Fuzzy Logic Based Backtesting System

Erandi Praboda(✉) and Thushari Silva

Faculty of Information Technology, University of Moratuwa,
Moratuwa, Sri Lanka
erandipraboda@gmail.com, thusharip@uom.lk

Abstract. Identification of favorable trading opportunities is crucial in financial markets as it could bring additional increment in profits. Backtesting is one of the process which consists of analyzing the past price movements and predict the best possible trading strategy. Current approaches focus only on exact matching of market values. In order to overcome deficiencies in current approaches, this research proposes a fuzzy logic based approximate matching approach by using the technical indicators and trading rules. The evaluation results demonstrate that finding approximate matching places for a particular trading strategy has a positive contribution to successful trading and an average trader can be successful in trading buy following those fuzzy logic based trading strategies.

Keywords: Backtesting · Fuzzy logic · Approximate matching · Exact matching

1 Introduction

In the financial market like the stock market, forex market and crypto currency market variety of technical indicators are used to make trading decisions [1]. A set of trading rules developed on top of the technical indicators is called as the trading strategies in the field of financial market. The process of checking the viability of these sets of rules by applying the on the historical data without risking any actual capital is called as backtesting. Backtesing is a practice that followed by the traders before they go to the real time trading. Traders will adjust their trading strategies based on the backtesting results to the direction where it will return more profit. Early days the backtesting was followed by series of manual calculations, but today with the enhancement in the field of software industry, different types of software have come to alive with the aim of automating this backtesting process. Even though various numbers of backtesting tools are commercially available since they are having some serious limitations, it is critically important to build a high-performance, high-fidelity backtester.

This research integrates the concept of analyzing the past price movement in the financial market with the elements of fuzzy logic in order to build a successful methodology to do the backtesting. Past price movements of the financial market can be analyzed by using technical indicators [1]. Technical indicator is a mathematical formula that computes a series of price based data points which represent a pattern over some period of time and assists traders to make selling and buying decisions [2]. Other

J. Hemanth et al. (Eds.): SLAAI-ICAI 2018, CCIS 890, pp. 15–31, 2019.
https://doi.org/10.1007/978-981-13-9129-3_2

than the basic functionality of the available back testing systems this tool will provide two main features including hybrid matching of trading strategy against the historical data and Maximum execution efficiency. Therefore, these two features which differentiate this Backtesting via hybrid matching module from existing back testing tools.

The rest of the paper is organized as follows: Sect. 2 outlines the review and research of relevant literature, followed by the proposed solution. The implementation is presented in the Sect. 4 and experimental results and analysis presented in Sect. 5. We conclude the paper with a conclusion.

2 Related Works

The process of testing trading strategy based on the historical data to ensure its feasibility before the trader risks any actual capital is known as backtesting [4]. This process is mainly based on the underlying theory that the trading strategy that was successful in the past is likely to be succeed in the future. Early studies of backtesting were mainly focused on testing one indicator at a time. At this edge the researchers use backtesting to determine the accuracy of an indicator when generating the buy and sell signals based on that indicator. After such studies were carried out over a hundred years in 1992 Brock et al. [5] conducted an experiment using 26 variations of moving average and trading range breaks. These two indicators were backtested on the Dow Jones Industrial Average over 100 years from 1887 to 1986 [5]. In 1998, a follow up study carried out by Bessembinder and Chain [6] and determined that when compensating for trading costs, the strategies analyzed by Brock were not significantly more profitable than a buy and hold approach. This implies that although the technical indicators are enriched of some kind predictive ability, they were unable to disprove the weak form of the efficient market hypothesis. Another follow up study conducted by Lo et al. in 2000 was able to implement a computer based model to detect analyzed the chart patterns from 1962 to 1996 on NYSE [7]. Since most of these studies are based on backtesting one indicator at once the researchers paid more attention to investigate whether the combine indicators can be used to evaluate the viability of the trading strategy. As a result of that in 2007 Elaine Loh used a combine model which is consisting of two indicators: moving average and a stochastic oscillator [8]. When this model was backtested on Asia Pacific stock exchanges, the authors could prove that when moving averages were accurate only 50% of the time, the method which used two indicators was accurate 75% of the time [8]. This study shows that combining indicators can eliminate the noise and increase the accuracy and efficiency of trading strategy. Therefore, it is necessary to develop a backtesting tool which provides the facility to backtest combining strategies. Thus, we followed the hybrid approach in the proposed method. Traders always required to backtest their algorithm with different set of parameters and based on different time frames and different time period and different currency pairs. An enhanced back testing tool which has the capability of backtesting a trading strategy with approximate and exact matching is presented below.

