University of Warsaw | Faculty of Economic Sciences

Working Papers
No. 10/2024 (446)

Predictive Modeling of Foreign Exchange Trading Signals Using Machine Learning Techniques

Sugarbayar Enkhbayar
Robert Ślepaczuk

Warsaw 2024

ISSN 2957-0506

UNIVERSITY OF WARSAW
FACULTY OF ECONOMIC SCIENCES
WORKING PAPERS

Predictive modeling of foreign exchange trading signals using machine learning techniques

Sugarbayar Enkhbayar[a], *Robert Ślepaczuk*[b]*

[a] *University of Warsaw, Faculty of Economic Sciences, Quantitative Finance Research Group*
[b] *University of Warsaw, Faculty of Economic Sciences, Quantitative Finance Research Group, Department of Quantitative Finance and Machine Learning*

* *Corresponding author: rslepaczuk@uw.edu.pl*

Abstract: This study aimed to apply the algorithmic trading strategy on major foreign exchange pairs and compare the performances of machine learning-based strategies and traditional trend-following strategies with benchmark strategies. It differs from other studies in that it considered a wide variety of cases including different foreign exchange pairs, return methods, data frequency, and individual and integrated trading strategies. Ridge regression, KNN, RF, XGBoost, GBDT, ANN, LSTM, and GRU models were used for the machine learning-based strategy, while the MA cross strategy was employed for the trend-following strategy. Backtests were performed on 6 major pairs in the period from January 1, 2000, to June 30, 2023, and daily, and intraday data were used. The Sharpe ratio was considered as a metric used to refer to economic significance, and the independent t-test was used to determine statistical significance. The general findings of the study suggested that the currency market has become more efficient. The rise in efficiency is probably caused by the fact that more algorithms are being used in this market, and information spreads much faster. Instead of finding one trading strategy that works well on all major foreign exchange pairs, our study showed it's possible to find an effective algorithmic trading strategy that generates a more effective trading signal in each specific case.

Keywords: machine learning, algorithmic trading, foreign exchange market, rolling walk-forward optimization, technical indicators

JEL codes: C4, C14, C45, C53, C58, G13

Acknowledgments: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors

INTRODUCTION

If a coin flip is used to make trading decisions, can it beat the benchmark? In quantitative finance, alpha represents the excess return generated by an investment over the relevant benchmark (Tulchinsky, 2019). The ultimate goal of most quantitative traders is to discover alpha using various methods such as statistics, econometrics, and machine learning. However, the characteristics of financial instruments and market participants make it more challenging to make optimal trading decisions. Many research studies showed that the distribution of price returns for most financial instruments closely resembles a Gaussian distribution, exhibiting leptokurtosis[1] characteristics. This means that the market is more likely to experience extreme positive or negative returns. Extreme negative returns significantly degrade Buy & Hold returns. On the other hand, human personality and emotions can greatly influence trading decisions and lead to unexpected losses. Because of these challenges, traditional long-only models are no longer effective. Therefore, algorithm-based trading systems have been widely adopted in the financial sector in recent decades.

The main objective of this study is to develop an algorithmic trading system using various machine learning models for major forex pairs using daily and intraday data. Algorithmic trading systems work by making automated buy and sell decisions based on the predictions of machine learning models. A system is defined as a collection of components functioning together as parts of a mechanism or interconnected network. Algorithmic trading systems help reduce errors caused by human emotions and time lag issues. Another objective of this study is to compare the performance of Machine Learning (ML) based strategies and traditional Trend Following (TF) strategies with a benchmark (B) strategy using economic and statistical significance tests. An independent t-test was used to compare the mean returns of different trading strategies. The mean returns are derived from observed returns for all out-of-sample periods. However, there are times when a strategy's return is positive and the benchmark's return is negative over a while. In such cases, it is inappropriate to rely only on statistical tests to compare the performance of trading strategies. Therefore, the Sharpe ratio (SR) was additionally considered as a measure of economic significance. In quantitative finance, strategies with higher SRs are preferred over strategies with higher mean returns. This is because the SR metric includes risk adjustment using the standard deviation (volatility) of return (Sharpe, 1966). Furthermore, a strategy with a high SR allows for higher returns at the end of the period compared to a strategy with a high mean return. In addition, the annualized Sharpe ratio (ASR) was used to compare multiple trading strategies based on different return methods and data frequencies.

[1] the distribution of price return have heavy tails when compared to normal distribution

To clarify the purpose of this study, the following hypotheses and research questions were examined:

- **Research Hypothesis 1. RH1. (ML vs. B)**: Machine Learning models produce trading strategies with a higher ASR and statistically higher mean returns than benchmark strategies.
- **Research Hypothesis 2. RH2. (TF vs. B)**: Trend Following models produce a trading strategy with a higher ASR and statistically higher mean return than the benchmark strategy.
- **Research Hypothesis 3. RH3. (ML with NN vs. ML without NN)**: Machine Learning models using Neural Network (NN) generate trading strategy with a higher ASR and statistically higher mean return than Machine Learning models that are not based on NN.

- **Research Question 1. RQ1. (Different Frequency)**: Is there any difference between using daily and intraday data in the performance of trading strategies based on machine learning?
- **Research Question 2. RQ2. (Different Trading Signals)**: Is there any difference in the performance of a machine learning-based trading strategy between providing two types of trading signals (buy and sell) and only one type of trading signal (only buy or only sell)?
- **Research Question 3. RQ3. (Different Return Methods)**: Is there any difference in the performance of machine learning-based trading strategies between using simple-return and log-return?

Research hypothesis 1 focused on comparing individual and integrated machine learning-based trading strategies to benchmark strategies. For individual strategies, the ASR and mean return were obtained from each machine learning model. However, for integrated strategies, the ASR and mean return were obtained by averaging the strategies in the same category, as shown in Table 2. The superior performance of neural network models (ANN, LSTM, GRU) compared to other machine learning models has been consistently observed in research articles related to algorithmic trading (Islam and Hossain, 2021). This is because these models use a large number of parameters such as weight, bias, number of layers, and number of units, which are facilitated by powerful computers. Therefore, the decision to include this observation was the reason for research hypothesis 3. A trading system that only provides buy or sell trading signals suffers from survivorship bias. The key caveat regarding survivorship bias is that our analysis is limited to non-bankruptcy stocks. This bias will weaken the performance of strategies that generate only sell signals and increase the performance of trading strategies that focus on only buy signals. To reduce the survivorship bias, all trading systems are evaluated in three scenarios: buy and sell, only buy, and only sell. Researchers often prefer to use log returns over simple returns for several reasons such as normality, mathematical convenience, and consistency. Miskolczi (2017) investigated how the choice of return type (simple and log return) affects the risk of the asset portfolio based on Hungarian daily stock prices from 2005 to 2015. They point out that the choice of return type can affect the riskiness of an asset portfolio. Therefore, Research Question 3 was formulated based on this observation.

In this study, daily and 4-hour frequency price data for the following six major currencies (EURUSD, GBPUSD, USDCHF, USDCAD, AUDUSD, NZDUSD) were downloaded from the ICmarket broker. The reason for choosing a major currency pair is their liquidity. These major pairs are linked to the US dollar and are among the most actively traded currencies. Two different methods were used for generating the target variable: simple return and log return. Additionally, our features include technical and statistical indicators. Technical indicators that are commonly mentioned in research articles and familiar to quantitative traders have been selected. Default settings were used for each technical indicator. Each trading strategy provides a predicted return and is converted into a trading signal using thresholds determined from the training set.

The main contributions and novelties of this study are as follows:

- Train and optimize various machine learning models for various currency instruments.
- Implement a rolling walk-forward optimization to solve the overfitting problem.
- Transform the output of machine learning models into trading signals using generation rules with specific thresholds.
- Implement a more realistic backtesting process that takes into account factors such as initial deposits, transaction costs, and trade volume.
- Perform economic and statistical significance tests to compare the performance of individual and integrated trading strategies.

The structure of the rest of this study is as follows. Section 1 provides a literature review of related studies. In Section 2, datasets and their statistics are described. Section 3 describes the implementation of the proposed methods in detail. In Section 4, empirical results will be described. The conclusion of the study will be presented in the last section.

1 LITERATURE REVIEW

In the 6th century BC, people relied on the barter system to trade, exchanging goods they needed for goods needed by others. Subsequently, gold coins emerged as the primary medium of trade, despite the difficulty of transportation. As a result, countries adopted the gold standard in the 1800s, allowing governments to redeem paper money for gold. However, the defeated European Union countries in World War I abandoned the gold standard to increase the money supply for war expenses. After World War II, the United Nations Monetary and Financial Conference was held in Bretton Woods, New Hampshire, USA, and adopted the policy of pegging the US dollar and all currencies to gold. However, in 1971 this system was discontinued and officially replaced by the free-floating system (Meier, 1971). Since then, empirical studies have been conducted on foreign exchange markets. This change helped to start online trading in the 1990s, and research articles have highlighted that the structure of the currency market has changed and become more complex.

Due to the rapid development of the foreign exchange market, it became the largest financial market in terms of daily turnover. It operates as an over-the-counter (OTC) market, active 24 hours a day, Monday through Friday. The main participants in the foreign exchange trading market include corporations, commercial banks, exchange brokers, and central banks (Jeffrey and Kenneth, 1990). The exchange rate indicates how many units of the quoted currency are equal to one unit of the base currency. Exchange rates are determined by the supply and demand of both the base and quoted currencies. According to traditional economic theory, the supply and demand of base and quoted rates are influenced by the monetary and fiscal policies of both countries. Fundamental analysis is the field that interprets changes in exchange rates in terms of changes in fundamental economic indicators. However, this concept began to lose its market influence after the 1980s.

The United States raised interest rates significantly between 1981 and 1984 and it attracted a large number of investors. According to the fundamental indicator theory, the dollar index increased strongly during this period. However, between 1984 and 1985, the real interest rate began to decline, indicating a downward trend in the U.S. dollar index along with other macroeconomic indicators. Surprisingly, despite these trends, the U.S. dollar index increased an additional 20 percent (Jeffrey and Kenneth, 1990). This interesting observation highlights an important point: traditional macroeconomic variables may have difficulty predicting short-term fluctuations in exchange rates, and may even fail to explain the causality behind such movements. This doubtful fact in the theoretical predictions of exchange rates is explained by factors such as high volatility, near-perfect efficiency, and the effect of noise in the currency market. Since the publication of this study, numerous research papers have been published, indicating that the influence of fundamental analysis in the currency market is declining and technical analysis is becoming more and more important in making trading decisions. Mark and Helen (1992) surveyed over 200 senior London-based forex dealers and found that 90 percent of these dealers consider technical analysis very important and incorporate it to varying degrees in their trading decisions. Some senior dealers emphasized technical analysis over fundamental analysis in short-term, while others used technical analysis to complement the results of fundamental analysis. In addition, Richard and Lee (1993) compared the performance of technical strategy based on moving averages with a bootstrapping strategy using data on the British pound, Canadian dollar, Japanese yen, and Swiss franc futures contracts from 1976 to 1990. They found that the technical trading strategies based on the moving average outperformed the bootstrapped simulation method for all futures contracts considered in terms of profitability and statistical significance. Charles Dow's development of the Dow Theory further expanded the application and scope of technical analysis. For instance, Brown et al. (1998) noted that Dow theory is consistently effective based on historical evidence, casting doubt even on the random walk and efficient market theory.

The basic concepts of machine learning models were first proposed in the 1950s, and since then many new models have been developed and refined with additional ideas. Due to the rapid development of these machine learning models, a large number of research papers were published in the 2000s that applied these models to financial trading decisions. For example, Yao and Tan (2000) used a neural network model to predict the weekly close price of five major currencies from 1984 to 1995 based on the following moving average MA5, MA10, MA20, MA60, and MA120 technical indicators. Trading signals were generated based on the predicted values and compared to two different benchmark strategies buy and hold, trend following strategy. For most currencies, buy-and-hold strategies outperformed trend-following strategies, but neural network-based strategies were superior to both buy-and-hold strategy and trend-following strategies. Similarly, Yu et al. (2007) summarized 45 papers on foreign exchange rate forecasting using artificial neural network (ANN) models and noted that although ANN can effectively predict foreign exchange rates, there were some negative results. However, they pointed out that the prediction performance can be improved by changing the factors affecting the performance of the ANN model. In addition, Nayak et al. (2019) applied support vector machine (SVM) and k-nearest neighbors (KNN) models for daily, weekly, and monthly exchange rates of Indian rupees with USD, GBP, and EUR, using multiple optimization methods (grid search, random search, binary genetic algorithm, particle swarm optimization, ant colony optimization, firefly algorithm, BAT optimization) combined for prediction. The research showed that BAT-SVM and BAT-KNN models provided the most accurate predictions in most cases.

However, with the widespread use of algorithmic trading systems based on computer algorithms, some researchers have shown that the rate of assimilation of news in the currency market has increased and become more efficient. For instance, Qi and Wu (2006) investigated the profitability of 2,127 technical indicator rules categorized into four groups: filter, moving average (MA), range break, and channel break, applied to seven currency pairs with daily USD data from 1973 to 1988. They pointed out that profitability has declined significantly in recent years. Chaboud et al. (2013) investigated the market impact of the growth popularity of algorithmic trading based on data from 2003 to 2007. As algorithmic trading became more active, it was observed that the opportunities for profit in the market decreased. In addition, this growth in algorithmic trading has helped reduce extreme volatility in the currency market, and studies have shown that algorithmic trading can quickly adjust exchange rates to new information, thereby improving market efficiency. Hsu et al. (2016) explored 21,000 trading strategies based on technical indicators to develop profitable trading strategies using daily data from 30 forex pairs. These strategies were divided into five groups: oscillator rule, filter rule, moving average rule, support resistance rule, and channel rule, according to the price combination of two groups: developed and emerging markets. Sharpe ratio and excess mean return metrics were utilized to compare these trading strategies. The results showed that moving average rules performed best for developed markets while filtering rule and support resistance rules performed best for emerging markets. Similarly, Fičura (2017) investigated the performance of the neural network (NN), k-nearest neighbors (KNN), and ridge regression models based on 10 exchange rate combinations between 1999 and 2015. They used features such as momentum (MOM), rate of change (ROC), relative strength index (RSI), commodity channel index (CCI), and stochastic oscillator, along with principal component analysis (PCA) for dimensionality reduction. Results showed that linear regression and KNN models with Manhattan distance are more efficient for most currency combinations. Moreover, it is emphasized that there is no single model that fits all combinations and each currency pair requires a different model.

