Machine Learning for Efficacy Improvements in Automated Decision-Making in Financial Trading

Using SigTech platform

Mittal Mehta

Master's Thesis
Master of Engineering - Big Data Analytics
April 20, 2022

MASTER'S THESIS	
Arcada University of Applied Sciences	
Degree Programme:	Master of Engineering - Big Data Analytics
Identification number:	8633
Author:	Mittal Mehta
Title:	Machine Learning for Efficacy Improvements in Automated Decision-Making in Financial Trading Using SigTech platform
Supervisor (Arcada):	Magnus Westerlund, Petri Nousiainen
Commissioned by:	Arcada University of Applied Sciences

Abstract:
Machine Learning (ML) for finance is a fruitful approach to detect patterns in data. However, when it comes to predicting financial markets based on financial data with a low signal-to-noise ratio, it has been shown to be a complex problem to solve. A low signal-to-noise ratio means that there is more irrelevant information in the data compared to actionable events, and if a model relies just on data to determine the underlying drivers, then it will most likely learn to infer noise. Instead of predicting financial markets directly, this work focuses on machine learning as a risk management tool that is taught to identify the price trend.
The paper explores novel machine learning areas applied to finance, including meta-labeling, fractional differentiation, and ensemble learning algorithms predicting five different financial instruments for five different periods. Furthermore, we inspect the impact of the strength of trend and distribution of the return of the financial security (target variable) on the financial return generated by the system developed in this study. Finally, the paper shows promising results for the developed trading strategy, both in terms of cumulative and risk-adjusted return on the assumption that the predictive instrument has a strong trend for a longer period, return distribution is negatively skewed, and has superior machine learning performance metrics, particularly F1-score. The trading system's ability to identify when good investment decisions occur, generated from signals using a naïve and human-understandable model, further offers a path to explainability for AI in finance.

Keywords:	Financial forecasting, machine learning, meta-labeling, fractional differentiation, time series analysis, feature engineering.
Number of pages:	76
Language:	English
Date of acceptance:	?

CONTENTS

FIGURES

TABLES

FOREWORD

This thesis originated from a fascination with exploring state-of-the-art technologies developed in the field of financial machine learning and utilizing them to create a better-performing trading strategy system. Financial machine learning algorithms can be convoluted and confusing. This research is aimed to develop a trading strategy system that is easy to understand and interpretable.

I would like to express my sincere gratitude to my supervisor, Magnus Westerlund for the continuous support of my study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in the research and writing of this thesis.

I cannot express enough thanks to my co-supervisor, Petri Nousiainen for his continuous support and advice. His previous work is the basis of this thesis and without his ideas and code, I wouldn't have been able to explore this topic.

1 INTRODUCTION

1.1 Introduction

The stock market plays a crucial role in modern economic and social life. Investors want to maintain or increase the value of their assets by investing in the stock of the listed company with higher expected earnings. As a listed company, issuing stocks is an important tool to raise funds from the public and expand the scale of the industry.

Stock market ups and downs affect the profit of investors. If market prices go up with available stock, investors profit with their purchased stocks. In other cases, investors have to face losses if the market goes down with available stock prices. Buyers buy stocks at low prices and sell stocks at high prices and try to get huge profits. Similarly, sellers sell their products at high prices for profit purposes. This act of buying and selling stocks is called trading.

Trading materializes in an overly competitive world because traders have constant pressure to make accurate decisions to maximize profits. Therefore, experienced traders rely on multiple sources of information, such as news, historical data, earning reports, and company insiders. Risk is high, and many variables needed to be considered. For that reason, some financial institutions rely purely on machines to make trades. With machine learning and artificial intelligence coming into the picture, old techniques of trading are becoming obsolete rapidly.

Machine learning is a vibrant subfield of computer science that draws on models and methods from statistics, algorithms, computational complexity, artificial intelligence, control theory, and a variety of other disciplines. One of the significant tasks of machine learning algorithms is to employ massive historical data and accurately predict the future picture. Fortunately, this task of machine learning correlates with the fundamental aspect of trading. The traders usually discover time and space-limited localized patterns and think about how to maneuver these patterns for greater return. These patterns are ever-changing, and the process of identifying these patterns entails a great deal of time and energy. Machine learning algorithms help in finding the patterns that can be combined with the intuition and experience of traders for accurate decisions. The main objectives of using machine learning for trading are to remove the emotional component from manual trading and to find inefficiencies in the market faster than possible for a human. (Chan 2019)

1.2 Background and Motivation

Machine learning and algorithmic automation play an increasingly prominent role in all steps of the investment process across asset classes, from idea generation and research to strategy formulation and portfolio construction, trade execution, and risk management (Jansen 2020). Machine learning is used in asset management for hedging, portfolio construction, detection of outliers and structural breaks, credit ratings, sentiment analysis, market making, bet sizing, securities taxonomy, risk management, detection of false investment strategies, and so on. However, a popular machine learning application in asset management is price prediction (de Prado 2020).

As emotional beings, subject to fears, hopes, and agendas, humans are not particularly good at making fact-based decisions, especially when those decisions involve conflicts of interest. In such situations, investors are better served by a machine making the decisions based on facts learned from complex data (de Prado 2018).

It is often advantageous to encode deterministic strategies as algoritmic rules for a high-frequency trading platform that can handle thousands of transactions a second. Algorithmic trading is an improvement on manual trading, and the bulk of trades happening today are algorithmic. Although, it still relies on a human being to identify relevant patterns and code an algorithm to take advantage of them (Kissell 2013).

Machine learning, in contrast, has several benefits compared to traditional rule-based algorithmic trading. Machine learning algorithms can spot patterns in large volumes of data. They are used to find associations in historical data that can then be applied to algorithmic trading strategies. Machine learning empowers traders to accelerate and automate one of the most complex, time-consuming, and challenging aspects of algorithmic trading, providing competitive advantage beyond rules-based trading.

The advantages of machine learning offer academic financial research exciting times in the future. Firstly, machine learning provides the power and flexibility needed to find dim signals in the sea of noise caused by arbitrage forces. Secondly, machine learning allows dividing the research process into two stages: a) searching for important variables without functional form and b) searching for a functional form that binds those variables. Thirdly, machine learning allows the possibility of simulating synthetic data (de Prado 2020).

Despite its many advantages, machine learning is no panacea. The flexibility and power of machine learning techniques have a dark side. When misused, machine learning algorithms will confuse statistical flukes with patterns, which makes them prone to overfitting. This fact, combined with the low signal-to-noise ratio that characterizes finance, all but ensures that careless users will produce false discoveries at an ever-greater speed. Many investors fail to grasp the complexity of ML applications to investments. This seems particularly true for discretionary firms moving into the quantamental (a portmanteau combining “quant”itative and fund”amental” investing) space. Their high expectations will not be met, not because ML failed, but because they used ML incorrectly. (de Prado 2018)

1.3 Purpose of Study

Academics and regulators have shown an increasing awareness that modeling solutions must be understandable to people ((Doshi-Velez 2017); (European Commission 2021)). Research regarding machine learning in finance has come under critique for being opaque.

When one focuses on data-driven methods - machine learning and pattern recognition models in particular - the inner workings of the model can be hard to understand. With the increasing prevalence and complexity of methods, business stakeholders at the very least have a growing number of concerns about the drawbacks of models, data-specific biases, and so on. Analogously, data science practitioners are often may be not aware of approaches emerging from the academic literature or struggle to appreciate the differences between different methods (V.Belle 2021). The lack of interpretability or auditability of AI and machine learning methods could become a macro-level risk. Similarly, widespread use of opaque models may result in unintended consequences (Basel 2017).

A common target for financial prediction is based on the basic premise that price discovery can be performed ahead of time. While professional human traders have for decades been somewhat successful in pattern recognition, for example, in predicting market trends, few have dared to claim an ability to predict the price at any given time in the future. Therefore, the predominant aim of research to perform price prediction is possibly misguided. Showing the validity of a system with an innate ability to perform functional estimation of non-linear price data argues against the efficient market hypothesis. An alternative approach is to emulate the success of human traders that often rely on naïve human-understandable technical analysis to gain insights into market behavior.

Building on the prior work (Nousiainen 2021), we explore a trading strategy system that uses a hybrid modeling approach. Initially, a primary naïve model produces a meta-label, then a secondary model (machine learning model) that has access to more complex features evaluates the primary's outcome. When the secondary model predicts whether the primary model will generate profit, the system records a trade. After experimenting with feature engineering techniques and two different ensemble algorithms (Random Forest and Light Gradient Boosting Machine (GBM)), we validate the trading system on five different financial instruments for five different forecast horizons.

Further, determining the best trading size (size of the bet) is considered essential, and leverage is needed to realize the full growth potential of an investment strategy (Chan 2006). A second benefit of the proposed system is that it provides a probability estimate for correctness, which can be used to determine leverage (i.e., borrowing on margin to buy stocks). Given that the trading strategy target is to maximize capital increase, we show the cumulative return, CAGR, and Sharpe ratio for the respective approaches. Additionally, we analyze the effect of strong trend presence and return divergence on the returns generated by the trading strategy.

1.4 Research Question and Hypothesis

This study is dedicated to the exploration of state-of-the-art technologies developed in the field of financial machine learning and constructing an automated decision-making trading strategy system. One of the biggest hurdles that machine learning faces today is a public trust and acceptance due to the "black box problem". The reason is that people mistrust what they do not understand.

The primary object of this research is to develop a human-readable machine learning-driven decision-making trading system. Additionally, the system developed should be scalable and repeatable. The research questions are following.

1. Define the architecture of a human-understandable automated decision-making trading system.

2. How does meta-labeling, fractional differentiation, different ensemble learning algorithms enable and improve the machine learning model in the trading system?

3. Assess the effect of the existence of a strong trend and return distribution on the financial performance of the trading system.

The research hypothesis is that machine learning can be used as a risk management tool that can enhance the financial performance of any trading strategy.

1.5 Limitations

The study is limited by the given data and some objective limitations of the considered models. Limitation of data means that we are using only the daily closing price data. The input data ignores intraday, weekly and monthly periods.

The primary model (Donchian Channel) used here is a trend following strategy, which relies on historical data and is prone to generating false breakout signals. When trying to enter at the start of a trend, trend following strategies may generate losing trades. It may enter in to a trade when there is not trend in the market. Furthermore, if there is huge difference between stop-loss and entry prices, it will need large amount of capital to initiate the trade (Faith 2007).

ADX is a lagging indicator as it is based on moving averages that react slowly to any price changes. ADX may not be a good indicator for less volatile stocks, and for the more volatile ones, there may be too many false signals generated. For this study, we have used the threshold limit of 25 but many experts prefer to use the limit of 20.

The machine learning algorithms should predict probabilities leading us to choose Random Forest and Light GBM machine learning algorithms as the secondary model. The secondary model (Light GBM) split the tree leaf- wise which can lead to overfitting as it produces much complex trees especially with a small dataset and the model depends on some random processes so results can differ from run to run.

For this study transaction costs and taxes has been ignored. Risk-free interest rate has been assumed to be zero.

2 LITERATURE REVIEW

2.1 Meta-labeling

In finance, the generation of trade signals are suggestions alerting to trading opportunities that arise for specific instruments on financial markets. These signals can be based on human analysis (manual signals) or generated by mathematical algorithms (automated signals). Regardless of how these signals are generated, blindly following trading signals without evaluating their correctness can be a recipe for disaster.

A technique proposed by De Prado (de Prado 2018) known as "Meta Labeling" can, as shown in the proposed setup, be used to evaluate the quality of trading signals. The idea is based on using a primary model with high recall, which is usually a naïve model (can be any econometric model, technical model, fundamental analysis-based model) with relaxed conditions for positive and negative classes to generate trading signals applied to price data. The primary model prediction (direction) becomes the target for the secondary model and new features (including but not limited to price data) are used as input for a more complex machine learning model (Wilcox 2019). The aim of the secondary model is for the machine learning algorithm to learn to replicate the primary model signal. Essentially, the secondary model is trained to recreate the signal using more complex data. The signal is converted to trade when the secondary model predicts that the primary model will return a profit.

Meta-labeling is particularly helpful when you want to achieve a higher F1-score. First, we build a model that achieves high recall, even if the precision is not particularly high. Second, we correct for the low precision by applying meta-labeling to the positives predicted by the primary model (de Prado 2018).

Meta-labeling will increase F1-score by filtering out the false positives, where the majority of positives have already been identified by the primary model. Stated differently, the role of the secondary machine learning algorithm is to determine whether a positive from the primary (exogenous) model is true or false. It is not its purpose to come up with a betting opportunity. Its purpose is to determine whether we should act or pass on the opportunity that has been presented by the primary model (de Prado 2018).

Meta-labeling is a very powerful tool to have in traders' arsenal, for four additional reasons. First, machine learning algorithms are often criticized as black boxes. Meta-labeling allows building a machine learning system on top of a white box (like a fundamental model founded on economic theory). This ability to transform a fundamental model into a machine learning model should make meta-labeling particularly useful to "quantamental" firms. Second, the effects of over-fitting are limited when you apply meta-labeling because machine learning will not decide the side of your bet, only the size. Third, by decoupling the side prediction from the size prediction, meta-labeling enables sophisticated strategy structures. Fourth, achieving high accuracy on small bets and low accuracy on large bets will ruin any trader. As important as identifying good opportunities is to size them properly, it makes sense to develop a machine learning algorithm solely focused on getting that critical decision (sizing) right. Meta-labeling machine learning models can deliver more robust and reliable outcomes than standard labeling models (de Prado 2018).

Meta-labeling provides the tradeoff between recall and precision, it not only improves classification metrics but also strategy metrics. Due to its ability to significantly improve the performance of various types of primary models, meta-labeling is a good case study of how machine learning can be applied in financial markets (Joubert 2022).

2.2 Fractional Differentiation

Machine learning algorithms demand data that is representative of the desired outcome. For financial engineering raw data is rarely used, but rather, data goes through a feature development process that aims to find a data form with higher relevance to the training of the machine learning algorithm (Sinan Ozdemir 2018). The quality of machine learning algorithms results largely depends on the quality of the available features (Guozhu Dong 2018).

Time series remember the past. The memory of a particular time series refers to the strength of the past values influencing future values. Strong memory means that past values have much information regarding the future values of a time series, and weak memory the converse. Many predictive models require a statistical consistency of the time series data called stationarity (Tsay 2005).

A stationary time series is one where the mean (average value) and variance (spread of values) are constant over time (P. & Granger 1973). The commonly used transformation to make a series stationary is differencing up to some order. One obtains first-order differencing by subtracting each value from the preceding one (extracting the rate of change). The second-order differencing repeats this process for the resulting series, and the same process continues for higher orders. These transformations can lead to stationarity, but the memory from the original series is erased (Tsay 2005).

A fractionally differenced time series retains the maximum amount of memory of the original series while fulfilling the stationary criteria. Hosking (Hosking 1981) originally introduced this and subsequent work by others concentrated on fast and efficient implementations for fractional differentiation for continuous stochastic processes (Noack & Nielsen 2014).

Recently, de Prado showed how to find the optimal balance between zero differentiation and fully differentiated time series (de Prado 2018). Walasek and Gajda (Walasek & Gajda 2021) evaluate and predict different stock indexes and conclude that a fractionally differentiated time series is stationary and keeps its memory, leading to a more accurate prediction using an Artificial Neural Network (ANN).

