occasions when the reinforcement learning model did not perform as well as the original approach. This highlights the intricacy of financial markets and the difficulties in creating universally successful trading algorithms.

The enhanced efficacy of this technical indicators methodology indicates that supplying the reinforcement learning model with a varied array of pre-processed market signals augments its capacity to identify pertinent patterns for trading choices. The integration of trend-following (MACD), momentum (RSI), and recent price fluctuations provides a more holistic perspective of market circumstances, facilitating the model's capacity to execute more educated trading choices.

The results provide a promising avenue for future study in the application of reinforcement learning to algorithmic trading, emphasizing the significance of feature selection and the possible integration of conventional technical analysis with sophisticated machine learning methodologies.

5.3.4 Overview

In conclusion of our examination of RL in algorithmic trading, we provide a thorough comparison of the top-performing models across reinforcement learning, machine learning (ML), and deep learning (DL) methodologies. Figure 20 presents a graphic comparison, elucidating the relative capabilities of each method over seven various trading strategies.

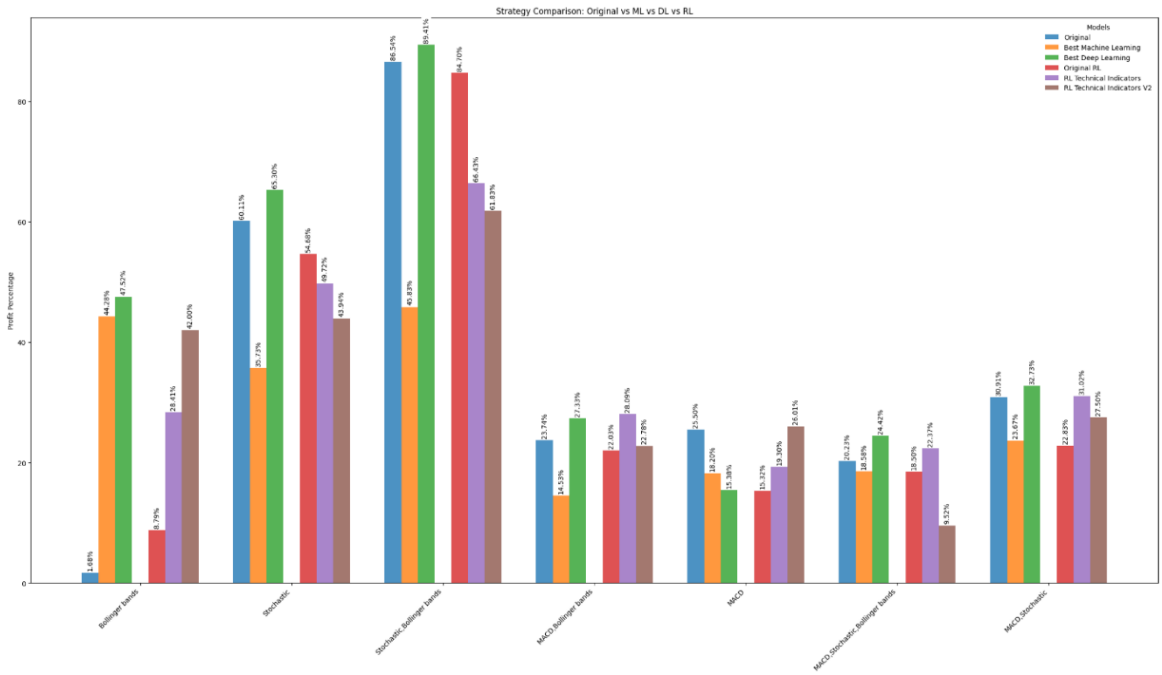

Figure 20: Reinforcement Learning Results Comparison

Precise analysis of the graph requires a juxtaposition of the performance of RL models (represented by purple and brown bars) against their original strategy performance (shown by red bars). The performances of ML and DL, shown by the orange and green bars respectively, must be compared with the performance of the original technique, represented by the blue bars. This comparison method allows us to determine which option exhibited the most significant improvement over the baseline approach.

This study demonstrates that reinforcement learning algorithms outperformed the leading machine learning models in six of the seven evaluated methodologies. This result is noteworthy, since it demonstrates the potential of reinforcement learning to comprehend complex market dynamics that may elude traditional machine learning techniques. The constant superior performance indicates that the sequential decision-making framework of reinforcement learning is suitably aligned with the dynamic characteristics of financial markets.

Nonetheless, when juxtaposing reinforcement learning with deep learning models, the outcomes are more intricate. Reinforcement learning techniques exceeded deep learning performance in just two instances, particularly in the fifth and sixth strategies shown in the graph. This result suggests that while reinforcement learning shows potential, deep learning methodologies consistently outperform it across most evaluated tactics.