In the 2010s, with the advancement of graphics processing units (GPUs) and the availability of large datasets, deep learning models with multiple layers and nonlinear hidden units became widely used in algorithmic trading. Islam and Hossain (2021) aimed to forecast the next 10 and 30-minute close prices for EURUSD, GBPUSD, USDCAD, and USDCHF between 2017 and 2020 using LSTM and GRU models. Additionally, they used a hybrid LSTM-GRU model consisting of two layers: the first layer was a GRU model with 20 neurons, and the second layer was an LSTM model with 256 neurons. The features were four statistical indicators: momentum, average price, range, and OHLC price. The results indicated that the hybrid GRU-LSTM model exhibits lower prediction errors (MAE, MSE, RMSE) compared to the GRU or LSTM models when used independently. Grudniewicz and Ślepaczuk (2023) compared performances of trading strategies based on different machine learning models. They used Neural Networks, K Nearest Neighbor, Regression Trees, Random Forests, Naive Bayes Classifiers, Bayesian Generalized Linear Models, and Support Vector Machines to predict returns of WIG20, DAX, S&P500, and CEE indices. These data covered the period from January 2002 to March 2023. Simple Moving Average (SMA), Moving Average Convergence Divergence (MACD), Stochastic Oscillator (SO), Relative Strength Index (RSI), and Williams' Percent Range (WPR) were considered as features. The results indicated that algorithmic strategies outperformed benchmark strategies in terms of risk-adjusted returns. Moreover, the best-performing models were the linear support vector machine and the Bayesian generalized linear model. Similarly, Yıldırım et al. (2021) investigated a novel hybrid model for predicting the directional movement of the EUR/USD currency pair. This hybrid model combines fundamental and technical data through two separate LSTM models: ME-LSTM and TI-LSTM. ME-LSTM incorporates fundamental data such as the Federal Reserve Fund Rate, inflation rates in the EU and the US, and closing values of the SP500 and DAX indices. TI-LSTM utilizes technical indicators including moving averages (MA), moving average convergence divergence (MACD), rate of change (ROC), relative strength index (RSI), Bollinger Bands (BB), and commodity channel index (CCI). The dataset covered the period from 2013 to 2018, with 80 percent training and 20 percent testing data. Their models were designed to predict whether the EUR/USD would rise or fall in the next day, three days ahead, and five days ahead. They used intelligent decision rules to effectively combine ME-LSTM and TI-LSTM predictions. The evaluation metric was the Sharpe ratio, which evaluated both profitability and accuracy. Their hybrid model showed better performance across all prediction horizons compared to individual LSTM models, indicating its effectiveness in forecasting directional movements in the forex market. Moreover, in recent decades LSTM and GRU models have been employed not only for analyzing exchange rates but also for other types of financial instruments such as bitcoin, stocks, bonds, and options. Kryńska and Ślepaczuk (2023) compared the performance of algorithmic trading strategies on daily, hourly, and 15-minute data for Bitcoin and the S&P 500 index. They combined LSTM models with a walk-forward approach to generate algorithmic signals for regression and classification problems. Research results showed that classification models produce more efficient trading strategies than regression models. Moreover, they found that regression models are more efficient when using daily data, while classification models are better when using intraday data.

To summarize, following the transition to the free-float system, fundamental indicators mainly were used in the currency market. However, technical indicators gained prominence over fundamental ones due to their inability to consistently explain exchange rate fluctuations. Throughout the evolution of machine learning, numerous new models have been applied to the currency market. Despite the currency market becoming more complex and efficient with the rapid advancement of machine learning, investigating its profitability remains a compelling challenge for academic research.

2 DATA DESCRIPTION

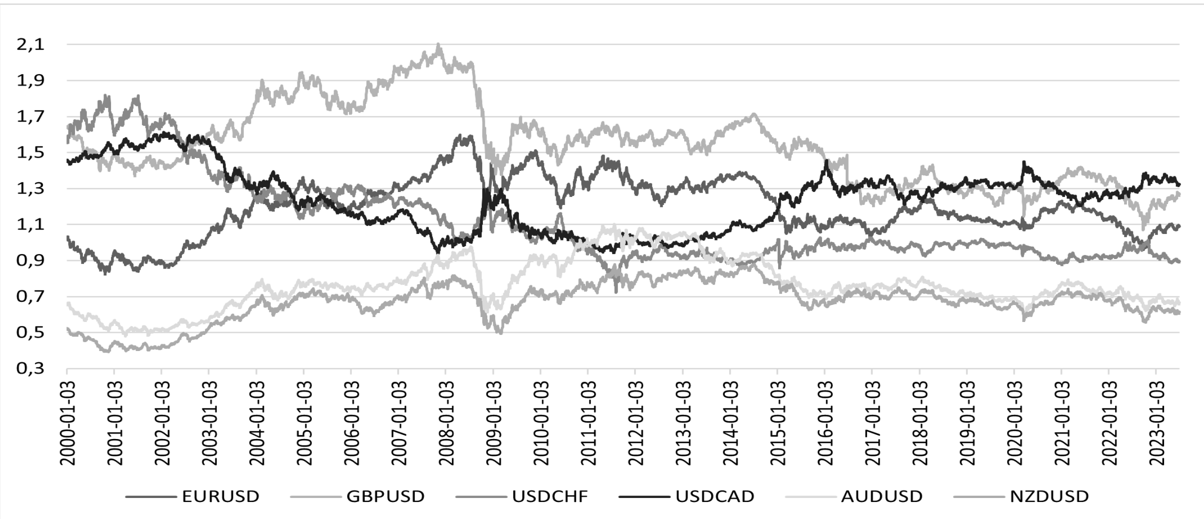
The currency market does not have a centralized place where trades are made at various points. Therefore, using the historical data of the broker we plan to trade with may lead to more realistic performance. In this study, the data of all currency pairs were downloaded from the IC-Market broker. The data for each currency pair includes the following information: timestamp, close, open, high, low, volume, and volatility from January 1, 2000, to June 30, 2023. Table 1 shows descriptive statistics for each forex pair with daily and 4-hour frequency. The p-values of the Jarque-Bera test (Jarque and Bera, 1980) are lower than the significance level (0.1) for each pair on daily and 4-hour frequency data, as observed in Table 1. The null hypothesis can be rejected, and the alternative hypothesis can be accepted based on this observation. This indicates that the distribution of returns for each pair is significantly different from the normal distribution. Moreover, mean returns are very close to zero. There are around 6000 observations of daily frequency and around 37000 observations of 4-hour frequency data for each pair. Figure 1 shows selected six foreign exchange rates.

Table 1: Descriptive statistics of daily and 4-hour frequency returns for all pairs

-	EURUSD		GBPUSD		USDCHF		USDCAD		AUDUSD		NZDUSD	
Stat	D	H4	D	H4	D	H4	D	H4	D	H4	D	H4
Obs	6171	36983	6170	36976	6171	36979	6170	36967	6170	36973	6170	36969
Mean$\times 10^{-6}$	23	4	-30	-5	-66	-11	4	.8	27	4	53	9
Std$\times 10^{-3}$	6.0	2.4	5.8	2.4	6.5	2.7	5.4	2.2	7.8	3.2	7.9	3.3
Min$\times 10^{-2}$	-2.7	-2.9	-8.3	-7.9	-12.6	-16.6	-3.8	-2.3	-8.2	-4.6	-6.8	-4.1
Max$\times 10^{-2}$	3.5	2.6	3.2	3.9	9.7	6.9	3.5	2.4	8.4	3.5	6.1	3.5
Skew	.1	.1	-.7	-1.0	-.7	-5.9	.2	-.0	-.3	-.3	.3	.2
Kurt	1.8	7.2	9.1	42	33	403	3.2	7.9	1.9	12.8	3.4	7.6
JB(p-value)	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

Note: Return was calculated using the simple return method.

Figure 1: Major six foreign exchange rates



Note: This figure shows the close price of six currency pairs in the period from January 1, 2000 to June 30, 2023.

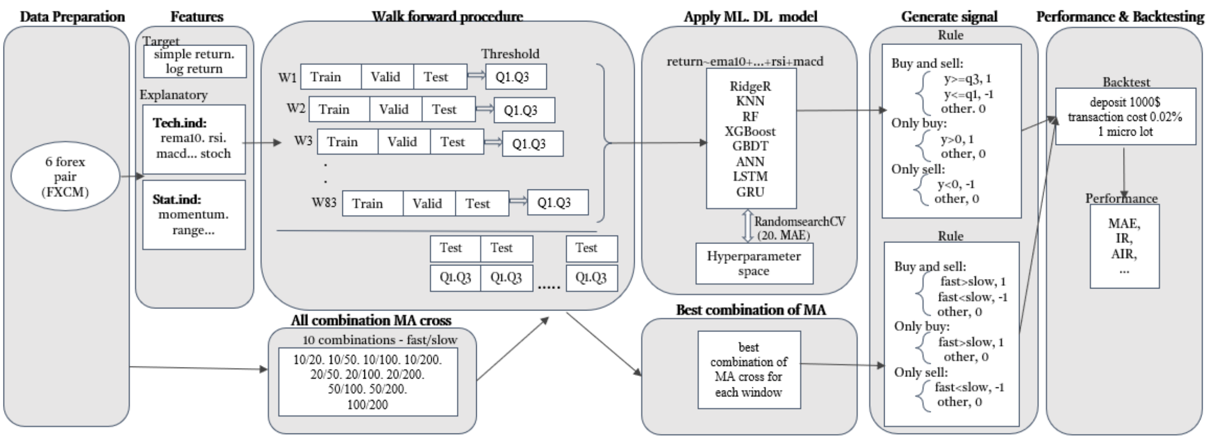
3 METHODOLOGY

3.1 Overall architecture

An architecture illustrating the machine learning model and the trading signal generation process is shown in Figure 2. More details about each primary component will be explained in the following sections 3.2-3.11. It consists of eight primary components:

- Data preparation - select the major foreign exchange pairs and import raw data from the broker using Python.
- Features - define the target variables of the regression problem and create features that include technical and statistical indicators.
- Walk forward optimization - define roll size and divide each window into training, validation, and testing periods.
- Model training and optimization - train a machine learning model on a given hyperparameter space and optimize the hyperparameters using the RandomsearchCV method with mean absolute error as a loss function.
- Generate signals - transform predicted returns into trading signals using thresholds (quartiles, given value) depending on the type of trading signal.
- All combinations of Exponential Moving Average (EMA) - generate all combinations of EMA and train them in the training period.
- Best combination of Exponential Moving Average (EMA) - find the best EMA combination with the highest SR.
- Backtesting - determine initial deposit, transaction cost, and trading volume to compare the performance of trading strategy including machine learning-based and trend-following strategies.

Figure 2: An architecture for training machine learning models and generating trading signal



Note: Machine learning-based strategies and traditional trend-following strategies were optimized by a rolling walk-forward approach.

3.2 Target variables and integrated categories

3.2.1 Return methods of the target variable

A regression problem was considered for this study because predicted returns were used to generate trading signals. Using today's high and low prices to predict today's return would meet a look-ahead bias because these prices cannot be determined until the market closes for the day. To avoid this look-ahead bias, the target variable and model equation were formulated according to the following equation. In this study, simple returns were utilized Eq (1) and log returns were employed Eq (2). Thus, the general equation Eq (3) of the regression problem is the same for both returns.

$$R_t = \frac{Close_{t+1} - Close_t}{Close_t} \tag{1}$$

$$R_t = \frac{Close_{t+1}}{Close_t} - 1 \tag{2}$$

$$\hat{R}_t = \beta_0 + \beta_i * X_{t,t-n} + e \tag{3}$$

where $Close_{t+1}$, $Close_t$ refers to close prices in periods t+1 and t. $Close_{t+1}$ indicates a close price of the next day or the next 4-hour price. Price information from periods t and t-n was utilized to predict the return between periods t and t+1, as per Eq (3).

3.2.2 Categorization of machine learning models into integrated categories

In this study, the performance of individual machine-learning models was considered. Additionally, individual models were divided into several integrated categories to facilitate overall performance comparisons. Table 2 shows which machine learning models belong to each integrated category.

Table 2: Category of integrated models

Category	Description	Individual models
ML^{A}	all ML models	Ridge, KNN, RF, XGBOOST, GBDT, ANN, LSTM, GRU
ML^{NN}	ML model with NN	ANN, LSTM, GRU
ML^{O}	ML model without NN	Ridge, KNN, RF, XGBOOST, GBDT

Note: ML^A denotes all machine learning models, ML^N denotes machine learning models with neural network, ML^O denotes machine learning models without neural network

3.3 Features

Fundamental indicators are acknowledged for their significant impact on the currency market, whereas technical indicators are recognized for their ability to explain exchange rate movements. Technical indicators help us understand how changes in fundamental indicators affect currency rates. Unexpected changes in fundamental indicators impact all financial instruments and determine trends. However, these fundamental indicators cannot be predictors, especially in the currency market. Their effects can only be explained after the currency rate has absorbed unexpected changes in fundamental indicator (Chan, 2017). As markets become more sophisticated, traditional models become less effective, and the influence of fundamental indicators also changes. Therefore, it is likely to be more effective to use technical indicators related to price, volume, and volatility rather than fundamental indicators in predicting future price rates. Based on this fact, only technical indicators were considered as features. Table 3 shows information about selected features and target variables, including settings.

Table 3: Category of features and selected settings

Category	Features	Setting
Target	simple return, log return	close price
Technical Indicator	EMA	10, 20, 50, 100, 200
	MACD	26, 12, 9
	RSI	14
	Stochastic Oscillator	14, 3
	CCI	20, .015
	BB	20, 2
	ATR	14
	WilliamR	14
	ADX	14
Statistical Indicator	Momentum	1
	Average price	1
	Range	1
	OHLC	1

Note: The table shows the abbreviated names of selected features. The Settings columns indicate the default settings for each feature.

There are many types of technical indicators such as trend-following, oscillator, and volatility indicators, each has a different purpose and offers insights into different aspects of price movements and market trends. Trend-following indicators can be used to determine the direction of the trend (up, down, sideways). Oscillator indicators can be used to identify some important thresholds that express overbought or oversold. Volatility indicators can be used to identify periods of high and low volatility. A detailed description of the technical indicator will be explained in the following sections 3.3.1-3.3.3

3.3.1 Trend following indicators

Exponential Moving Average

The Exponential Moving Average (EMA) is one of the most popular moving averages for generating trading signals and relying on crossovers and divergences from historical averages. Unlike a simple moving average, it gives more weight to the most recent data. Essentially, this means that the latest price information carries a higher weight. Commonly used EMAs include EMA10, EMA20, EMA50, EMA100, and EMA200, and the EMA formula is defined by the following equation (Tinghino, 2008):

$$EMA = Close\frac{2}{n+1} + \frac{\sum_{j=1}^{n} Close_j}{n}(100 - \frac{2}{n+1}) \tag{4}$$

Moving Average Convergence Divergence

The Moving Average Convergence Divergence (MACD) is a tool used to determine the direction and strength of a trend consisting of three moving averages with different lengths. Gerald Appel invented it in the 1970s, and Thomas Aspray extended it by introducing the histogram component in 1986. The MACD consists of the signal line, MACD line, and MACD histogram. The MACD line represents the difference between the slowest and medium-moving averages, while the signal line represents the slowest-moving average. The default MACD setting is usually 26, 12, and 9. The MACD formula defined by Tinghino (2008) and Appel (2005) is expressed by the following equation:

$$MACD = SMA(n_1) - SMA(n_2) \tag{5}$$

$$SignalLine = SMA(n_{sig}, MACD) \tag{6}$$

where n_1, n_2, n_{sig} indicates number of days; $n_1 < n_2$

Average Directional Index

The Average Directional Index (ADX) is used to assess the strength and direction of a trend. It consists of three lines: DMI (Directional Movement Index), +DI (Positive Directional Indicator), and -DI (Negative Directional Indicator). Welles Wilder invented it in 1978, the ADX is represented on a scale from 0 to 100. A higher ADX value indicates a stronger trend, while a lower ADX value indicates a weaker or sideways trend. The default setting of ADX is typically 14. The formula (Tinghino, 2008) for ADX is defined in the following equation.

$$+DI = (\frac{Smoothed + DM}{ATR}) * 100 \tag{7}$$

$$-DI = (\frac{Smoothed - DM}{ATR}) * 100 \tag{8}$$

$$DX = (\frac{|+DI - -DI|}{|+DI + -DI|}) * 100 \tag{9}$$

$$ADX = (\frac{(PriorADX * 13) + CurrentADX}{14}) \tag{10}$$

$$PH = PreviousHigh \tag{11}$$

$$+DM = CurrentHigh - PH \tag{12}$$

$$-DM = PreviousLow - CurrentLow \tag{13}$$

3.3.2 Oscillator indicators

Relative Strength Index

The Relative Strength Index (RSI) is an oscillator indicator and is used to measure the rate of change in price direction. It was developed by J.Welles Wilder in 1978. RSI takes values from 0 to 100. If the RSI is close to 0, it indicates a weaker trend. If RSI approaches 100, it indicates a stronger trend. The default RSI setting is usually 14. The RSI formula defined by Tinghino (2008) and Wilder (1978), is expressed by the following equation:

$$RSI = 100 - \frac{100}{1 + \frac{AverageGain}{AverageLoss}} \tag{14}$$

Stochastic Oscillator

The Stochastic Oscillator (STOCH) consists of two lines, ranging from 0 to 100, and serves to identify trends and reversals. It helps identify overbought and oversold conditions using predefined thresholds. George Lane invented it in the 1950s, STOCH shows the %K and %D lines. The %K line represents the ratio of the closing price (K) to the observed price range over a specified number of bars in the historical period. On the other hand, %D is a smoothed moving average of %K, intended to diminish false signals while remaining aligned with the overall trend. The default setting for STOCH is typically 14, 3, and 3. The formula (Tinghino, 2008) for the stochastic oscillator is outlined in the following equation.