Assume that a time series X runs throughout time t and it is not stationary:

$$X = \{X_t, X_{t-1}, X_{t-2}, \ldots, X_{t-k}, \ldots\}$$

Computing the differences between consecutive observations is an approach to achieve a stationary time series:

$$\nabla X_t = X_t - X_{t-1}$$

By defining a backshift operator B as $B^k X_t = X_{t-k}$ for $k \geq 0$ and $t > 1$, the above formula, first-order differentiation, can be expressed as:

$$\nabla X_t = X_t - X_{t-1} = X_t - BX_t = (1-B)X_t$$

Concerning the situation when differenced data will not appear to be stationary, it may be demanded to difference the data a second time to obtain a stationary series:

$$\nabla^2 X_t = \nabla(\nabla X_t) = \nabla(X_t - X_{t-1}) = (X_t - X_{t-1}) - (X_{t-1} - X_{t-2}) = X_t - 2X_{t-1} + X_{t-2}$$

which using the backshift operator may be represented as:

$$\begin{aligned}
&(1-B)^2 = 1 - 2B + B^2 \\
&B^2 X_t = X_{t-2} \\
&(1-B)^2 X_t = X_t - 2X_{t-1} + X_{t-2}
\end{aligned}$$

More generally, for order of differentiation d we have:

$$\nabla^d X_t = (1-B)^d X_t$$

For a real number d using a binomial formula:

$$(1+x)^d = \sum_{k=0}^{\infty} \begin{pmatrix} d \\ k \end{pmatrix} x^k$$

the series can be expanded to:

$$(1-B)^d = \sum_{k=0}^{\infty} (-B)^k \prod_{i=o}^{k-1} \frac{d-i}{k-i} = 1 - dB + \frac{d(d-1)}{2!} B^2 - \frac{d(d-1)(d-2)}{3!} B^3 + \ldots$$

The current value in the time series is the function of all the past values that occurred before this time point. To each past value, a weight ω_k is assigned:

$$X_t = \sum_{k=0}^{\infty} \omega_k X_{t-k}$$

The application of fractional differentiation on time series allows deciding each weight for each corresponding value. All weights calculated by fractional derivative can be expressed as:

$$\omega = \left\{1, -d, \frac{d(d-1)}{2!}, -\frac{d(d-1)(d-2)}{3!}, \ldots, (-1)^k \prod_{i=0}^{k-1} \frac{d-i}{k!}, \ldots\right\}$$

When we consider d as a positive integer number, there is a point when d is equal to k, then $d-k=0$ and:

$$\prod_{i=0}^{k-1} \frac{d-i}{k!} = 0$$

which leads to the conclusion that memory beyond that point is removed. In first-order differencing (d=1), weights follow as (shown in Figure 1):

$$\omega = \{1, -1, 0, 0, \ldots\}$$

The general-purpose approach of coefficient for various orders of differencing is depicted

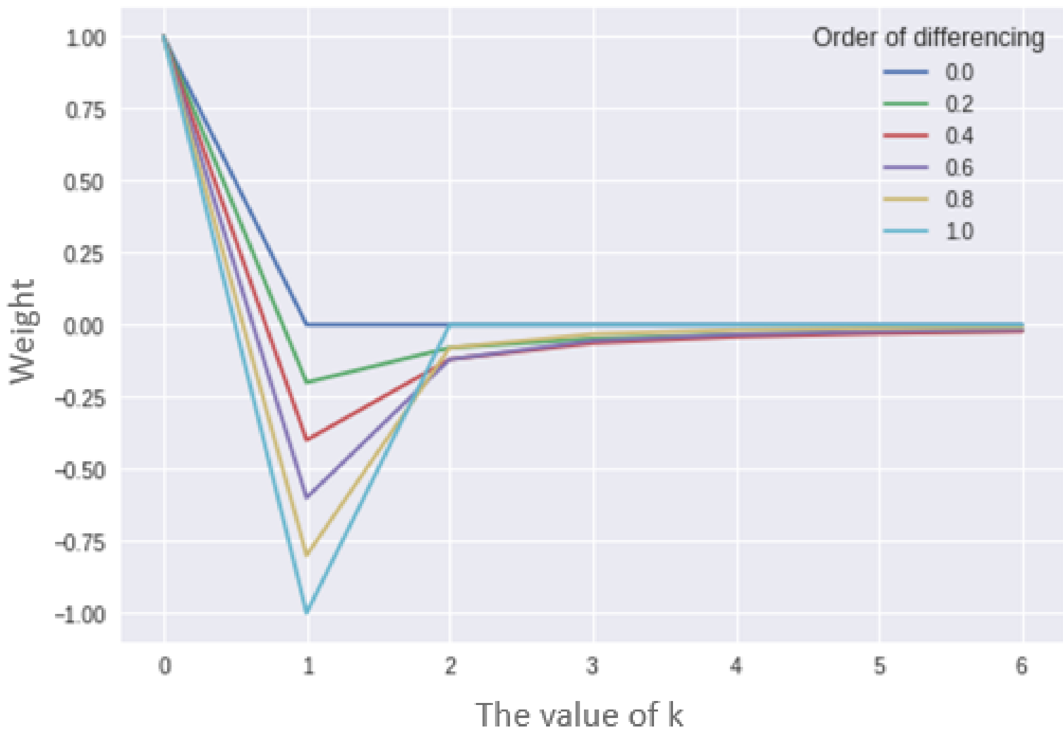

Figure 1. Weights of the lag coefficients for various values of k.

in Figure 1 where each line is related to particular order of differencing ($d \in [0,1]$). For example, if $d = 0.25$ and k is always an integer number, all weights achieved values other than 0 which means that the memory is going to be preserved. From the above derivation, the iterative formula for the weights of the lags can be deduced [3]:

$$\omega_k = -\omega_{k-1} \cdot \frac{(d-k+1)}{k}$$

where ω_k is the coefficient of backshift operator B^k. For the first-order differentiation, we have: $\omega_0 = 1, \omega_1 = -1. \omega_k = 0$ for $k > 1$.

In conclusion, the primary intention to use the fractional differentiation is finding the fraction d, which is considered the minimum number needed to achieve stationarity, while keeping the maximum volume of memory in analyzed time series (Walasek & Gajda 2021).

2.3 Ensemble Learning Models

Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by taking a (weighted) vote of their predictions (Dietterich 2000). Two well-known ensemble learning methods are boosting (Peter Bartlett 1998) and bagging (Breiman 1996) of classification trees. In boosting, successive trees give extra weight to points incorrectly predicted by earlier predictors. In the end, a weighted vote is taken for prediction. In bagging, successive trees do not depend on earlier trees, each is independently constructed using a bootstrap sample of the data set. In the end, a simple majority vote is taken for prediction.

Random forests are ensembles of decision trees that add a layer of randomness to bagging. In addition to constructing each tree using a different bootstrap sample of the data, random forests change how the classification or regression trees are constructed. In standard trees, each node is split using the best split among all variables. In a random forest, each node is split using the best among a subset of predictors randomly chosen at that node. This somewhat counterintuitive strategy turns out to perform very well compared to many other classifiers, including discriminant analysis, support vector machines, and neural networks and is robust against overfitting (Breiman 2001).

Light GBM is a fast, distributed, high-performance gradient boosting framework based on a decision tree algorithm, used for ranking, classification and, many other machine learning tasks. It implements two novel techniques: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to deal with a large number of data instances and a large number of features respectively (K. Guolin 2017). Light GBM has many advantages, including sparse optimization, parallel training, multiple loss functions, regularization, bagging, and early stopping. It grows trees leaf-wise. It chooses the leaf it believes will yield the most significant decrease in loss (Joseph 2020).

Researchers and practitioners tend to argue that stock market prediction is more effective with algorithms that are themselves nonlinear, adaptive, and do not assume a fixed functional form (Ritika Chopra 2021). The efficacy of the forecasts tends to improve when multiple classifiers are organized in serial, conditional, hybrid or parallel combinations (Wang & Chan 2006).

Decision trees or the more advanced version called random forest is the preferred machine learning method for trading because it does not have as many parameters to fit as a neural network, thus reducing the danger of data snooping bias. The output of a decision tree is a set of conditional decision rules, which are much easier to interpret than the nonlinear functions that neural networks use. On the other hand, K-Nearest Neighbors (KNN) or logistic regression is too simple, they do not capture a lot of the nonlinear dependence between different input features and the output return (Chan 2020a).

2.4 Kelly Criterion

Gambling in all forms, whether it be in blackjack, sports, or the stock market, must begin with a bet. There are fascinating parallels between strategy games and investing. Some of the best portfolio managers are excellent poker players, perhaps more so than chess players. The machine learning algorithm can achieve high accuracy, but if we do not size our bets properly, our investment strategy will inevitably lose money (de Prado 2018).

Kelly (Kelly 1956) in his original paper, highlighted how a gambler could maximize his profit by reinvesting his winnings. He developed a formula that outputs the optimal position size as a percentage of capital, given the winning probability factor and the win/loss ratio. Thorp (Thorp 1969) extended upon Kelly's original gambling strategy by expanding it to the stock market and financial derivatives.

$$f^* = p - \frac{q}{b}$$

where f^* is the fraction of the current bankroll to wager, p is the probability of a win, q is the probability of a loss and b is the amount gained with a win.

We use the probability estimates of the secondary model to derive the size of the bet (de Prado 2018).

3 THEORETICAL BACKGROUND

3.1 Method Evaluation Metrics

To evaluate the performance of the secondary machine learning model we use accuracy, precision, recall, F1-score, confusion metrics, and AUC-ROC curve (Wikipedia 2021a).

A confusion matrix is a table that is used to define the performance of a classification algorithm. The confusion matrix consists of four basic numbers that are used to define the measurement metrics of the classifier. These four numbers are:

1. TP (True Positive): TP represents the number of profitable trades which have been correctly classified.

2. TN (True Negative): TN represents the number of non-profitable trades which have been correctly classified.

3. FP (False Positive): FP represents the number of misclassified trades as profitable but they are actually non-profitable trades. FP is also known as a Type I error.

4. FN (False Negative): FN represents the number of misclassified trades as non-profitable but they are actually profitable. FN is also known as a Type II error.

Accuracy is defined as the ratio of true positives and true negatives to all positive and negative observations. Accuracy tells us how often we can expect our machine learning model will correctly predict an outcome out of the total number of times it made predictions.

Precision is the ratio of true positive to the sum of true positive and false positive. Precision represents the machine learning model's ability to correctly predict the positives out of all the positive predictions it made.

The recall is the ratio of true positive to the sum of true positive and false negative. The recall represents the machine learning model's ability to correctly predict the positives out of actual positives. In other words, it measures how good the machine learning model is at identifying all actual positives out of all positives that exist within a dataset.

F1-score is a harmonic mean of precision and recall score. This is a useful metric in the scenarios where choosing either precision or recall score can result in a compromise in terms of the model giving high false positives and false negatives respectively.

The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR against FPR at various threshold values. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

3.2 System Validation Metrics

To evaluate the performance of the trading strategy and validate the system developed in this study, we use cumulative log return, Compound Annual Growth Rate (CAGR) (Wikipedia 2021b), and Sharpe ratio (Sharpe 1966).

The compound annual growth rate (CAGR) is the rate of return that would be required for an investment to grow from its beginning balance to its ending balance, assuming the profits were reinvested at the end of each period of the investment's life span.

$$\text{CAGR} = \left(\frac{V_{\text{final}}}{V_{\text{begin}}}\right)^{1/t} - 1$$

Where, V_{begin} is the beginning value, V_{final} is the final value and t is time in years.

The Sharpe ratio was developed by Nobel laureate William F. Sharpe (Sharpe 1966) and is used to help investors understand the return of an investment compared to its risk. The ratio is the average return earned in excess of the risk-free rate per unit of volatility or total risk. Volatility is a measure of the price fluctuations of an asset or portfolio.

$$S_a = \frac{E\left[R_a - R_b\right]}{\sigma_a}$$

Where, E is the expected value, R_a is the asset return, R_b is the risk-free return and σ_a is the standard deviation of the asset excess return.

4 METHODOLOGY

To summarize the claims for state-of-the-art techniques in both academic and professional literature, we have devised a method to test such a trading system. The section below details the proposed approach to a trading system. First, we start by discussing the architectural demands for the trading strategy. Then we specify parameters for the primary model. Finally, we discuss the efficacy of the chosen techniques for establishing the secondary model and method to determine the strength of the trend on a given trading instrument.

4.1 Trading Strategy System Architecture

The trading strategy system is shown in Figure 2 and is derived from (Nousiainen 2021) with some variation in decision logic. The system is implemented on the SigTech platform. SigTech is a cloud-hosted and Python-based platform, which integrates trading backtest engine and analytics with curated investment-grade datasets covering equities, rates, FX, commodities, and volatility.

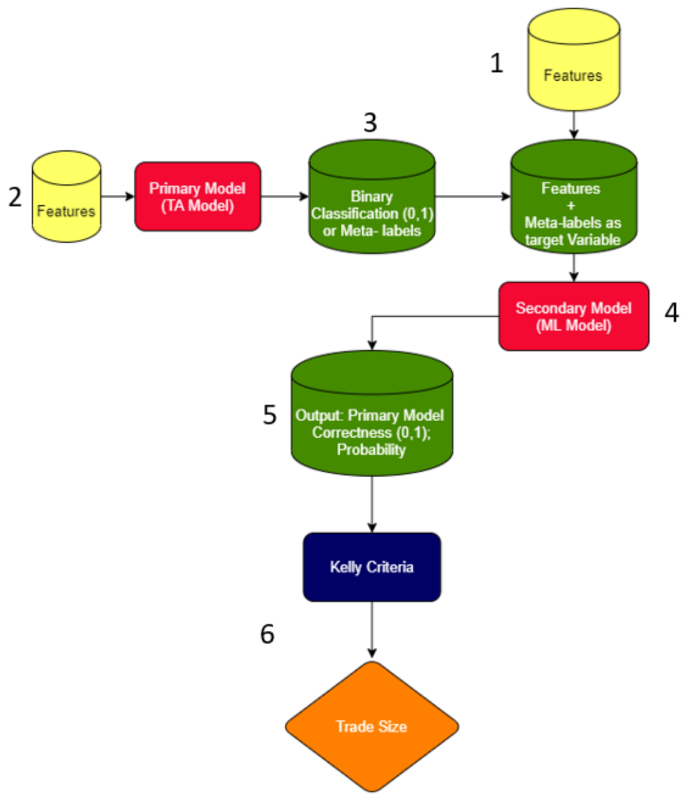

Figure 2. Trading Strategy System

The trading strategy system is designed upon the insights of trading practitioners. In (Chan 2020b), Chan discusses how to predict the conditional probability of profit using a certain Tail Reaper strategy. This utilizes a random forest model trained with features that include a comprehensive list of financial instruments in the market with historical price data. The target label can be defined as an implied "meta-labeled" market return that indicates profit (true: 1) or loss (false: 0). Hence, the model should predict if the trade is profitable with the probability of the same to be use to calculate the size of the bet.

Further work in (Hudson & Thames 2020) investigates how a technical analysis-based meta-labeling model architecture improves strategy performance metrics by filtering out false positives. They also detail a process and metrics for predicting the side of the bet (long or short).

The basic idea behind meta-labeling has two models.

- The Primary Model determines whether one should open a long or short position on a given day.
- The Secondary Model is the exogenous model that will predict if the primary model was correct.

Table 1 demonstrates the trading strategy system structure applied in this study.

1. We start with the data pre-processing stage, where we extract data of 17 rolling futures from the SigTech platform, calculate the one-day percentage return of those futures and create technical indicators which can be used as input features in the secondary model.
2. We chose the Donchian channel (DC) technical analysis model as a primary model for this study because it is a well-known human-understandable trend-following strategy (Singh 2020). The model is optimized to maximize the cumulative daily percentage returns, for a 50-day channel, with a ten-day mean exit. The primary Donchian channel model produces binary labels showing whether the primary model

will generate profit or not referred to as meta-labels.

3. For the secondary model, the random forest has been proposed in (de Prado 2018) and (Chan 2020b) and we also use this as our base secondary model. The input features for the secondary model are a set of 48 features, including 17 rolling front future price data, 17 one-day percentage returns of each future contract, 14 technical analysis indicators, and a meta-labeled target.

4. The secondary model predicts whether the primary model will return a profit (1) or not (0) and the statistical probability of the prediction.