In summary, while reinforcement learning has the potential to improve algorithmic trading strategies relative to conventional machine learning methods, it does not consistently surpass deep learning techniques. This comparative research indicates that a hybrid methodology, using the advantages of both reinforcement learning and deep learning, may provide the most resilient and versatile trading algorithms. Future study may examine hybrid models and analyze the particular market circumstances and strategic attributes that preferentially support one method over another.

5.4 Technical Imputs on Deep Learning

Building on the encouraging outcomes achieved via reinforcement learning using technical indicators, we investigated the feasibility of incorporating these RL methodologies into our deep learning models, which had previously shown superior overall performance. This chapter analyzes the combined method by comparing the performance of models using technical indicators with both absolute and relative price inputs.

It is crucial to acknowledge that the need to exclude some initial transactions for the computation of technical indicators renders the original profit outcomes in this study not directly comparable to those from prior deep learning methodologies. This constraint must be acknowledged while analyzing the findings.

Surprisingly, the integration of reinforcement learning techniques inside the deep learning framework failed to improve the original trading strategies in most cases. This result suggests that the direct use of successful reinforcement learning procedures in deep learning models may not guarantee enhanced performance in algorithmic trading.

A significant difference arose between the models using absolute prices and those utilizing relative pricing for technical indicators:

1. Absolute Prices (Figure 21): The algorithms often reverted to anticipating only positive trades ("all yes" forecasts) instead of exhibiting sophisticated decision-making. This behavior signifies a failure to identify critical patterns in market data, implying overfitting or inadequate generalization from the training data.
2. Relative Prices (Figure 22): Conversely, models using relative pricing exhibited more variability in predictions, indicating an effort to identify trends within the market data. Although these models did not consistently surpass the original methods, they exhibited a more advanced strategy for trade prediction in comparison to the absolute price models.

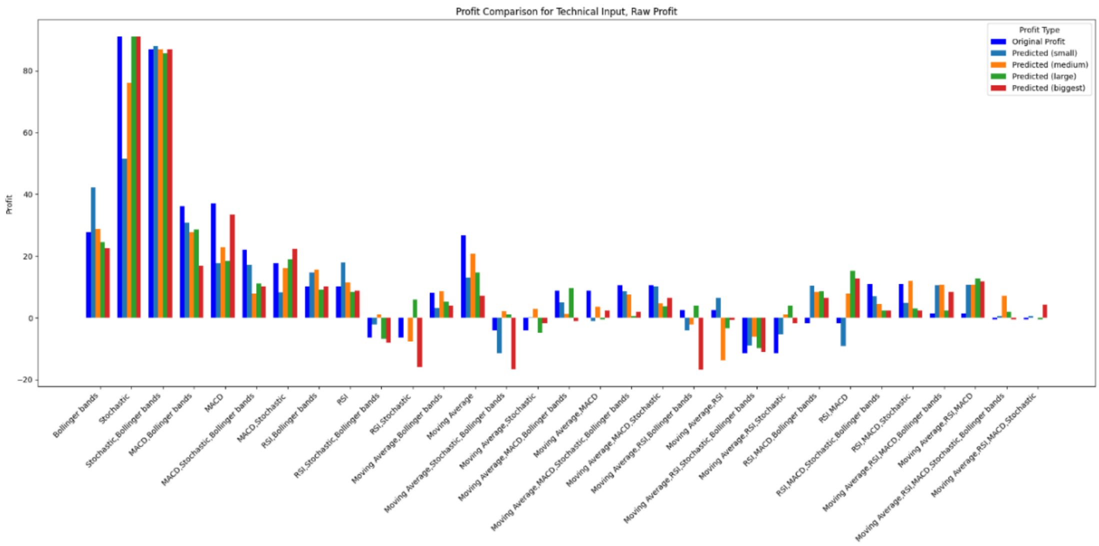

Figure 21: Results Deep Learning with Technical Indicators (Absolute Prices)

