Algorithmic Trading using Machine Learning for Forex CFDs and its Deployment.

Kapil Parekh
200941505
Dr. Luk Arnaut
M.Sc. Big Data Science

***Abstract*— The rise of expert advisors in the market has significantly transformed the trading environment. The market's efficiency has risen as a result of its widespread acceptance. Foreign currency markets, in particular, have grown very efficient, leaving little to no room for conventional investors to stand in the game. The price of a financial instrument adapts quickly in response to fresh public information. With the introduction of Technical Indicators, which give a variety of use cases such as anticipating the trend in the market as well as the strength of the trend, its adoption has grown, and many algorithmic traders have begun to employ it. Because of the high number of transactions that might result from the use of technical indicators, using them alone is insufficient and unproductive. To make better use of it, more and more models based on machine learning are being developed, including technical indications as features. I will try back-testing machine learning-based models to anticipate instrument price trends and deploy them using an API offered by one of the interactive brokers.**

Keywords*— Trend Prediction, Support vector machine, Multilayer perceptron, K-Nearest Neighbors, Forex, CFD.

I. INTRODUCTION

To study the efficiency of the market we have to understand the Efficient Market Hypothesis (EMH). It states that it is impossible for traders to generate an alpha which is the excess return with respect to the benchmark index for an instrument if the market shows strong form of efficiency. All markets show different form of efficiency [2]. The difficulty of generating an alpha increases with the increase in market efficiency. Technical analysis as a concept has given rise to many technical indicators such as Bollinger Bands, Relative Strength Index, Moving Average Convergence Divergence etc. Some technical indicators show frequent signals which are less efficient and some show less frequent signals but are comparatively more efficient. If a collection of technical indicators that provide more frequent signals is picked, the model becomes biased toward the more frequent ones. As a result, there should be a tradeoff between the Indicator choices. The definition of a good model is determined by an investor's risk aversion. To limit the risk of an investment, the timing of the transaction, the choice of instruments, and the technique for initiating a position all play an essential role [1], [8]. Along with this, to have an edge in the market, dynamic target profit and stop losses should be used which changes based on volatility of the market which is absent in many of the algorithmic trading strategies [3], [7]. In the world of automation machines play a huge part in assisting and optimizing the decision making process of humans. Machine learning has become indissoluble in today's world and we are using it in our routine life even without knowing it. Similarly in financial markets, machine learning algorithms are used because they maneuver through data and estimate the future market picture. They are incredibly useful in optimizing the human decision making process [11], [12], [14]

The immense complexity of trading, along with the growing relevance of algorithmic trading, have resulted in a massive Quant Industry populated by quantitative analysts, developers, and data scientists. A quant is a professional who uses statistics and mathematical principles in the realm of finance and risk management, and they are in great demand in various hedge funds, investment banks, and other institutions.

II. SUPPORT VECTOR MACHINE (SVM)

Support vector machine is used for multiclass classification as well as regression and is a supervised machine learning algorithm. In a linear binary classifier, a plane is defined which separates positive and negative outcomes. Similarly in SVM, data is separated using a hyperplane. If there are N dimensions, hyperplane will be an N-1 dimension subspace.

A margin is an optimal line created by maximizing distance between the nearest points of the classes. Since the goal is to maximize the margin, SVM with linear kernel is also known as maximum margin classifier. The margin line only depends on the support vectors which are nearest data points in either classes and not on every point.

Because the disadvantage of margin lines is that they can only be utilized optimally for linearly separable data. To overcome this, I have used a high order polynomial to increase the feature space. This can be accomplished by employing a function kernel.

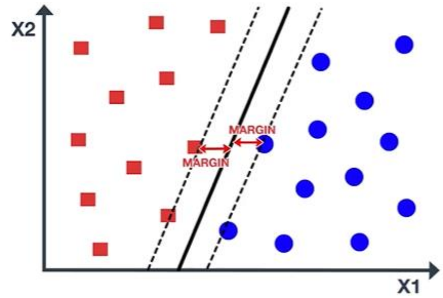


Fig. 1. Margin in SVM

I. RBF kernal

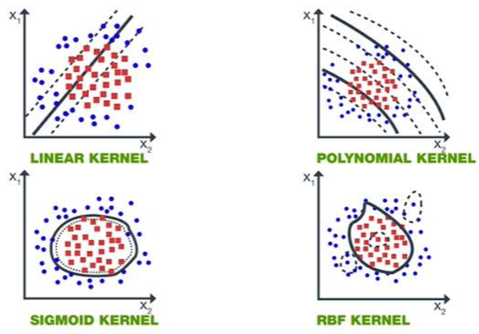


Fig. 2. Visualization of different kernels

There are four types of kernels in SVM out of which I have used the RBF kernel for my model so let's dive into the parameters used in this kernel, which are c and gamma. The c parameter is the tradeoff between the correct classification of data and smoothening of decision boundary. The number of support vectors increases with value of c. If the value is too large the model will over fit. While gamma parameter lets us control how far the influence of a single training example reaches.

III. K-NEAREST NEIGHBOR AND MULTILAYER PERCEPTRON

K-Nearest Neighbor or kNN algorithm is used widely for various classification and regression problems [5], [4]. According to kNN the nearest k points to the new point are taken and the class of the new point is assigned same as that of majority of k neighbors. In order to compute the nearest neighbors, algorithms such as ball_tree, kd_tree and brute force search is used.

In finance, neural network applications aren't well defined [6]. Unlike many other data, finance related data are widely available. But with its availability it shows a lot of inconsistency due to the congenital nature of big data which makes it unusable without efficient cleaning. Traditional methods used such as forward fill and backward fill for dealing with missing data doesn't work with data of high granularity. To counter the trivial procedures, techniques such as interpolation is used which contains linear, multivariate and spline interpolation. Since the data is still uncertain, limitations on financial data analysis with traditional statistical methods are raised. To overcome these limitations, highly advanced applications of machine learning which Neural Networks are. One such feedforward artificial neural network is the Multilayer Perceptron (MLP), which from collection of input points generates a set of output. The layers inside the multilayer perceptron is a directed graph wherein several input and output layers are connected with each other. With its efficiency in the financial research, neural networks, even though not easily interpretable received a lot of importance [6].