5. The system will execute a trade when the secondary model predicts 1 meaning that the primary model will return a profit. No trade will be executed when the secondary model predicts 0.

6. Utilizing the probability generated by the machine learning model and Kelly criterion, we determine the size of the trade.

Table 1. Trading Strategy System Architecture

Step	**Description (Base Model)**
1	Features (48): (17 Rolling Futures, 17 1-D Percentage Returns, 14 Technical Indicators)
2	Primary Model (Technical Analysis Model): Input: S&P500 E-mini future (Target feature, 5636 close days, 2000-01-04 to 2021-10-01) Model: Donchian Channel (Channel 50, exit 10) Output: Trade signal
3	Labeling / Binary Classification [0,1] Binary classification of Donchian Channel output (1= Profit, 0= Loss i.e. Meta-labels [0,1])
4	Secondary Model (Machine Learning Model): Input: Features Target Variable: Meta-labels[0,1] Model: Random Forest Output: Prediction (Profit=1 or Not=0) and probability of next day's prediction
5	Rule: If the Secondary Model returns true (1) Then do the trade (positive) else no trade (negative)
6	Trade size: Using Kelly criterion and probability generated by secondary model

4.2 Priming of the Primary Model

The quality of a trading decision is often determined as correctness in direction, aka. hit ratio. If the probability of a specific trading signal is P(A) 0.5 it suggests that the trading system is useless, as a flip of a coin could reach a similar level. In our case, the primary model provides trading signals, based on a well-known technical indicator, the Donchian channel. The primary model was chosen to capture trend-related signals and should thus work best on such instruments that show a clear trend.

We experiment with five different instruments and five different prediction periods to validate the whole system. We optimize the Donchian channel entry and exit parameters for each target instrument based on out-of-sample data. Table 2 shows the optimized parameters for the respective instruments.

Table 2. Parameters of the Primary Model

Target Variable	Channel	Exit
S&P 500	50	10
Nasdaq	20	10
Nikkei	60	10
Euro	35	10
Gas Oil	60	10

4.3 Secondary Model Evaluation

As discussed in the previous sub-section, the primary model is expected to provide trend-related signals. Using a naïve primary model means that it delivers a human-understandable outcome. However, the naïve model cannot be expected to work on unseen data without prior validation, as it lacks a non-linear and dynamic behavior. This suggests that the naïve model cannot be used for investment decisions. Therefore, the chosen primary model should be viewed as a flip-of-a-coin model, with a correctness probability of close to 0.5.

While this line of thought is true in many cases, it does not explain what happens with our invested capital. For example, if the primary model correctly identifies significant price shifts, it may mean that a profit/loss account would still show a profit while the hit ratio is close to 0.5. Further, to evaluate the outcome of the primary we train a secondary model that makes a judgment. We also independently examine the efficacy of each novel idea of improving a trading system discussed in the literature section, i.e., meta-labeling, different ensemble machine learning algorithms, feature engineering steps like fractional differentiation and the impact of additional features (macro indicators), and Kelly criterion for determining the size of the trade.

Figure 3 describes the process of evaluating the secondary model and techniques applied in this research.

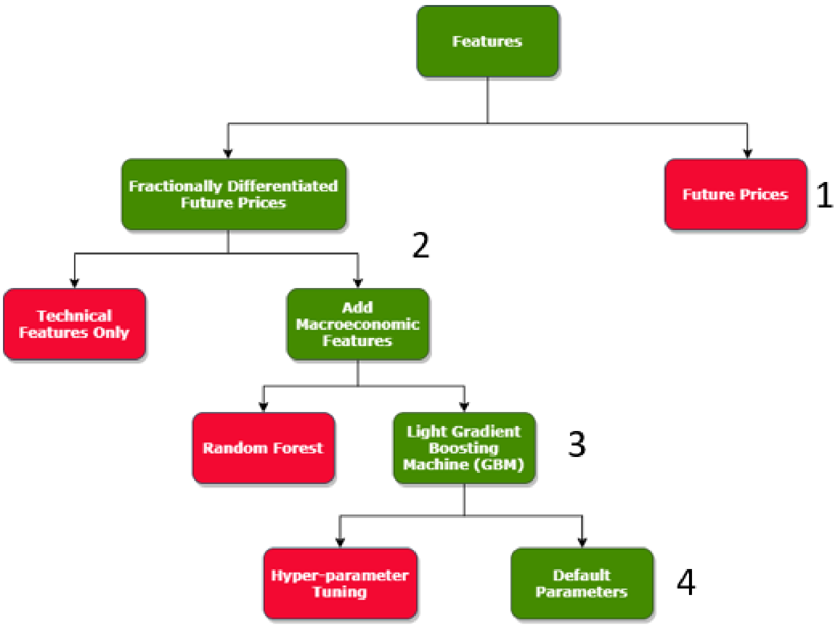

Figure 3. Secondary Model Evaluation Process

1. We examine the base model's performance that inputs a set of 48 features and uses the random forest as the secondary model.

2. We perform the feature engineering step where we fractionally differentiate the futures and add 27 new macroeconomic indicators as features, bringing input features total to 75. We evaluate the impact of feature engineering techniques and choose the one which performs better for the next analysis.

3. We compare the performance of the random forest-based model after feature engi-

neering with the light GBM model.

4. We tune hyperparameters of our preferred machine learning model and examine the results.

4.4 Trend Strength Indicator

In order to determine whether the strength of a trend of a given instrument affects the results obtained in this study, we use the Average Directional Index (ADX) to measure the duration of a strong trend for the different forecasting instruments.

ADX was developed by J. Welles Wilder (Wilder 1978), a technical analysis indicator that measures the magnitude or strength of a trend, but not the actual direction of the latter. ADX indicator measures strong or weak trends. This can be either a strong uptrend or a strong downtrend. It does not tell if the trend is up or down, it just helps to understand the strength of the current trend.

The ADX is a combination of two other indicators, the positive directional indicator (abbreviated +DI) and the negative directional indicator (-DI). The ADX combines them and smooths the result with a smoothed moving average.

To calculate +DI and -DI, we need price data consisting of high, low, and closing prices for each period. First we need to calculate the directional movement (+DM and -DM) (ADX 2021):

UpMove = today's high − yesterday's high
DownMove = yesterday's low − today's low
if UpMove > DownMove and UpMove > 0, then +DM = UpMove, else +DM = 0
if DownMove > UpMove and DownMove > 0, then -DM = DownMove, else -DM = 0

After selecting the number of periods (Wilder used 14 days originally and we used the same for this study), +DI and -DI are:

+DI = 100 times the smoothed moving average of (+DM) divided by average true range

-DI = 100 times the smoothed moving average of (-DM) divided by average true range

The smoothed moving average is calculated over the number of periods selected, and the average true range is a smoothed average of the true ranges. Then:

ADX = 100 times the smoothed moving average of the absolute value of (+DI − -DI) divided by (+DI + -DI)

Variations of this calculation typically involve using different types of moving averages, such as an exponential moving average, a weighted moving average or an adaptive moving average (ADX 2021). For this study, we have used an exponential moving average.

Table 3 shows the ADX indicator scale values. ADX values help traders identify the strongest and most profitable trends to trade. The values are also important for distinguishing between trending and non-trending conditions. Many traders will use ADX readings above 25 to suggest that the trend is strong enough for trend-trading strategies. Conversely, when ADX is below 25, many will avoid trend-trading strategies (Schaap 2021).

Table 3. ADX Indicator Scale

ADX Value	Trend Strength
0-25	Absent or Weak Trend
25-50	Strong Trend
50-75	Very Strong Trend
75-100	Extremely Strong Trend

The direction of the ADX line is important for reading trend strength. When the ADX line is rising, the trend strength is increasing, and the price moves in the direction of the trend. When the line is falling, the trend strength is decreasing, and the price enters a period of retracement or consolidation (Schaap 2021).

When the +DMI is above the −DMI, prices are moving up, and ADX measures the strength of the uptrend. When the −DMI is above the +DMI, prices are moving down, and ADX measures the strength of the downtrend. Figure 4 is an example of an uptrend reversing to a downtrend. As shown in figure 4 ADX rises during the uptrend, when +DMI was above −DMI. When the price reversed, the -DMI crossed above the +DMI,

Figure 4. ADX Line Example by Investopedia

and ADX rises again to measure the strength of the downtrend (Schaap 2021).

ADX calculations for this study have been based on an exponential moving average of price range expansion over 14 days (Adithyan 2021). Figure 5 shows the ADX line, + DI, -DI, and closing price of the S&P500 for the test period. ADX line above the 25 shows the strong trend and below 25 shows no strength in trend.

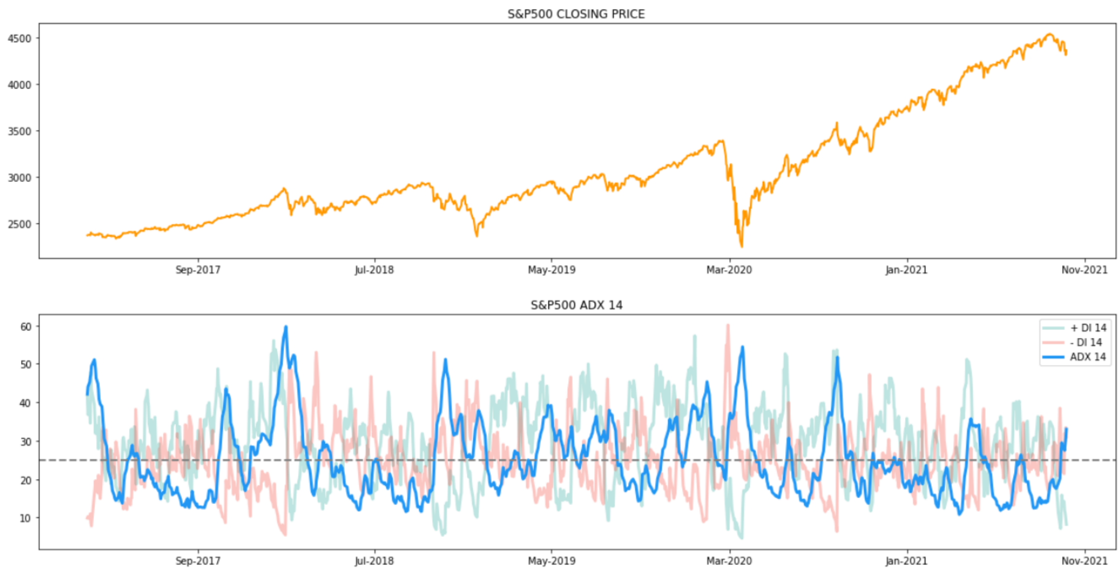

Figure 5. S&P500 Average Directional Indicator

5 DATA AND FEATURE ENGINEERING

To study price behavior on financial markets demands data that is both representative and reliable with high integrity. The SigTech platform provides the data that is ready to use for analysis.

In total, we use 75 features (17 futures, 17 one-day returns, 14 technical indicators, 27 macroeconomic indicators), and to develop our system, we evaluate market performance against the E-mini (S&P500 rolling future) from 2000-01-04 to 2021- 10-01 this includes 5636 days of closing prices.

During validation, we broaden the scope of the experiment with additional target futures: Nasdaq, Nikkei, Euro, and Gas oil. For all experiments, we split the data into training and test sets, 80% and 20% respectively.

In summary, the data used is for longer time period with different categories.

Table 4. Input Features - Instruments used for Modeling

No.	Input Feature	Description	Type
1	TU	US 2YR Note	Fixed Income
2	TY	US 10YR Note	Fixed Income
3	US	US Long Bond	Fixed Income
4	CT	Cotton	Commodity
5	LC	Live Cattle	Commodity
6	BO	Soybean Oil	Commodity
7	QS	Gas Oil	Commodity
8	HO	NY Harbor ULSD Heating Oil	Commodity
9	BP	British Pound	Currency
10	EC	Euro	Currency
11	CD	Canadian Dollar	Currency
12	JY	Japanese Yen	Currency
13	AD	Australian Dollar	Currency
14	NV	New Zealand Dollar	Currency
15	ES	E-mini S&P 500 Index	Index
16	NQ	E-mini Nasdaq 100 Index	Index
17	NX	Nikkei 225 Index	Index

The input features contain a data set of 17 rolling futures i.e., three fixed income futures, six currency futures, five commodity futures, and three equity index futures. In addition, we include 14 technical analysis indicators calculated on the underlying price and 27 macroeconomic indicators. Tables 4, 5 and 6 show the list of instruments, technical and macroeconomic indicators used for this study.

Table 5. Input Features - Technical Analysis Indicators

No.	Technical Indicators	Formula
1	Momentum (default 10)	$C_t - C_{t-10}$
2	Rate of Change (default 14)	$C_t / C_{(t-n)} x100$
3	Disparity (default 5)	$C_t / MA_5 x100$
4	Disparity (default 10)	$C_t / MA_{10} x100$
5	Price Oscillator	$MA_5 - MA_{10} / MA_5$
6	Relative Strength Index (default 14)	$100 - 100 / \left(1 + \frac{\sum_{i=0}^{n-1} xUp_{t-i}}{n} / \frac{\sum_{i=0}^{n-1} xDw_{t-i}}{n}\right)$
7	Moving Average (default 5)	$\left(\sum_{i=1}^{5} xC_{t-i+1}\right) / 5$
8	Deviation Rate (default 6)	$\frac{C_t - MA_6}{MA_6} x100\%$
9	Psychological Line (default 12)	$PSY_{12} = (A/12) x100\%$
10	Average of Return (default 5)	$\left(\sum_{i=1}^{5} xSY_{t-i+1}\right) / 5$
11	Average of Return (default 4)	$\left(\sum_{i=1}^{4} xSY_{t-i+1}\right) / 4$
12	Average of Return (default 3)	$\left(\sum_{i=1}^{3} xSY_{t-i+1}\right) / 3$
13	Average of Return (default 2)	$\left(\sum_{i=1}^{2} xSY_{t-i+1}\right) / 2$
14	Average of Return (default 1)	$ASY_1 = SY_{t-1}$

Table 6. Input Features - Macroeconomic Indicators

No.	Macroeconomic Indicators	Macroeconomic Indicators
1	Consumer Price Index All (CPI All)	Producer Price Index All Commodities (PPI)
2	Harmonized Unemployment	Producer Price Index Mining Industries
3	Business Confidence Index (BCI)	Producer Price Index Manufacturing Industries
4	Consumer Confidence Index (CCI)	USA GDP
5	Composite Leading Indicator (CLI)	Belgium GDP
6	Consumer Price Index Urban (CPI Urban)	Switzerland GDP
7	Non-Farm Payroll	Germany GDP
8	Rent Prices US	France GDP
9	Nominal House Price Indices US	Italy GDP
10	Real House Price Indices US	Netherland GDP
11	Price To Rent Ratio US	Sweden GDP
12	Standardised Price-Rent Ratio US	UK GDP
13	Standardised Price-Income Ratio US	Japan GDP
14	Price To Income Ratio	

The time series is transformed to introduce stationarity during feature engineering using the fractionally differentiated method. Figure 6, shows the raw S&P500 price series and the fractionally differentiated version (calculated with the coefficient d(0.5)).

