Predictive Modeling of Forex Trading Signals Using regression and classification methods

Sugarbayar Enkhbayar[1]
[1] *University of Warsaw, Faculty of Economics*
s.enkhbayar@student.uw.edu.pl

Abstract:
Main purpose of this project is applying machine learning regression and classification methods for creating trading signals on foreign exchange markets. Many trading firms use advanced machine learning models with financial theory for predicting exact prices and determining trends. Most research articles indicate that machine learning models with neural networks predict prices with higher accuracy than traditional models.

Keywords: algorithmic trading, regression, classification

1. Introduction

Central purpose of this project is using machine learning models for algorithmic trading on forex pairs. I applied KNN, RandomForest, XGBoost and Neural network models for creating trading signals. In other words, every model created a trading system that gives us buy or sell signals. If the price increases, we should buy or long position. If the price decreases, we should sell or short position.

Another purpose is comparing performance of machine learning strategy to benchmark strategy and trend following strategy. I assumed a buy and hold strategy as a benchmark strategy. For trend following strategy, I used multiple MA cross strategy.

For the regression problem, our target variable is just simple return and I used a quantile of train set for generating trading signals. For classification problems, I used a newly created target variable. In other words, if the return is higher than 0, it will be 1, otherwise it will be 0.

I put forward the following hypothesis and research question for this project. First hypothesis related to efficient market hypothesis theory. Main idea of this theory is that traders can't make profit using historical technical indicators. Because technical indicators won't work and it fully reflects all past markets. I compared mentioned strategies on hypothesis 2 and hypothesis 3. We can use some statistical performance metrics for comparing machine learning models. But we can't say anything about profit using these indicators. So I used financial performance metrics IR for comparing strategies. Higher IR is a better strategy.

- Hypothesis 1: We can reject EMH. We can make profit using technical indicators.
- Hypothesis 2: The ML strategy outperform B&H strategy
- Hypothesis 3: The ML strategy outperform Trend Following Strategy
- Research question: Which ML strategy generates a trading signal with the highest information ratio (IR)?

2. Exploratory Data Analysis

Every financial product has its own characteristics. So I decided to use different forex pairs for regression and classification problems.

For the regression problem, the target variable is EURUSD daily simple return. We used close price to calculate the return of price. For the classification problem, the target variable is 1 or 0. 1 means today's return is higher than yesterday's return. 0 means today's return is lower than yesterday's return. All data was downloaded from MetaTrader 5 platform using my own trading account. 2 forex pairs' range started from 1st of January 2000 to 30th of June 2023. Each pair has 6172 observations.

Table: Number of data

Pair	Daily	Date range
EURUSD	6172	2000.01.01 - 2023.06.30
GBPUSD	6171	

Traders mainly use the following 3 types of indicators for decision making. They are fundamental indicators, technical indicators and statistical indicators.

- Fundamental indicators related with economic news, announcements;
- Technical indicators related with mathematical formula that use price's information such as close, open, high, low, volume, spread and volatility
- Statistical indicators related with some statistical formula

But I focused on only technical and statistical indicators on this project. I used default options for all technical indicators.

Table: Summary of variables

Category	Regression task	Classification task
Target	simple return (close)	new target (0, 1)
Explanatory (technical)	ema {10. 20. 50. 100. 200} macd {26. 12. 9} rsi {14} stochastic {14. 3} cci {20. 0.015} bollinger band {20. 2} atr {14} william R {14} adx {14}	ema {10. 20. 50. 100. 200} macd {26. 12. 9} rsi {14} stochastic {14. 3} cci {20. 0.015} bollinger band {20. 2} atr {14} william R {14} adx {14}
Explanatory (statistical)	momentum. avg price. range. ohlc	momentum. avg price. range. ohlc

Note: technical indicator was calculated using own functions and pandas ta library with default settings