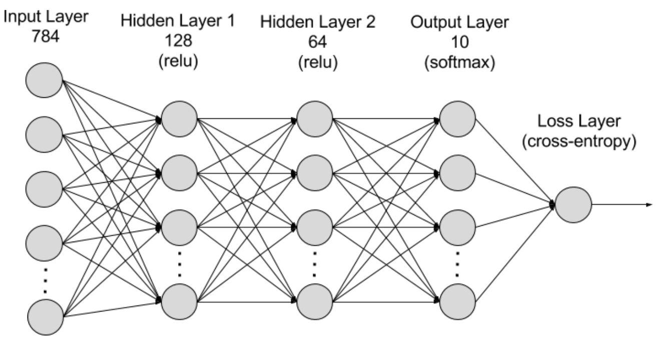


Fig. 3. Multilayer perceptron

As the name itself suggests, perceptron are used to observe and infer like humans do but much faster. In multilayer perceptron there are at least 3 layers with minimum 1 input layer 1 hidden layer and 1 output layer wherein each node in a layer is connected with a certain weight to each node of the subsequent layer. For the process of learning, with each iteration that is with entry of each data points the weights with which the nodes are connected, changes. To decide whether to activate a neuron, an activation function is used. One such activation function which is used in deep learning is rectifier linear unit (ReLU) $f(x) = x^+ = \max(0, x)$ which is just a max function between the variable x and 0. There are other variants of ReLU such as

Leaky ReLU: $f(x) = \begin{cases} x & If\ x > 0 \\ 0.01x & else \end{cases}$

Parametric ReLU: $f(x) = \max(ax, x)$

Gaussian Error Linear Unit: $f(x) = x * \Phi(x)$

All these variants are also used as activation functions among many more.

IV. RELATED WORKS

A. *Trading using only technical indicators*

In this trading method two indicators at a time were grouped together and a position was taken based on the signals provided by these Indicators. The Indicators used were Moving Average Convergence Divergence (MACD), Relative strength index (RSI), Average Directional Index, Renko Charts, Average True Range and On Balance Volume.

For trading, API from an Interactive broker was used to create trades, find current profit and balance among other details. The results showed returns in Indian stock market [3]. From the given results and the time format on which the trading was done it was understood that returns was the only performance measure considered which isn't sufficient. One might get positive returns depending on the amount of risk taken. If the risk taken is high we can get a positive return for a short period of time but the factors such as Sharpe ratio which provides information about the returns obtained on a unit of risk taken. Similarly there is alpha, Beta, Maximum Drawdown which should be considered. Positive returns using technical indicators alone may be possible if the market shows a weak form of market efficiency such as the Indian stock market [2].

B. *SVM based model for predicting foreign currency market rates*

In the model created various moving averages were taken as a group of technical Indicators as well as the features of the model along with currency pair's current close price. The model was evaluated based on Normalized mean square error, Mean absolute error, Directional symmetry and correct up/down trend.

Based on the deviation of actual and forecasted values NMSE and MAE were calculated. It can be seen that the NSME was quite high for one currency and others were within acceptable range. It was observed that the selection of kernel while training the SVM model played an important part but not a single kernel dominated results on all currencies. For forecasting the correct up and down trend the linear kernel was incapable of providing the correct result. Polynomial and RBF kernels were the dominating ones in terms of correct trend prediction [9].

Major takeaways from this approach is that each currency shows different performance with change in kernels and no kernel dominates with all currencies. To overcome this we can select a group of currencies to trade with, which performs well with the RBF kernel since it performed well among others.

Since only moving averages were used as features it questions the reliability of this model at the current market situation. Also since all currencies were paired with Australian dollars (AUD) the parameters needn't be standardized. But for trading different currency pairs, and using different technical indicators the features need to be standardized using the standard scalar.

C. *Exchange Rate Prediction using Multilayer Perceptron*

In this paper, gold prices as external factor was considered. The fact that forex market is one of the biggest market is highlighted [10]. Three currency pairs namely GBP/USD, EUR/USD and USD/JPY were scrutinized. Gold prices were made available using yahoo finance. Weekly missing data when the market doesn't trade were filled using moving averages which again highlights the fact that ffill and bfill methods aren't efficient enough. Training dataset consisted of 2395 data points and testing dataset had 355 data points.

For evaluating the model, mean square error (MSE), NMSE are used. Using gold as an external factor which was an input in the MLP model, improved the forecast ability which was again proved by the performance measures.

In this paper RBF neural network model was also tested and possibility of creating a hybrid model was highlighted which was a part of future scope.

This method of creating hybrid model is also known as ensemble learning which is an approach in which two or more models are combined to predict the outcome usually to seek better performance.

D. *Stock Price Model using RBF-SVM Algorithm*

Again for downloading data, yahoo finance was used in this paper. C, gamma and n_epoch was set to 0.7,1 and 30 respectively. While building the model, factors such as learning time was considered. Feature values were standardized. The drawbacks of this approach is again no other features except the ohlc data as well as adjusted close are taken [11].