$$\%K = SMA(100 * \frac{CurrentClose - LowestLow}{HighestHigh - LowestLow}) \tag{15}$$

$$\%D = SMA(K, periodD) \tag{16}$$

Commodity Channel Index

The Commodity Channel Index (CCI) is used to determine trend direction as well as overbought and oversold levels by comparing price changes to moving averages. Donald Lambert invented it in 1980, CCI values range from 100 to -100. A CCI above 0 indicates that the price is higher than the historical average. Conversely, a CCI below 0 indicates that the price is below its historical average. The default CCI setting is usually 20. The formula for CCI, as outlined by (Tinghino, 2008) is presented in the following equation:

$$TypicalPrice(TP) = \sum_{i=1}^{p} (\frac{High + Low + Close}{3}) \tag{17}$$

$$MA = \frac{\sum_{i=1}^{p} TP}{P} \tag{18}$$

$$MD = \frac{\sum_{i=1}^{p} |TP - MA|}{P} \tag{19}$$

$$CCI = \frac{TP - MA}{.015 * MD} \tag{20}$$

Williams R

The Williams %R indicator was invented by Larry Williams and is used to identify overbought and oversold conditions while measuring momentum. It takes a value between 0 and -100. When Williams %R is close to -100, it suggests oversold conditions. The default setting for Williams %R is usually 14 periods. The formula for Williams %R defined by Tinghino (2008) is given by the following equation:

$$\%R = \frac{HighestHigh - CurrentClose}{HighestHigh - LowestLow} * -100 \tag{21}$$

3.3.3 Volatility indicators

Bollinger Band

The Bollinger Bands (BB) indicator measures volatility and consists of three lines that relate to the price's mean and standard deviation. Bollinger Band was invented by John Bollinger in the 1980s (John, 2002), it consists of a middle line indicating a simple moving average, and upper and lower bands showing two standard deviations of price. As the volatility increases, the bands widen accordingly. The default setting for Bollinger Bands is usually 20 periods with a standard deviation of 2. The formula (Tinghino, 2008) for BB is outlined in the following equation.

$$MiddleBand = SMA(20) \tag{22}$$

$$UpperBand = SMA(20) + (2 * std) \tag{23}$$

$$LowerBand = SMA(20) - (2 * std) \tag{24}$$

Average True Range

The Average True Range (ATR) indicator serves to measure volatility. A high ATR value indicates high volatility, while a low ATR value indicates low volatility. The default setting for ATR is usually 14 periods. The formula for calculating the Average True Range (ATR), as outlined by Tinghino (2008), is expressed in the following equation:

$$TR = MAX[(H - L), abs(H - C_p), Abs(L - C_p] \tag{25}$$

$$ATR = \frac{1}{n} * \sum_{i=1}^{n} TR_i \tag{26}$$

where TR is the particular true range and n is the period employed.

3.3.4 Statistical indicators

Researchers often use additional statistical indicators obtained from price data for algorithmic trading research. The following four statistical indicators were added as features based on Islam and Hossain (2021).

$$Momentum_t = Open_t - Close_t \tag{27}$$

$$AVG_t = \frac{Low_t + High_t}{2} \tag{28}$$

$$Range_t = High_t - Low_t \tag{29}$$

$$OHLC_t = \frac{Open_t + Close_t + Low_t + High_t}{4} \tag{30}$$

3.4 Machine learning-based strategy

In this study, supervised machine learning algorithms such as Ridge Regression, K-Nearest Neighbor, Random Forest, Extreme Gradient Boosting, Gradient Boosting Decision Trees, Artificial Neural Networks, Long Short Term Memory, and Gated Recurrent Unit models were considered based on a literature review. These models are trained using technical and statistical indicators as features, and the target variables include both simple and log returns. Each model was optimized within a predefined hyperparameter space in each rolling walk-forward method's window. The models and hyperparameter spaces are described in sections 3.4.1 and 3.4.2.

3.4.1 Machine learning models

Ridge Regression

Arthur E. Hoerl and Robert W. Kennard invented the Ridge regression (Ridge) model in 1970 (Hoerl and Kennard, 1970). It is a method used in regression problems and is an extension of ordinary least square regression by adding a penalty (regularization) term to the loss function. This penalty is used to shrink large coefficients of the model towards zero and reduce the overfitting. Ridge regression aims to minimize the regularized loss function to estimate the coefficients. The loss function is defined as follows:

$$argmin \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{31}$$

where λ is the penalty parameter. The advantage of ridge regression is that it handles multicollinearity, reduces overfitting, and provides stable coefficients. A disadvantage of ridge regression is that it can lead to underfitting.

K Nearest Neighbors

Fix and Hodges (1989) invented K Nearest Neighbors (KNN). It is a nonparametric model that predicts the target variable by averaging the values of its k-nearest neighbors. There are various distance measures such as Euclidean and Manhattan to determine the nearest neighbors. The following formula shows the principle of KNN regression:

$$\hat{y_0} = \frac{1}{K}\sum_{i=1}^{K} y_i \tag{32}$$

where y_i represents the target value of i^{th} nearest neighbor. K represents the optimal number of K. The advantages of KNN regression are that is easy to understand and implement, can be used for nonlinear relationships, and does not require assumptions.

Random Forest

Tin Kam Ho invented the Random Forest (RF) model in 1995 (Ho, 1995). This is a special case of bagging technique that produces an ensemble of independent decision trees. This model is built on different subsamples randomly selected from the full training set with replacements for each tree. The final prediction is the mean of the predictions made for each tree. The following formula shows the principle of Random Forest regression:

$$\hat{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{33}$$

where N is the total number of trees in the forest. y_i shows the prediction made by the i^{th} tree. The advantages of random forest are that it provides highly accurate predictions, is robust to overfitting, and provide feature importance. The disadvantages of random forests are that they are computationally expensive, memory intensive, uninterpretable, and highly parameterized.

Extreme Gradient Boosting

Tianqi Chen and Carlos Guestrin designed the Extreme Gradient Boosting (XGBoost) model in 2016 (Chen and Guestrin, 2016) and it is an advanced implementation of a gradient boosting algorithm that aims to minimize loss function by gradient descent algorithm. It builds many trees and each new tree fits the residual errors of the previous tree. The following formula shows the principle of the XGBoost model:

$$\hat{y} = \sum_{i=1}^{N} n f_i(x) \tag{34}$$

where N is the total number of weak trees, $f_i(x)$ is a prediction of the i^{th} weak tree and n is the shrinkage factor that controls the contribution of each weak tree to the final prediction. The advantages of XGBoost are highly accurate prediction, high optimization for learning speed and efficiency, and avoiding overfitting and improving performance. The disadvantages of XGBoost are that it is more complex and computationally intensive for large data.

Gradient Boosting Decision Trees

Leo Breiman and Jerome H. Friedman invented the Gradient Boosting Decision Trees (GBDT) model (**?**). It is an ensemble learning method that uses the principle of gradient boosting. It trains decision trees using the residual error from the previous trees to improve performance. The final predictions are the mean predicted values of all decision trees. GBDT is quite similar to the XGBoost model. But GBDT is slower, less efficient, and less effective in practice than XGBoost. The following formula shows the principle of the GBDT model:

$$\hat{y} = \sum_{i=1}^{N} f_i(x) \tag{35}$$

where N is the total number of decision trees. $f_i(x)$ shows predicted value of the i^{th} decision tree.

Artificial Neural Network

The Artificial Neural Network (ANN) model was inspired by the biological neural networks in the human brain. It is a computational model consisting of interconnected nodes, or neurons, organized into layers. Each neuron receives input signals, processes them through a selected activation function, and transmits the output to the neurons of the next layer. The activation function is a non-linear transform element of neurons. Weights are interconnections between layer nodes. In general, this model learns from data by adjusting the weights of connections between neurons through a process called back-propagation that aims to minimize a chosen loss function. One of the most important steps is to define the architecture of the neural network, including the number of layers, number of neurons, and activation function.

$$\hat{y} = f(\beta + \sum_{i=1}^{n} (x_i * w_i)) \tag{36}$$

where x_i is inputs, w_i is weights, β is bias, f is activation functions, and n is the number of neurons. In this study, only the tanh (tangent hyperbolic) activation function was considered, along with Gradient Stochastic Descent and Root Mean Square Propagation as optimizers. The reason for choosing the tanh function is range of return is between -1 and 1. The advantage of ANN is that it works well with complex non-linear relationships and is highly adaptable. Disadvantages of ANN are complexity, computationally intensivity, potential overfitting, and difficult interpretation.

Long Short Term Memory

Long Short Term Memory (LSTM) model was invented by Hochreiter, S. and Schmidhuber, J. in 1997 (Hochreiter and Schmidhuber, 1997). It is an advanced type of Recurrent Neural Network (RNN). In an LSTM network, at each specific time step, both the current input and the output from the previous time step are fed into the LSTM unit. The LSTM unit processes this information and generates an output, which is then passed to the next time step. This iterative process continues, enabling the LSTM to capture and learn patterns over sequential data. LSTMs are equipped with specialized memory cells designed to retain information over extended periods. This feature allows them to adeptly capture intricate dependencies within sequential data, offering a marked improvement over traditional RNNs. In an LSTM network, there are three essential components known as gates: the input gate, the forget gate, and the output gate. Each gate plays a crucial role in controlling the flow of information within the network. The input gate decides how to incorporate the current input and the previous internal state to update the internal memory. The forget gate determines the extent to which the previous internal state should be retained or discarded. Lastly, the output gate governs the impact of the internal memory on the overall output of the network. The following formula shows the principle of the LSTM model:

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \tag{37}$$

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \tag{38}$$

$$\bar{C}_t = tanh(W_c * [h_{t-1}, x_t] + b_C) \tag{39}$$

$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t \tag{40}$$

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{41}$$

$$h_t = o_t * tanh(C_t) \tag{42}$$

where f_t - the forget fate output. i_t - the input gate output. o_t - the output gate output. $\bar{C}_t$ - candidate cell state. C_t - updated cell state. h_t - output of the LSTM at time step t. $\sigma - sigmoid function.W_{f,i,c,o}$ - weight matrices. $b_{f,i,C,o}$ - bias vectors. x_t - input at time step t. The advantages of LSTMs include their ability to effectively capture long-term dependencies, learn complex patterns, and their versatility for various tasks. However, LSTMs come with some drawbacks, such as being computationally intensive, having a complex architecture, challenging interpretability of the learning process, and susceptibility to overfitting.

Gated Recurrent Units

The gated Recurrent Units (GRU) model was invented by Kyunghyun Cho in 2014 and it is a type of Recurrent Neural Network (RNN) (Kyunghyun et al., 2014). The training process of GRU is similar to LSTM. However, GRU has only two gates an update gate and a reset gate. The update gate controls how much of the previous memory should be retained and how much of the new memory should be added to the current state. The reset gate determines how much of the past information should be forgotten when computing the current state. The formulas for GRU are as follows:

$$r_t = \sigma(W_r * [h_{t-1}, x_t]) \tag{43}$$

$$z_t = \sigma(W_z * [h_{t-1}, x_t]) \tag{44}$$

$$\bar{h_t} = tanh(W_h * [r_t \odot h_{t-1}, x_t) \tag{45}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \bar{h_t} \tag{46}$$

where r_t-reset gate activation, z_t-update gate activation, $\bar{h_t}$-candidate hidden state, h_t-updated hidden state at time step t, σ-sigmoid function, $\odot$ - element wise multiplication, $W_{r,z,h}$-weight matrices, h_{t-1} -previous hidden state, x_t-input at time step t. The advantages of GRU are easier to interpret and train compared to LSTM, is faster, and requires fewer resources. The disadvantage of GRU is the limited ability to capture long-term dependencies.

3.4.2 Hyperparameter tuning process

Hyperparameter tuning is an important step in the development of machine learning models, which greatly affects their performance. It finds the optimal hyperparameters that control the learning process of the model. Before the hyperparameter tuning process, feature scaling was applied using a min-max scaler within each rolling walk-forward window for all features. A min-max scale is a data preprocessing technique used to transform numerical features with a certain range, usually between 0 and 1. This ensures that all features have the same scale and prevents features with larger magnitudes from dominating the learning process. The formula for min-max scaling is as follows:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{47}$$

where X_{scaled} is the scaled value of feature X, X_{min} is the minimum value of feature X, X_{max} is the maximum value of feature.

The RandomizedSearchCV method was used for hyperparameter tuning with 20 iterations. This method randomly selects hyperparameter values from a predefined hyperparameter space. It is particularly effective in high-dimensional hyperparameter spaces compared to a grid search. During the hyperparameter tuning process, mean absolute error (MAE) was utilized as the loss function. MAE measures the mean absolute difference between predicted return and true return values in the training data. The main goal of optimizing the machine learning model is to minimize this loss function and thereby improve the performance of the model. The MAE formula is expressed as follows:

$$MAE = \frac{1}{n} * \sum_{i=1}^{n} |y_i - \hat{y}_i|^2 \tag{48}$$

where y_i is the true return value, $\hat{y}_i$ is the predicted return value, and n represents the number of observations.

Table 4 shows the hyperparameter space considered for each model. One important advantage of using RandomizedSearchCV is its ability to define the distribution for hyperparameter values rather than specific values. Both uniform and log-uniform distributions were used for certain hyperparameters. In a hyperparameter with a uniform distribution, all values within a specified range have an equal probability of being chosen. Conversely, for a hyperparameter with a log-uniform distribution, values are sampled uniformly on a logarithmic scale. This implies that values closer to the lower bound of the range have a higher likelihood of being selected.

Table 4: Hyperparameter spaces for each model on daily and 4-hour frequency

Model	Hyperparameter	Range	Dimension
Ridge	lambda	[.00001, 1]	uniform
	kernel	linear/rbf/poly	multivalue
KNN	n neighbors	[1, 20]	integer
	p	[1, 3]	integer
RF	n estimator	[20, 200]	integer
	max features	[5, 30]	integer
	max depth	[1, 6]	integer
	min samples split	[2, 30]	integer
XGBoost	n estimators	[20, 200]	integer
	learning rate	[.001, .5]	loguniform
	max depth	[8, 15]	integer
	gamma	[.001, .02]	uniform
GBDT	n estimators	[20, 200]	integer
	learning rate	[.0001, .5]	loguniform
	max depth	[1, 5]	integer
ANN, LSTM, GRU	no.hidden.layer	[1, 3]	integer
	no.neurons	[5, 40]	integer
	activation function	tanh	fixed
	dropout rate	.2	fixed
	optimizer	SGD/RMSProp	multivalue
	learning rate	[.001, .05]	loguniform
	momentum	[.1, .4]	uniform
	batch size	32/64/128	multivalue
	epoch	10/20/30	multivalue

Note: Table shows hyperparameter spaces for each model. The hyperparameter column shows the names of specific hyperparameters for each model. Range - the range of hyperparameter space. Dimension - types of hyperparameter space. For uniform and log-uniform, hyperparameter spaces are generated using min and max values in brackets. For integer dimension, hyperparameter space is between two values in brackets. For multivalue, a hyperparameter is selected from given values in brackets. The choice of hyperparameter space based on literature review extended with uniform and log-uniform distribution

Table 5 shows the optimal range of hyperparameters selected across all rolling walk-forward windows for both daily and 4-hour frequency data.