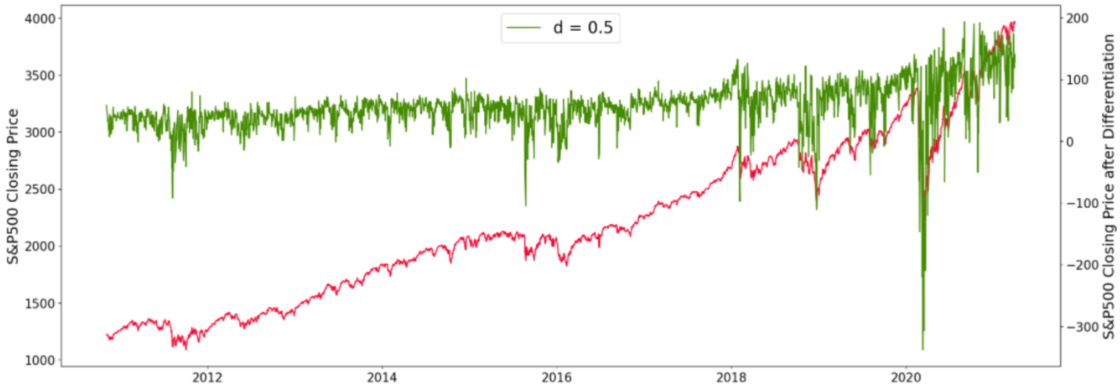

Figure 6. S&P500 Original Price Series and Fractionally Differentiated Series

The Augmented Dickey-Fuller test (ADF test) is a common statistical measure used to determine whether a given time series is stationary or not. Since the null hypothesis of ADF test assumes the presence of a unit root, the p-value obtained by the test should be less than the significance level (say 0.05) to reject the null hypothesis. Thereby, inferring that the series is stationary (Verma 2021).

Table 7 shows the ADF statistic and p-value for the S&P500 price series, one-day return (first-order differentiation, d=1), and fractionally differentiated price series (d=0.5). The original price series of the S&P500 index has the ADF statistics of 1.9167, while the differentiated equivalent has this statistic equal to -4.7415. At a 95% confidence level, the fractionally differentiated time series has a p-value of 0.0001 similar to the first-order differentiated series.

Table 7. ADF statistics and p-value of S&P500

Price Series	**ADF Statistics**	**p-value**
S&P500 E-mini	1.9167	0.9986
S&P500 E-mini 1-d Return (d=1)	-33.5818	0.0000
S&P500 E-mini Frac Diff (d = 0.5)	-4.7415	0.0001

This shows that the fractionally differenced series is not only stationary but holds considerable memory of the original series as well.

The Light GBM model can provide relative importance scores for each input feature. The feature score informs how useful input features are at predicting a target variable.

In Figure 7, we plot the features sorted by feature score for the prediction of S&P500. The

RSI technical indicator is the most important feature as per the feature score. The least important features were Standardized (housing) price to rent ratio, Price to income ratio, SE (Sweden) GDP, CH (Switzerland) GDP, and CPI U (consumer price index urban).

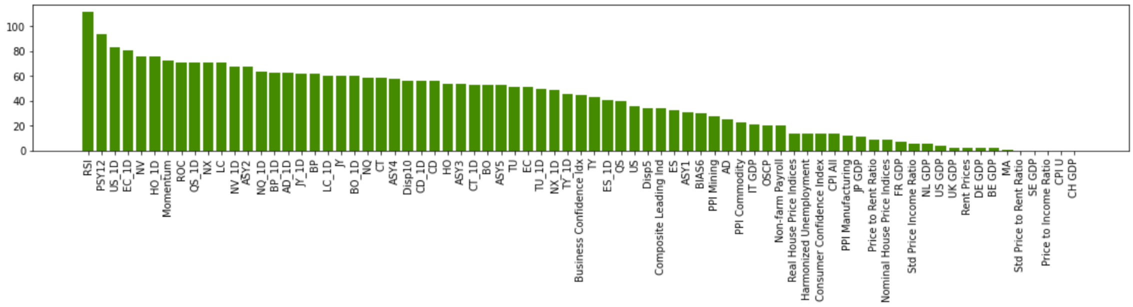

Figure 7. Feature Importance Ranking of Light GBM (with 75 features)

While outside the scope of this paper, the feature importance score can further be used for feature selection, to discard features with low scores, and keep features with higher scores.

6 METHOD DEMONSTRATION

6.1 Primary Model

The Donchian Channel technical analysis model is selected to be the primary model in the trading strategy process because it is a trend-following strategy. Trend following strategy means that we buy the asset when the price trend goes up and sell the asset when the price trend goes down according to the strategy signals (Nousiainen 2021).

Trend following is a systematic investment strategy that aims to seek profit from long, medium, or short-term trends in markets based on the assumption that price trends tend to endure. The strategy does not attempt to predict the price of an asset but rather aims to buy an asset when its price is trending up and sell when its price is trending down (Nousiainen 2021).

While the trend is up, traders may assume it will continue until there is evidence that points to the contrary. Such evidence could include lower swing lows or highs, the price breaking below a trendline, or technical indicators turning bearish. While the trend is up, traders focus on buying, attempting to profit from a continued price rise (Nousiainen 2021).

When the trend turns down, traders focus more on selling or shorting, attempting to minimize losses or profit from the price decline. Most (not all) downtrends do reverse at some point, so as the price continues to decline, more traders begin to see the price as a bargain and step in to buy. This could lead to the emergence of an uptrend again. Whereas, in the stable price action, the trend is sideways (Nousiainen 2021).

When asset prices are stable, Donchian Channels narrow to show a constricted range and limited profit potential for day traders. When market prices fluctuate to a larger degree, Donchian Channel readings will widen and this tends to be the type of environment that lends itself well to trading gains in both directions (bullish and bearish). A disadvantage of the Donchian Channel strategy is that it may generate losses with false breakouts while searching for a price signal to trigger an entry trade to join a long trend for given security (Nousiainen 2021).

The Donchian Channel strategy, in this study, is optimized. For S&P500 the optimized channel length is 50 days, and the exit length is ten days. The optimized channel parameters maximized the cumulative daily returns of the strategy (Nousiainen 2021).

Donchian Channel Trading Strategy Signals Explained in Figure 8 (Nousiainen 2021).

Figure 8. Donchian Channel Trading Strategy Overview

A - Long entry signal generated, when upper channel (UC) breakout 50 days high, at the price 1000.

B - Long exit signal generated, when the price hit 10 days average price, at 1125 (A-B profit)

C - Short entry signal generated, when lower channel (LC) breakout 50 days low, at the price 1110.

D - Short exit signal generated, when the price hit 10 days average price, at 1120 (C-D loss).

E - Long entry signal generated, when upper channel (UC) breakout 50 days high, at the price 1150.

Trading position can be no position, long, or short. A long position means the purchase of an asset with the expectation it will increase in value, a bullish attitude. A short position means a trading technique in which an investor sells a security with plans to buy it later, a bearish attitude. No position means not making any trade and staying out of the market.

Table 8. The Structure of the Donchian Channel Trading Strategy Coding

No.	Signal	Description
1	Entry signal Enter long Enter short	Enter long when close > the 50-days high close or enter short when close < the 50-days low close
2	Exit signal Exit long Exit short	Exit long if the close < the 10-days avg price, or exit short when the close > the 10-days avg price
3	No trading position	Before entry and after exit signals

The coding structure of the Donchian channel trading strategy as shown in table 8 has an entry signal; open a long position when the close is higher than the highest close during the 50-days channel, open short position when the close is lower than the lowest close during the 50-days channel. An exit signal; close a long position when the close is lower than the 10-days average price, close short position when the close is higher than the 10-days average price, and when no position between the exit and entry signals, stay out of the market (Nousiainen 2021).

Table 9 shows the coding logic for the trading signal generation based on the Donchian Channel primary model.

1. On day 1, when the close price 1489 > the 50 day high of 1478 generates a long entry signal (True in column Long entry)

2. On day 4, when the close price 1504 < the 10 day average of 1516 generates a long exit signal (True in column Long exit)

3. No trade has been executed on days 5 and 6 as no new long or short entry signal is generated.

4. On day 7, when the close price 1426 < the 50 day low of 1432 generates a short entry signal (True in column Short entry)

Table 9. The Primary Model Trading Signal Generation Logic for Coding

Position	Close	50 d high	50 d low	10 d avg	Long entry	Short entry	Long exit	Short exit	Signal
Long entry									
Day 1	1489	1478	1338	1401	T	F	F	T	1
Day 2	1478	1489	1338	1409	F	F	F	T	1
Day 3	1530	1555	1338	1505	F	F	F	T	1
Day 4	1504	1555	1338	1516	F	F	T	F	0
Long exit									
No position									
Day 5	1515	1555	1338	1519	F	F	T	F	0
Day 6	1528	1555	1338	1522	F	F	F	T	0
Short entry									
Day 7	1426	1523	1432	1455	F	T	T	F	-1
Day 8	1416	1523	1426	1451	F	T	T	F	-1
Day 9	1352	1523	1344	1400	F	F	T	F	-1
Day 10	1401	1523	1344	1390	F	F	F	T	0
Short exit									

5. On day 10, when the close price 1401 > the 10 day average of 1390 generates a short exit signal (True in column Short exit)

6.2 Meta-labeling Binary Classification

In the current setup, we want to build a secondary machine learning model that predicts whether the primary model will return profit or not. The supervised learning algorithms require that the rows in X are associated with an array of labels or values y so that those labels or values can be predicted on unseen features samples. The meta-labels will be used as the y value for the secondary machine learning model.

Table 10 demonstrate the meta-labeling trading logic applied in this study.

1. We shift the trading signal generated by the primary model by one day to calculate the daily log return of the underlying (as shown in column signal adj for return calc). This is needed as the signal will be generated based on the closing price, the trade will be initiated on the next day at the close price.

2. Next, we calculate the percentage daily log return of the underlying.

Table 10. Meta-labeling Coding Logic

Day	Close	Signal	Signal adj for return calc	Daily % log ret of the underlying	Signal adj X daily % log ret of the underlying	Daily % log ret of the trading strategy	Meta-labeling (profit=1 else 0)
1	1489	1					
2	1478	1	1	-0.74%	1 X -0.74%	-0.74%	0
3	1530	1	1	3.46%	1 X 3.46%	3.46%	1
4	1504	0	1	-1.71%	1 X -1.71%	-1.71%	0
5	1515	0	0	0.73%	0 X 0.73%	0.00%	0
6	1528	0	0	0.85%	0 X 0.85%	0.00%	0
7	1426	-1	0	-6.91%	0 X -6.91%	0.00%	0
9	1352	-1	-1	-4.63%	-1 X -4.63%	4.63%	1
10	1401	0	-1	3.56%	-1 X 3.56%	-3.56%	0

3. To calculate the percentage return of the trading strategy, we multiply the adjusted signal by the daily percentage log return of the underlying. If the signal is long and the daily percentage log return is positive or if the signal is short and the daily percentage log return is negative, we make a profit. On the other hand, if the signal is short and the daily percentage log return is positive or if the signal is long and the daily percentage log return is negative, we make a loss.

4. Once we have the daily percentage log return of the trading strategy we label these returns; if positive as 1 else 0. These labels are called meta-labels which we use as the y values in the secondary model.

This process turns our problem into purely binary classification. The label values are {0,1} as opposed to the primary model signal values {-1,0,1}. The machine learning algorithm will be trained to decide whether to take the bet or pass. When the predicted label is 1, we can use the probability of this secondary prediction to derive the size of the bet, where the side (sign) of the position has been set by the primary model (de Prado 2018).

6.3 Secondary Model

There are many kinds of machine learning models; they can be shallow models or deep models. One of the shallow learning models is logistic regression, which only predicts

Table 11. The Secondary Model Training Logic

Day	Features									Target
	Instr. & 1-D return			Tech. Indicators			Macro Indicators			
	TU	TU-1D	...	RSI	MA	...	CPI	BCI	...	Meta-label
1	99.078	0.006	...	65.886	99.001	...	71.219	100.852	...	1
2	99.148	0.001	...	68.894	99.002	...	71.219	100.852	...	0
3	99.219	0.001	...	67.328	99.002	...	71.219	100.852	...	0
4	99.109	-0.001	...	65.239	99.002	...	71.219	100.852	...	1
5	98.961	-0.001	...	66.168	99.001	...	71.219	100.852	...	1
6	98.898	-0.001	...	65.565	99.001	...	71.219	100.852	...	0
7	99.055	0.002	...	68.369	99.001	...	71.219	100.852	...	1
8	98.969	-0.001	...	62.129	98.010	...	71.219	100.852	...	0
9	98.979	0.000	...	63.448	98.010	...	71.219	100.852	...	0
10	98.898	-0.001	...	60.120	98.009	...	71.219	100.852	...	1

two classes. It is a shallow model because it is linear, non-hierarchical and it doesn't work well with a lot of features. Applying linear or logistic regression on a large, disparate set of features to predict a label usually fails, because many relationships cannot be captured by a linear model. On the other hand, complicated deep learning algorithms such as LSTM can indeed take into account the time series dependence of the features and labels more readily, but they run a serious risk of data snooping due to a large number of parameters to fit (Chan 2020b).

As we need a model which is not too simple nor complex we choose the ensemble learning model for this study. The nonlinear co-dependences between the predictors can be discovered and utilized by ensemble learning models. For example, maybe when VIX <= 15, the 1-day SPY return isn't useful for predicting the probability of profit of the trade. But when VIX >= 15, 1-day SPY return is very useful. This type of relationship is best discovered using "supervised" hierarchical learning algorithms called ensemble learning models (Chan 2020b).

Ensemble learning models like random forest and light GBM may discover the hypothetical relationship between VIX, 1-day SPY return, . . . , and whether the short vol trade will be profitable as illustrated in figure 9 (Chan 2020b).

We use random forest and light GBM as the secondary model. The secondary model

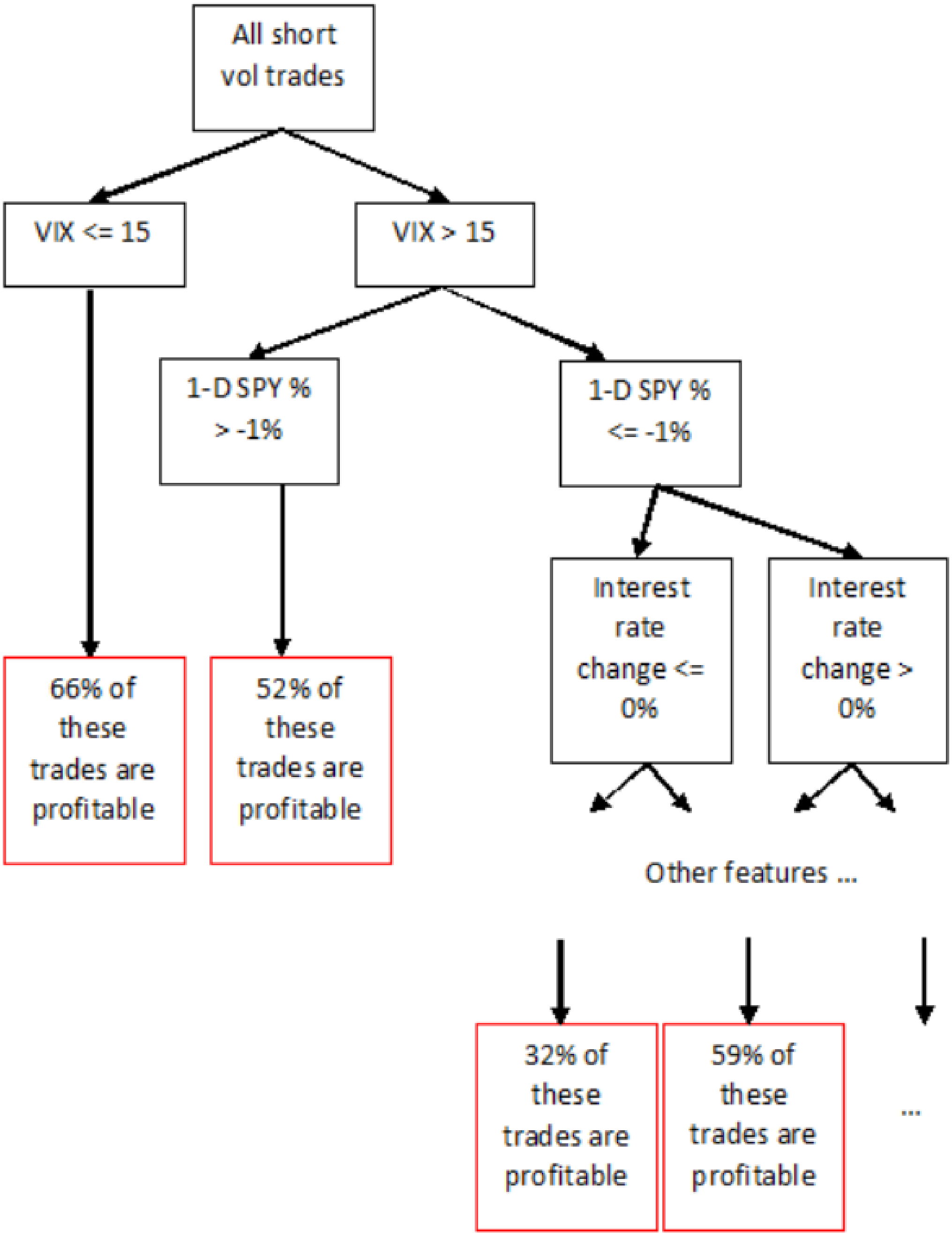

Figure 9. Classification Tree Generated by predictnow.ai (Chan 2020b)

inputs 75 features (17 futures, 17 one-day returns, 14 technical indicators, 27macroeconomic indicators) and meta-label as the target variable predicting whether the primary model will generate profit or not. Table 11 shows the coding logic for training the secondary model.