3 Proposed Solution

The proposed method uses the fuzzy logic to design new tools to backtest trading strategies. In this proposed solution high performance is achieved by using multi-threads. In order to achieve high accuracy both exact and approximate matching are followed. As illustrated in Fig. 1 Hybrid matching module consists of a Data feeder, Chart Generator and Indicator Calculator which calculate the relevant indicators based on the trading strategy. Both exact matching algorithm and approximate matching algorithm generate the buy and sell signals and as the final step these buy and sell signals are used to calculate the final back testing results. The major components of this proposed system can be further detailed as follows.

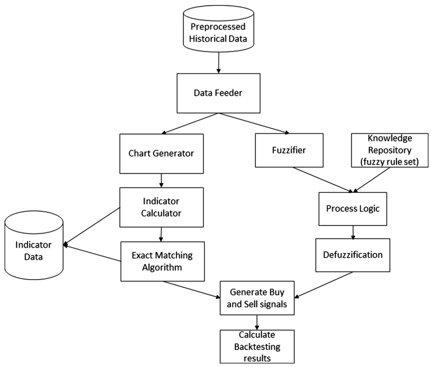

Fig. 1. Overall design of Hybrid Matching Module

3.1 Data Management Module

Data feeder manages access to historical price data from the API through the internet. There is a separate API that has been developed for this research work and it provides necessary preprocessed historical data within a specific time period accommodating clients' request. The developed REST API is capable of sending the relevant price data within a specific date range. Forex historical data consists of Date, Time, Open, High, Low, Close and Volume. This system has an internal cache of historical data which enhances the performance by minimizing the access to the external API. When the data feeder is triggered, it searches in the internal cache first and if the requested data is not available.

3.2 Chart Generating Module

The main objective of the Chart Generating module is plot the historical price data against the date, plot the indicators against the date and time and indicate the sell and buy signals on price data chart.

3.3 Indicator Calculator

This sub module is dedicating for calculating the necessary indicators for a particular trading strategy. It consists of the necessary logic to calculate the technical indicators.

3.4 Exact Matching Module

Under the exact matching module five standard buy and sell trading models are developed based on their relevant technical indicators. The technical indicators used under this module can be listed as follows.

3.4.1 Moving Average Cross Over

This is a simple but powerful trading strategy used by the most of the forex traders. It has been studied extensively with encouraging results, most notably study by Brock et al. in 1992 [5]. Moving average can be calculated by using following formula here, where Price (n) denote the closing price on n^{th} day.

$$\text{Moving Average} = \sum\nolimits_{n-5}^{n} \frac{price(n)}{5} \tag{1}$$

The most commonly used moving average cross over trading strategy has been built from using the short term moving average as well as the long term moving average. If the short term moving average crosses over the long term moving average the trading strategy will generate a buy signal and market is thought to be trending up on the other hand, if the short term moving average crosses below the long term moving average sell signal is generated considering that the uptrend has been replaced by the down trend [11].

3.4.2 Moving Average Convergence/Divergence (MACD)

MACD is a very popular indicator used in technical analysis to measure the momentum in security. In late 1970 this indicator was introduced by Gerald Apple. The MACD denotes the difference between two given moving averages. Upward momentum indicates when the short term moving average is above the long term moving average or when the MACD is positive. On the other hand, when the MACD is negative, the short term moving average is below the long term moving average and suggest a downward momentum [12]. MACD can be constructed by using followings.

$$\text{MACD} = \text{SEMA - LEMA} \tag{2}$$

Where SEMA is the Short term EMA of closing prices and LEMA is the Long term EMA of closing prices (LEMA).