Table 5: Selected hyperparameter values for each model on daily and 4-hour frequency

Model	Hyperparameter	H4	Daily
Ridge	alpha	]0, .2[	]0, .1]
KNN	n neighbors	]3, 6[	]0, 5[
	p	1	1
RF	n estimator	]100, 175[	]125, 175[
	max features	]20, 30[	]20, 25[
	max depth	5	5
	min samples split	]0, 10[	]0, 10[
XGBoost	n estimators	]125, 200[	]125, 200[
	learning rate	]0, .1[	]0, .1[
	max depth	]8, 14[	]8, 14[
	gamma	]0, .003[	]0, .002[
GBDT	n estimators	]100, 200[	]100, 200[
	learning rate	]0, .1[	]0, .1[
	max depth	]3, 4[	]2, 4[
ANN, LSTM, GRU	no.hidden.layer	]1, 2[	]1, 2[
	no.neurons	]5, 40[	]5, 40[
	activation function	tanh	tanh
	dropout rate	.2	.2
	optimizer	SGD/RMSProp	SGD/RMSProp
	learning rate	]0, .01[	]0, .01[
	momentum	].1, .5[	].1, .5[
	batch size	64	64
	epoch	30	30

Note: Table shows selected optimal hyperparameters for each models

3.5 Trend following strategy

Trend following (TF) strategy is one of the oldest and the most popular trading strategies in financial markets. It is based on the principle that the market continuously moves up or down through the gradual dissemination of information. According to this strategy, the more traders make decisions based on market news, the more likely the trend is to intensify and create a strong trend. The speed at which the market absorbs information determines the duration of the trend. The trading signals for the trend-following strategy are defined as follows:

- If the price has an uptrend, traders should consider buying (taking a long position) the asset.
- Conversely, if the price has a downtrend, traders should consider selling (taking a short position) assets.

In this study, the exponential moving average (EMA) was utilized as an indicator to identify trading signals for the trend-following strategy. Moving averages are commonly used as smoothing tools to reduce noise in financial data. There are various ways to use EMA to generate trading signals, but in this study, only fast and slow EMA crossover was considered. Specifically, EMA10, EMA20, EMA50, EMA100, and EMA200 were used with a walk-forward approach. This indicates that 10 combinations of fast and slow EMAs were investigated in each rolling walk-forward window. The Sharpe ratio was used to optimize the moving average crossover strategy in each window. A buy signal occurs when the fast EMA crosses above the slow EMA, indicating an uptrend. Conversely, a sell signal occurs when the fast EMA crosses below the slow EMA, indicating a downtrend. Table 6 shows a detailed algorithm describing the EMA crossover strategy.

Table 6: Pseudocode of trend following strategy

Algorithm: EMA cross strategy

```
Inputs: ema10, ema20, ema50, ema100, ema200, train, test
Output: trading signal
ema_i ←fast ema
ema_j ←slow ema
for train, test in each window:
    find combination[ema_i, ema_j] = argmax_{ema_i,ema_j} SR(train)
    for each combination of ema_i, ema_j :
        if ema_i > ema_j :
            trading signal=1
        elif ema_i < ema_j :
            trading signal=-1
        else
            trading signal=0
    apply best combination of ema_i, ema_j totest
    SR,trading signal=F(test, ema_i, ema_j)
        if ema_i > ema_j :
            trading signal=1
        elif ema_i < ema_j :
            trading signal=-1
        else
            trading signal=0
```

Note: Inputs are combinations of moving averages. Outputs are trading signals such as buy, sell, and hold. The loop function runs on each window of the rolling walk-forward approach. The combination of the moving average with the highest Sharpe Ratio was defined as the best combination on each window.

3.6 Rolling walk forward optimization

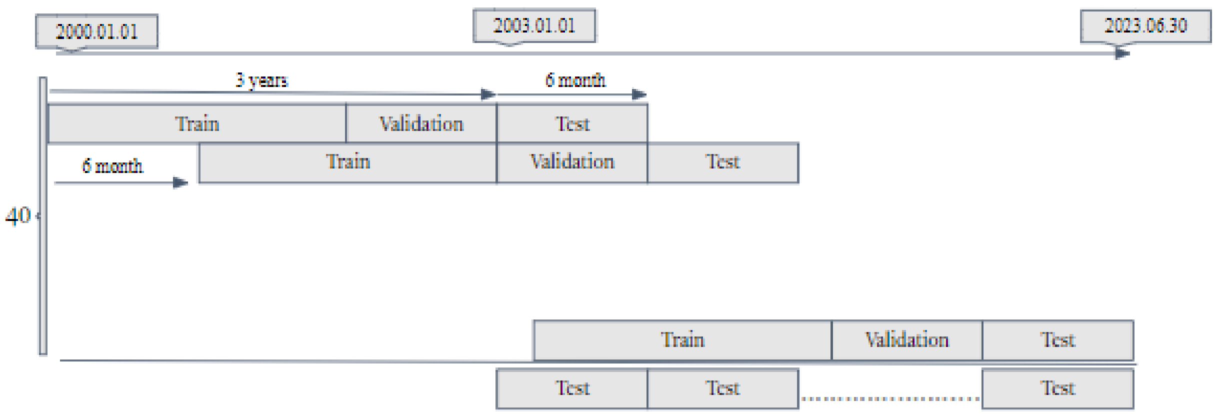
One of the most important challenges in algorithmic trading is the presence of data-snooping bias. This bias occurs when a large number of parameters are used to generate trading strategies using for example machine learning models that perform well on historical data. However, there is no guarantee that past market conditions will repeat themselves in the future. Using a large number of parameters often leads to overfitting. Overfitting artificially inflates backtest performance, and poses significant risks when applied to real-world markets. Therefore, a trade-off between high accuracy and overfitting must be carefully considered during model training.

The term ”parameters” includes a variety of concepts beyond just the hyperparameters of the machine learning model. It includes factors such as:

- Hyperparameters of the machine learning model
- Fixed ranges of in-sample and out-of-sample data
- Trading volume size and transaction cost size
- Thresholds that transform predicted returns into entry and exit signals
- Settings of technical indicators

To avoid overfitting problems and data-snooping bias, a rolling walk-forward approach was implemented. This approach entails iterative training of the model on a fixed training set, tuning hyperparameters on a fixed validation set, and subsequently making predictions on a fixed test set. The parameters of the rolling walk-forward approach are illustrated in Figure 3. The dataset is divided into 40 windows, each window containing different train, validation, and test sets. The rolling size is set at six months and in each window, the train, validation, and test sets advance by six months until the end of the dataset.

Figure 3: Rolling Walk Forward approach used in this study



Note: For each window, the training machine learning process was executed.

Table 7 shows number of data for the rolling walk-forward approach. The training set is 68 percent, the validation set is 18 percent, and the test set is 14 percent of the total data in one rolling walk-forward approach's window for both daily and 4-hour frequency data. For daily frequency data, 600 days of price data were used for the training set, followed by 156 days for the validation set, and 126 days for the test set in each window. This method was applied to machine learning model training, hyperparameter tuning, and prediction processes iteratively on a window-by-window basis to provide a reliable evaluation of model performance.

Table 7: Train, validation, and test sets period in one window

-	Train	Validation	Test
Percentage	68	18	14
Daily	600	156	126
4 Hour	3600	936	756

Note: This table shows the number of observations for the train, validation, and test datasets.

3.7 Transforming rule

In this study, both machine learning-based trading strategies and trend-following trading strategies were investigated. For a machine learning-based strategy, trading signals were generated according to the rules defined in Eqs (49-51). Meanwhile, for the trend-following strategy, trading signals were generated according to the rules in Eqs (52-54). According to the above rules, the $signal_t$ produces three different outputs: 0, 1, and -1. A value of 1 represents a buy signal, -1 represents a sell signal, and 0 represents a hold signal.

Machine Learning strategy

To generate trading signals, the predicted returns were compared to the first and third quartiles of the training set. For buy and sell signals, if the predicted return is higher than the third quartile of the training set, it will trigger a buy signal. Conversely, if the predicted return is lower than the first quartile of the training set, it will trigger a sell signal. Otherwise, it will generate a hold signal. For only the buy signal, if the predicted return is higher than 0, it will create a buy signal. Otherwise, it will generate a hold signal. Similarly, for only a sell signal, if the predicted return is lower than 0, it will create a sell signal. Otherwise, it will generate a hold signal.

Buy and sell signal:

$$signal_t = \begin{cases} \hat{y} \geq Q3, 1 \\ \hat{y} \leq Q1, -1 \\ otherwise, 0 \end{cases} \tag{49}$$

Only buy signal:

$$signal_t = \begin{cases} \hat{y} > 0, 1 \\ otherwise, 0 \end{cases} \tag{50}$$

Only sell signal:

$$signal_t = \begin{cases} \hat{y} < 0, -1 \\ otherwise, 0 \end{cases} \tag{51}$$

where Q3 is the third quartile of the train set, Q1 is the first quartile of the train set, and $\hat{y}$ is the predicted return.

Trend Following strategy

Fast and slow exponential moving averages (EMAs) were used to generate trading signals. For buy and sell signals, if the fast EMA is higher than the slow EMA, it will trigger a buy signal. Conversely, if the fast EMA is lower than the slow EMA, it will generate a sell signal.

Otherwise, it will represent a hold signal. For only the buy signal, if the fast EMA is higher than the slow EMA, it will generate a buy signal. Otherwise, it will create a hold signal. Similarly, for only the sell signal, if the fast EMA is lower than the slow EMA, it will generate a sell signal. Otherwise, it will represent a hold signal.

Buy and sell signal:

$$signal_t = \begin{cases} EMA_{fast} > EMA_{slow}, 1 \\ EMA_{fast} < EMA_{slow}, -1 \\ otherwise, 0 \end{cases} \tag{52}$$

Only buy signal:

$$signal_t = \begin{cases} EMA_{fast} > EMA_{slow}, 1 \\ otherwise, 0 \end{cases} \tag{53}$$

Only sell signal:

$$signal_t = \begin{cases} EMA_{fast} < EMA_{slow}, -1 \\ otherwise, 0 \end{cases} \tag{54}$$

where EMA_{fast} is exponential moving average with smaller setting, EMA_{slow} is exponential moving average with larger setting.

3.8 Backtest assumption

Algorithmic trading systems require three basic rules: when to enter a trade, when to exit a trade, and what position to take. An exit strategy is considered as crucial as an entry strategy. In this study, entry signals are generated based on the transformation rule outlined in section 3.7. Whenever the entry strategy changes, it serves as the signal to exit the current trade. The position size remains fixed at 1 micro lot for each trade.

The primary objective of this study is to generate trading signals with the highest Sharpe Ratio (SR), rather than precisely predicting returns that closely match real returns. Therefore, when calculating the Sharpe Ratio (SR), transaction costs should be taken into account. The performance of a strategy without transaction costs can significantly diverge from reality, especially as the trading frequency increases. The assumption of backtesting is detailed in the following bullet points:

- Initial trading capital starts at 1000 USD.
- Lot size is 1 micro lot (1000 units).
- Pip value is 0.1 USD per pip.
- Transaction cost is a fixed 0.02 percent.
- Leverage, stop loss, and take profit size are not considered in this study to maintain simplicity.

For many financial instruments, high and low prices have more noise than open and close prices. This noise can be caused by very small orders transacted at high or low prices and misreported ticks (Chan, 2009). Therefore, open and close prices were used for determining the holding period rather than high and low prices. In other words, all trading signals considered in this study will hold positions from the open price at the time of the signal to the close price before a different signal is given.

3.9 Performance metrics

Various performance metrics were employed to evaluate the effectiveness of the trading strategy. These metrics include net profit, number of trades, percentage of buy trades, Sharpe ratio, and annualized Sharpe ratio.

The Sharpe Ratio (Sharpe, 1966) is one of the most widely used performance metrics for comparing different strategies. It is based on the calculation of the mean return, standard deviation of return, and the cost of carrying a position. In this research, the position carrying fee was assumed to be zero. Therefore, the Sharpe ratio simplifies to the ratio between the mean and standard deviation of returns. There are some commonly used thresholds for the Sharpe ratio as a rule of thumb. If a strategy's Sharpe ratio is less than 1, the strategy may not be suitable as a stand-alone strategy. If the Sharpe ratio is higher than 2, the strategy is profitable almost every month. If the Sharpe ratio is higher than 3, the strategy's profitability is almost guaranteed every day (Chan, 2009). The formula for calculating the Sharpe Ratio is as follows:

$$SR = \frac{R_p - R_f}{\sigma_p} = \frac{R_p}{\sigma_p} \tag{55}$$

where R_p is mean return of trading strategy, R_f is carrying fee, σ_p is the standard deviation of return.

The annualized Sharpe Ratio (ASR) can be used to effectively compare the performance of different strategies. It is adjusted according to the data frequency N. There are 252 trading days in the year, and N represents the number of windows based on the data frequency per day. For daily frequency data, N is equal to 1. For 4-hour frequency data, N is equal to 6. A trading strategy with a higher Annualized Sharpe Ratio (ASR) is better. The formula for calculating the ASR is as follows:

$$ASR = SR * \sqrt{252} * \sqrt{N} \tag{56}$$

3.10 Independent t-test

An independent t-test (Student, 1908) was used to compare the mean return of different strategies. The rationale for choosing the independent t-test is based on the assumption that these trading strategies are independent and unrelated to each other. In all cases where the t-test was used, the Null hypothesis stated that the mean returns of the two unrelated strategies are equal, while the Alternative hypothesis stated that the mean return of the first strategy is higher than the mean return of the second strategy:

$$\begin{cases} H_0 : \bar{R}_{strategy1} = \bar{R}_{strategy2} \\ H_A : \bar{R}_{strategy1} > \bar{R}_{strategy2} \end{cases} \tag{57}$$

where H_0 is the null hypothesis. H_A is an alternative hypothesis. $\bar{R}$ is mean return of strategy. If the p-value of the independent t-test is lower than the significance level (typically set at 0.1), the null hypothesis can be rejected and the alternative hypothesis can be accepted at a 90 percent significance level. This rejection suggests that the mean return of the first strategy is higher than the mean return of the second strategy.

3.11 Software and running time

Kaggle's free GPU was utilized to train our machine-learning models. It took 3 hours and 30 minutes to complete the model training, hyperparameter tuning, and prediction process using the daily frequency data of all models. In addition, it took 7 hours and 50 minutes to complete the model training, hyperparameter tuning, and prediction process using the 4-hour frequency data of all models. Table 8 shows details of the hardware specifications, while Table 9 displays the running time of each model on both daily and 4-hour frequency data. The GRU, LSTM, and ANN models require more runtime compared to the other models.

Table 8: Hardware specifications of Kaggle

Info	Value
RAM	29GB
Disk	73GB
GPU	GPU T4x2, GPU P100, TPU VM v3-8
Memory	15GB

Source: https://www.kaggle.com/page/GPU-tips-and-tricks

Table 9: Runtime by minutes

Models	Daily	4 Hour
Ridge	.5m	.7m
KNN	1.7m	37m
RF	4.2m	20m
XGBoost	1.3m	1.6m
GBDT	7.1m	37m
ANN	22m	54m
LSTM	88m	171m
GRU	81m	147m

Note: This table shows the running minutes for each model for daily and 4-hour frequency datasets.

4 EMPIRICAL RESULTS

In this chapter, we focused on comparing the results obtained from individual and integrated trading strategies with the benchmark strategies using the annualized Sharpe ratio as economic significance and the independent t-test as statistical significance. The benchmark strategy is considered as buying the instrument at the beginning of the out-of-sample period and selling it at the end. Each individual and integrated strategy is considered across six scenarios, including two different return methods and three different trading signals, as discussed in Sections 3.1 through 3.7. These six scenarios are presented as follows:

- Buy and Sell signals with the simple return: The target variable was calculated using the simple return, and trading strategies generate buy and sell signals.
- Only Buy signal with the simple return: The target variable was calculated using the simple return, and trading strategies generate only buy signals.
- Only Sell signal with the simple return: The target variable was calculated using the simple return, and trading strategies generate only sell signals.
- Buy and Sell signals with the log return: The target variable was calculated using the log return, and trading strategies generate buy and sell signals.
- Only Buy signal with the log return: The target variable was calculated using the log return, and trading strategies generate only buy signals.
- Only Sell signal with the log return: The target variable was calculated using the log return, and trading strategies generate only sell signals.