6.4 Trade Size Prediction

Finding the probability of profit of the next trade is very crucial for any trader. While looking at the historical trades (live or backtest) and counting the winners and losers, can give a percentage of winning trades, say 60%. But, this is not the probability of profit of the next trade (Chan 2020b).

This 0.6 is what may be called an unconditional probability of profit. It is the same for every trade that we will ever make (unless our winning ratio changes significantly in the future), so it is not helpful as a guide to whether to take the next specific trade or not. It can tell whether to trade this strategy in general but it can't do so on a trade-by-trade basis. The latter is the conditional probability of profit. As the adjective suggests, this probability is conditioned on the specific market environment at the time when we expect to trade (Chan 2020b).

For example, we are trading a short volatility strategy which can be algorithmic, or even discretionary. If we are trading it during a very calm market, it is likely that the conditional probability of profit would be quite high. If we are trading during a financial crisis, it could be very low. The conditions that can determine the probability may even be quantifiable by many factors like the level of VIX, the recent SPY returns, the interest rate change, etc. Although these factors were not taken into consideration while building the trading strategy but they may have an impact on the conditional probability of profit (Chan 2020b).

The only known way to compute this conditional probability is machine learning. Random forest and light GBM models provide the class prediction as well as the probability of getting that class prediction. Class prediction gives the most likely outcome of the trade: profit or not and the probability gives the likelihood of that outcome. This probability is the conditional probability of the next trade and can be used to size the trade (Chan 2020b).

The Kelly Criterion is one of the many allocation techniques that can be used to manage money effectively. Kelly criterion helps to find out the optimal fraction of investment to be made in a single bet and consequently in a series of bets when the probability and the net outcomes are known. The Kelly criterion not only works at its finest when we know the actual probability and net income of our bets, but it is also superior to any essentially different strategy when we just know the probability distribution of the returns.

Table 12. The Kelly Criterion Calculation Logic, Last Ten Days

Date	**Daily % log ret of the primary model**	**Meta-label**	**ML model pred**	**Probability (0,1)**	**Kelly criterion calculation**	**Trade size**
9/17/2021	-0.00952	0	1	0.42, 0.58	(0.58 - 0.5) X 2	0.157
9/20/2021	-0.01662	0	1	0.12, 0.88	(0.88 - 0.5) X 2	0.750
9/21/2021	-0.00115	0	0	0.60, 0.40	No Trade	NA
9/22/2021	0.00938	1	1	0.38, 0.62	(0.62 - 0.5) X 2	0.243
9/23/2021	0.01232	1	1	0.49, 0.51	(0.51 - 0.5) X 2	0.012
9/24/2021	0.00175	1	1	0.26, 0.74	(0.74 - 0.5) X 2	0.472
9/27/2021	-0.00287	0	0	0.63, 0.37	No Trade	NA
9/28/2021	-0.02019	0	0	0.56, 0.44	No Trade	NA
9/29/2021	0.00144	1	0	0.53, 0.47	No Trade	NA
9/30/2021	-0.01195	0	0	0.60, 0.40	No Trade	NA

Table 12 demonstrates the calculation logic of bet size using the Kelly criterion. When the secondary model predicts that the primary model will return a profit (in column ML model pred as 1 we initiate a trade. Inputting the probability provided by the machine learning model in the Kelly criteria we calculate the size of the trade.

7 RESULTS

This section presents results for the trading system's efficacy improvements. First, we explore different methods like feature engineering and ensemble learning models to improve the secondary machine learning model.

Then, we determine the strength of a trend of forecasting instruments used in this study using the Average Directional Index (ADX) technical indicator.

Lastly, to validate the system developed in this study, we experiment with multiple target variables; Nasdaq, Nikkei, Euro, and Gas Oil. Additionally, we evaluate system performance on different prediction periods; 2 days, 3 days, 5 days (a week), and 20 days (a month).

7.1 Method Exploration

First, we check the performance of the base model. We use 48 input features (17 futures, 17 one-day returns, 14 technical analysis features), and we use the random forest as the secondary model predicting the next day investment decision for E-mini S&P500 futures. The test accuracy, precision, recall, and F1-score of the base model are shown in figure 10.

```
Accuracy:  0.564
Precision: 0.6052
Recall:    0.4975
F1-score:  0.5461
```

Figure 10. Accuracy, Precision, Recall, and F1-score of the Random Forest Model

The confusion matrix (figure 11) shows the values of the secondary model, profit or not, prediction; the true positive (TP) 350, the true negative (TN) 304, the false-negative (FN) 198, and the false positive (FP) 307 of the 1159 test days in total.

The receiver operating characteristic curve (figure 12) presents the area under the curve (AUC) of 0.58 in value. The result shows a slight value in the secondary model probability prediction compared to the 0.50 value of no skill.

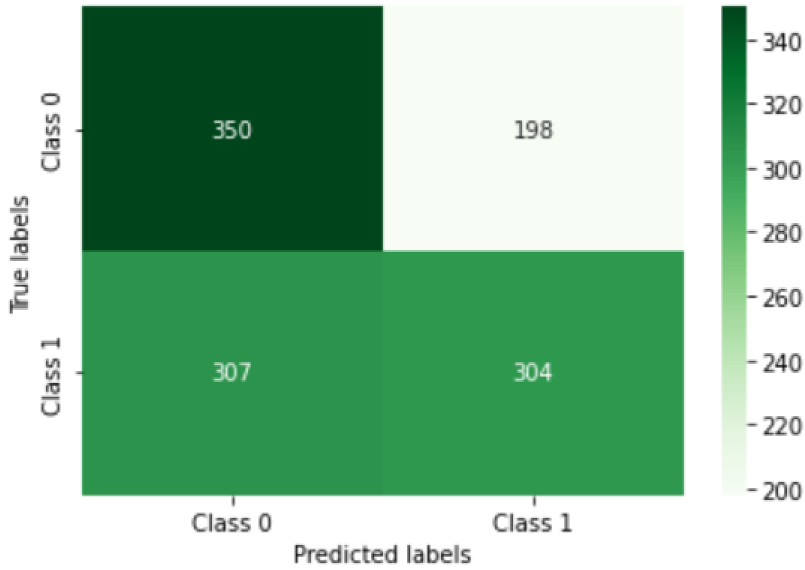

Figure 11. The Confusion Matrix of the Random Forest Secondary Model Profit or Not Prediction

Figure 12. The ROC Curve of the Probability Prediction of the Random Forest model

We assess the effect of feature engineering where we fractionally differentiate future prices and add 27 new macroeconomic indicators as features leading to 75 input features. Feature engineering leads to a decrease in accuracy and precision score while recall and F1-score have improved.

```
Accuracy:  0.542
Precision: 0.5514
Recall:    0.5523
F1-score:  0.5519
```

Figure 13. Accuracy, Precision, Recall, and F1-score of the Random Forest Model after Feature Engineering

The confusion matrix, as shown in Figure 14, shows; the true positive (TP) 301, the true negative (TN) 327, the false-negative (FN) 266, and the false positive (FP) 265. We see improvement in true negative and false positive while true positive and false positive also worsens AUC value drops mildly to 0.57.

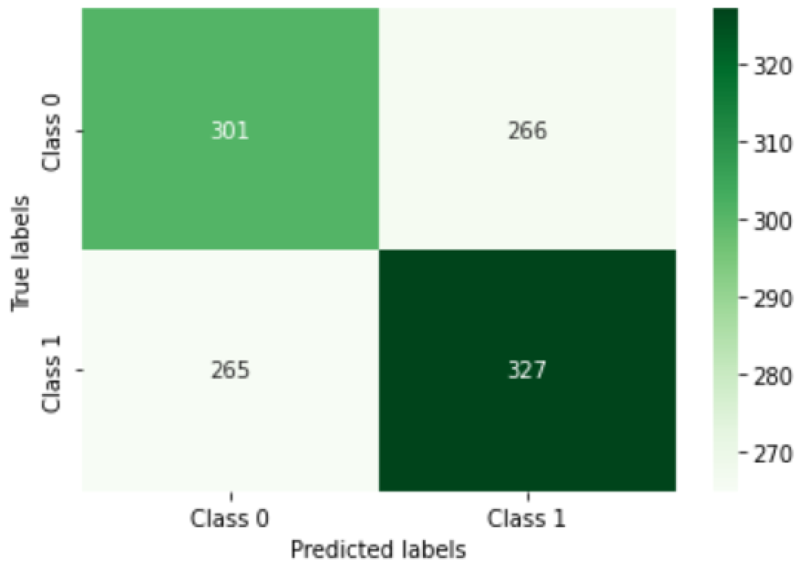

Figure 14. The Confusion Matrix of the Random Forest Secondary Model after Feature Engineering Profit or Not Prediction

In finance, it is essential to find out when to trade as well as when not to make a trade to preserve the capital. Therefore, it is of paramount importance to improve the F1-score. As we saw improvement in the F1-score of the model due to the feature engineering step, we will use it in our next analysis.

To evaluate the effects of different ensemble algorithms, we fit our data set to the random

Figure 15. The ROC Curve of the Probability Prediction of the Random Forest model after Feature Engineering

forest model and the light GBM model with default hyperparameters. Figure 16 shows improvement in accuracy, precision, recall, and especially F1-score against the random forest model.

```
Accuracy:  0.578
Precision: 0.5871
Recall:    0.5861
F1-score:  0.5866
```

Figure 16. Accuracy, Precision, Recall, and F1-score of the Light GBM Model after Feature Engineering

Figure 17 shows the confusion metrics of the light GBM model. True positive (323) and true negative (347) value increases while a false negative (244) and false positive (245) value decreases compare to the random forest model. AUC value also sees significant improvement from 0.57 to 0.61 (Figure 18).

The light GBM model leads to better results across all the measured metrics, we choose the light GBM model as the secondary machine learning model for this study.

To tune hyper parameters of the light GBM model we used GridSearchCV with values for num leaves as 7,12,30,60; max depths as 3,4,5,6,-1; n estimators as 500,1000,5000;

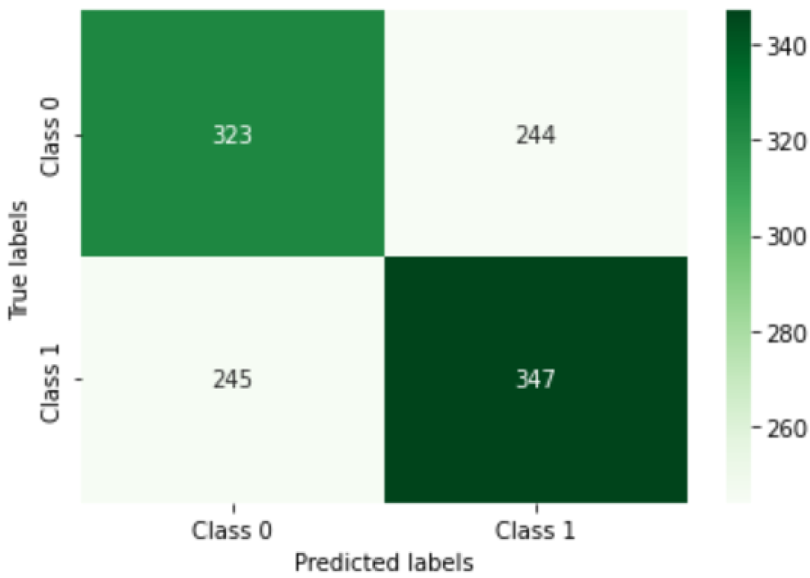

Figure 17. The Confusion Matrix of the Light GBM Secondary Model after Feature Engineering Profit or Not Prediction

Figure 18. The ROC Curve of the Probability Prediction of the Light GBM model after Feature Engineering

learning rates as 0.001, 0.002, 0.0046, 0.01, 0.021, 0.046, 0.1, 0.215, 0.464, 1. Best indicator values were num leaf= 7, max depth = 3, n estimator = 1000 and learning rate = 0.01.

Tuning the hyperparameters marginally improves the recall and F1-score but slightly worsens accuracy and precision (Figure 19). True negative and false positive mildly increase, whereas true positive and false negative worsen slightly (Figure 20). AUC curve value is the same as the light GBM model with default parameters (Figure 21).

```
Accuracy:  0.577
Precision: 0.5847
Recall:    0.5946
F1-score:  0.5896
```

Figure 19. Accuracy, Precision, Recall, and F1-score of the Light GBM Model after Hyper-parameter Tuning

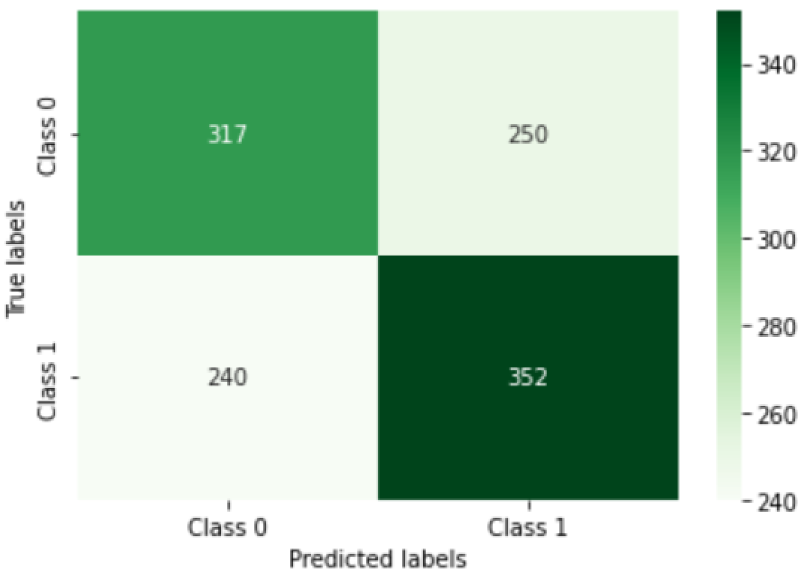

Figure 20. The Confusion Matrix of the Light GBM Secondary Model after Hyper-parameter Tuning Profit or Not Prediction

As there is no enhancement in our model due to hyperparameter tuning, we stayed with the light GBM model with default parameters.

Table 13 summarizes the key evaluation metrics used for the machine learning model; accuracy, precision, recall, and F1-score. Random forest without feature engineering

Figure 21. The ROC Curve of the Probability Prediction of the Light GBM model after Hyper-parameter Tuning

step gives the lowest F1-score. The light GBM model with default parameters gives the highest accuracy and satisfactory F1-score, leading us to choose it as the secondary machine learning model for this study.