9 day EMA of the MACD line (Signal Line)

$$\text{EMA} = \text{Price (T)} * \text{m} + \text{EMA (T} - 1) * (1 - \text{m}) \tag{3}$$

When T represent day and k is the no of days that the exponential moving average should be calculated. m is computed as m = 2/(k + 1). According to the rules when the

MACD crosses above the signal line buy signal is generated. The sell signal is generated when the MACD crosses below the signal line.

3.4.3 Bollinger Bands

This method is used to compare the relative price level and the volatility over a particular period of time. The volatility of the price level is computed by taking the standard deviation of the security prices. The middle line of the Bollinger Bands is computed by taking the k-period moving average of the price series. Then by calculating xσ distance from the middle line it is possible to obtain the upper line, while obtaining the lower line by taking –xσ distance, where x is a positive constant and σ is the standard deviation computed over a moving window of K periods [3]. When the close price of the currency pair crosses above the signal line it system will indicate a sell signal and while close price crosses below the signal line it will generate a buy signal.

3.4.4 Relative Strength Index (RSI)

RSI is a most popular and useful momentum oscillator developed and introduced by J. Welles Wilder in 1978 [3]. Basically, this indicator compares the magnitudes of the recent gain and recent losses and it converts to a number between 0 to 100. If the RSI is greater than 70 it is considered as a good place to sell a particular currency fair and according to the RSI trading strategy if the RSI value is above 70 it indicate a best place to sell and if it is below 30 then it indicate a buy signal. Following equation are used to calculate RSI value.

$$\mathrm{RSI} = 100 - \frac{100}{1 + \mathrm{RS}} \tag{4}$$

$$\text{Average Gain} = \text{Total Gain}/\mathrm{n} \tag{5}$$

$$\text{Average Loss} = \text{Total Losses}/\mathrm{n} \tag{6}$$

$$\mathrm{RS} = \frac{\text{Average Gain}}{\text{Average Loss}} \tag{7}$$

3.4.5 Stochastic Oscillator

It is a momentum indicator comparing the closing price of a security to the range of its prices over a certain period of time. It can be calculated by using the following formula. Buy and Sell signals are created when the %Y crosses through a three-period moving average, which is called the %C.

$$\%\mathrm{Y} = 100\,(\mathrm{D} - \mathrm{L14})/(\mathrm{H14} - \mathrm{L14}) \tag{8}$$

where D - the most recent closing price, L14 – Low value of the 14 previous trading sessions, H14 – The highest price traded during the 14 period, %Y - Current market rate for a particular currency pair and %C – 3 period moving average of %Y.

3.5 Approximate Matching

The main purpose of this module is to find out the places within historical data where a particular trading strategy has been applied with some slight differences and evaluate the viability of a trading strategy based on those approximate matching places. In order to find out those approximate matching place fuzzy logic has been used. Here the Mamdani fuzzy inference system is followed as the fuzzy inference system [13]. Therefore, this module consists of several sub components: Fuzzification, Knowledge Repository, and Logic Processing Unit. Under this module five fuzzy trading strategies which correspond to the trading strategies mention under Exact Matching Module were developed. Table 1 lists these five fuzzy trading strategies with the relevant parameters.

Table 1. Fuzzy trading strategies

Trading strategy name	Parameters
Fuzzy simple moving average crossover	• Short-term window size
	• Long-term window size
	• Fuzzy Threshold value
Fuzzy MACD	• Short-term window size
	• Long-term window size
	• Signal line window size
	• Fuzzy Threshold value
Fuzzy Bollinger bands strategy	• Lookback period
	• Band standard deviations
	• Fuzzy Threshold value
Fuzzy RSI	• Lookback period
	• Fuzzy Threshold value
Fuzzy stochastic oscillator	• %Y – first (?) Lookback period
	• %C- second (?) Lookback period
	• Fuzzy Threshold value

3.5.1 Fuzzification

The process of transforming real scalar (crisp) values into fuzzy linguistic variable using fuzzy membership function stored in the fuzzy knowledge base is known as fuzzification [14]. In this research following input variables and parameters were used for the fuzzification process under different trading strategy and the input variables of under different trading strategy can be defined as follows. Several input variables has been normalized and scaled to − 100 to 100 [15].