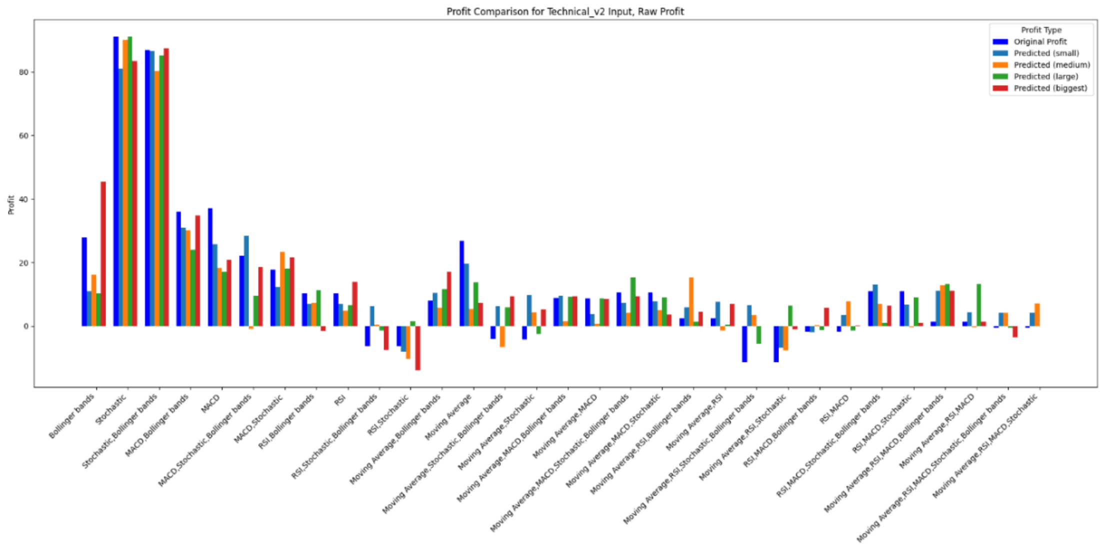

Figure 22: Results Deep Learning with Technical Indicators (Relative Prices)

The significant disparity in performance between absolute and relative pricing models highlights the critical role of input representation in financial modeling. Relative pricing seem to provide a more solid basis for pattern detection, consistent with conclusions from earlier chapters.

Although these findings are unsatisfactory for performance enhancement, they provide significant insights into the difficulties of integrating several machine learning paradigms inside algorithmic trading. The inability of this method to exceed other deep learning techniques underscores the intricacy of financial markets and the challenge of creating universally successful trading algorithms.

6. Conclusions

This work has analyzed the use of several machine learning techniques in algorithmic trading, focusing on a comparison of conventional machine learning (ML), deep learning (DL), and reinforcement learning (RL) approaches. Through extensive testing and study, we have gained substantial insights into the strengths and limitations of these approaches in financial markets. This part encapsulates our principal results, examines their ramifications for algorithmic trading, rigorously assesses our techniques, revisits our original research inquiries, and contemplates the practical consequences for traders and investors.

Our research has produced some notable discoveries that enhance the comprehension of machine learning applications in algorithmic trading:

6.1. Comparative Analysis of Performance Among ML, DL, and RL Methodologies

Deep Learning models significantly surpassed conventional machine learning methods in the majority of trading strategies. In six of the seven methods analyzed, at least one deep learning architecture exhibited higher performance relative to both the original strategy and the top machine learning model.

Reinforcement Learning algorithms demonstrated favorable outcomes, surpassing the leading machine learning models in six of seven tactics. Nonetheless, RL exceeded DL performance in just two instances, suggesting that while RL has promise, DL methodologies continue to retain a superiority in most evaluated techniques.

Traditional machine learning techniques, while establishing a reliable baseline, were often less adept at capturing intricate market dynamics in comparison to deep learning and reinforcement learning approaches.

6.2. Influence of Data Preprocessing Techniques:

The selection of data representation significantly affected model efficacy. Relative price inputs regularly surpassed absolute price inputs across several model architectures, especially in deep learning models.

In our deep learning studies, the raw profit methodology, when integrated with relative price inputs, demonstrated the most reliable and promising results, with at least one network design surpassing the profitability of the original strategy across almost all trading strategies. Unexpectedly, within the realm of reinforcement learning using technical indicators, the use of relative prices did not provide a distinct advantage over absolute prices, hence questioning our prior beliefs on the superiority of relative price data.

6.3. Efficacy of Technical Indicators

The integration of technical indicators (MACD and RSI) as input features in reinforcement learning models yielded more favorable and consistent outcomes compared to prior RL studies using just price data.

The incorporation of technical indicators into deep learning models yielded suboptimal outcomes. Models using absolute pricing often tended to anticipating just good results, neglecting to account for market intricacies. Although models using relative pricing exhibited more variability in predictions, they continually failed to surpass the performance of the original techniques.

The results indicate that while technical indicators might provide significant insights, their efficacy may be contingent upon the particular learning methodology and its integration within the model architecture.

6.4. Initial Hypothesis

We have substantiated our original hypothesis that deep learning methodologies may improve algorithmic trading techniques. Although we did not see significant enhancements in profitability across all tactics, our findings indicate a distinct advancement above conventional procedures and original ideas in several instances. These principal results emphasize the intricacy of using machine learning in financial markets and reinforce the need of meticulous attention in model selection, data preparation, and feature engineering. They highlight the promise of sophisticated methods such as deep learning and reinforcement learning in identifying intricate market trends that older approaches may overlook. This indicates that we have recognized a viable direction for more study and enhancement in algorithmic trading using sophisticated machine learning methodologies, hence creating new opportunities for optimizing trading strategy efficacy.