4.1 Individual strategies

Table 10 and Table 11 show the performance of individual trading strategies, trend-following strategy, and the benchmark strategy for EURUSD on daily and 4-hour frequencies in six scenarios. For each scenario, we can see the following metrics Net Profit (NT), Number of Trades (NT), Transactional Cost (TC), and Annualized Sharpe Ratio (ASR). Column M presents machine learning-based individual strategies, trend-following strategies, and benchmark strategies. Tables showing the performance of the other currency pairs are available in the appendix section in Tables from 17 to 26.

We can see the all trading strategies based on intraday (4-hour) data produce higher net profit, more trading entries, and more transaction cost than daily data. Trading strategies with **bold** ASR indicate that they outperformed benchmark strategies. For example, a trading strategy based on an Artificial Neural Network (ANN) model for EURUSD daily data generated 775 trading signals, yielding a net profit of $12,763 with $191 as transactional costs, resulting in a Sharpe ratio of 0.49, while generating both buy and sell signals.

Table 10: Performance metrics for Individual strategies for EURUSD daily data

			Buy and Sell				Only Buy				Only Sell			
TR	Str	M	NP($)	NT	TC($)	ASR	NP($)	NT	TC($)	ASR	NP($)	NT	TC($)	ASR
S	ML^O	RIDGE	6933	1929	477	-.25	6783	1363	338	-.12	6861	1363	338	-.14
		KNN	4523	1178	290	-.29	7398	1012	250	-.13	6713	987	245	-.06
		RF	7078	1950	481	-.22	6899	1392	345	-.22	6746	1400	347	-.17
		XGBOOST	4281	1226	308	-.31	6926	1398	346	-.20	6560	1394	345	-.19
		GBDT	7007	1907	471	-.19	6762	1402	347	-.13	6809	1393	345	-.16
	ML^{NN}	ANN	11405	771	189	-.09	8080	403	100	.15	7204	451	113	-.14
		LSTM	11132	220	56	.1	8179	83	21	.04	7613	69	16	-.2
		GRU	13030	254	64	.1	7370	149	37	**.24**	6931	103	26	-.09
	TF	MAs	14386	77	19	.17	7235	38	10	**.31**	6915	40	10	.01
	B	B/H	14449	2	0	.22	14449	2	0	.22	14449	2	0	.22
L	ML^O	RIDGE	6943	1930	477	-.25	6795	1362	338	-.15	6902	1364	338	-.13
		KNN	4643	1157	284	-.12	7272	1004	249	-.14	6800	982	243	-.04
		RF	7053	1948	481	-.21	6828	1398	346	-.15	6763	1394	345	-.17
		XGBOOST	3607	1134	285	-.34	7037	1396	346	-.18	6661	1395	346	-.14
		GBDT	6839	1918	474	-.23	6849	1397	346	-.21	6799	1392	345	-.13
	ML^{NN}	ANN	12763	775	191	**.49**	6725	390	97	.08	7278	433	106	.02
		LSTM	10848	218	53	**.42**	7116	101	24	**.24**	6925	125	31	.04
		GRU	11056	251	64	-.01	5713	138	34	.00	6506	137	34	-.12
	TF	MAs	14386	77	19	.17	7235	38	10	**.31**	6915	40	10	.01
	B	B/H	14449	2	0	.22	14449	2	0	.22	14449	2	0	.22

Table 11: Performance metrics for Individual strategies for EURUSD 4-hour data

			Buy and Sell				Only Buy				Only Sell			
TR	Str	M	NP	NT	TC	ASR	NP	NT	TC	ASR	NP	NT	TC	ASR
S	ML^O	RIDGE	17159	11407	2816	-.02	14429	8040	1989	-.22	15991	8028	1986	**.73**
		KNN	11615	6879	1702	-.06	14721	5820	1442	.03	16053	5765	1427	**.86**
		RF	16923	11285	2788	-.01	14303	8191	2027	-.28	16119	8153	2018	**.79**
		XGBOOST	11897	7812	1947	.16	14733	8045	1989	-.28	15761	8124	2010	**.78**
		GBDT	16842	11219	2769	-.04	14442	8154	2018	.1	15973	8108	2006	**.78**
	ML^{NN}	ANN	27991	1324	331	.05	15438	753	183	-.12	17656	1046	259	.04
		LSTM	30643	93	23	.12	16905	23	5	**.39**	16554	89	21	-.03
		GRU	30775	23	6	.02	15821	13	3	**.25**	18931	65	16	-.12
	TF	MAs	30789	593	147	.01	15417	267	66	.16	15375	315	77	.0
	B	B/H	31180	2	0	.22	31180	2	0	.22	31180	2	0	.22
L	ML^O	RIDGE	17170	11403	2815	.0	14441	8042	1989	-.12	15970	8044	1990	**.71**
		KNN	11701	6871	1702	-.08	15014	5767	1428	-.04	16082	5738	1420	**.81**
		RF	16967	11363	2808	-.05	14370	8147	2016	-.07	16156	8154	2018	**.77**
		XGBOOST	11905	7711	1919	**.27**	14807	8108	2005	.02	15566	8087	2000	**.83**
		GBDT	16871	11330	2796	-.04	14525	8105	2005	.0	16032	8137	2013	**.78**
	ML^{NN}	ANN	26733	1700	425	-.12	12176	709	183	-.18	15596	980	240	.0
		LSTM	30888	17	4	.11	15611	38	9	-.1	12582	101	24	-.25
		GRU	31130	73	18	**.34**	15698	11	3	.0	15235	32	7	.1
	TF	MAs	30789	593	147	.01	15417	267	66	.16	15375	315	77	.0
	B	B/H	31180	2	0	.22	31180	2	0	.21	31180	2	0	.22

Note: ASR in bold indicates higher SR than Benchmark strategy. S - simple return. L - log return. ML^O - ML without NN. ML^{NN} - ML with NN.
Source: Own study

Each currency pair required a different individual machine-learning model tailored to its characteristics as outlined in Table 10, 11, and 17-26. These models are designed to generate trading signals with the highest Sharpe ratio for each specific case. The following list summarized the individual trading strategies that outperformed the benchmark for daily and 4-hour data frequency across six scenarios. In other words, the strategies utilizing machine learning models (specified within parentheses) outperformed the benchmark strategies for the given cases.

Trading strategies using simple return:

- Buy and Sell

  D: usdchf - (lstm), usdcad - (ann, gru, tf), audusd - (gru)
  4H: gbpusd - (knn, ann), usdcad - (ann, lstm, gru)
- Only buy

  D: eurusd - (gru, tf), gbpusd - (lstm), usdchf - (gru), usdcad - (ann, gru, tf)
  4H: eurusd - (lstm, gru), gbpusd - (gru), usdchf - (knn, rf), usdcad - (gbdt, lstm, gru), audusd - (tf), nzdusd - (ann, gru)
- Only sell

  D: gbpusd - (gru), usdchf - (ann, lstm)
  4H: eurusd - (ridge, knn, rf, xgboost, gbdt), gbpusd -(ridge, knn, rf, xgboost, gbdt), usdchf - (ridge, knn, rf, xgboost, gbdt), usdcad - (ridge, knn, rf, xgboost, gbdt, ann), audusd - (ridge, knn, rf, xgboost, gbdt), nzdusd - (ridge, rf, xgboost, gbdt)

Trading strategies using log return:

- Buy and Sell

  D: eurusd - (ann, lstm), gbpusd - (lstm, gru), usdchf - (gru)
  4H: eurusd - (gru), gbpusd - (rf), usdcad - (lstm, tf), usdcad - (lstm)
- Only buy

  D: eurusd - (lstm, tf), usdchf - (gbdt)
  4H: gbpusd - (xgboost, gbdt), usdchf - (rf, gbdt), usdcad - (tf), usdcad - (ridge, knn, gru), audusd - (tf)
- Only sell

  D: no instances
  4H: eurusd - (ridge, knn, rf, xgboost, gbdt), gbpusd - (ridge, knn, rf, xgboost, gbdt), usdchf - (ridge, knn, rf, xgboost, gbdt, ann, lstm), usdcad - (ridge, knn, rf, xgboost, gbdt, ann, lstm), audusd - (ridge, knn, rf, xgboost, gbdt), nzdusd - (ridge, knn, rf, xgboost, gbdt)

Based on the aforementioned summary of individual strategies, it is evident that the ANN and LSTM, GRU models consistently offer trading strategies with higher Sharpe ratios than the benchmark in the majority of cases. Furthermore, machine learning models excluding neural networks prove to be more effective (with higher Sharpe ratios) in generating only sell signals when compared to ANN, LSTM, GRU models, and benchmarks. In only a few instances the trend-following strategy outperformed the benchmark in terms of economic significance.

4.2 Integrated strategies

In the previous section, we assessed the performance of individual machine learning-based strategies. In this section, the performance of integrated strategies was compared to the benchmark strategy. The ASR of integrated strategies represents the mean ASR of the strategies assigned to one category of Table 2. Table 12 shows the economic significance results of the integrated trading strategy, trend-following strategies, and benchmark strategies across the six scenarios discussed in the previous section. For instance, the integrated strategy employing machine learning with neural network models produced a trading strategy with a Sharpe ratio of 0.26, generating both buy and sell signals based on GBPUSD daily data using the log return method as outlined in Table 12.

Table 12: Performance metrics of integrated trading strategies for all pairs

			Buy and Sell					Only Buy					Only Sell				
			Int			Ind		Int			Ind		Int			Ind	
		Tr	ML^A	ML^{NN}	ML^O	TF	B	ML^A	ML^{NN}	ML^O	TF	B	ML^A	ML^{NN}	ML^O	TF	B
EURUSD	D	S	-.14	.04	-.25	.17	.22	-.05	.14	-.16	**.31**	.22	-.14	-.14	-.15	.01	.22
EURUSD	D	L	-.03	.3	-.23	.17	.22	-.06	.11	-.17	**.31**	.22	-.08	-.02	-.12	.01	.22
EURUSD	4H	S	.03	.06	.01	.01	.22	-.02	.17	-.13	.16	.22	**.48**	-.04	**.79**	0	.22
EURUSD	4H	L	.05	.11	.02	.01	.22	-.06	-.09	-.04	.16	.22	**.46**	-.08	**.78**	0	.22
GBPUSD	D	S	-.08	-.06	-.1	.15	.22	.05	.2	-.03	.22	.22	.07	.07	.07	-.07	.22
GBPUSD	D	L	.05	**.26**	-.08	.15	.22	-.01	.01	-.02	.22	.22	-.04	-.23	.08	-.07	.22
GBPUSD	4H	S	.15	.14	.16	-.02	.21	.07	.2	0	-.02	.21	**.71**	.1	**1.08**	-.13	.21
GBPUSD	4H	L	.09	-.01	.14	-.02	.21	.04	0	.06	-.02	.21	**.65**	-.03	**1.06**	-.13	.21
USDCHF	D	S	-.03	-.01	-.04	-.16	.12	-.03	.06	-.09	-.1	.12	0	.12	-.07	-.23	.12
USDCHF	D	L	-.02	.04	-.05	-.16	.12	-.02	-.04	-.01	-.1	.12	-.07	-.06	-.07	-.23	.12
USDCHF	4H	S	.01	.03	-.01	.12	.12	.02	-.01	.04	-.07	.12	**.46**	-.16	**.83**	-.32	.12
USDCHF	4H	L	-.04	.02	-.08	.12	.12	-.01	.01	-.01	-.07	.12	**.57**	**.16**	**.81**	-.32	.12
USDCAD	D	S	-.01	**.32**	-.2	**.14**	.12	-.08	**.14**	-.21	**.16**	.12	-.12	-.17	-.08	.03	.12
USDCAD	D	L	-.12	.1	-.26	**.14**	.12	-.16	-.09	-.2	**.16**	.12	-.05	.02	-.09	.03	.12
USDCAD	4H	S	0	.3	-.17	-.17	.12	.01	.07	-.03	-.08	.12	**.51**	-.01	**.83**	-.24	.12
USDCAD	4H	L	-.08	.05	-.15	-.17	.12	.09	.17	.04	-.08	.12	**.54**	.11	.8	-.24	.12
AUDUSD	D	S	-.07	.2	-.24	-.07	.28	-.08	.19	-.25	.19	.28	-.27	-.17	-.32	-.21	.28
AUDUSD	D	L	-.16	-.03	-.24	-.07	.28	-.12	.07	-.23	.19	.28	-.23	-.09	-.31	-.21	.28
AUDUSD	4H	S	.06	.08	.05	.17	.29	0	.05	-.04	**.31**	.29	**.41**	-.08	**.7**	-.04	.29
AUDUSD	4H	L	.07	.06	.08	.17	.29	.01	.06	-.02	**.31**	.29	**.43**	-.01	**.69**	-.04	.29
NZDUSD	D	S	-.1	-.11	-.1	.00	.37	.06	.17	-.01	.14	.37	-.24	-.22	-.25	-.31	.37
NZDUSD	D	L	-.11	-.04	-.15	.00	.37	.07	.19	.01	.14	.37	-.24	-.27	-.22	-.31	.37
NZDUSD	4H	S	.11	.25	.03	.03	.4	-.04	**.41**	-.31	.26	.4	.23	-.21	**.49**	-.23	.4
NZDUSD	4H	L	.01	-.04	.04	.03	.4	-.11	.2	-.3	.26	.4	.26	-.17	**.51**	-.23	.4

Note: Performance metrics tested from January 1, 2003 to June 30, 2023. These numbers show the average annualized Sharpe ratio. Tr: different return. ML^A: mean ASR of all machine learning-based strategies. $ML^N N$: mean ASR of machine learning with NN-based strategies. ML^O: mean ASR of machine learning without NN-based strategies. TF: ASR of trend-following strategy. ASR is in bold indicates a higher ASR than the benchmark strategy.
Source: Own study

We can observe the following general findings from the performance of integrated trading strategies:

- For generating buy and sell signals, there are two instances (out of 24 instances) of machine learning using neural network-based integrated strategies (ML^{NN}) generated trading signals with higher annualized Sharpe Ratios (ASR) compared to the benchmark strategy ASR = $(.26, .32)$. Additionally, two instances (out of 24 instances) of trend-following (TF) strategies produced trading signals with higher ASRs than the benchmark strategies ASR = $(.14, .14)$.
- For generating only buy signal, two instances (out of 24 instances) of machine learning utilizing neural network-based integrated strategies (ML^{NN}) generated trading signals with higher annualized Sharpe Ratios (ASR) compared to the benchmark strategy ASR=(.14, .41). In addition, there are six instances (out of 24 instances) of trend-following (TF) strategies producing trading signals with higher annualized Sharpe ratio (ASR) than benchmark strategy ASR=(.31, .31, .16, .16, .31, .31).
- For generating only sell signal, there are ten instances (out of 24 instances) of all machine learning-based integrated strategies (ML^{A}) generated trading signals with higher ASRs than benchmark strategy ASR=(.48, .46, .71, .65, .46, .57, .51, .54, .41, .43). Additionally, there are eleven instances (out of 24 instances) of machine learning without NN-based integrated strategy (ML^{O}) generated trading signals with higher ASR than benchmark strategy ASR=(.79, .78, 1.08, 1.06, .83, .81, .83, .7, .69, .49, .51).

From the results presented in Table 12, there are almost no integrated trading strategies that outperformed benchmark strategy for generating buy and sell, and only buy signals. But trading strategies based on all machine learning models (ML^{A}) or machine learning models without neural network (ML^{O}) outperformed benchmark strategy for generating only sell signal. All of these successful (outperformed benchmark) strategies generated trading signals on intraday (4-hour) data.