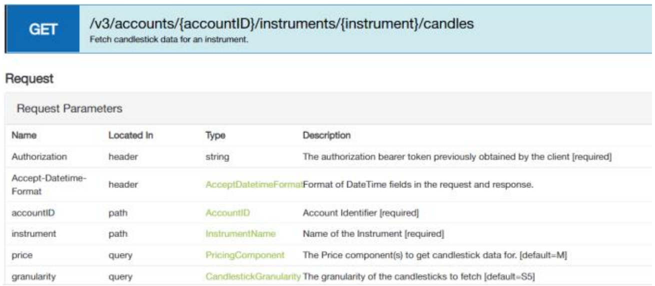
V. WORKING WITH API

In order to get reliable data we shall use Oanda API. Oanda is a renowned interactive broker who provides low spreads, 50:1 leverage for practice accounts which can be set to 500:1 depending on the country a person is trading in. For United Kingdom, the leverage provided in practice account by default is 50:1. Oanda provides REST API or Restful API. We shall be requesting the server using HTTP requests and shall parse the response using JSON.

For using the API service provided by the broker, there is a need to generate a token which will be used for authentication along with account ID. For fetching historical data and streaming real-time data, two different base URL's has to be used. The access token and account id are required parameters under header argument and other parameters such as start date, end date, instrument list etc. are passed inside the params argument while sending a GET request to the server. The response is then parsed using JSON which will return the data in form of a dictionary.

Alternatively one can use a proprietary "tpqoa" package in python which provides limited functionalities but is easier to use. In this approach, I have used both the wrapper package as well as GET request method to fetch the data depending on the case for example to get the data of past 5000 candles till the current price, GET request is used and to get historical data for a particular period of time, the wrapper module is used. While streaming the data using the "pricing" endpoint, it was observed that it just printed the tick price and didn't return anything. Since we wanted to convert tick data to Open, High, Low, Close value data frame, I used the concept of Inheritance by super method in python to access the methods of parent class which returns tick data and resampled the data to convert it into the required data frame and returned it.

It was observed that due to frequent server outages, empty responses were being sent which lead to JSONDecoder error which couldn't be handled with just one try, except block hence the blocks has to be looped until a response is fetched. This is very important to not lose the control of our trades in case of network or server outages while deploying the model. We have to make sure that the model is prone to outages and can continue to work where it stopped. The list of possible HTTP errors are listed in the API end point documentation which can be used as a reference. An example of the format in which request has to be made for fetching candlestick data is as follows.


GET /v3/accounts/{accountID}/instruments/{instrument}/candles
Fetch candlestick data for an instrument.

Request

Request Parameters

Name	Located In	Type	Description
Authorization	header	string	The authorization bearer token previously obtained by the client [required]
Accept-Datetime-Format	header	AcceptDatetimeFormat	Format of DateTime fields in the request and response.
accountID	path	AccountID	Account Identifier [required]
instrument	path	InstrumentName	Name of the Instrument [required]
price	query	PricingComponent	The Price component(s) to get candlestick data for. [default=M]
granularity	query	CandlestickGranularity	The granularity of the candlesticks to fetch [default=S5]

Fig. 4. API documentation

VI. UNDERSTANDING OUR DATASET

A. *Understanding Forex CFD data*

CFD or contract for differences are one of the financial derivatives wherein the difference between the opening and closing price of the trades are finally settled without having the instrument from which the contract is made delivered to you. Similarly in forex there are contracts to buy and sell currency pairs where buying the contract essentially means to buy the first currency and sell the second from the given pair.

Price movements are the price of one currency with respect to the base or the second currency. All prices are given in the form of the "Bid" and "Ask" prices and difference between the bid and ask price is called the spread which is measured in pips. This spread is a cost to the investor and it depends on the current market liquidity and broker's commission. "Pip" is the smallest increment in a forex currency pair. Taking an example of a currency pair with bid price 1.36279 and ask price 1.36285, the spread is of 0.6 pips.

Spread near to market closing or next day starting are very high. This happens around 5 PM EST. The forex market is a 24-hour market which closes at 5 PM EST on Friday and opens at 5 PM EST on Sunday. The trader trades in terms of lots, the smallest lot size is 0.01 which means to buy/sell a contract with 1000 units of underlying instrument.

B. Feature Addition

Any trading algorithm before deployment requires backtesting. For backtesting our model we shall use data with 10 minute granularity. The reason for using 10 minute granularity is because the default granularities are 1 minute, 5 minute, 15 minutes and so on. Majority of trades arising in the market are based on calculations performed on these granularities [13]. So as to have a slight edge over the market I am using a different granularity. For deploying the model, all tick data are appended to the last row and then resampled to not lose the price for each tick and avoid slippage arising due to latency related issues. It is advised to use technical analysis on a short term view for creating a particular position in the market than having a long term view since, as the time passes after the position is created, the reliability of technical analysis decreases due to addition of many fundamental factors related to an instrument and thus the prices rapidly adjusts to it and the role of technical analysis is minimized.

Along with Open, High, Low, Close (OHLC) data, I have calculated the cumulative maximum and minimum values also known as day high and day low value of the prices since the day started till the last tick received. For example if 30 minutes have passed since the day started the value of day's high will be the maximum value of the instrument in 30 minutes. Similarly till the end of the day, the day high and day low values are computed and assigned to the data frame columns.

1. Relative Strength Index

Many believe that the most essential part of technical analysis is using technical indicators. But in reality the reason why technical analysis has gained importance is because of the math behind it. One such technical indicator used in technical analysis is the RSI indicator. RSI is a momentum based indicator which identifies the overbought and oversold situation of an instrument. Since this indicator is an oscillator, its values range from 0-100. The periods wherein there is no price gain are counted as 0 in average gain computation and similarly if there is price gain it will be 0 in average loss computation. It is counted for 13 past periods and current gain and losses are added to the formula. It is not necessary that we sell when the instrument is overbought and buy when it is oversold. The choice remains of the strategy of an investor. We can consider the values given by RSI as strength of the predicted momentum. This indicator will be one of the feature of our dataset and hence its values are computed

$$RSI = 100 - \left(\frac{100}{1+\frac{(Previous\ Average\ Gain*(period-1)+current\ gain)}{Previous\ Average\ Loss*(period-1)+current\ loss)}} \right)$$