Table 13. Method Validation by Machine Learning Metrics for S&P500

Method	Accuracy	Precision	Recall	F1-Score
Random Forest	0.564	0.6052	0.4975	0.5461
Feature Engg (RF)	0.542	0.5514	0.5523	0.5519
Light GBM	0.578	0.5871	0.5861	0.5866
Light GBM HT	0.577	0.5847	0.5946	0.5896

7.2 Trend Strength and Return Distribution

7.2.1 Trend Strength

For all the forecasting instruments we calculate the Average Directional Index indicator and assess how many days in the test period we have a strong trend and a weak or no trend.

Trend distribution by the number of days in the test period of 1159 days for the different target variables is shown in Table 14. Nikkei and Gas Oil futures have a strong trend for the highest number of days respectively. Nasdaq has the highest weak or no trend as per the ADX indicator.

Table 14. Trend Distribution for Different Target Variables for the Test Period

Target Variable	Strong Trend No. of Days	Weak Trend No. of Days
S&P500	364	795
Nasdaq	342	817
Nikkei	495	664
Euro	413	746
GasOil	492	667

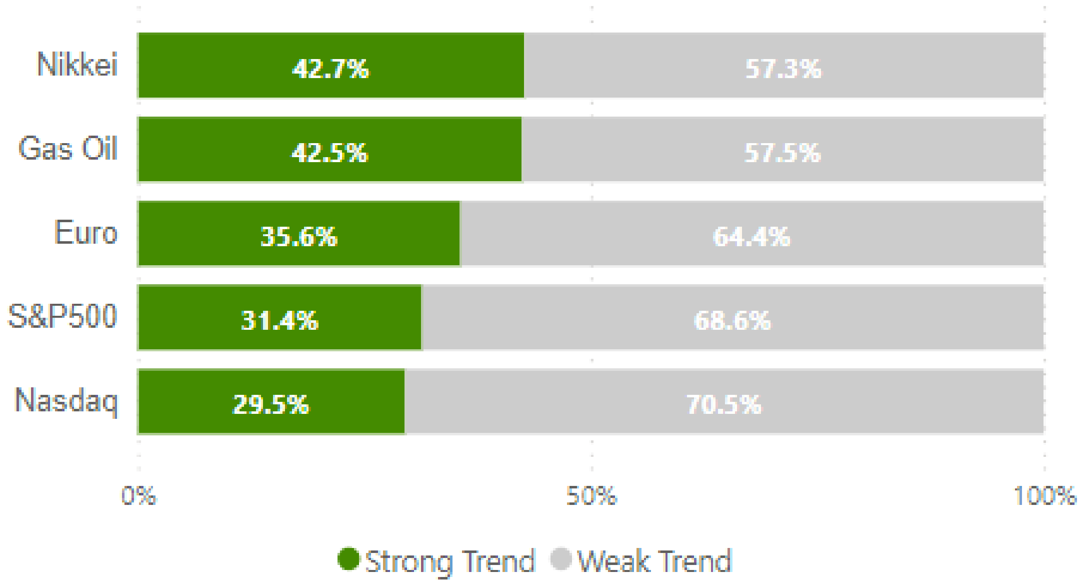

Figure 22. % Trend Distribution for Different Target variables for the Test Period

Figure 22 displays the percentage share of the strong trend and the weak trend for the test periods of all five target variables. The chart has been sorted by the target variables based on the highest share of the strong trend namely Nikkei, Gas Oil, Euro, S&P500, and Nasdaq.

As shown by table 14 and figure 22, we can observe that Nikkei and Gas Oil have a strong trend for a large number of days while Nasdaq has a strong trend for a relatively lower number of days.

7.2.2 Return Distribution

To visualize the return distribution of different forecasting instruments and time horizons, we use the Box and Whisker plot, also called as Box plot. The reason for using a box plot rather than a histogram is that we can show all the target variables in one single plot, which is much easier to read.

In order to examine the dispersion of the return, we compare the interquartile range (the box lengths). The longer the box the more dispersed the return while the smaller the box the less dispersed the return. The extreme values show the overall spread at the end of two whiskers. This shows the range of scores (another type of dispersion). Larger ranges indicate wider distribution, that is, more scattered data. The line in the box indicates the median return value. When the median line within the box is not equidistant from the hinges, the data is skewed. The points outside the ends of the whiskers are outliers.

Figures 23, 24, 25, 26, 27 show the return distribution for different forecasting securities and return periods. The rectangles show the range of returns from the first (the 25th percentiles) to the third (the 75th percentiles). The vertical line inside the rectangles is the median return. The horizontal lines that extend from the top and bottom of the rectangles, up to the "whiskers" (the vertical lines), reflect the returns above and beyond the interquartile range i.e., returns that fall between the first and third performance quartiles, as shown by the rectangles. The outliers are the gray dots at the outer reaches, beyond the whiskers.

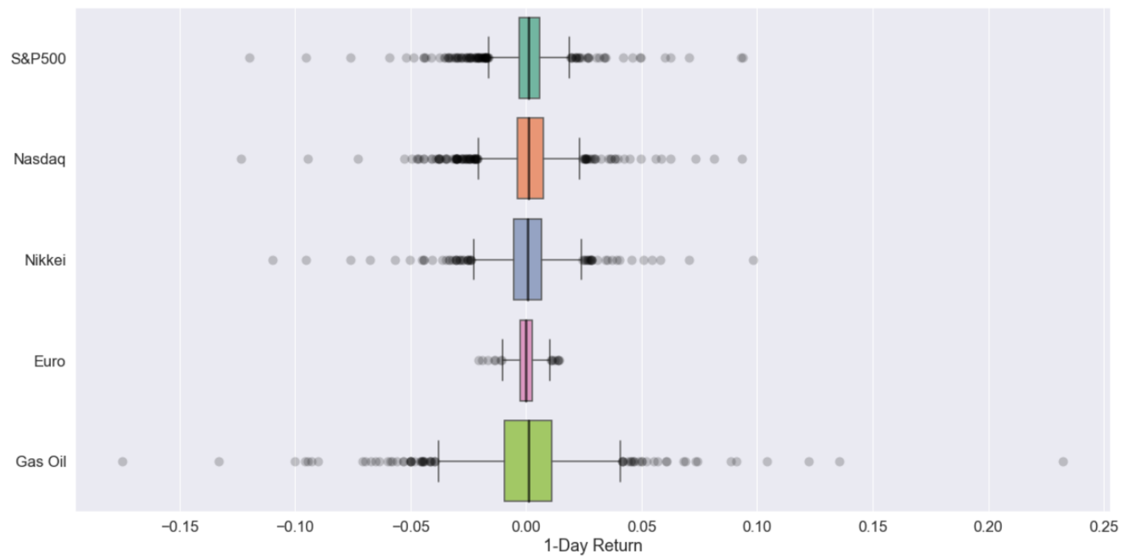

Figure 23. 1-Day Return Distribution for Different Target variables for the Test Period

Figure 23 visualizes the 1-day return distribution for different target variables for the test period. Gas Oil has the highest median value of 0.001245 followeed by Nasdaq, S&P500, Nikkei, and Euro.

Return distribution of Euro is symmetric because median is zero and both half-boxes are the same length. All other target variables are a little left-skewed, meaning fewer returns to the negative side.

Gas Oil has the most dispersed return distribution meaning that it can 'over-perform' and 'under-perform' compare to other target variables. The return distribution range of the Euro is narrow which is understandable as there is not much volatility in currency prices while Gas Oil being a commodity class has wild swings in prices.

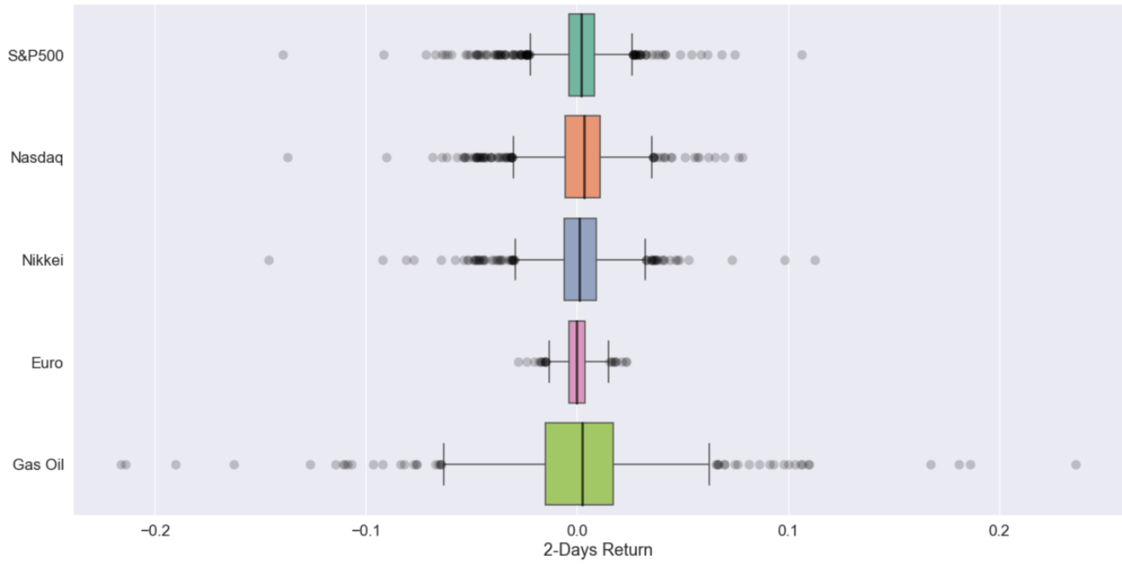

Figure 24. 2-Days Return Distribution for Different Target variables for the Test Period

Nasdaq has the highest median return of 0.003325 meaning greater variability in the return and Euro has the lowest median return of zero implying symmetrical distribution for a 2-days return as shown in figure 24. All the target variables have a higher median value than the 1-day return.

Same as the 1-day and 2-days return distribution of Euro is symmetrical for 3-days return period. Nasdaq has the highest median return of 0.004519 tailed by Gas Oil, S&P 500, Nikkei, and Euro as shown in figure 25.

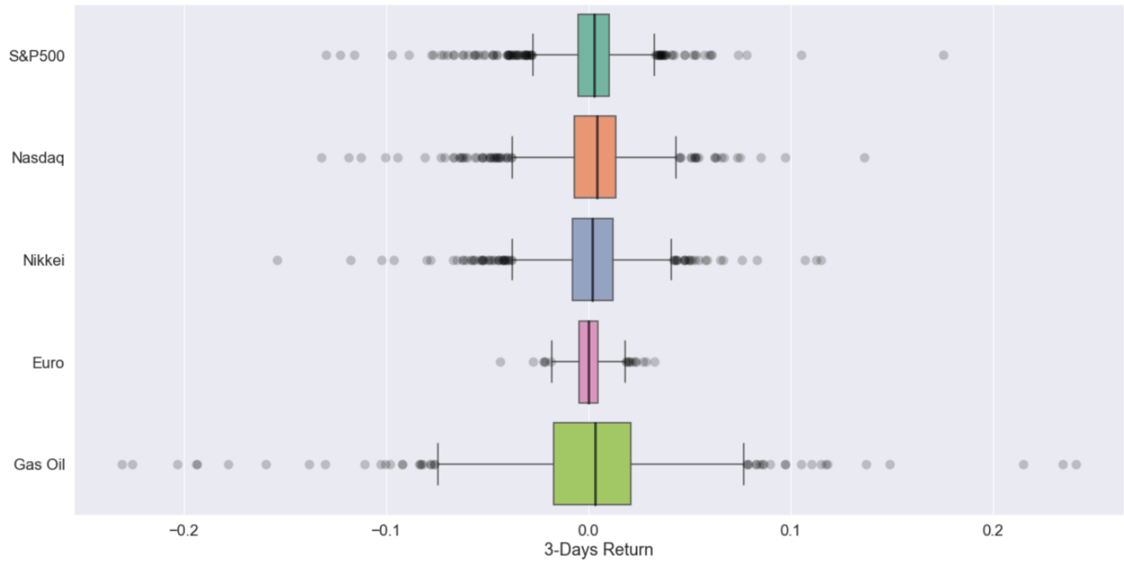

Figure 25. 3-Days Return Distribution for Different Target variables for the Test Period

5-days return distribution for different target variables is shown in Figure 26. Gas Oil returns are widely dispersed followed by Nasdaq, Nikkei, S&P500, and Euro. The highest median value observed by Nasdaq succeeded by S&P500, Gas Oil, Nikkei, and Euro. The median value for Euro is 0.00251 pointing towards an almost symmetrical return distribution.

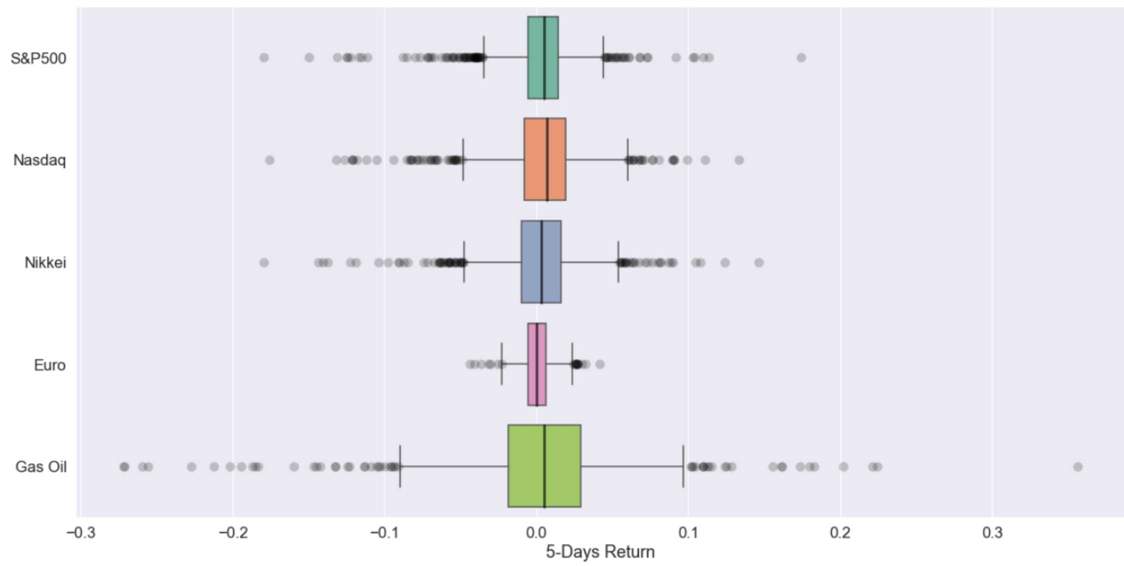

Figure 26. 5-Days Return Distribution for Different Target variables for the Test Period

Figure 27 gives the 20-days return distribution overview. Nasdaq and Gas Oil returns are left-skewed with the median value of 0.025265 and 0.022101 respectively and also S&P500 returns are mildly skewed towards the left. While they are symmetrical for

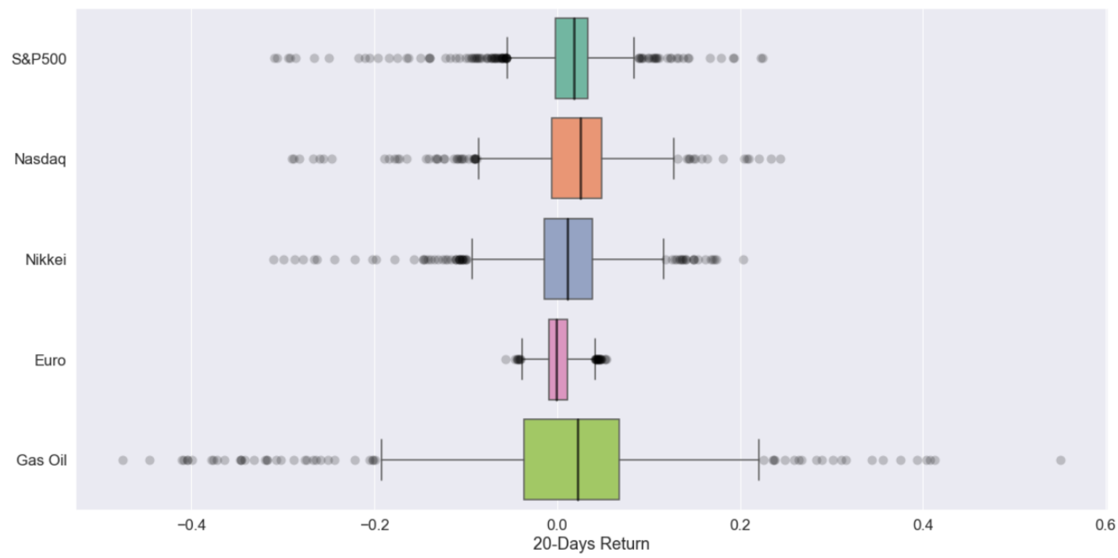

Figure 27. 20-Days Return Distribution for Different Target variables for the Test Period

Nikkei and Euro.