7. Future Work

This research has shown several prospects for more investigation and improvement. These key areas aim to address the shortcomings found in our current approaches and to explore new opportunities for enhancing algorithmic trading methods using advanced machine learning techniques.

7.1 Advanced Data Preprocessing Methods

Our study highlighted the significance of data representation, especially the efficacy of relative pricing inputs; yet, there is potential for improvement in data preparation methodologies.

- Investigate sophisticated feature engineering techniques to elucidate intricate market dynamics. This may include formulating novel technical indicators or constructing composite features that amalgamate several market signals.
- Examine non-traditional data sources in addition to conventional pricing and volume metrics. This may include sentiment research from financial news, social media trends, or macroeconomic statistics to provide a more complete perspective on market circumstances.

7.2 Hybrid Models

Our findings indicated that distinct methodologies (ML, DL, RL) exhibited differing degrees of efficacy across various tactics. Subsequent endeavors may concentrate on integrating these methodologies to capitalize on their individual advantages:

- Create models that amalgamate deep learning and reinforcement learning methodologies. Utilizing deep neural networks to analyze market data and produce characteristics, which are then used by reinforcement learning agents to execute trading choices.
- Investigate methods to integrate conventional financial models and domain expertise into machine learning frameworks. This may include using outputs from recognized financial models as inputs for machine learning models or creating bespoke loss functions that embody financial concepts.

7.3 Robustness and Generalization

Future research should concentrate on enhancing the resilience and generalization capabilities of ML-based trading strategies to guarantee their practical usefulness.

- Perform comprehensive evaluations of models under various market situations, including bull markets, bear markets, and phases of heightened volatility. This will assist in identifying tactics that are robust to changing market conditions.
- Broaden the testing to include several asset types not addressed in this research. This may include fixed income assets, commodities, or cryptocurrencies to evaluate the generalizability of the methodologies.
- Formulate strategies to alleviate overfitting, such regularization techniques tailored for financial time series data or innovative cross-validation procedures that consider the temporal characteristics of financial markets.

7.4 Interpretability and Explainability

The growing complexity of machine learning models necessitates enhanced interpretability and explainability, especially in the financial industry.

- Investigate methods to improve the transparency of deep learning and reinforcement learning models. This may include approaches such as attention mechanisms or SHAP (SHapley Additive exPlanations) values to ascertain the most critical components in the model's determinations.

- Develop visualization tools that effectively communicate the reasoning behind a model's trading decisions to human traders and investors. This might boost trust in machine learning methodologies and encourage their use in real-world trading scenarios.

7.5 Integration of Risk Management

Although our present study concentrated mostly on profitability, further investigations should prioritize risk management.

- Integrate risk measurements directly into model goals. This may include formulating multi-objective optimization strategies that equilibrate return and risk, or crafting bespoke loss functions that punish undue risk-taking.
- Examine the use of machine learning in risk assessment and management. This may include creating models to forecast market volatility or to detect future market collapses, which might then be incorporated into trading methods.

7.6 Applications of High-Frequency Trading

Considering the rising significance of high-frequency trading in financial markets, further study may investigate the applicability of our results to this area:

- Modify the most promising models from this research for ultra-low latency settings. This entails improving model architectures and inference methods to satisfy the rigorous temporal limitations of high-frequency trading.
- Examine the use of reinforcement learning in high-frequency decision-making. This may include creating reinforcement learning settings that replicate high-frequency market microstructure and training agents to function in these rapid circumstances.

Bibliography

Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques–Part II: Soft computing methods. Expert Systems with Applications, 36(3), 5932-5941.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and regression trees. CRC press.

Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

Dixon, M., Klabjan, D., & Bang, J. H. (2017). Classification-based financial markets prediction using deep neural networks. Algorithmic Finance, 6(3–4).

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

Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. Expert Systems with Applications, 38(8), 10389–10397.

Hawkins, D. M. (2004). The Problem of Overfitting. In Journal of Chemical Information and Computer Sciences (Vol. 44, Issue 1, pp. 1–12).

Heaton, J. B., Polson, N. G., & Witte, J. H. (2017). Deep learning for finance: deep portfolios. Applied Stochastic Models in Business and Industry, 33(1), 3-12.

Kim, K. J., & Lee, W. B. (2004). Stock market prediction using artificial neural networks with optimal feature transformation. Neural Computing and Applications, 13(3), 255–260.

Lawrence, R. (1997). Using neural networks to forecast stock market prices. University of Manitoba.

López de Prado, M. (2018). Advances in financial machine learning. John Wiley & Sons.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, 5(4).

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning.

Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6).

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088).

Scholkopf, B., & Smola, A. J. (2002). Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press.