Table: Summary of variables

Category	Variable	Equation
Technical indicator	EMA	$(M(t) - EMA(M, t-1, \tau)).\frac{2}{\tau+1} + EMA(M, t-1, \tau)$
	MACD	$MACD\ \ Line: (12\ \ Day\ \ EMA - 26\ \ Day\ \ EMA) \quad Signal\ \ Line: 9\ \ Day\ \ EMA\ \ of\ \ MACD\ \ Line$
	RSI	$RSI = 100 - \frac{100}{1 + (averagegain/averageloss)}$
	STOCH	$\%K = \frac{C - L_{14}}{H_{14} - L_{14}} * 100 \qquad \%D_t = \frac{K1+K2+K3}{3}$
	CCI	$CCI = \frac{Typical\ \ Price - 20\ \ Period\ \ SMA\ \ of\ \ TP}{0.015 * Mean\ \ Deviation}$ $Typical\ \ Price\ \ (TP) = \frac{High + Low + Close}{3}$
	BB	$BB_t = SMA_{20} \pm stdev_{20} * 2$
	ATR	$ATR_\tau = ATR_{\tau-1} * (period - 1) + \frac{TR_t}{period}$
	William R	$R = \frac{\max(high) - close}{\max(high) - \min(low)} * -100$
	ADX	$+DI = 100 * EMA\left(\frac{+DMI}{Average\ \ True\ \ Range}\right) \quad DI = 100 * EMA\left(\frac{-DMI}{Average\ \ True\ \ Range}\right) \quad ADX = 100 * EMA\left(Absolute\ \ Value\ \ of\ \ \left(\frac{+DI - -DI}{+DI + -DI}\right)\right)$
Statistical indicator	Momentum	$momentum=O-C$
	Average Price	$avg\_price=(L+H)/2$
	Range	$range=H-L$
	OHLC	$ohlc=(H+L+C+O)/4$

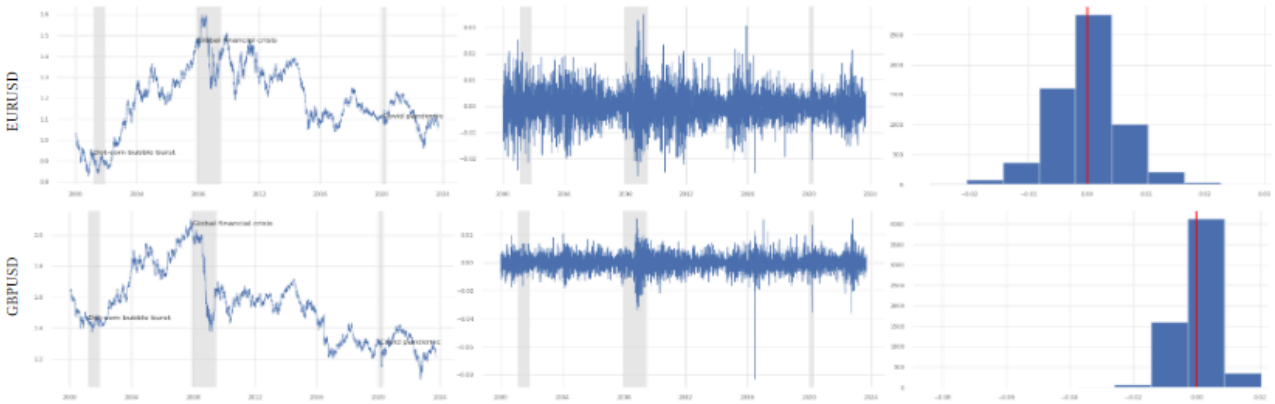
Note: explanatory variables are selected based on multiple research article. But mainly focused on following articles "**Algorithmic financial trading with deep convolutional neural network**" by Omer Berat Sezer. 2018. "**Application of machine learning in quantitative investment strategies on global stock market**" by Jan Grudniewicz. Robert ślepaczuk. 2021. "**An ensemble Architecture Incorporating Machine Learning Models and Genetic Algorithm Optimization for Forex Trading**" by Leonard Kin yung Loh. 2022

Following table shows us 3 big economic recessions in the US. There is only one reason why I mentioned it here. Structural changes like an economic recession affects trading strategy. We can lose all profit during an economic recession. We can see EURUSD and GBPUSD prices decreased during this period. Also, standard deviation of return reached maximum value during this period. Return of the following 2 forex pairs is close to 0.

Table: Summary of variables

Economic recessions	Date range
Dot-com bubble bursts	2001.03-2001.11
Global financial crisis	2007.12-2009.06
Covid pandemic	2020.02-2020.04

Figure: Close price dynamic. return dynamic. return histogram



Following 2 tables show us descriptive statistics of regression and classification problems. P value of jarque is lower than 0.05. It means the return of EURUSD is significantly different from normal distribution. And the mean return is close to 0. Also GBPUSD's target variable is balanced. So we can use it for further analysis.