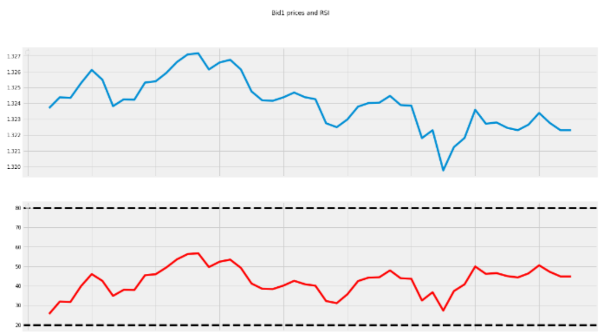


Fig. 5. RSI

2. Average Directional Index (ADX)

Like RSI, ADX is also a momentum based indicator this indicator along with the strength, it also gives the direction of momentum. This indicator is made from 3 parts namely the positive direction line +DI, negative direction line –DI and the ADX strength which gives the strength of the signal.

Fig. 6. ADX Indicator signal

Note that the indicator's signals aren't always reliable especially for an efficient market since this information is available to everyone and we can't generate an alpha based on publically available information in an efficient market. To prove this, I have created an algorithmic trading strategy based on just the ADX indicator with MQL4 language which is built on C++ and since it is a low level language, the backtesting of the strategy is very fast.

The logic behind our algorithm is to trade when +DI< -DI with shift =1 i.e., previous candle price and +DI> -DI for shift=0 and "ADX Main" which is an oscillator which gives strength of the direction set to >25.

As we can observe, that for a period of 4 months, the profitable trades were 48.89% and loss trades are 51.11% with net loss of 506.88 GBP on 10000 GBP initial deposit with a maximum drawdown of 5.07%. With this we can understand that in Forex CFD market, trading with technical indicators alone may be unprofitable. This doesn't mean that technical analysis as a concept is flawed. Rather we can say that the complexity of using it is increased and to understand the use of it more efficiently and to find out inferences which humans alone can't figure out, we use machine learning and so for our model I have input 3 values provided by this indicator for each data point to our dataset by adding 3 more columns.

TABLE I. ADX Backtesting

General Report	Values	Specific Report	Values
Initial deposit	10000	Gross profit	888.96
Net profit	-506.88	Gross Loss	-1395.84
Profit factor	0.64	Profit trades	48.89%
Total trades	180	Loss trades	51.11%
		Consecutive wins	2
		Consecutive losses	2

3. Moving Average Convergence Divergence and Bollinger Bands.

The visualizations given below help us understand these indicators in a better way.

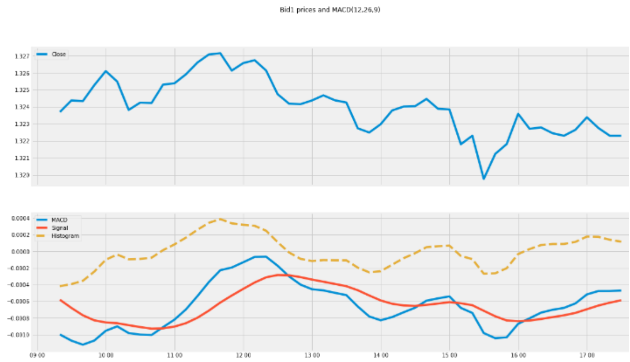


Fig. 7. MACD and close price

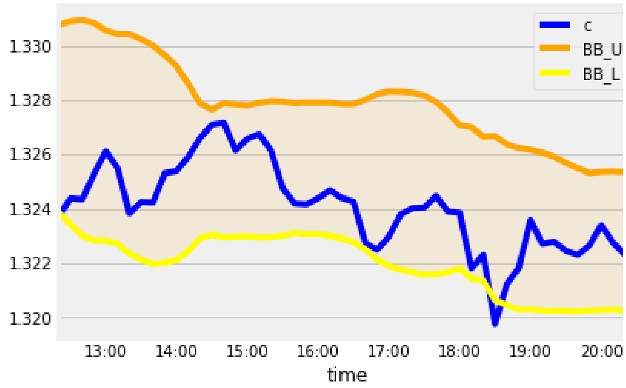


Fig. 8. Bollinger Bands

MACD converts indicators which are used to follow trend into a momentum oscillator. MACD is calculated by the subtracting two Exponential moving averages of length 26 and 12 which are called as the long term EMA and short term EMA respectively. To interpret the direction of the momentum, we calculate EMA of length 9 on the calculated MACD line and name it as signal line. The difference between the MACD line and signal line is used to calculate MACD histogram. Bollinger bands are volatility based indicators used to plot 2 times standard deviation of a simple moving average on close price. The simple moving average used is for a period of 20 and standard deviation used is 2 which can be arbitrarily set. Since forex markets are believed to be very volatile hence I have set the standard deviation to 2.3. To examine the efficiency of MACD and Bollinger bands together in predicting the trends correctly below is the backtesting results of the sell biased trading strategy created using only this pair of indicators. The logic used behind this approach is that if Current high price is above the upper Bollinger band and MACD (shift 1) > Signal (shift 1) and MACD (shift 0) < Signal (shift 0) which means the MACD line has just crossed the Signal line downwards i.e. diverged. We can understand from the results below that since this Strategy was tested for 8 months on USD/CHF total net Profit is positive with only 36 trades. With 77.78% trades in profit and 22.22% trades in loss. Note that this strategy is just tested on a single currency pair so it may only use around 150 GBP for initiating a trade even though 10000 is deposited. From these performance measures I can understand that this is a fairly good strategy with rare signals but can be deployed concurrently on multiple currency pairs to increase the amount of signals but the chances of loss will increase. I have added the values of these indicators to the dataset of each currency pairs.