As the time horizon of the return increases, we observe an increase in the dispersion of return distribution. 20-days return shows the highest variability whereas 1-day return shows lower variability. This makes sense as financial instruments tend to have greater price movement for a longer period of time than the shorter time period.

7.3 System Validation

In pursuance of validating the system applied in this study, the experiment was performed on multiple target variables; Nasdaq, Nikkei, Euro, and Gas oil. Furthermore, to observe the effectiveness of the system on different prediction periods we evaluated the performance of various forecasting horizons; 2 days, 3 days, 5 days (a week), and 20 days (a month).

Table 15 shows the machine learning evaluation metrics for various target variables and forecasting periods. The highest accuracy and F1-score were achieved for one-day forward prediction by Nasdaq, S&P 500, and Gas and oil were close second and third respectively. Nasdaq's 20 days prediction period had the lowest accuracy and Euro 2 days prediction had the lowest F1-score. One-day prediction period has higher accuracy and F1-score for all the variables compared to other prediction periods. 20 days prediction

Table 15. Validation of Machine Learning Model

Target	Metric	Prediction Period No. of Days				
		1-D	2-D	3-D	5-D	20-D
S&P 500	F1-Score	0.587	0.531	0.538	0.512	0.503
	Accuracy	0.578	0.51	0.531	0.507	0.484
Nasdaq	F1-Score	0.592	0.521	0.539	0.543	0.469
	Accuracy	0.591	0.5	0.521	0.531	0.459
Nikkei	F1-Score	0.519	0.507	0.497	0.468	0.482
	Accuracy	0.544	0.544	0.518	0.495	0.505
Euro	F1-Score	0.536	0.448	0.479	0.486	0.461
	Accuracy	0.548	0.476	0.515	0.509	0.484
Gas Oil	F1-Score	0.572	0.518	0.552	0.519	0.536
	Accuracy	0.569	0.506	0.545	0.5	0.521

period seems the most difficult to predict as most of the target variables have the lowest accuracy and F1-score for this period. It makes sense, as it is easier to predict for a shorter period than a longer period.

The cumulative return of all the target variables and prediction periods by the strategy are shown in table 16. We compare the return generated by Buy and Hold (buy a stock at the beginning of the test period and sell at the end of the test period), the primary model, the secondary model, and the secondary model using the Kelly criterion.

This study aims to build a system that utilizes machine learning to boost the return generated by the primary model. The buy and hold strategy is the passive investment strategy that aligns with Efficient Market Hypothesis. It states that all known information about investment securities is already factored into the price and an active investor cannot be any more effective than one who buys and holds. It is good to compare the return generated by the buy and hold strategy with the primary and the secondary model return.

The buy and hold strategy outperforms the primary model for S&P500, Nasdaq, Nikkei, and Euro for all the prediction periods. Whereas the return of the primary model for Gas Oil exceeds the buy and hold strategy return.

Compared to the 1-day cumulative return generated by the primary model, the 1-day return of the secondary model and the secondary model using a probability for leverage is higher for S&P500, Nikkei, Euro, and Gas Oil. Gas Oil has the highest rate of return while Nasdaq underperforms the primary Donchian Channel model.

For S&P500, the return of the secondary model is higher than the primary model when the prediction periods are 1-day and 20-days. 2-days, 3-days and 5-days returns of the secondary model are less than the primary model.

In the case of Nasdaq, the secondary model underperforms for all the prediction periods. We only see improvement in the return compared to the primary model for the 20-day period when we use the Kelly criterion for leverage.

The secondary model outperforms the primary model for all the prediction periods for Nikkei. The secondary model generated the highest return for the 2-days period and the lowest for 1 day period. Return of the secondary model using leverage surpasses the secondary model for the period of 1-day, 2-days, and 3 days.

As for Euro, the secondary model performs better than the primary model for all the prediction periods. The highest improvement is observed for the 2-days period and the lowest is for the 20-days period. Usage of leverage gives better results for 1-day, 2-days, 5-days, and 20-days. For the 3-days prediction period, we do not see any significant change in the return due to the use of leverage (Kelly criterion).

Gas Oil returns of the secondary model beat the primary model for prediction periods are 1-day and 20-days. 2-days, 3-days, and 5-days generate subdued returns compared to the primary model for both the secondary model and the secondary model with leverage. Application of Kelly criterion for the prediction periods 1-day and 20-days give a significant boost to the return of Gas and Oil.

To summarize, the secondary model seems to outperform the primary model for all the forecasting instruments except Nasdaq for a 1-day prediction period. For the periods of 2-days, 3-days, and 5-days the secondary model generates better returns only for Euro and Nikkei. The system leads to superior financial performance for the secondary model compared to the primary model for all target variables except Nasdaq when the prediction period is 20-days.

Table 16. Validation of System by Cumulative Return

Target	Strategy	Cumulative Return				
Prediction Period		1 Day	2 Days	3 Days	5 Days (1 Week)	20 Days (1 Month)
SP500	Buy and Hold	0.602	0.602	0.602	0.603	0.613
	Primary Model Donchian Channel	0.4996	0.4996	0.4996	0.5002	0.5108
	Secondary Model (Profit or Not)	0.5943	0.1322	0.1816	0.2660	0.5643
	Secondary Model (Probability)	0.7802	0.2761	0.2116	0.2793	0.7181
Nasdaq	Buy and Hold	0.996	0.996	0.996	0.996	1.008
	Primary Model Donchian Channel	0.4617	0.4617	0.4617	0.4620	0.4735
	Secondary Model (Profit or Not)	0.2266	-0.0701	-0.0728	0.2699	0.4342
	Secondary Model (Probability)	0.2472	-0.2066	-0.0564	0.3698	0.6074
Nikkei	Buy and Hold	0.446	0.446	0.446	0.443	0.467
	Primary Model Donchian Channel	-0.1089	-0.1089	-0.1089	-0.1115	-0.1195
	Secondary Model (Profit or Not)	-0.0157	0.2201	0.0965	0.0716	-0.0472
	Secondary Model (Probability)	-0.0063	0.2389	0.2010	0.0503	-0.0506
Euro	Buy and Hold	0.082	0.082	0.082	0.088	0.085
	Primary Model Donchian Channel	-0.0300	-0.0300	-0.0300	-0.0246	-0.0274
	Secondary Model (Profit or Not)	0.0464	0.0894	-0.0201	0.0195	-0.0229
	Secondary Model (Probability)	0.0723	0.1318	-0.0202	0.0474	-0.0042
Gas Oil	Buy and Hold	0.332	0.332	0.332	0.321	0.302
	Primary Model Donchian Channel	0.4919	0.4919	0.4919	0.5032	0.5214
	Secondary Model (Profit or Not)	0.7082	0.3384	0.3712	0.3157	0.6265
	Secondary Model (Probability)	1.0688	0.3793	0.4744	0.5088	0.9370

The secondary model that uses probability to input in Kelly criterion to generate the trade size gives a better return than the secondary model for most prediction labels and periods. Hence, we can conclude that by utilizing probabilities to generate trade size, we achieved much better financial results.

Table 17 shows the CAGR of S&P500, Nasdaq, Nikkei, Euro, and Gas Oil for the prediction periods of 1-day, 2-days, 3-days, 5-days, and 20-days.

Table 17. Validation of System by CAGR in %

Target	Strategy	CAGR				
Prediction Period		1 Day	2 Days	3 Days	5 Days (1 Week)	20 Days (1 Month)
SP500	Buy and Hold	14.384	14.384	14.384	14.410	14.558
	Primary Model Donchian Channel	9.472	9.472	9.472	9.482	9.516
	Secondary Model (Profit or Not)	10.976	2.821	3.806	5.408	10.559
	Secondary Model (Probability)	13.747	5.605	4.398	5.654	12.911
Nasdaq	Buy and Hold	24.877	24.877	24.877	24.923	25.183
	Primary Model Donchian Channel	8.839	8.839	8.839	8.850	8.922
	Secondary Model (Profit or Not)	4.666	-1.600	-1.666	5.478	8.263
	Secondary Model (Probability)	5.058	-5.023	-1.278	7.277	10.967
Nikkei	Buy and Hold	10.390	10.390	10.390	10.449	10.876
	Primary Model Donchian Channel	-2.595	-2.595	-2.595	-2.573	-2.671
	Secondary Model (Profit or Not)	-0.352	4.473	2.015	1.592	-1.078
	Secondary Model (Probability)	-0.140	4.804	4.091	1.138	-1.157
Euro	Buy and Hold	1.980	1.980	1.980	1.971	1.883
	Primary Model Donchian Channel	-0.556	-0.556	-0.556	-0.566	-0.664
	Secondary Model (Profit or Not)	1.015	1.927	-0.452	0.431	-0.561
	Secondary Model (Probability)	1.569	2.799	-0.454	1.040	-0.141
Gas Oil	Buy and Hold	7.409	7.409	7.409	7.119	7.018
	Primary Model Donchian Channel	9.607	9.607	9.607	9.844	9.871
	Secondary Model (Profit or Not)	12.967	6.985	7.565	6.321	11.531
	Secondary Model (Probability)	18.003	7.730	9.385	9.623	15.989

The result obtained is similar to the cumulative return shown in table 16 except it shows the annualized growth rate. The highest CAGR by the secondary model (12.967%) and for the secondary model that uses leverage (18.003%) is achieved for Gas Oil for a 1-day prediction period. 3-days period for Nasdaq gives the lowest CAGR of -1.666% for the secondary model. The secondary model using the Kelly criterion gives the lowest return for the 2-days period for Nasdaq.

Table 18 shows the Sharpe ratio for the target variables with periods. The Sharpe ratio helps to determine the investment choice to deliver the highest returns while considering risk. The greater a portfolio's Sharpe ratio, the better its risk-adjusted-performance. A negative Sharpe ratio either means the risk-free rate is greater than the portfolio's return, or the portfolio's return is expected to be negative.

The buy and hold strategy outperforms the primary model for all the financial instruments and all the prediction periods. It means that the risk is paying off in the form of above-average returns for buy and hold strategy.

S&P500 has a Sharpe ratio above the buy and hold and the primary model for the period of 1-day and 20-days for the secondary model and the secondary model with leverage. 2-days prediction period has the lowest Sharpe ratio of 0.1842.

Sharpe ratio for the 20 days surpasses the Sharpe ratio of the primary model for Nasdaq. We observed a negative Sharpe ratio for the prediction period of 2-days and 3-days implying that the return is below the risk-free rate meaning negative return.

Nikkei has a negative Sharpe ratio for a 1-day and 20-days period. The Sharpe ratio is highest compared to the primary model for the period of 2-days, followed by 3-days and 5-days. The primary model has a negative Sharpe ratio for all the periods.

The Sharpe ratio is negative for all the periods for the primary model in the case of the Euro. The Sharpe ratio exceeds the primary model for the periods of 1-day, 2-days, and 5-days. Again we have a negative Sharpe ratio for 3-days and 20-days.

Gas Oil's Sharpe ratio significantly exceeds the primary model for a 1-day and 20-days

Table 18. Validation of System by Sharpe Ratio

Target	Strategy	Sharpe Ratio				
Prediction Period		1 Day	2 Days	3 Days	5 Days (1 Week)	20 Days (1 Month)
SP500	Buy and Hold	0.7777	0.7777	0.7777	0.7779	0.7831
	Primary Model Donchian Channel	0.5613	0.5613	0.5613	0.5622	0.5752
	Secondary Model (Profit or Not)	0.9528	0.1842	0.3419	0.4174	0.8514
	Secondary Model (Probability)	0.9682	0.2990	0.3180	0.3407	0.8184
Nasdaq	Buy and Hold	1.0704	1.0704	1.0704	1.0707	1.0775
	Primary Model Donchian Channel	0.4406	0.4406	0.4406	0.4410	0.4529
	Secondary Model (Profit or Not)	0.3076	-0.0907	-0.1029	0.3391	0.6056
	Secondary Model (Probability)	0.2480	-0.2129	-0.0624	0.3717	0.6714
Nikkei	Buy and Hold	0.5739	0.5739	0.5739	0.5758	0.5933
	Primary Model Donchian Channel	-0.1150	-0.1150	-0.1150	-0.1178	-0.1264
	Secondary Model (Profit or Not)	-0.0238	0.3472	0.1717	0.1028	-0.0668
	Secondary Model (Probability)	-0.0072	0.3148	0.2825	0.0578	-0.0574
Euro	Buy and Hold	0.3212	0.3212	0.3212	0.3195	0.3063
	Primary Model Donchian Channel	-0.0983	-0.0983	-0.0983	-0.0804	-0.0900
	Secondary Model (Profit or Not)	0.2115	0.4424	-0.0919	0.0921	-0.1055
	Secondary Model (Probability)	0.2660	0.5218	-0.0731	0.1769	-0.0157
Gas Oil	Buy and Hold	0.3765	0.3765	0.3765	0.3684	0.3654
	Primary Model Donchian Channel	0.3185	0.3185	0.3185	0.3260	0.3384
	Secondary Model (Profit or Not)	0.6362	0.3148	0.3317	0.2873	0.5758
	Secondary Model (Probability)	0.7257	0.2799	0.3237	0.3682	0.6829

period giving us a better risk-adjusted return. It mildly improves for 3 days but decreases for 2-days and 3-days.

S&P500 has the highest Sharpe ratio for the secondary model and the secondary model using the Kelly criterion for the period of 1-day. Gas Oil also has a better Sharpe ratio than both buy and hold and the primary model for 1-day and 20-days. The worse Sharpe ratio is observed for Nasdaq 2-days for the secondary model with Kelly criterion. This insulates that the target variable Gas Oil and S&P500 perform very well after the risk adjustment. On the contrary, Nasdaq performs poorly after the risk adjustments.

Examining the trend strength, return dispersion of prediction securities, the machine learning model performance metrics, cumulative return, CAGR, and the Sharpe ratio we observe that to generate better financial return besides improving the secondary machine learning model performance metrics, we need a target variable which has stronger trend and also negatively skewed (left-skewed) return distribution. All the previously discussed factors can affect the magnitude of the financial return generated by the system developed in this study.

The system seems to work well compared to the primary model on Nikkei and Euro for all the time periods as they had a stronger trend for a relatively higher number of days.

Gas Oil generates a much better return on investment than any other underlying for the prediction period 1-day and 20-days due to higher dispersion of the returns meaning it has many wild price movements leading to a higher return.

S&P500 outperforms the primary model marginally as machine learning performance metrics were satisfactory and returns are favorably distributed for the prediction period 1-day. When the prediction period is 20-days for the same security despite having poor machine learning metrics and lower trend strength we are getting higher returns due to the interquartile range of the return being above zero leading to superior returns.

Nasdaq has the best machine learning model metrics in our study, but it under-performs the primary model due to the lowest trend strength compared to other forecasting labels and highly disbursed return (second to gas Oil).