Table: Descriptive statistic of target variable (regression)

	EURUSD
Stat	Daily
Count	6171
Mean (10^{-6})	23
Std (10^{-3})	6.0
Min (10^{-2})	-2.7
Max (10^{-2})	3.5
Skew	0.1
Kurt	1.8
JB(pvalue)	0.00

3. Methodology

Rolling walk forward approach:

When we train machine learning models with time series data, overfitting is the biggest problem. So to avoid overfitting, I decided to use a rolling walk forward approach. Here is the walk forward approach's selected parameters.

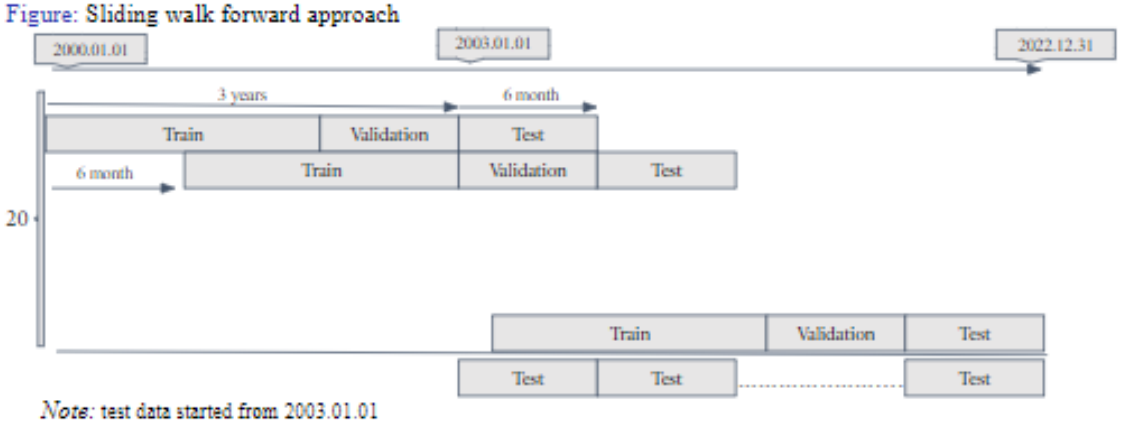
- Train set is 600 days
- Validation set is last few day if train set and 156 days
- Test set is 252 days or 1 year
- Sliding size is 252 days or 1 year

Main idea of this methodology is rolling trains, validation and test sets. Number of this approach's window is 20. It means all training, hyperparameter tuning and prediction processes will be run 20 times. Following figure shows the main procedure of this method.

- Firstly, we train models on a train set. Then we find optimal hyperparameters based on the validation set. Finally, we can predict the return using a test set and save it.
- Now move on to the next window. This time the train, validation and test size's range changed. But the procedure is the same.
- We follow this process 20 times.

Test set started from 1st of January 2003.

Figure: Sliding walk forward approach



Note: test data started from 2003.01.01

Machine learning models:

Some machine learning models require scaling. Because it affects the model's performance and training time. So I used a min-max scaler for each window.

Also, I used RandomSearchCV with uniform and log-uniform distribution for hyperparameter tuning. Loss function means absolute error. There is only one reason why I chose this method. When we use the gridsearchcv method, we have to determine the hyperparameter scape. But we don't know the correct hyperparameter spaces. But when we use randomizedsearchCV, the hyperparameter will be selected randomly from the given distribution. If a parameter is selected from a log-uniform distribution, it has a high probability of being chosen from first values in range. If a parameter is selected from uniform distribution, it has the same probability of choosing all ranges.

Following table shows us hyperparameter spaces of each machine learning model on regression and classification problems. I used the same hyperparameter for both problems.

Table: Hyperparameter spaces for classification and regression

Models	Hyperparameters	Range	Dimension	Source
KNN	n_neighbors	[1, 20]	integer	Thi-Thu Nguyen(2019)
	p	[1, 3]	integer	
RF	n_estimators	[20. 200]	integer	Thi-Thu Nguyen(2019)
	max_features	[5, 30]	integer	
	max_depth	[1, 6]	integer	
	min_samples_split	[2, 30]	integer	
XGBoost	n_estimators	[20, 200]	integer	Robert Ślepaczuk (2023)
	learning_rate	[0.001, 0.5]	loguniform	
	max_depth	[8, 15]	integer	
	gamma	[0.001, 0.02]	uniform	

Note: uniform dist - all values within a specified range have an equal probability of being selected. loguniform dist - is a distribution where the logarithm of the values is uniformly distributed.