TABLE II. MACD and RSI backtesting

General Report	**Values**	**Specific Report**	**Values**
Initial deposit	10000	Gross profit	289.96
Net profit	49.70	Gross Loss	-240.26
Profit factor	1.21	Profit trades	77.78%
Absolute drawdown	61.14	Loss trades	22.22%
Total trades	36	Consecutive wins	4
		Consecutive losses	1

Other features added to our dataset will be 2 moving averages namely simple moving average with period 10 and 30 and the signals generated using these indicators after the signal generation process below.

C. *Signal Generation*

In the signal generation process, the signals for each technical indicators will take values only when certain conditions are met with respect to the values of the technical indicators. If the conditions are not met the signal column for that particular indicator will take value 0. Given below are the conditions for each technical indicators. Note that the technical indicator's signal doesn't decide the actual signal of the trade and it will only indicate the signal that the particular technical indicator is favoring. Value "1" represents a buy signal and value "-1" represents a sell signal
While "0" represents no signal.

TABLE III. Signal condition

Features	**Signal Condition**
Bollinger bands (BB)	If Upper BB < close, signal =1 If Lower BB > close, signal = -1 If Upper BB > close <Lower BB, signal=0
ADX Indicator	If ADX>25 and ADX +DI < -DI, signal = -1 If ADX>25 and ADX +DI > -DI, signal = 1 Else signal =0
RSI	If RSI >70, signal = 1 If RSI<30 signal = -1
MACD	If MACD (shift 1) < MACD_signal (shift 1) and MACD (shift 0) > MACD_signal (shift 0), signal = 1 If MACD (shift 1) > MACD_signal (shift 1) and MACD (shift 0) < MACD_signal (shift 0), signal = -1 Else signal = 0
Simple moving averages	If SMA_10 (shift 1) < SMA_30 (shift 1) and SMA_10 (shift 0) > SMA_30 (shift 0), signal = 1 If SMA_10 (shift 1) > SMA_30 (shift 1) and SMA_10 (shift 0) < SMA_30 (shift 0), signal = -1 Else signal = 0

VII. TRAINING DATASET

A training dataset is used to train the model. Based on the values given in the training dataset, any machine learning model would traverse through the dataset and make their set of inferences. Support vector machines learn from train data in order to classify new data to a particular class by creating a hyperplane, K- nearest neighbors uses the train dataset to create centroids in order to classify the new data to the nearest neighbor etc.

Since I am concerned about obtaining signals to initiate a trade and the values of signals can be buy or sell, the problem in hand is a classification problem. To label financial time series data, there are different methods used depending on the strategy for which the model is built. To label a financial time series data is complex because to label a data point using three classes buy, sell and no position involves ambiguity.

Consider a stock price of 100 to better comprehend it. This stock's price can move up, down, or sideways. It can achieve a price of 120 without falling below 100, and it can reach 120 after falling below 80. In both cases, the price of 120 is reached, but if the trader has set a stop loss of 85 while holding a buy position in the market, the trade will close at 85 rather than 120, and because the trade has incurred a loss with the buy position, the correct label for that data point should've been sell rather than buy. We may conclude from this that the classes of the training dataset are certain to vary for a financial time series data based on a trader's target profit and stop loss. If we labelled each data point in the dataset according to the trader's target profit and stop loss, each data point would be assigned a class, which would be either buy or sell. As a result, a pre-requisite must be established for when to commence a transaction, whether buy or sell, and then input the dataset to the model for training.

While setting a prerequisite on when to trade, we have to keep in mind the importance of technical indicators as well. Technical indicators aren't the standalone factors leading to success but it helps to understand the data just like Meta data which provides more information of data. The technical indicators such as RSI will only be able to tell if the instrument is overbought when it reaches at least the price of 70 and similarly at least the value of 30 in case of oversold. Similarly ADX gives strength of the momentum only when it reaches value greater than 25 and the +DI and –DI lines overlap each other. This doesn't mean that there won't be any trend in the market if certain values of these indicators are met but it would only mean that those indicators won't have any say on that particular trend.

For the approach covered in this paper I've created two training datasets for each currency pairs namely 'GBPUSD', 'CADCHF', 'EURGBP', 'GBPCHF', 'EURNZD', 'SGDCHF', 'AUDSGD', 'USDCHF', 'AUDCAD', 'GBPNZD', 'CADSGD', 'NZDUSD' and 'AUDNZD'. In order to adjust with server time and daylight saving, we shall convert the time column to UTC+3:00 since this is the time zone in which the day start and day end is decided in forex trading.

For the first training dataset, the prerequisite for trading is set such that a position will be created only when at least 3 indicators show a signal. This signal provided by individual indicators can be buy or sell and the signal values of these indicators needn't be the same. The target profit and stop loss set for making the training dataset are dynamic based on the current volatility for that particular instrument in the market. To measure the volatility I am using the difference between the day high and day low value at a given point in time which acts as an input to the profit and max loss functions which returns the value of target profit and stop loss as output. The calculation of profit and loss in forex depends on the base currency used. For example if I buy EURNZD, the formula would be, the difference in the trade close price and trade open price multiplied by number of units purchased which in our case is 3000 and the calculation is divided by the current price of GBPNZD.
To understand the distribution of data and figure out whether the indicator signals coincide with actual signal I have plotted a stack bar plot for each technical indicator.