We can infer from the results obtained here that the system used in this study holds true for various financial instruments and different prediction periods provided the target variable we choose tends to show a strong trend for a longer period of time, return is negatively skewed and has strong machine learning model performance metrics (especially F1-score). If the conditions mentioned above are met, the machine learning model leads to much better financial returns than the primary model.

8 CONCLUSIONS

In this paper, we have explored state-of-the-art techniques for automated decision-making in financial trading. Our work focuses on the use of machine learning for efficacy improvements in detecting trade signals and converting these signals into trades. The examined techniques include meta-labeling, fractional differentiation, ensemble learning, and bet sizing. With each technique independently analyzed.

The devised system is modular by design and uses a hybrid modeling setup. The primary model is a human-understandable rule-based algorithm that is often used to identify price trends during manual trading. However, when traded on the signals created by the primary model are often lead to a sub-par performance.

The results obtained by this study show that machine learning can be an effective tool for risk management. The secondary machine learning model yields superior financial return than the primary Donchian Channel model. The financial instrument has a strong trend for a longer period, return distribution is left-skewed and dispersed, and has satisfactory machine learning model metrics, especially F1-score. Based on the idea of meta-labeling we show that a secondary model can be trained to supervise the primary model. The secondary model performs relatively poorly on the accuracy, and F1-score suggests that the models often disagree. However, when evaluated on decision outcome, the combination of models perform very well.

Examining the results of fractional differentiation, we infer that it improves model performance (F1-score) which is crucial for machine learning in finance. Comparing the random forest algorithm with the light GBM model shows that light GBM significantly boosts the F1-score. In addition to that, we see enhancement in the accuracy as well.

The study shows that the trading strategy system employed in this paper is scalable, repeatable, and interpretable. By changing parameters like target variable and prediction length, we could still reproduce similar results, given that the setup used in this study remains the same.

8.1 Summary of Research Questions

This study is based on the hypothesis that machine learning can be used as a risk management tool to enhance any trading strategy's financial performance. The research questions were developed to test the hypothesis of this study.

Research question 1. Define the architecture of a human-understandable automated decision-making trading system.

We devised a system architecture that is modular by design and uses a hybrid modeling setup. The trading strategy system is designed upon the insights of trading practitioners on the SigTech platform with the Python programming language utilizing their data.

The primary model is a human-understandable rule-based algorithm often used to identify price trends during manual trading. The Donchian Channel primary model generates an entry signal, an exit signal , and no position signal. The trading signals are defined as long (1), short (-1), and no position (0).

The secondary machine learning model predicts whether the primary model will return profit or not rather than predicting the side. Instead, the secondary model predicts whether the primary model succeeds or fails at a particular prediction (a meta-prediction).

If the signal generated by the primary model is long and the daily percentage log return is positive or if the signal is short and the daily percentage log return is negative, we make a profit. On the other hand, if the signal is short and the daily percentage log return is positive or if the signal is long and the daily percentage log return is negative, we make a loss. Once we have the trading strategy's daily percentage log return, we label these returns; if positive as 1, else 0. These labels are called meta-labels which we use as the y values in the secondary model. We utilize meta-labeling to turn our problem into binary classification. The label values are {0,1} as opposed to the primary model signal values {-1,0,1}.

When the secondary model predicts that the primary model will return a profit (labeled as 1) we initiate a trade. Inputting the probability provided by the machine learning model in the Kelly criteria we calculate the size of the trade.

The study shows that the trading strategy system employed in this paper is scalable, repeatable, and interpretable. By changing parameters like target variable and prediction length, we could still reproduce similar results, given that the set-up used in this study remains the same.

Research question 2. How does meta-labeling, fractional differentiation, different ensemble learning algorithms enable and improve the machine learning model in the trading system?

We examined and analyzed techniques including meta-labeling, fractional differentiation, and ensemble learning to determine if they were useful at delivering better forecasts.

Based on the idea of meta-labeling we show that a secondary model can be trained to supervise the primary model. The meta-labeling method trains the secondary model on the prediction outcomes of a primary model. The secondary model performs relatively poorly on accuracy and F1-score, suggesting that the models often disagree. However, when evaluated on decision outcome, the combination of models perform very well. The primary model can be any trading strategy that labels the output as profit or loss and the machine learning algorithm will predict whether the primary model will generate the profit or not.

In this study, we have explored how fractional differentiation can be applied to financial time series data that has memory, i.e., future values depending on the past. The fractional differentiation was useful at providing a framework to achieve stationarity without losing memory in time series. The ADF statistics and p-value of the time series data suggest that fractionally differenced time series retains the maximum amount of memory of the original series while fulfilling the stationary criteria. Examining the results of fractional differentiation, we infer that it improves model performance (F1-score) which is crucial for machine learning in finance. Fractional differentiation is a tool that should be used whenever one knows that the underlying time series has memory.

To evaluate the effect of different ensemble learning algorithms on the forecasting ability of the system developed in this study, we compared the performance metrics between the random forest and the light GBM machine learning model. The results obtained in this study show that compared to the random forest algorithm, the light GBM model significantly boosts the F1-score. In addition to that, we see enhancement in the accuracy when the light GBM model is used as the secondary model as opposed to the random forest model.

The empirical research results of the implemented trading strategy system show that meta-labeling, fractional differentiation, and using the light GBM model can improve in the forecasting ability of the system developed in this study.

Research question 3. Assess the effect of the existence of a strong trend and return distribution on the financial performance of the trading system.

The system's financial results developed in this study show that the secondary model seems to outperform the primary model for all the forecasting instruments except for Nasdaq for a 1-day prediction period. For the periods of 2-days, 3-days, and 5-days the secondary model generates better returns only for Euro and Nikkei. The system leads to superior financial performance for the secondary model compared to the primary model for all target variables except for Nasdaq when the prediction period is 20-days.

We analyzed the impact of the existence of a strong trend in the target instrument and return distribution on the result of this study. We use the Average Directional Index (ADX) to measure the duration of a strong trend for the different forecasting instruments. ADX indicator measures strong or weak trends, it does not tell if the trend is up or down, it just helps to understand the strength current trend.

Nikkei and Gas Oil have a strong trend for a large number of days followed by Euro and S&P500 while Nasdaq has a strong trend for a relatively lower number of days. Comparing the return distribution of various prediction periods shows that as the time horizon of the return increases, we observe an increase in the dispersion of return distribution. The 1-day return shows lower variability in the return distribution in contrast to that 20-days return shows the highest variability.

Nasdaq has the highest median value for all the periods except 1-day. Gas Oil, S&P500, Nikkei, and Euro have the next best median value in the respective order. We observed that the return distribution of the Euro is almost symmetrical for all the prediction periods.

The system seems to work well compared to the primary model on Nikkei and Euro for all the periods as they had a stronger trend for a relatively higher number of days. Gas Oil generates a much better return on investment than any other underlying for the prediction period 1-day and 20-days due to higher dispersion of the returns meaning it has many wild price movements leading to a higher return. In our study, Nasdaq has the best machine learning model performance metrics, but it under-performs the primary model due to the lowest trend strength compared to other forecasting labels and highly disbursed returns. The results obtained show that the existence of the trend and return distribution indeed impact the financial performance of the trading strategy.

8.2 Concluding Discussion

To summarize, based on the implemented trading strategy system, results show that by exploring the ways of constructing a hybrid model it is possible to achieve an explainable solution that is scalability, repeatability, and interpretability. The primary model can be any other technical analysis-based, econometric, or fundamental analysis-based model. The set of features used here can also be changed to any liquid financial instruments, fundamental features, any other features that deem related to the target variable stock. Additionally, as shown in this research the target instrument and prediction period could be switched too. All these parameters can be changed making the setup developed in this thesis flexible and repeatable.

In conclusion, we show support that meta-labeling is an important technique for financial trading. The technique empowers a machine learning algorithm to improve a trading strategy as proposed, and the hybrid modeling provides an improved human understanding of automated decision-making.

On the personal side, the author has learned extremely useful concepts like time series analysis, meta-labeling, fractional differentiation, and machine learning models. This research further helped the author become competent in python programming by acquiring knowledge of modularizing python code. This work has been an excellent introduction to academic research in the field of finance that has been the long-time passion of the author.

8.3 Recommendation for Future Research

This work has further room for improvement by adding more features or performing feature selection. In particular, a feature selection method, cluster-based multivariate discriminant analysis (cMDA), has been argued to improve predictive performance, feature stability, and model interpretability (de Prado 2020). The study (Chan & Man 2020) shows that the cMDA selected features' stability and interpretability are superior to MDA selected features. Therefore, feature selection may further develop and enhance the performance of the trading strategy system.

This work can be further enhanced by experimenting with the different primary models which can be econometric model, fundamental analysis based model, different technical analysis based model (Bollinger Bands, RSI, Moving Average, or any other), or even machine learning model.

REFERENCES

Adithyan, Nikhil. 2021, *Algorithmic Trading with Average Directional Index in Python*. Available at Wikipedia https://medium.com/codex/algorithmic-trading-with-average-directional-index-in-python-2b5a20ecf06a.

ADX, Wikipedia. 2021, *Average directional movement index — Wikipedia, The Free Encyclopedia*. Available at Wikipedia https://en.wikipedia.org/wiki/Average_directional_movement_index.

Basel, Switzerlan d: Financial Stability Board. 2017, *Artificial Intelligence and Machine Learning in Financial Services–Market Developments and Financial Stability Implication, Technical Report*. Available at https://www.fsb.org/wp-content/uploads/P011117.pdf.

Breiman, L. 1996, Bagging predictors, *Machine learning*, vol. 24, no. 2, pp. 123–140.

Breiman, L. 2001, Random forests, *Machine learning*, vol. 45, no. 1, pp. 5–32.

Chan, Ernest. 2006, How much leverage should you use?, available at http://epchan.blogspot.com/2006/10/how-much-leverage-should-you-use.html.

Chan, Ernest. 2019, *Machine learning In trading Q&A*, [Quant Insti Webinar]. Available at https://blog.quantinsti.com/machine-learning-webinar-11-june-2019/.

Chan, Ernest. 2020a, *An interview with Dr. Ernest Chan*, [CSSA]. Available at https://cssanalytics.wordpress.com/2020/11/09/an-interview-with-dr-ernest-chan/.

Chan, Ernest. 2020b, What is the probability of profit of your next trade?, available at http://epchan.blogspot.com/2020/07/what-is-probability-of-profit-of-your.html.

Chan, Ernest & Man, Xin. 2020, Cluster-based Feature Selection, *Market Technician*, vol. 90, , pp. 11–22.

Dietterich, T. 2000, Ensemble Methods in Machine Learning, *Multiple Classifier Systems. MCS 2000*, vol. 1857, , pp. 1–15.

Doshi-Velez, B. Kim. 2017, Towards a rigorous science of interpretable machine learning, *arXiv preprint arXiv:1702.08608*.

European Commission. 2021, *Regulatory framework proposal on artificial intelligence*. Available at https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai.

Faith, Curtis. 2007, *Way of the Turtle: The Secret Methods that Turned Ordinary People into Legendary Traders*, McGraw-Hill.

Guozhu Dong, Huan Liu. 2018, *Feature Engineering for Machine Learning and Data Analytics*, CRC Press/Taylor & Francis Group.

Hosking, J. R. M. 1981, Fractional differencing, *Biometrika*, vol. 68, no. 1, pp. 165–176.

Hudson & Thames. 2020, Meta labeling (a toy example), available at https://hudsonthames.org/meta-labeling-a-toy-example/.

Jansen, Stefan. 2020, *Machine learning for algorithmic trading, 2nd edn.*, Packt Publishing, available at https://www.perlego.com/book/1694603/machine-learning-for-algorithmic-trading-pdf.

Joseph, Manu. 2020, The gradient boosters IV: lightgbm, available at https://deep-and-shallow.com/2020/02/21/the-gradient-boosters-iii-lightgbm/.

Joubert, Jacques. 2022, Meta-Labeling: Theory and Framework (Pre-print), *Available at SSRN 4032018*.

K. Guolin, T. Finley T. Wang W. Chen W. Ma et al., Q. Meng. 2017, Lightgbm: A highly efficient gradient boosting decision tree, *Advances in neural information processing systems*, vol. 30, , pp. 3146–3154.

Kelly, J. L. 1956, A new interpretation of information rate, *The Bell System Technical Journal*, vol. 35, , p. 917–926.

Kissell, Robert. 2013, *The science of algorithmic trading and portfolio management*, Academic Press.

Noack, A. & Nielsen, M. 2014, A fast fractional difference algorithm, *Journal of Time Series Analysis*, vol. 35, no. 5, pp. 428–436.

Nousiainen, Petri. 2021, *Exploration of a trading strategy system based on meta-labeling and hybrid modeling using the SigTech Platform*, Master's thesis, Arcada University of Applied Sciences.

P., Newbold & Granger, C.W.J. 1973, Spurious regressions in econometrics, *Journal of Econometrics*, vol. 2, no. 2, pp. 111–120.

Peter Bartlett, Wee Sun Lee Robert E. Schapire, Yoav Freund. 1998, Boosting the margin: a new explanation for the effectiveness of voting methods, *The Annals of Statistics*, vol. 26, no. 5, pp. 1651–1686.

de Prado, Marcos Lopez. 2018, *Advances in financial machine learning, 1st edn.*, Wiley, available at https://www.perlego.com/book/993133/advances-in-financial-machine-learning-pdf.

de Prado, Marcos Lopez. 2020, *Machine learning for asset managers, 1st edn.*, Cambridge University Press.

Ritika Chopra, Gsgsn Deep Sharma. 2021, Application of Artificial Intelligence in Stock Market Forecasting: A Critique, Review, and Research Agenda, *Journal of Risk and Financial Management*, vol. 14, no. 11, p. 526.

Schaap, Candy. 2021, *ADX: The Trend Strength Indicator*. Available at https://www.investopedia.com/articles/trading/07/adx-trend-indicator.asp#citation-2.

Sharpe, W. F. 1966, Mutual Fund Performance, *The Journal of Business*, vol. 39, no. 1, pp. 119–138.

Sinan Ozdemir, Michael Smith, Divya Susarla. 2018, *Feature Engineering Made Easy. 1st edn.*, Packt Publishing, available at https://www.perlego.com/book/578772/feature-engineering-made-easy-pdf.

Singh, Navdeep. 2020, Donchian channels in technical analysis trading guide, available at https://bit.ly/3HMvSSU.

Thorp, E. O. 1969, Optimal gambling systems for favorable games, *Review of the International Statistical Institute*, vol. 37, , pp. 273–293.

Tsay, Ruey S. 2005, *Analysis of financial time series*, vol. 543, John Wiley & sons.

V.Belle, I. Papantonis. 2021, Principles and practice of explainable machine learning, *Frontiers in big Data*, p. 39.

Verma, Yugesh. 2021, *Complete Guide To Dickey-Fuller Test In Time-Series Analysis*. Available at shorturl.at/yASTU.

Walasek, R. & Gajda, J. 2021, Fractional differentiation and its use in machine learning, *International Journal of Advances in Engineering Sciences and Applied Mathematics*, vol. 13, , pp. 270–277.

Wang, Jar-Long & Chan, Shu-Hui. 2006, Stock market trading rule discovery using two-layer bias decision tree, *Expert Systems with Applications*, vol. 30, , pp. 605–611.

Wikipedia. 2021a, *Confusion Matrix — Wikipedia, The Free Encyclopedia*. Available at Wikipedia https://en.wikipedia.org/wiki/Confusion_matrix.

Wikipedia, CAGR. 2021b, *Compounded Annual Growth Rate — Wikipedia, The Free Encyclopedia*. Available at Wikipedia https://en.wikipedia.org/wiki/Compound_annual_growth_rate.

Wilcox, Paul. 2019, A Quantamental Approach Using Labeling for Stock Trading, available at https://lucenaresearch.com/2019/03/27/quantamental-approach-to-stock-trading/.

Wilder, J. Welles. 1978, *New Concepts in Technical Trading Systems*, Trend Research.