Table: Hyperparameter spaces for classification and regression

Models	Hyperparameters	Range	Dimension	Source
ANN	no.hidden layer	[1, 3]	integer	Hyperparameter: - *based on multiple article* - *small range of hyperparameter space* Architecture: - *based on following online article https://www.kdnuggets.com/2019/11/designing-neural-networ*
	no.neurons	[5, 40]	integer	
	activation function	ReLU/sigmoid	fixed	
	dropout rate	0.2	fixed	
	optimizer	SGD / RMSProp	multivalue	
	learning rate	[0.001, 0.05]	loguniform	
	momentum	[0.1, 0.4]	uniform	
	batch size	32 / 64 / 128	multivalue	
	epoch	10 / 20 / 30	multivalue	

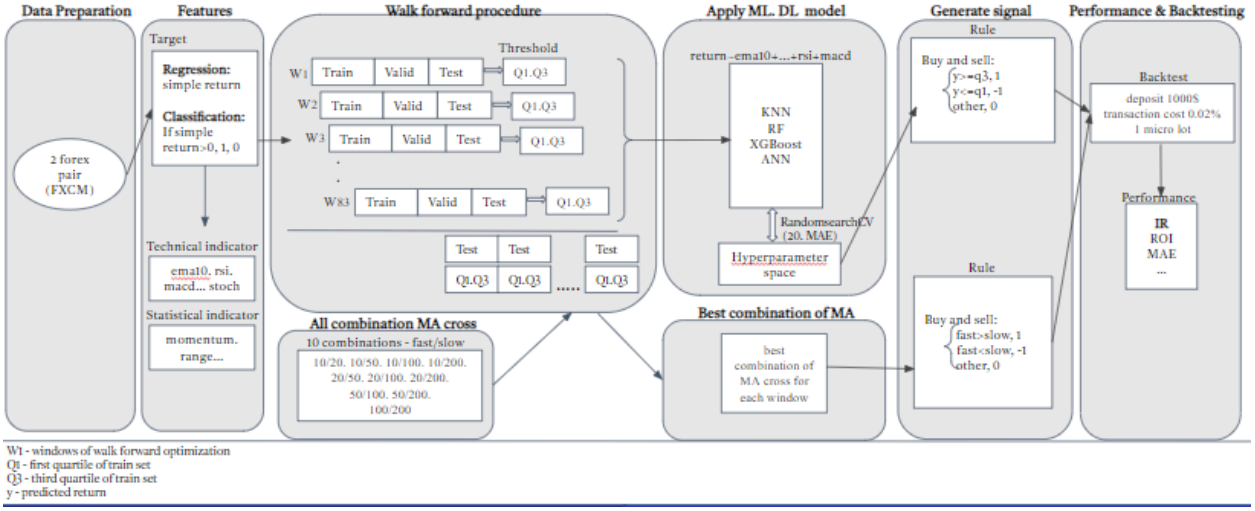
Note: **keras** with loss = 'mae' and metrics = 'mean absolute error'. SGD - Stochastic Gradient Descent. RMSProp - Root Mean Square Propagation. Stacked LSTM and stacked GRU. The stacked LSTM is an extension to this model that has multiple hidden LSTM layers where each layer contains multiple memory cells

Architecture of model:

Following figure shows the architecture of this project. In other words, we can see each step from the following figure. I would like to highlight the generating rule of each trading strategy.

- For a regression problem, if the predicted return is higher than the 3rd quartile of the train set, it will give us a buy signal. If predicted return is lower than 1st quartile of train set, it will give us sell signal

Figure: Architecture of thesis



Performance metrics:

We have to use financial performance metrics for comparing trading strategies. Following table shows us financial performance metrics that were used in this project. I mainly focused on Information Ratio metrics. IR is the ratio between mean return and standard deviation of return. Higher IR is better performance.

Category	Metrics	Formula	Criteria
Investing Metric	Sharpe Ratio (SR)	$SR = (R_p - R_f)/\sigma_p$	**higher** is better
	Information Ratio (IR)	$IR = (R_p - R_b)/\sigma_e$	**higher** is better
	Max Drawdown (MD)	$MD = (Peak - Through)/Peak$	**lower** is better
Additional metric	Rate on Investment (ROI)	$ROI = Net\ Profit\ /\ Initial\ Investment\ *\ 100$	**higher** is better
	Profit Factor (PF)	$PF = Gross\ Profit/Gross\ Loss$	**higher** is better
	Win Loss Ratio (WLR)	$WLR = Number\ of\ winning\ trades/Number\ of\ losing\ trades$	**higher** is better
	Winning Percentage (WP)	$WP = Number\ of\ winning\ trade/Total\ Number\ of\ Trades\ *\ 100$	**higher** is better
Regression metric	Mean Absolute error (MAE)	$MAE = (\sum\|Actual - Predicted\|)/Number\ of\ instances$	**lower** is better

4. Empirical result

Following table shows us selected optimal hyperparameters of each model on regression and classification problems. It takes 40 minutes to train following 4 machine learning models.