In the next step, we used an independent t-test to verify the robustness of the previous findings observed from the performance of integrated trading strategies. The independent t-test was used as a measure of statistical significance. The mean return of integrated strategies and trend-following strategies was compared to the mean return of benchmark strategies using an independent t-test. Table 13 presents the p-values indicating the statistical significance of integrated strategies, including trend-following and benchmark strategies, across both daily and 4-hour frequency. If the p-value is lower than the significance level (0.10), the null hypothesis is rejected, and the alternative hypothesis is accepted, as indicated in the second row of Table 13. This rejection implies that the mean return of the first-mentioned strategy surpasses the mean return of the second-mentioned strategy.

Table 13: Statistical significance (Independent T-test) for all pairs

			Buy and Sell					Only Buy					Only Sell				
		Tr	ML^A>B	ML^{NN}>B	ML^O>B	TF>B	ML^{NN}>ML^O	ML^A>B	ML^{NN}>B	ML^O>B	TF>B	ML^{NN}>ML^O	ML^A>B	ML^{NN}>B	ML^O>B	TF>B	ML^{NN}>ML^O
EURUSD	D	S	.93	.72	.95	.55	.12	.84	.65	.87	.53	.17	.9	.9	.88	.76	.49
	D	L	.9	.49	.95	.55	**.03****	.87	.71	.88	.53	.17	.86	.81	.86	.76	.33
	4H	S	.77	.7	.77	.73	.38	.94	.67	.94	.64	**.06***	.41	.81	.15	.76	.98
	4H	L	.73	.61	.77	.73	.3	.69	.85	.68	.64	.33	.43	.86	.16	.76	1
GBPUSD	D	S	.87	.85	.83	.59	.45	.75	.63	.79	.61	.25	.74	.77	.71	.83	.57
	D	L	.76	.5	.82	.59	.14	.79	.77	.78	.61	.46	.85	.96	.7	.83	.9
	4H	S	.74	.73	.63	.72	.73	.83	.61	.83	.78	.17	.29	.7	.11	.86	.99
	4H	L	.82	.82	.64	.72	.82	.48	.84	.13	.78	.92	.39	.85	.12	.86	1
USDCHF	D	S	.74	.74	.6	.81	.71	.92	.63	.92	.76	**.08***	.7	.6	.74	.84	.25
	D	L	.69	.66	.68	.81	.43	.14	.72	.14	.76	.87	.74	.74	.74	.84	.46
	4H	S	.7	.67	.7	.36	.44	.39	.7	.38	.73	.62	.48	.81	.27	.89	1
	4H	L	.73	.67	.75	.36	.33	.77	.7	.76	.73	.24	.38	.59	.27	.89	.97
USDCAD	D	S	.76	.61	.85	.59	.46	.81	.6	.87	.59	.12	.76	.79	.74	.68	.63
	D	L	.84	.65	.88	.59	**.09***	.85	.77	.87	.59	.25	.72	.69	.74	.68	.36
	4H	S	.75	.47	.83	.82	**.08***	.42	.68	.41	.75	.59	.46	.72	.3	.82	.99
	4H	L	.8	.69	.82	.82	.3	.3	.58	.3	.75	.7	.43	.64	.31	.82	.99
AUDUSD	D	S	.93	.63	.96	.86	**.05****	.91	.64	.96	.7	**.06***	.98	.94	.98	.94	.18
	D	L	.96	.85	.96	.86	.19	.92	.78	.95	.7	.14	.97	.91	.98	.94	.17
	4H	S	.75	.47	.83	.82	**.08***	.42	.68	.41	.75	.59	.46	.72	.3	.82	.99
	4H	L	.82	.82	.78	.65	.56	.17	.84	.17	.62	.83	.51	.86	.26	.87	.99
NZDUSD	D	S	.93	.63	.96	.86	**.05****	.91	.64	.96	.7	**.06***	.98	.94	.98	.94	.18
	D	L	.97	.94	.96	.87	.32	.87	.79	.9	.82	.26	.98	.99	.98	.99	.59
	4H	S	.87	.75	.89	.86	.19	.97	.68	.98	.77	**.02****	.82	.98	.55	.98	.99
	4H	L	.94	.95	.89	.86	.64	.97	.78	.98	.77	**.04****	.81	.98	.53	.98	.99

Note: Performance metrics tested from January 1, 2003 to June 30, 2023. These numbers show the p-value of the independent t-test. The 2nd row of the table shows the alternative hypothesis. the p-value in bold indicates a p-value that is lower than .1. ML^A: the mean return of all machine learning-based strategies. ML^{NN}: the mean return of machine learning with NN-based strategies. ML^0: the mean return of machine learning without NN-based strategies. TF: the return of a trend-following strategy. B: the return of the benchmark strategy

Based on the findings in Table 13, the following conclusions can be drawn for all pairs:

- For generating buy and sell signals, statistical significance indicated that there are no integrated trading strategies (ML^A, ML^{NN}, ML^O, TF) that generated trading strategies with higher mean return than benchmark strategy for all instances including different return method and different data frequency for all major pair, as all p-values from independent t-tests are higher than 0.1.
  However, there are 6 instances (out of 24 instances) p-value=(.03, .09, .08, .05, .08, .05) that trading strategy based on machine learning models with neural network (ML^{NN}) generated trading strategy with higher mean return than a trading strategy based on machine learning models without neural network(ML^O), as p-values are lower than 0.1.
- For generating only buy signal, statistical significance indicated that there are no integrated trading strategies (ML^A, ML^{NN}, ML^O, TF) that generated trading strategies with higher mean return than benchmark strategy for all instances including different return method and different data frequency for all major pair, as all p-values from independent t-tests are higher than 0.1.
  However, there are 6 instances (out of 24 instances) p-value=(.06, .08, .06, .06, .02, .04) that trading strategy based on machine learning models with neural network (ML^{NN}) generated trading strategy with higher mean return than a trading strategy based on machine learning models without neural network(ML^O), as p-values are lower than 0.1.
- For generating only sell signal, statistical significance indicated that there are no integrated trading strategies (ML^A, ML^{NN}, ML^O, TF) that generated trading strategies with higher mean return than benchmark strategy for all instances including different return method and different data frequency for all major pair, as all p-values from independent t-tests are higher than 0.1.
  However, there are no instances (out of 24 instances) of trading strategy based on machine learning models with neural network (ML^{NN}) generated trading strategy with higher mean return than trading strategy based on machine learning models without neural network(ML^O), as p-values are higher than 0.1.

4.3 Performance evaluation of trading strategy based on all machine learning models: different data frequencies, signal types, and return methods

In this section, the mean return of trading strategies based on all machine learning models within the context of the factors considered in the research question will be described, utilizing both economic and statistical significance. Each trading strategy was presented in six scenarios: two different return methods and three different trading signal types.

For research question 1, it is evident from Table 14 that trading strategies based on all machine learning models on intraday (4-hour) data produced trading signals with a higher annualized Sharpe Ratio (ASR) compared to those based on daily frequency data in six scenarios.

Table 14: Economic significance ASR for different data frequency

		Buy and Sell		Only Buy		Only Sell	
		D	H4	D	H4	D	H4
EURUSD	simple	-.14	**.03**	-.05	-.02	-.14	**.48**
EURUSD	log	-.03	**.05**	-.06	-.06	-.08	**.46**
GBPUSD	simple	-.08	**.15**	.05	**.07**	.07	**.71**
GBPUSD	log	.05	**.09**	-.01	**.04**	-.04	**.65**
USDCHF	simple	-.03	**.01**	-.03	**.02**	0	**.46**
USDCHF	log	-.02	-.04	-.02	-.01	-.07	**.57**
USDCAD	simple	-.01	0	-.08	**.01**	-.12	**.51**
USDCAD	log	-.12	-.08	-.16	**.09**	-.05	**.54**
AUDUSD	simple	-.07	**.06**	-.08	0	-.27	**.41**
AUDUSD	log	-.16	**.07**	-.12	**.01**	-.23	**.43**
NZDUSD	simple	-.1	**.11**	.06	-.04	-.24	**.23**
NZDUSD	log	-.11	**.01**	.07	-.11	-.24	**.26**

Note: Performance metrics tested from January 1, 2003 to June 30, 2023. The table shows the mean annualized Sharpe ratio of all machine learning-based strategies on daily and intraday frequency data. Sharpe ratio in bold indicates higher ASR than daily data and higher than zero.
Source: Own study

Table 15 shows the economic and statistical significance of trading strategies based on all machine learning models that generated three different trading signals: buy and sell, only buy, and only sell. For research question 2, based on Table 15, the following findings are drawn for all pairs:

- Economic significance:
  There is only one instance (out of 24 instances) ASR=(.05) when strategies generating buy and sell signals exhibit higher ASR than other types.
  There are two instances (out of 24 instances) ASR=(.06, .07) when strategies generating only buy signals produce trading signals with higher ASR than other types.
  There are thirteen instances (out of 24 instances) ASR=(.48, .46, .07, .71, .65, .46, .57, .51, .54, .41, .43, .23, .26) when strategies generating only sell signals produce trading signals with higher ASR than other types.
- Statistical significance:
  There are two instances (out of 24 instances) p-value=(.06, .08) when strategies gener-

ating buy and sell signals generate trading signals with higher mean returns than those generating only buy signals.

There are four instances (out of 24 instances) p-values=(.1, .02, .08, .05) when strategies generating only sell signals produce trading signals with higher mean returns than those generating buy and sell signals.

There are three instances (out of 24 instances) p-values=(.06, .08, .07) when strategies generating only sell signals produce trading signals with higher mean returns than those generating only buy signals.

Table 15: Comparison table of strategies for economic significance, ASR

			ASR			Independent t-test					
			B&S	Only Buy	Only Sell	B&S>B	B>B&S	B&S>S	S>B&S	B>S	S>B
EURUSD	D	simple	-.14	-.05	-.14	.53	.47	.49	.51	.47	.53
EURUSD	D	log	-.03	-.06	-.08	.44	.56	.54	.46	.59	.41
EURUSD	4H	simple	.03	-.02	**.48**	**.06***	.94	.9	.1	.94	**.06***
EURUSD	4H	log	.05	-.06	**.46**	.32	.68	.86	.14	.69	.31
GBPUSD	D	simple	-.08	.05	**.07**	.71	.29	.73	.27	.52	.48
GBPUSD	D	log	**.05**	-.01	-.04	.45	.45	.45	.45	.45	.45
GBPUSD	4H	simple	.15	.07	**.71**	.17	.83	.74	.26	.83	.17
GBPUSD	4H	log	.09	.04	**.65**	.8	.2	.82	.18	.49	.51
USDCHF	D	simple	-.03	-.03	0	**.08***	.92	.66	.34	.92	.08
USDCHF	D	log	-.02	-.02	-.07	.87	.13	.47	.53	.13	.87
USDCHF	4H	simple	.01	.02	**.46**	.62	.38	.9	**.1***	.39	.61
USDCHF	4H	log	-.04	-.01	**.57**	.24	.76	.98	**.02****	.77	.23
USDCAD	D	simple	-.01	-.08	-.12	.53	.47	.62	.38	.66	.34
USDCAD	D	log	-.12	-.16	-.05	.41	.59	.79	.21	.79	.21
USDCAD	4H	simple	0	.01	**.51**	.59	.41	.92	**.08***	.42	.58
USDCAD	4H	log	-.08	.09	**.54**	.7	.3	.95	**.05****	.31	.69
AUDUSD	D	simple	-.07	-.08	-.27	.5	.5	.25	.75	.26	.74
AUDUSD	D	log	-.16	-.12	-.23	.59	.41	.42	.58	.34	.66
AUDUSD	4H	simple	.06	0	**.41**	.6	.4	.77	.23	.4	.6
AUDUSD	4H	log	.07	.01	**.43**	.83	.17	.88	.12	.17	.83
NZDUSD	D	simple	-.1	**.06**	-.24	.82	.18	.33	.67	.11	.89
NZDUSD	D	log	-.11	**.07**	-.24	.79	.21	.29	.71	.11	.89
NZDUSD	4H	simple	.11	-.04	**.23**	**.1***	.9	.6	.4	.92	**.08***
NZDUSD	4H	log	.01	-.11	**.26**	.16	.84	.8	.2	.93	**.07***

Note: B-only buy signal. S-only sell signal. B&S-buy and sell signal. ASR in bold indicates the highest ASR of 3 types of trading signal. p-value in bold indicates a p-value that is lower than 0.1.

Table 16 presents the economic and statistical significance of trading strategies based on all machine learning models utilizing different return methods. Economic and statistical significance analyses were conducted for three different trading signal types: buy and sell, only buy, and only sell. From the economic significance, it is observed that trading strategies employing the simple return approach sometimes yield trading signals with higher ASR (Annualized Sharpe Ratio) compared to those using the log return approach. However, in other instances, trading strategies employing the log return approach produce signals with higher ASR than those utilizing the simple return method. Moreover, the statistical significance analysis does not conclusively determine which return approach contributes to generating trading signals with a higher mean return.

Table 16: Performance of machine learning based strategy on different return approach

-	-	Buy and Sell				Only Buy				Only Sell			
		Economic		Statistical		Economic		Statistical		Economic		Statistical	
		ASR		p-value		ASR		p-value		ASR		p-value	
		S	L	L>S	S>L	S	L	L>S	S>L	S	L	L>S	S>L
EURUSD	D	-.14	-.03	.42	.58	-.05	-.06	.53	.47	-.14	-.08	.39	.61
EURUSD	4H	.03	**.05**	.45	.55	-.02	-.06	.13	.87	**.48**	.46	.53	.47
GBPUSD	D	-.08	**.05**	.3	.7	**.05**	-.01	.56	.44	**.07**	-.04	.68	.32
GBPUSD	4H	**.15**	.09	.72	.28	**.07**	.04	.17	.83	**.71**	.65	.66	.34
USDCHF	D	-.03	-.02	.36	.64	-.03	-.02	**.04****	.96	0	-.07	.59	.41
USDCHF	4H	**.01**	-.04	.57	.43	**.02**	-.01	.78	.22	.46	**.57**	.27	.73
USDCAD	D	-.01	-.12	.48	.52	-.08	-.16	.59	.41	-.12	-.05	.4	.6
USDCAD	4H	0	-.08	.6	.4	.01	**.09**	.46	.54	.51	**.54**	.42	.58
AUDUSD	D	-.07	-.16	.61	.39	-.08	-.12	.52	.48	-.27	-.23	.42	.58
AUDUSD	4H	.06	**.07**	.51	.49	0	**.01**	.18	.82	.41	**.43**	.38	.62
NZDUSD	D	-.1	-.11	.42	.58	.06	**.07**	.48	.52	-.24	-.24	.47	.53
NZDUSD	4H	**.11**	.01	.71	.29	-.04	-.11	.51	.49	.23	**.26**	.47	.53

Note: Performance metrics for tested algorithmic trading strategies from January 1, 2003 to June 30, 2023. This table shows the annualized Sharpe ratio of all machine learning-based strategies based on different return approaches. Additionally, it shows the p-value of the independent T-test using the mean return of all machine learning-based strategies based on different return approaches. L indicates log-return. S indicates a simple return. The Sharpe ratio in bold indicates a higher Sharpe ratio than other return approaches and higher than 0. The P-value in bold indicates a p-value that is lower than 0.1.
Source: Own study

CONCLUSIONS

There is a noticeable lack of research applying algorithmic trading strategies to the foreign exchange market compared to studies devoted to Bitcoin and stocks. This might be due to the perception that the foreign exchange market is more efficient than other financial instruments. Additionally, some research articles often overlook crucial issues like model optimization, overfitting, and algorithmic trading biases. Therefore, our study encompassed various machine learning models, walk-forward optimization, a more reliable model creation process, solutions for biases, different return methods, and different trading signals. In this study, Ridge Regression, K Nearest Neighbors, Random Forest, Extreme Gradient Boosting, Gradient Boosting Decision Trees, Artificial Neural Networks, Long Short Term Memory, and Gated Recurrent Units models were used. The main objective of this study is to develop an algorithmic trading system for major foreign exchange rates and compare performances of machine learning-based strategy, trend following strategy to benchmark strategy. We used the Annualized Sharpe ratio metric as economic significance, and the independent t-test as statistical significance to compare various trading strategies. Six major foreign exchange rates (EURUSD, GBPUSD, USDCHF, USDCAD, AUDUSD, NZDUSD) used in this study were downloaded from ICmarket broker in a daily and 4-hour frequency data from 2000 to 2023.