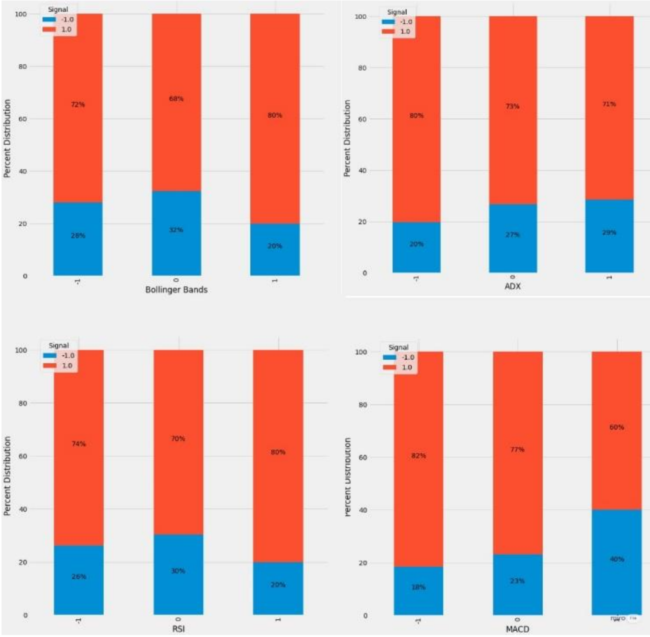


Fig. 9. Stack plot for training dataset 1

In this strategy of labeling the actual class for the dataset I have used a big stop loss as compared to the target profit but within the justifyable range and because of this the indicator's didn't coincide with the actual signal. This method might show good accuracy but will still fail to generate overall profits because a single loss trade would walk over multiple profitable trades. Also this method would neglect the importance of features in our model and would be biased to the target profit and stop loss. If the values of the features doesn't relate with the actual label, the machine learning model wouldn't be able to make good inferences and will result in a flawed model. Many traders prefer a big stop loss over target profit but so as to not set a bias over the choice of strategy we shall also test another strategy with target profit greater than the stop loss. So even in case of accuracies around 50% we will still be able to generate an alpha.

For the second training dataset I have set the prerequisites of trades with 3 minimum signals and cummulative value of the signal greater than 1 but with a big target profit and a small stop loss. Let us now identify if the actual signals coincide with that of the Indicator signals.

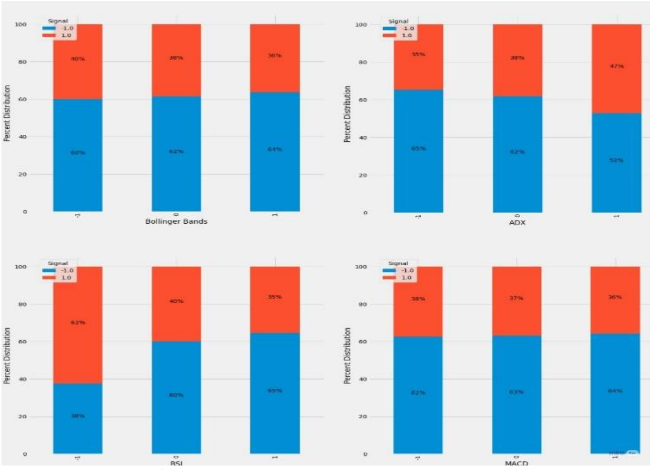


Fig. 10. Stack plot for training dataset 2

From the above figure it is clear that the actual values coincide with the feature indications and importance of features is retained. Now in the training process the model will make reliable inferences and predict the classes effeciently.

VIII. METHODOLOGY

A. Hyperparameter tuning

Adjustable parameters, often known as hyperparameters, are used in machine learning algorithms. Setting the hyperparameters values before training the model is critical for developing powerful, accurate, and efficient models. Because there is always a tradeoff between bias and variance, hyperparameters are employed to aid us in the tradeoff, preventing model underfitting and overfitting.

In case of SVM with rbf kernal, c and gamma are the hyper parameters. These parameters are used to create a decision boundary. Similarly for K-NN the hyperparameters optimised are leaf_size, p, for MLP the hyperparameters optimised are activation function, solver which is used for weight optimization, alpha is the penalty for wrong classification also known as a regularization parameter and learning rate. Following are the results for the set of hyperparameters for 8 out of 13 currency pair dataset for SVM and classification report for 3 currency pairs.

To make the table concise I have renamed the currency pairs as B1, B2 etc. Precision is the percentage of predictions which are correctly classified. Recall is the percentage of positive cases. F1 score is the percentage of positive predictions that were accurate. Since we are using randomizedsearchcv, the values of parameter may change with each iteration.

TABLE IV. Hyperparameter tuning for SVM training dataset 2

	B1	B2	B3	B4	B5	B6	B7	B8
C	100	1000	1000	1000	100	1000	10	10
Kernel	rbf	rbf	rbf	Rbf	rbf	rbf	rbf	Rbf
Gamma	1.0	0.1	1.0	1.0	1.0	0.01	0.01	1.0

TABLE V. Classification report for SVM training dataset 2

Classification Report	**-1.0**	**1.0**	**Accuracy**
Precision B1	0.609756	0.416667	0.566038
Recall B1	0.781250	0.238095	0.566038
f1-score B1	0.684932	0.303030	0.566038
Support B1	32.000000	21.000000	0.566038
Precision B2	0.27777	0.346154	0.318182
Recall B2	0.227273	0.409091	0.318182
f1-score B2	0.2500	0.375000	0.318182
Support B2	22.000000	22.00000	0.318182
Precision B3	0.740742	0.421053	0.608696
Recall B3	0.645161	0.53333	0. 608696
f1-score B3	0.689655	0.470588	0. 608696
Support B3	31.000000	15.000000	0. 608696

Since I have used SVM in the deployment phase, given above were some specific details of the classification report for what was observed while testing each currency pair. Given below is the cumulative classification report for different models including the SVM model for training dataset 1 which I have created with big stop loss. It was observed that for all models the accuracy of predicting trades in the training dataset 1 was greater than 70 but since I was using big stop loss, it ended up losing money even with high accuracy. Given that all trades will run concurrently, I have taken true positives, false negatives, false positives and true negatives for all currency pairs and calculated average scores for other models.