Model	Hyperparameter	Regression	Classification
KNN	n_neighbors	[3, 5]	[8,14]
	p	1	3
RF	n_estimators	[60, 160]	[25, 90]
	max_features	[16, 22]	[7, 16]
	max_depth	[4, 5]	[1,3]
	min_samp_split	]12, 20]	[3, 14]
XGBOOST	gamma	]0.001, 0.003]	[0.001, 0.009]
	learning rate	]0.005, 0.2]	[0, 0.1]
	max_depth	[11, 13]	[11, 13]
	n_estimators	[95, 155]	[20, 175]

Model	Hyperparameter	Regression	Classification
ANN	no.hidden layer	1, 2	1, 2
	no.neurons	[10 15]	[10, 35]
	optimizer	sgd/rmsp	sgd/rmsp
	learning rate	]0.001, 0.02]	]0, 0.01]
	momentum	[0.3, 0.5]	[0.1, 0.3]
	batch size	64	64
	epoch	30	30
	activation	ReLU / sigmoid	ReLU / sigmoid
	dropout rate	0.2	0.2

When we compare trading strategies, we need a backtesting process. We can see the backtest's assumption from the next part.

- Initial deposit is 1000$
- Always trade by 1 micro lot
- Pip value is 0.1$ per pip
- Transactional cost is constant 0.02%
- No leverage and no stop loss

Following 2 tables show us the result of trading strategies on regression and classification problems. Each strategy indicates gross profit, number of trade, percentage of buy trading, transactional cost, information ratio, return on investment and mean absolute error(accuracy).

IR value with red color indicates best 3 strategy.

- For regression problems, benchmark strategy has the highest IR. This strategy earned 14 thousand dollars during the testing period.
- For classification problems, benchmark strategy has the highest IR. This strategy earned 17 thousand dollars during the testing period.

Strategy	Model	Gross profit	Num Trade	Buy %	Win trade	Trans Cost	IR (10^-3)	ROI %	MAE (10^-3)
ML	KNN	4380	1077	50.1%	502	267	**-12.4**	-52%	2.8
	RF	6923	1873	50.5%	898	464.5	**-12.5**	-58%	0.7
	XGBOOST	4044	1195	51.9%	558	298.4	**-18.0**	-61%	1.8
	ANN	11813	980	47.9%	469	242.3	**2.7**	-3%	19.7
Trend following	MAs	14156	68	50.0%	32	17.4	**12.9**	36%	-
Benchmark	B&H	14158	2	50.0%	0	0.42	**13.2**	36%	-

Strategy	Model	Gross profit	Num trade	Buy %	Win trade	Trans Cost	IR (10^-3)	ROI %	Accuracy
ML	KNN	17444	1787	50.0%	925	557	**7.6**	16%	0.79
	RF	17245	2575	50.0%	1306	803.7	**1.7**	-24%	0.97
	XGBOOST	17210	2588	50.0%	1310	807.2	**-0.1**	-33%	0.96
	ANN	16944	1473	50.0%	739	461.2	**-18.1**	-85%	0.68
Trend following	MAs	17490	71	50.7%	43	21.3	**9.8**	26%	-
Benchmark	B&H	17558	2	50.0%	0	0.6	**12.3**	40%	-

5. Conclusion

I focused on creating algorithm trading strategies using 4 different machine learning models on chosen 2 forex pairs. Also, I considered rolling walk window approach and randomizedsearchCV with 2 different distributions. These methods improved models accuracy. Another purpose was comparing performance of different trading strategies. In this part, I will verify my hypothesis and research question using results from empirical analysis.

- I can accept the first hypothesis. Because if we choose the right model, we can make profit using technical indicators. Final 2 tables showed it.
- I reject the second hypothesis. Because ML trading strategy's highest IR is lower than Bechmark's IR.
- I reject the third hypothesis. Because ML trading strategy's highest IR is lower than Trend Following strategy's IR.
- For the regression problem, ANN strategy generated trading signals with highest IR. for classification problem, KNN strategy generated trading signals with highest IR.