Based on the empirical results, we can observe at least one individual machine learning-based strategy that outperforms the benchmark in each case in terms of economic significance. Moreover, for integrated strategies, we observe that some integrated strategies (ML^A, ML^{NN}, ML^O) outperform the benchmark in generating only sell signals in terms of economic significance. Therefore, we have to reject the first research hypothesis because it is verified positively only with regard to economic significance. The empirical results presented in Table 13 demonstrated that ML^{NN} outperformed ML^O in generating buy and sell signals or only buy signals. So we do not have the basis to reject the third research hypothesis when generating buy and sell or only buy signals. However, we reject the third research hypothesis when generating only sell signals. For trend-following strategies, we considered the TF strategy the same in both individual and integrated strategies in our analysis. Therefore, there are only a few instances where trend-following strategies outperformed the benchmark in terms of economic significance when generating buy and sell signals or only buy signals. However, the statistical significance analysis indicated that trend-following strategies did not consistently outperform the benchmark in all instances. Based on these findings, we can reject the second research hypothesis.

The following findings were discovered for the research questions: all machine learning-based strategies generated trading signals with higher ASR for intraday (4-hour) data compared to daily data. Additionally, trading strategies generated signals with higher ASR for only sell signals compared to buy and sell signals, or only buy signals, in terms of economic significance alone (RQ1). There is no difference in the performance of trading strategies between using different return methods (RQ3). Similar to studies cited in the literature review, our results showed that individual ANN, LSTM, and GRU models based on neural networks generate more effective trading signals in terms of economic significance compared to other individual models (RQ2). However, this assertion cannot be confirmed in terms of statistical significance. Furthermore, our results discovered that the mean performance of integrated strategies without neural networks, or the mean performance of all machine learning models, performs quite well, particularly in the scenario of generating only sell signals. Moreover, our results showed a different finding from some studies that indicate the Moving Average Crossover strategy is more effective than the benchmark strategy. However, it was observed that trading strategies based on moving average crossovers are not efficient at all.

There are very few studies that consider a variety of cases with currency pair data like the one discussed in this study. This study made contributions to the field of algorithmic trading by training and optimizing various machine-learning models across different currency pairs. The implementation of a rolling walk-forward optimization technique addressed the pervasive issue of overfitting, enhancing the robustness of trading strategies. Moreover, the transformation of model outputs into trading signals through predefined generation rules, and the economic and statistical significance tests were used for evaluating the performance of both individual and integrated trading strategies, contributing to a deeper understanding of their performance in real-world scenarios. Overall, this study advanced algorithmic trading by providing comprehensive solutions to prevalent challenges.

There are several ways to expand upon this study. Firstly, it would be advantageous to incorporate sensitivity analysis to address data snooping bias and evaluate the robustness of trading strategies. Secondly, employing ensemble methods could enhance the effectiveness of trading strategies. Thirdly, experimenting with different parameter settings for technical indicators might boost the performance of trading strategies. Lastly, exploring various loss functions could influence the machine learning model optimization.

BIBLIOGRAPHY

Appel, G. (2005). *Technical Analysis: Power Tools for Active Investors*. Financial Times Prentice Hall books. Financial Times/Prentice Hall.

Brown, S. J., Kumar, A., and Goetzmann, W. N. (1998). The dow theory: William peter hamilton's track record re-considered. Available at SSRN: https://ssrn.com/abstract=58690.

Chaboud, A., Chiquoine, B., Hjalmarsson, E., and Vega, C. (2013). Rise of the machines: Algorithmic trading in the foreign exchange market. *Journal of Finance*, 69:2045–2084. FRB International Finance Discussion Paper No. 980.

Chan, E. (2009). *Quantitative Trading: How to Build Your Own Algorithmic Trading Business*. Wiley Trading. Wiley.

Chan, E. (2017). *Machine Trading: Deploying Computer Algorithms to Conquer the Markets*. Wiley Trading. Wiley.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

Chojnacki, K. and Ślepaczuk, R. (2023). Ensembled lstm with walk forward optimization in algorithmic trading. *SSRN Electronic Journal*. Available at SSRN: https://ssrn.com/abstract=4516294.

Donald, L. (1980). Commodity channel index: Tool for trading cyclic trends. *Commodities Magazine*, (219).

Fix, E. and Hodges, J. L. (1989). Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3):238–247.

Fičura, M. (2017). Forecasting foreign exchange rate movements with k-nearest-neighbour, ridge regression and feed-forward neural networks. *SSRN Electronic Journal*. Available at SSRN: https://ssrn.com/abstract=2903547.

Grudniewicz, J. and Ślepaczuk, R. (2023). Application of machine learning in algorithmic investment strategies on global stock markets. *Research in International Business and Finance*, 66:C.

Ho, T. K. (1995). Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ICDAR '95, pages 278–282, M. IEEE Computer Society.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hoerl, A. and Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hsu, P.-H., Taylor, M. P., and Wang, Z. (2016). Technical trading: Is it still beating the foreign exchange market? *Journal of International Economics*, 102(C):188–208.

Islam, M. and Hossain, E. (2021). Foreign exchange currency rate prediction using a gru-lstm hybrid network. *Soft Computing Letters*, 3:100009.

Jarque, C. and Bera, A. (1980). Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, 6(3):255–259.

Jeffrey, F. and Kenneth, F. (1990). Chartists, fundamentalists, and trading in the foreign exchange market. *American Economic Review*, 80(2):181–185.

John, B. (2002). *Bollinger on Bollinger Bands*. McGraw-Hill Education.

Kryńska, K. and Ślepaczuk, R. (2023). Daily and intraday application of various architectures of the lstm model in algorithmic investment strategies on bitcoin and the s&p 500 index. Available at SSRN: `https://ssrn.com/abstract=4628806` or `http://dx.doi.org/10.2139/ssrn.4628806`.

Kyunghyun, C., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Lukas, M. (2010). The use of technical analysis by fund managers: International evidence. *Journal of Banking & Finance*, 34(11):2573–2586.

Mark, T. and Helen, A. (1992). The use of technical analysis in the foreign exchange market. *Journal of International Money and Finance*, 11(3):304–314.

Meier, G. M. (1971). The bretton woods agreement – twenty-five years after. *Stanford Law Review*, 23(2):235–275.

Menkhoff, L. and Taylor, M. P. (2007). The obstinate passion of foreign exchange professionals: Technical analysis. *Journal of Economic Literature*, 45(4):936–972.

Miskolczi, P. (2017). Note on simple and logarithmic return. *Applied Studies in Agribusiness and Commerce*, 11(1-2):127–136.

Nayak, R. K., Mishra, D., and Rath, A. K. (2019). An optimized svm-k-nn currency exchange forecasting model for indian currency market. *Neural Comput. Appl.*, 31(7):2995–3021.

Neely, C. J. and Weller, P. A. (2011). Technical analysis in the foreign exchange market. (2011-001B). Available at SSRN: `https://ssrn.com/abstract=1734836` or `http://dx.doi.org/10.2139/ssrn.1734836`.

Qi, M. and Wu, Y. (2006). Technical trading-rule profitability, data snooping, and reality check: Evidence from the foreign exchange market. *Journal of Money, Credit and Banking*, 38(8):2135–2158.

Richard, L. and Lee, T. (1993). The significance of technical trading-rule profits in the foreign exchange market: a bootstrap approach. *Journal of International Money and Finance*, 12(5):451–474.

Robert, P. and Perry, K. (1992). *Design, testing, and optimization of trading systems*. John Wiley & Sons, Inc., USA.

Sharpe, W. (1966). Mutual fund performance. *The Journal of Business*, 39(1):119–138.

Student (1908). The probable error of a mean. *Biometrika*, 6(1):1–25.

Tinghino, M. (2008). *Technical Analysis Tools: Creating a Profitable Trading System*. Bloomberg Professional. Bloomberg Press.

Tulchinsky, I. (2019). *Introduction to Alpha Design*. John Wiley Sons, Ltd.

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

Yao, J. and Tan, C. L. (2000). A case study on using neural networks to perform technical forecasting of forex. *Neurocomputing*, 34:79–98.

Yıldırım, D., Toroslu, I., and Fiore, U. (2021). Forecasting directional movement of forex data using lstm with technical and macroeconomic indicators. *Financial Innovation*, 7(1):1.

Yu, L., Wang, S., and Lai, K. K. (2007). *Foreign-Exchange-Rate Forecasting With Artificial Neural Networks*. Number 978-0-387-71720-3 in International Series in Operations Research and Management Science. Springer.

APPENDIX

Table 17: Performance metrics for Individual strategies for GBPUSD Daily

			Buy and Sell				Only Buy				Only Sell			
TR	Str	M	NP	NT	TC	ASR	NP	NT	TC	ASR	NP	NT	TC	ASR
S	ML^{O}	RIDGE	9076	1978	613	-.08	8668	1300	404	-.02	9004	1298	403	.1
		KNN	6284	1155	361	-.09	9016	928	290	.02	8892	951	297	.06
		RF	9115	1986	616	-.04	8666	1325	411	-.06	8804	1325	411	.06
		XGBOOST	5076	1251	380	-.17	8766	1327	412	-.04	8755	1326	411	.07
		GBDT	9040	1993	616	-.09	8647	1327	412	-.08	8913	1327	412	.04
	ML^{NN}	ANN	15457	850	265	.08	8873	376	118	.21	8524	401	128	.08
		LSTM	13297	208	63	-.11	10138	83	25	**.23**	8424	76	24	-.13
		GRU	15054	264	80	-.15	9404	91	27	.16	7244	72	23	**.25**
	TF	MAs	17840	79	24	.15	9101	45	14	.22	8983	46	14	-.07
	B	B/H	17954	2	1	.22	17954	2	1	.22	17954	2	1	.22
L	ML^{O}	RIDGE	9159	1970	610	-.06	8671	1299	403	-.01	9008	1297	403	.11
		KNN	6327	1120	350	-.01	8639	972	305	.06	9003	962	301	.05
		RF	9181	1971	611	-.03	8683	1319	409	-.04	8909	1326	412	.08
		XGBOOST	6123	1466	451	-.22	8936	1323	410	-.05	8973	1321	410	.08
		GBDT	9020	1968	609	-.09	8652	1322	411	-.07	8853	1327	412	.05
	ML^{NN}	ANN	15694	828	254	.05	8825	460	145	.03	8802	403	126	-.3
		LSTM	13954	261	83	**.39**	9727	77	23	.05	7861	138	43	-.15
		GRU	16393	211	60	**.35**	7314	104	31	-.05	9363	73	22	-.22
	TF	MAs	17840	79	24	.15	9101	45	14	.22	8983	46	14	-.07
	B	B/H	17954	2	1	.22	17954	2	1	.22	17954	2	1	.22

Table 18: Performance metrics for Individual strategies for GBPUSD H4

			Buy and Sell				Only Buy				Only Sell			
TR	Str	M	NP	NT	TC	ASR	NP	NT	TC	ASR	NP	NT	TC	ASR
S	ML^{O}	RIDGE	22158	11452	3545	.14	18427	7936	2465	-.21	20973	7939	2466	**1.08**
		KNN	15712	6903	2134	**.26**	19245	5601	1751	.18	20312	5602	1753	**.96**
		RF	21385	11177	3460	.21	18301	8023	2493	-.22	20918	8008	2487	**1.14**
		XGBOOST	14368	7740	2406	.02	19265	7870	2449	.09	19770	7735	2408	**1.09**
		GBDT	21615	11180	3455	.15	18086	7984	2481	.15	21193	8017	2492	**1.1**
	ML^{NN}	ANN	35173	1868	543	**.27**	19986	664	202	.11	21305	915	279	.15
		LSTM	38819	59	18	.3	25131	15	5	.15	16468	112	33	-.03
		GRU	38325	26	8	-.14	15998	10	3	**.34**	14810	11	3	.18
	TF	MAs	38618	581	182	-.02	18521	238	76	-.02	18851	392	121	-.13
	B	B/H	39286	2	1	.21	39286	2	1	.21	39286	2	1	.21
L	ML^{O}	RIDGE	22167	11478	3552	.14	18423	7936	2465	.07	20980	7943	2467	**1.09**
		KNN	15697	6922	2142	.12	19163	5560	1736	-.28	20257	5602	1751	**.86**
		RF	21579	11203	3466	**.22**	18063	8023	2492	-.03	21022	8016	2490	**1.12**
		XGBOOST	14896	8164	2534	.06	18625	7848	2439	**.28**	20779	7923	2464	**1.12**
		GBDT	21802	11210	3473	.18	18292	8013	2489	**.25**	21154	7976	2478	**1.11**
	ML^{NN}	ANN	36476	1606	495	.15	19836	1007	316	-.22	16232	628	194	-.25
		LSTM	38107	58	20	-.21	17928	69	20	.11	20547	15	5	.15
		GRU	37800	180	49	.03	19176	46	16	.11	18457	18	6	.02
	TF	MAs	38618	581	182	-.02	18521	238	76	-.02	18851	392	121	-.13
	B	B/H	39286	2	1	.21	39286	2	1	.21	39286	2	1	.21

Table 19: Performance metrics for Individual strategies for USDCHF Daily

			Buy and Sell				Only Buy				Only Sell			
TR	Str	M	NP	NT	TC	ASR	NP	NT	TC	ASR	NP	NT	TC	ASR
S	ML^O	RIDGE	5881	1892	395	**.15**	5806	1365	285	-.26	6342	1364	285	-.08
		KNN	3752	1069	226	-.15	5917	913	193	-.06	6342	927	195	-.07
		RF	5850	1875	392	-.04	5985	1395	292	-.23	6207	1381	288	-.05
		XGBOOST	4182	1384	291	-.16	6023	1379	288	.08	6141	1384	289	-.08
		GBDT	5946	1897	397	-.01	5998	1388	290	.03	6244	1391	290	-.09
	ML^{NN}	ANN	9882	748	157	-.12	6374	377	79	-.01	7167	389	82	**.19**
		LSTM	8415	260	54	**.18**	6894	128	28	.06	7742	155	32	**.2**
		GRU	9664	243	51	-.09	5426	119	25	**.14**	4876	88	18	-.04
	TF	MAs	12347	75	15	-.16	5202	41	9	-.1	7101	43	9	-.23
	B	B/H	12676	2	0	.12	12676	2	0	.12	12676	2	0	.12
L	ML^O	RIDGE	5872	1890	395	.04	5818	1366	286	-.2	6320	1364	285	-.08
		KNN	3776	1057	224	-.18	5856	945	199	-.04	6355	943	198	-.05
		RF	5917	1878	392	-.04	6024	1381	289	-.03	6213	1383	289	-.06
		XGBOOST	4417	1367	290	-.06	5887	1383	289	-.06	6237	1392	291	-.05
		GBDT	5846	1893	395	-.03	5903	1387	290	**.28**	6173	1379	288	-.12
	ML^{NN}	ANN	10552	687	143	-.06	5645	364	77	-.09	6150	427	91	-.09
		LSTM	9367	234	50	.04	5558	147	29	-.03	8266	93	19	-.12
		GRU	10514	280	63	**.15**	5328	85	17	.02	4895	146	30	.03
	TF	MAs	12347	75	15	-.16	5202	41	9	-.1	7101	43	9	-.23
	B	B/H	12676	2	0	.12	12676	2	0	.12	12676	2	0	.12