TABLE VI. Hyperparameter tuning for SVM training dataset 1

	B1	B2	B3	B4	B5	B6	B7	B8
C	10	10	10	1000	10	1000	10	10
Kernel	rbf	rbf	rbf	Rbf	rbf	rbf	rbf	Rbf
Gamma	0.01	0.01	0.01	1.0	0.01	1	0.1	0.01

TABLE VII. Classification report for SVM training dataset 1

Classification Report	**-1.0**	**1.0**	**Accuracy**
Precision	0.1259	0.8181	0.793231
Recall	0.074976	0.960133	0.793231
F1-score	0.0872	0.88028	0.793231
support	17	74	0.793231

TABLE VIII. Hyperparameter tuning for KNN training dataset 2

	B1	B2	B3	B4	B5	B6	B7	B8
Leaf_size	38	30	26	7	9	9	20	49
P	1	2	1	1	1	1	1	2
n_neighbors	20	14	14	22	5	1	24	3

TABLE IX. Classification report for KNN training dataset 2

Classification Report	**-1.0**	**1.0**	**Accuracy**
Precision	0.59	0.445	0.5646
Recall	0.7407	0.3148	0. 5646
F1-score	0.646499	0.33345	0. 56465
support	31	22	0. 5646

Let a = (50, 100, 50), b = (50, 50, 50), c = (100,)

TABLE X. Hyperparameter tuning for MLP training dataset 2

	B1	B2	B3	B4	B5
Hidden_layer_size	a	b	b	a	C
Acivation	relu	relu	tanh	tanh	relu
solver	sgd	adam	adam	sgd	adam
Alpha	0.04	0.04	0.04	0.03	0.03
Learning_rate	adapt	adapt	adapt	adapt	const

TABLE XI. Classification report for MLP training dataset 2

Classification Report	**-1.0**	**1.0**	**Accuracy**
Precision	0.6253	0.4958	0.5847
Recall	0.7158	0.3810	0.5847
F1-score	0.6604	0.4069	0. 5847
support	38	26	0. 5847

IX. DEPLOYMENT OF MODEL

For deployment, I have used SVM classification model. Since there are two approaches one being the risk averse approach (Training dataset 2) and other one being a risky approach (Training dataset 1). When the model is executed on Oanda's platform, trades are sent automatically through the REST API which is used to send orders, receive acknowledgement of the order, current unrealized P/L account balance, margin used etc. While deploying the model in Oanda it was observed that there were many server outages. Due to which the model lost control over some trades. Since the functions inside the wrapper package are made by third-party, it didn't handle exception well so most of the execution was done using direct request response method by referring to the documentations for various endpoints given by the broker itself.

Since the trading at broker's end wasn't reliable so I continued with the model with training dataset 2 which was deployed locally but fetched real time ticks. While running the model locally, the tick data was appended to another variable and resampling of the data was done for each tick to convert to 10 minute granularity. At real-time, price of every 10th minute data was appended to the main data frame after that the values of the features were calculated. These features are used to generate feature restricted signals. Since the prerequisite for 2nd trading strategy was to initiate a position only when the summation of all the signal values is greater than 1 and at least 3 indicators show either 1 or -1. The choice of prerequisite was done after several permutations, and the best condition was selected. Whenever the data point passes the pre-requisite condition, its values are then passed on to the SVM model. Based on the optimal hyper parameters for each currency pair, the data point is classified to either buy or sell by the SVM model. A dictionary is maintained to store the predicted signals, price of the trade when it was opened, differential count which is nothing but difference between day high and day low values are all stored inside these dictionaries under the instrument name which is the key for accessing underlying values. The values of target profit and stop loss are stored in the same row at which the trade was opened. For each iteration, the dictionary is traversed and based on the differential count as well as the open price of the trade, the profit and loss conditions are evaluated. After the trade is closed, the actual value of the signal is then stored in "signal" column at that particular time on which the trade was closed. Based on the Target profit and Stop loss values, the total profit is calculated.

X. RESULTS AND DISCUSSION

Total 36 trades were executed from a period of 12/11/2021 to 18/11/2021 with the second strategy which even in testing phase showed around 56 % accuracy even when the target profit was set higher than the stop loss. Given below is the classification report.

TABLE XII. Classification report (SVM) training dataset 2

Classification Report	-1.0	1.0	Accuracy
Precision	0.4545	0.64285	0.52777
Recall	0.6666	0.42857	0.52777
F1-score	0.5405	0.5142	0.52777
support	15	21	0.52777

Total P/L obtained: 94.77 GBP with initial investment of 1000 GBP.

	label	Predicted_Signal	Actual_signal	Target_Profit	Stop_Loss	Trade_Open_time	Trade_close_time	P/L
0	gbpusd	-1.0	1.0	5.2356	-3.618999	2021-11-12 11:40:00+03:00	2021-11-13 00:00:00+03:00	-3.618999
1	gbpusd	-1.0	1.0	5.2356	-3.618999	2021-11-15 04:50:00+03:00	2021-11-18 00:00:00+03:00	-3.618999
2	gbpusd	-1.0	-1.0	5.2356	-3.618999	2021-11-15 08:30:00+03:00	2021-11-18 16:20:00+03:00	5.235600
0	cadchf	-1.0	1.0	5.2356	-3.618999	2021-11-15 18:20:00+03:00	2021-11-16 00:00:00+03:00	-3.618999
1	cadchf	-1.0	-1.0	10.7796	-5.004990	2021-11-15 20:30:00+03:00	2021-11-17 19:50:00+03:00	10.779600
0	eurgbp	-1.0	-1.0	5.2356	-3.618999	2021-11-15 05:50:00+03:00	2021-11-15 18:00:00+03:00	5.235600
0	gbpchf	1.0	1.0	10.7796	-5.004990	2021-11-12 09:00:00+03:00	2021-11-13 00:00:00+03:00	10.779600
1	gbpchf	1.0	1.0	10.7796	-5.004990	2021-11-12 11:40:00+03:00	2021-11-16 00:00:00+03:00	10.779600
2	gbpchf	1.0	1.0	10.7796	-5.004990	2021-11-12 11:50:00+03:00	2021-11-17 00:00:00+03:00	10.779600
3	gbpchf	1.0	1.0	5.2356	-3.618999	2021-11-15 08:30:00+03:00	2021-11-18 00:00:00+03:00	5.235600
4	gbpchf	1.0	-1.0	5.2356	-3.618999	2021-11-15 18:10:00+03:00	2021-11-18 15:30:00+03:00	-3.618999
5	gbpchf	1.0	-1.0	5.2356	-3.618999	2021-11-16 04:50:00+03:00	2021-11-19 12:20:00+03:00	-3.618999
0	eurnzd	-1.0	-1.0	10.7796	-5.004990	2021-11-12 04:10:00+03:00	2021-11-12 20:10:00+03:00	10.779600
1	eurnzd	-1.0	-1.0	10.7796	-5.004990	2021-11-12 04:20:00+03:00	2021-11-12 22:50:00+03:00	10.779600