Fuzzy Moving Average Crossover Strategy

$$\text{NMA} = 100 * ((\text{FMA} - \text{SMA})/\text{LMA}) \tag{9}$$

Where NMA- Normalized moving average, FMA- Fast Moving Average or short term moving average,
SMA- Slow Moving Average or long term moving average
Fuzzy MACD strategy

$$\text{fuzzyMACDInput} = 100 * (\text{MACD} - \text{SignalLine}/\text{MACD}) \tag{10}$$

Fuzzy Bollinger Bands Strategy

$$\text{fuzzyInputUpperBand} = 100 * (\text{close} - \text{upperBand}/\text{close}) \tag{11}$$

$$\text{fuzzyInputlowerBand} = 100 * (\text{close} - \text{lowerBand}/\text{close}) \tag{12}$$

Fuzzy RSI Strategy

$$\text{fuzzyRSIOne} = 100 * (\text{RSI} - 70)/\text{RSI} \tag{13}$$

$$\text{fuzzyRSITwo} = 100 * (\text{RSI} - 30)/\text{RSI} \tag{14}$$

Fuzzy Stochastic Oscillator

$$\text{fuzzyStochasticInput} = \%Y - \%C \tag{15}$$

Figure 2 represents the fuzzy membership function which is used in the fuzzification process of Fuzzy Simple Moving Average cross over strategy and Fuzzy MACD strategy. As illustrated in the Fig. 2 triangular membership function has been used, it will be defined based on the fuzzy Threshold values.

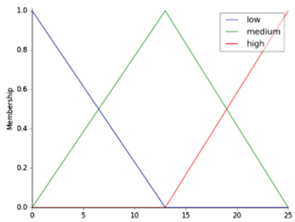

Fig. 2. Fuzzy Membership Function for input variables

3.5.2 Fuzzy Rules

Normally there are three types of decisions that can be used in the process, they are riskless choice, decision making under uncertainty and risky choice. The system is trying to implement decision making under uncertainty. These Decisions are made based on fuzzy rules. Fuzzy rules allow approximate reasoning by including the facts about rules and linguistic variables according to the fuzzy set theory [16]. Usually fuzzy rules are expressed in terms of an IF-THEN statement [17]. The knowledge base used by different trading strategy can be defined as Table 2.

Table 2 Fuzzy rule base

Trading strategy	Fuzzy rules
Fuzzy moving average crossover strategy	IF SMA IS High THEN Signal IS Buy
	IF SMA IS Normal THEN Signal IS Hold
	IF SMA IS Low THEN Signal IS Sell
Fuzzy MACD strategy	IF fuzzyMACDInput IS High THEN Signal IS Buy
	IF fuzzyMACDInput IS Normal THEN Signal IS Hold
	IF fuzzyMACDInput IS Low THEN Signal IS Sell
Fuzzy Bollinger bands strategy	IF fuzzyInputUpperBand IS High THEN Signal IS Buy
	IF fuzzyInputUpperBand Normal THEN Signal IS Hold
	IF fuzzyInputUpperBand IS Low THEN Signal Hold
	IF fuzzyInputlowerBand IS High THEN Signal IS Sell
	IF fuzzyInputlowerBand Normal THEN Signal IS Hold
	IF fuzzyInputlowerBand IS Low THEN Signal Hold
Fuzzy RSI strategy	IF fuzzyRSIOne IS High THEN Signal IS Buy
	IF fuzzyRSIOne Normal THEN Signal IS Hold
	IF fuzzyRSIOne IS Low THEN Signal Hold
	IF fuzzyRSITwo IS High THEN Signal IS Sell
	IF fuzzyRSITwo IS Normal THEN Signal IS Hold
	IF fuzzyRSITwo IS Low THEN Signal Hold
Fuzzy stochastic oscillator	IF fuzzyStochasticInput IS High THEN Signal IS Buy
	IF fuzzyStochasticInput IS Normal THEN Signal IS Hold
	IF fuzzyStochasticInput IS Low THEN Signal IS Sell