Table 20: Performance metrics for Individual strategies for USDCHF H4

			Buy and Sell				Only Buy				Only Sell			
TR	Str	M	NP	NT	TC	ASR	NP	NT	TC	ASR	NP	NT	TC	ASR
S	ML^O	RIDGE	14432	11115	2311	.0	12828	8020	1672	-.08	14422	8011	1670	**.9**
		KNN	10303	6760	1407	.08	13417	5559	1163	**.38**	13768	5561	1163	**.61**
		RF	14135	11114	2310	-.13	12912	8175	1704	**.22**	14264	8189	1707	**.89**
		XGBOOST	9346	7210	1508	.08	13177	8144	1698	-.2	14037	8108	1692	**.87**
		GBDT	14472	11205	2334	-.06	13011	8154	1699	-.1	14240	8150	1699	**.87**
	ML^{NN}	ANN	23208	1930	392	-.07	11221	785	156	-.05	11805	670	134	-.05
		LSTM	27276	116	26	.12	13566	16	4	.0	15010	46	11	-.23
		GRU	27351	34	7	.03	12538	46	11	.0	10933	18	3	-.2
	TF	MAs	27024	842	173	.12	12567	402	83	-.07	14204	448	92	-.32
	B	B/H	27666	2	0	.12	27666	2	0	.12	27666	2	0	.12
L	ML^O	RIDGE	14392	11139	2316	-.03	12839	8038	1675	-.19	14399	8034	1674	**.89**
		KNN	10222	6883	1430	-.16	13530	5583	1166	-.29	13861	5581	1167	**.56**
		RF	14128	11092	2307	-.05	13005	8191	1707	**.24**	14216	8182	1705	**.91**
		XGBOOST	9674	7588	1597	-.06	13311	8107	1691	-.12	14098	8167	1703	**.9**
		GBDT	14433	11202	2330	-.09	13065	8159	1701	**.28**	14111	8134	1695	**.81**
	ML^{NN}	ANN	24625	1962	413	.01	16312	1309	274	-.1	16707	793	165	**.28**
		LSTM	27434	20	4	.03	12699	14	3	.03	15583	11	2	**.13**
		GRU	27416	18	4	.03	12958	11	2	.1	14885	65	12	.05
	TF	MAs	27024	842	173	.12	12567	402	83	-.07	14204	448	92	-.32
	B	B/H	27666	2	0	.12	27666	2	0	.12	27666	2	0	.12

Table 21: Performance metrics for Individual strategies for USDCAD Daily

			Buy and Sell				Only Buy				Only Sell			
TR	Str	M	NP	NT	TC	ASR	NP	NT	TC	ASR	NP	NT	TC	ASR
S	ML^O	RIDGE	6629	1936	468	-.27	6333	1349	324	-.25	6331	1349	324	-.08
		KNN	4864	1170	284	-.1	6436	987	237	-.09	6502	1006	242	-.05
		RF	6662	1946	471	-.23	6404	1369	329	-.21	6267	1378	331	-.08
		XGBOOST	4146	1244	300	-.21	6358	1373	330	-.27	6350	1371	329	-.1
		GBDT	6622	1944	471	-.2	6358	1381	331	-.25	6217	1380	331	-.11
	ML^{NN}	ANN	11629	781	189	**.38**	6367	436	105	**.16**	5694	446	106	-.15
		LSTM	8950	241	58	-.07	7340	81	20	-.06	6484	124	30	-.1
		GRU	12283	166	39	**.63**	7553	64	15	**.3**	3595	124	29	-.22
	TF	MAs	13172	87	22	**.14**	6413	52	13	**.16**	6601	50	12	.03
	B	B/H	13127	2	1	.12	13127	2	1	.12	13127	2	1	.12
L	ML^O	RIDGE	6636	1933	467	-.26	6290	1353	325	-.27	6332	1343	322	-.08
		KNN	4590	1190	289	-.14	6344	964	232	-.03	6362	996	239	-.1
		RF	6720	1960	474	-.22	6364	1378	331	-.23	6226	1378	331	-.11
		XGBOOST	4011	1323	318	-.43	6280	1379	331	-.27	6364	1367	328	-.08
		GBDT	6660	1970	476	-.24	6379	1376	330	-.23	6256	1376	330	-.08
	ML^{NN}	ANN	11718	801	193	-.11	5655	405	97	-.15	7439	430	105	.07
		LSTM	9821	242	57	**.38**	6333	100	25	-.09	6353	74	18	.07
		GRU	12049	236	58	.04	6057	57	14	-.03	7116	108	28	-.08
	TF	MAs	13172	87	22	**.14**	6413	52	13	**.16**	6601	50	12	.03
	B	B/H	13127	2	1	.12	13127	2	1	.12	13127	2	1	.12

Table 22: Performance metrics for Individual strategies for USDCAD H4

			Buy and Sell				Only Buy				Only Sell			
TR	Str	M	NP	NT	TC	ASR	NP	NT	TC	ASR	NP	NT	TC	ASR
S	ML^O	RIDGE	16627	11449	2771	-.13	13591	7924	1902	.12	15406	7930	1903	**.86**
		KNN	11966	6967	1686	-.19	13742	5654	1354	-.35	15376	5688	1363	**.66**
		RF	16431	11403	2760	-.16	13504	8045	1931	.12	15515	8048	1931	**.86**
		XGBOOST	11191	7792	1878	-.22	13498	7957	1910	-.21	15639	7981	1916	**.86**
		GBDT	16502	11409	2760	-.16	13586	8053	1932	**.19**	15409	8042	1930	**.91**
	ML^{NN}	ANN	27887	1182	291	**.15**	12472	867	210	-.2	17505	481	113	**.18**
		LSTM	29849	21	5	**.5**	11390	47	11	**.25**	16660	11	3	-.15
		GRU	29616	19	5	**.26**	13296	9	2	**.15**	14671	9	2	-.07
	TF	MAs	29093	620	143	-.17	14903	311	72	-.08	14021	365	86	-.24
	B	B/H	29581	2	1	.12	29581	2	1	.12	29581	2	1	.12
L	ML^O	RIDGE	16596	11440	2769	-.13	13578	7926	1902	**.17**	15405	7924	1902	**.86**
		KNN	12074	7080	1714	-.18	13593	5661	1358	**.16**	15366	5730	1374	**.54**
		RF	16340	11368	2752	-.13	13566	8050	1932	-.31	15535	8050	1931	**.87**
		XGBOOST	11299	7965	1914	-.15	13963	7760	1853	.07	15858	7818	1871	**.84**
		GBDT	16477	11446	2768	-.17	13659	8034	1928	.09	15483	8016	1923	**.87**
	ML^{NN}	ANN	27126	1665	386	.1	12695	1572	360	-.02	16801	646	161	**.28**
		LSTM	29342	95	22	**.16**	13025	20	5	.04	13239	47	12	**.19**
		GRU	28875	61	16	-.1	16518	14	3	**.5**	14161	22	6	-.14
	TF	MAs	29093	620	143	-.17	14903	311	72	-.08	14021	365	86	-.24
	B	B/H	29581	2	1	.12	29581	2	1	.12	29581	2	1	.12

Table 23: Performance metrics for Individual strategies for AUDUSD Daily

			Buy and Sell				Only Buy				Only Sell			
TR	Str	M	NP	NT	TC	ASR	NP	NT	TC	ASR	NP	NT	TC	ASR
S	ML^O	RIDGE	5905	1993	318	-.24	5843	1353	217	-.28	5304	1351	217	-.35
		KNN	3675	1097	173	-.29	6387	985	157	-.19	4705	995	159	-.3
		RF	5958	2013	320	-.23	6083	1371	220	-.31	5093	1369	219	-.36
		XGBOOST	4468	1567	249	-.24	6027	1378	221	-.27	5136	1364	219	-.32
		GBDT	5854	1928	306	-.18	6075	1351	216	-.2	5191	1365	219	-.29
	ML^{NN}	ANN	10277	836	135	.13	6506	373	59	.08	4620	319	50	-.12
		LSTM	7573	243	40	.08	7788	124	20	.27	6042	146	24	-.2
		GRU	9757	201	31	**.38**	6972	109	17	.22	6263	176	29	-.19
	TF	MAs	11642	114	18	-.07	6593	62	10	.19	4968	58	9	-.21
	B	B/H	11990	2	0	.28	11990	2	0	.28	11990	2	0	.28
L	ML^O	RIDGE	5901	2000	319	-.26	5842	1353	217	-.29	5323	1355	217	-.36
		KNN	3382	1034	164	-.35	6361	991	158	-.2	4834	995	159	-.31
		RF	5923	1999	317	-.22	6062	1363	218	-.25	5128	1363	218	-.31
		XGBOOST	4360	1538	246	-.22	6069	1360	218	-.2	5188	1356	217	-.26
		GBDT	5929	1974	313	-.16	6074	1353	217	-.21	5218	1354	217	-.30
	ML^{NN}	ANN	10182	820	130	.11	6397	383	62	.07	4824	459	73	-.19
		LSTM	7368	228	36	-.09	5509	114	18	.18	5752	138	23	-.16
		GRU	9233	263	44	-.09	6087	154	25	-.03	5526	192	31	.08
	TF	MAs	11642	114	18	-.07	6593	62	10	.19	4968	58	9	-.21
	B	B/H	11990	2	0	.28	11990	2	0	.28	11990	2	0	.28

Table 24: Performance metrics for Individual strategies for AUDUSD H4

			Buy and Sell				Only Buy				Only Sell			
TR	Str	M	NP	NT	TC	ASR	NP	NT	TC	ASR	NP	NT	TC	ASR
S	ML^O	RIDGE	14747	11718	1868	.05	12977	7811	1253	.07	13855	7811	1253	**.71**
		KNN	9950	7084	1118	.13	13668	5849	937	-.22	13285	5851	937	**.57**
		RF	14418	11571	1839	.02	13123	7933	1272	.02	13680	7923	1271	**.74**
		XGBOOST	11611	9414	1509	.02	13440	7927	1272	.04	13503	7943	1274	**.75**
		GBDT	14220	11349	1800	.04	13270	7836	1257	-.09	13615	7867	1262	**.74**
	ML^{NN}	ANN	23891	1922	295	.18	12143	1126	186	-.17	15713	640	96	-.1
		LSTM	26578	113	19	.1	13169	10	2	.11	13716	15	2	-.27
		GRU	25928	58	9	-.04	14301	12	2	.21	13510	59	12	.13
	TF	MAs	26851	636	105	.17	13673	273	46	**.31**	13169	345	57	-.04
	B	B/H	27089	2	0	.29	27089	2	0	.29	27089	2	0	.29
L	ML^O	RIDGE	14712	11711	1867	.01	12952	7818	1254	.18	13835	7819	1254	**.7**
		KNN	9679	6887	1086	.16	13565	5818	931	-.23	13401	5801	928	**.63**
		RF	14354	11463	1820	.09	13096	7925	1271	.15	13640	7945	1275	**.73**
		XGBOOST	11234	9247	1477	.05	13307	7951	1276	-.24	13612	7981	1280	**.68**
		GBDT	14337	11337	1796	.11	13168	7880	1264	.05	13666	7833	1257	**.69**
	ML^{NN}	ANN	23156	1930	317	.25	12494	1350	214	.02	13315	709	112	.01
		LSTM	26454	104	19	.02	11417	83	12	.27	10623	15	3	.02
		GRU	26457	69	10	-.1	9648	8	1	-.11	15309	10	2	-.05
	TF	MAs	26851	636	105	.17	13673	273	46	**.31**	13169	345	57	-.04
	B	B/H	27089	2	0	.29	27089	2	0	.29	27089	2	0	.29

Table 25: Performance metrics for Individual strategies for NZDUSD Daily

			Buy and Sell				Only Buy				Only Sell			
TR	Str	M	NP	NT	TC	ASR	NP	NT	TC	ASR	NP	NT	TC	ASR
S	ML^{O}	RIDGE	5620	1965	275	-.15	5497	1316	185	-.03	5113	1312	185	-.25
		KNN	3189	1067	149	.0	5728	996	140	.09	4915	1012	142	-.15
		RF	5556	1992	278	-.16	5546	1355	191	-.02	5037	1346	190	-.26
		XGBOOST	4429	1613	225	-.03	5540	1354	191	-.05	4980	1356	191	-.32
		GBDT	5350	1914	267	-.17	5574	1350	190	-.06	5000	1348	190	-.28
	ML^{NN}	ANN	8825	836	117	-.29	4500	390	55	.16	6011	411	57	-.3
		LSTM	7810	208	30	.08	5380	114	16	.23	5924	114	17	-.33
		GRU	8096	314	43	-.1	5412	101	14	.13	6120	130	18	-.03
	TF	MAs	10801	130	19	.0	5903	71	10	.14	4798	75	11	-.31
	B	B/H	11156	2	0	.4	11156	2	0	.37	11156	2	0	.37
L	ML^{O}	RIDGE	5615	1967	275	-.15	5495	1315	185	-.03	5103	1314	185	-.26
		KNN	3382	1059	149	.02	5814	966	136	.14	5025	990	139	-.18
		RF	5506	1990	278	-.2	5560	1347	190	-.01	5021	1342	189	-.25
		XGBOOST	4303	1546	216	-.2	5612	1346	189	-.04	4961	1339	189	-.23
		GBDT	5322	1945	272	-.22	5532	1335	188	-.04	5000	1324	186	-.2
	ML^{NN}	ANN	8023	816	115	.05	5881	452	63	.22	4557	431	60	-.26
		LSTM	6765	265	38	-.38	4637	143	20	.11	5089	99	13	-.37
		GRU	9742	164	24	.21	5535	94	13	.21	6843	110	15	-.2
	TF	MAs	10801	130	19	.0	5903	71	10	.14	4798	75	11	-.31
	B	B/H	11156	2	0	.37	11156	2	0	.37	11156	2	0	.37

Table 26: Performance metrics for Individual strategies for NZDUSD H4

			Buy and Sell				Only Buy				Only Sell			
TR	Str	M	NP	NT	TC	ASR	NP	NT	TC	ASR	NP	NT	TC	ASR
S	ML^{O}	RIDGE	13699	11702	1638	.06	12147	7790	1097	-.35	12799	7795	1097	**.54**
		KNN	8525	6542	910	.02	12637	5698	800	-.24	12347	5741	806	.37
		RF	13378	11569	1618	.01	12061	7936	1117	-.3	12839	7942	1118	**.5**
		XGBOOST	10888	9687	1353	.06	12148	7939	1117	-.32	12755	7918	1114	**.54**
		GBDT	13125	11365	1590	.0	12108	7857	1107	-.35	12710	7880	1109	**.51**
	ML^{NN}	ANN	20058	1935	279	.33	11212	1591	221	**.41**	10656	1040	145	-.38
		LSTM	24059	90	13	.1	10123	74	10	.39	10770	79	11	-.19
		GRU	25136	91	12	.31	12973	11	2	**.44**	13363	12	2	-.05
	TF	MAs	24938	714	103	.03	13199	330	48	.26	11591	390	56	-.23
	B	B/H	25397	2	0	.4	25397	2	0	.4	25397	2	0	.4
L	ML^{O}	RIDGE	13694	11699	1638	.06	12151	7792	1097	-.35	12802	7790	1096	**.55**
		KNN	8672	6669	929	.06	12732	5716	803	-.13	12566	5696	800	**.48**
		RF	13427	11565	1619	.03	12134	7935	1117	-.31	12803	7944	1118	**.49**
		XGBOOST	10620	9578	1340	-.01	12108	7947	1118	-.34	12691	7921	1115	**.51**
		GBDT	13260	11408	1596	.04	12127	7836	1103	-.35	12729	7865	1107	**.53**
	ML^{NN}	ANN	18985	2262	323	-.07	12551	948	138	.16	11613	1105	149	.03
		LSTM	24776	123	18	-.02	13240	25	3	.20	10705	74	12	-.33
		GRU	24801	58	7	-.04	15609	9	1	.24	15750	9	1	-.19
	TF	MAs	24938	714	103	.03	13199	330	48	.26	11591	390	56	-.23
	B	B/H	25397	2	0	.4	25397	2	0	.40	25397	2	0	.4

University of Warsaw
Faculty of Economic Sciences
44/50 Długa St.
00-241 Warsaw
www.wne.uw.edu.pl
ISSN 2957-0506