Fig. 10. Snapshot of Result

For the limited time the training dataset 1 strategy was running, the trades were buy skewed, similar to what was found during the testing phase, and hence the model was unable to appropriately identify the transactions. The initial strategy's high accuracy was due to the feature relevance being overlooked. When the stop loss and profit settings were changed from dynamic to static, the accuracies dropped dramatically.

It was only after a thorough evaluation of all aspects that may affect the model, while keeping in mind the tradeoff between bias and variance, that an alpha could be generated. The stacked bar plot was critical in establishing the dependability of technical indicators for a certain model. Despite the fact that the model was trained on 13 currency pairings, there is a need for additional processing power or adaptation to low level languages such as MQL4, which is based on C++ and allows us to train our models on massive amounts of data.

For future scope, since the addition of dynamic target profits and stop losses increased the efficiency of the model, similarly we can build or modify existing technical indicators for example in case of bollinger bands, when price movements are low the distance between upper band and lower band decreases leading to consolidation and prices moving in one direction. To counter this one can adjust the multiplier of standard deviation which was 2 in default case and 2.3 in my case to ever changing with respect to current volume. This will be a new learning opportunity for the model to find inferences using volume and bollinger bands.

ACKNOWLEDGEMENT

Special thanks to Dr. Luk Arnaut for his advice, time and consideration. I am honored to have his assistance and work under his guidance. Thanks to Dr. Jesús Requena for covering such an amazing course on machine learning at the Queen Mary University of London.

REFERENCES

[1] Shen, J., Shafiq, M.O. Short-term stock market price trend prediction using a comprehensive deep learning system. *J Big Data* **7,** 66 (2020).

[2] Ioannis N. Kallianiotis, 2018. "How Efficient is the Foreign Exchange Market?," Athens Journal of Business & Economics, *Athens Institute for Education and Research* , vol. 4(3), pages 293-326, July.

[3] T. Salkar, A. Shinde, N. Tamhankar and N. Bhagat, "Algorithmic Trading using Technical Indicators," 2021 International Conference on Communication information and Computing Technology (ICCICT),2021,pp. 1-6, doi: 10.1109/ICCICT50803.2021.9510135.

[4] Yauheniya Shynkevich, T.M. McGinnity, Sonya A. Coleman, Ammar Belatreche, Yuhua Li, Forecasting price movements using technical indicators: Investigating the impact of varying input window length, Neurocomputing, Volume 264, 2017, Pages 71-88,

[5] Ban, Tao & Zhang, Ruibin & Pang, S. & Sarrafzadeh, Abdolhossein & Inoue, Daisuke. (2013). Referential kNN Regression for Financial Time Series Forecasting. 601-608. 10.1007/978-3-642-42054-2_75.

[6] Ahmed, Riyaz. (2021). NEURAL NETWORKS IN FINANCE: A DESCRIPTIVE SYSTEMATIC REVIEW. Indian Journal of Finance and Banking. 5. 1-27. 10.46281/ijfb.v5i2.997.

[7] G. S. Atsalakis and K. P. Valavanis, "Surveying stock market forecasting techniques – Part II: Soft computing methods," Expert Systems with Applications, vol. 36, no. 3, pp. 5932–5941, Apr. 2009.

[8] Z. Liu, Z. Dang and J. Yu, "Stock Price Prediction Model Based on RBF-SVM Algorithm," 2020 International Conference on Computer Engineering and Intelligent Control (ICCEIC), 2020, pp. 124-127, doi: 10.1109/ICCEIC51584.2020.00032.

[9] Kamruzzaman, Joarder & Sarker, Ruhul & Ahmad, Iftekhar. (2003). SVM based models for predicting foreign currency exchange rates. 557- 560. 10.1109/ICDM.2003.1250976.

[10] Fathian, Mohammad & Kia, Arash. (2012). Exchange rate prediction with multilayer perceptron neural network using gold price as external factor. Management Science Letters. 2. 561-570. 10.5267/j.msl.2011.12.008.

[11] Żbikowski, Kamil. (2014). Using Volume Weighted Support Vector Machines with walk forward testing and feature selection for the purpose of creating stock trading strategy. Expert Systems with Applications. 42. 10.1016/j.eswa.2014.10.001.

[12] K. Khare, O. Darekar, P. Gupta and V. Z. Attar, "Short term stock price prediction using deep learning," *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 2017, pp. 482-486, doi: 10.1109/RTEICT.2017.8256643.

[13] J. Fior and L. Cagliero, "Exploring the Use of Data at Multiple Granularity Levels in Machine Learning-Based Stock Trading," *2020 International Conference on Data Mining Workshops (ICDMW)*, 2020, pp. 333-340, doi: 10.1109/ICDMW51313.2020.00053.

[14] Ryś, Przemysław & Slepaczuk, Robert. (2019). Machine Learning Methods in Algorithmic Trading Strategy Optimization – Design and Time Efficiency. Central European Economic Journal. 5. 206-229. 10.1515/ceej-2018-0021.