3.5.3 Logic Processing Unit

This takes the output of the fuzzification process and the knowledge repository as the inputs. Fuzzified inputs are applied to the antecedents of the fuzzy rules. If a given fuzzy rule has multiple antecedents the fuzzy operator (AND or OR) is used to obtain a single number that represents the results of the antecedent evaluation [18]. The output of the Logic Processing Unit is a signal on a normalized domain on which two different fuzzy sets, BUY, HOLD, and SELL are defined. Figure 3 defines the output membership function used in this system.

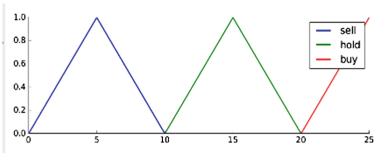

Fig. 3. Fuzzy membership function for output variables.

3.5.4 Defuzzification

The process of mapping the fuzzy space defined over an output universe of discourse into non-fuzzy action. The method used in this work is the centroid of area (COA) [19]. This method provides crisp values based on the center of the gravity of the fuzzy set. Equation 16, defines the formula that is used for the defuzzification when the membership function is discrete [20].

$$F = \frac{\sum_{i}^{K} \mu(Zi)(Zi)}{\sum_{i}^{K} \mu(Zi)} \tag{16}$$

Here the defuzzified value has been denoted as F, Zi indicates the sample element, K represents the number of elements in the sample and $\mu(Zi)$ is the membership function.

3.6 Calculating Backtesting Results

After identifying the Buy, Sell and Hold signals by using the approximate matching and exact matching modules the next step is a calculation of the backtesting results. The main aim of this module is presenting the viability of the trading strategy with some statistical figures like equity curve which can be understood by the user easily. This receive a set of signals and create a series of positions by allocating against the cash component. In order to construct the Mark to market (MTM) portfolio followings are calculated [21].

Positions for every day;

$$\text{Holdings} = \text{Position} * \text{Closing Price} \tag{17}$$

$$\text{Cash} = \text{Initial Capital} - \text{Total Holdings} \tag{18}$$

$$\text{MTM} = \text{Cash} + \text{Positions} * \text{Closing Price} \tag{19}$$

4 Implementation

This section presents the implementation of the proposed solution mentioned in Sect. 4. The rest of the explanation will be elaborated by considering the moving average cross over strategy as the trading strategy, but the actual proposed solution is extended to backtest five standard trading strategies.

4.1 Preliminaries

The following methods are assumed to be declared in advance when deriving the algorithms.

get(url, params): Allow to send an api request to get the data from the server, which is located at the web address provide by the property *url* and allow to pass the relevant parameters by setting the values to the *params* property. A json object containing the relevant price data is returned by the API as a response to this get request. If values has not been set for the property *params* then default values will be set.

rolling_mean(value, window_size): Calculate moving average of value based on the window size.*generate_signals()*: This method is responsible for generating BUY and SELL signals (1and 0) based on the rules of trading strategy as explained under Exact Matching Module. As an example, if the trading strategy is Moving Average Cross Over, Then the implementation of this method will be as follows.

```
if (signals['short_mavg'] > signals['long_mavg']){
signals['signal'] =1.0 }
else{
signals['signal'] =0.0 }
```

cumsum(): Calculate the cumulative sum.

pct_change(): Return the percent change over given number of period. By default the period that used to shift for making percent change is 1.

Antecedent(universeVariable, label): This function will input the variables to the fuzzy control system. There are two parameters one is a one dimensional array and the other one is the name of the universe variable or array.

Consequent(universeVariable, label): This function will output the variables to the fuzzy control system. There are two parameters one is a one dimensional array and the other one is the name of the universe variable or array.

$$f(y, d, e, f) = \begin{cases} 0, & y \le d \\ \frac{y-d}{e-d}, & d \le y \le e \\ \frac{f-y}{f-e}, & e \le y \le f \\ 0, & f \le y \end{cases} \tag{20}$$

The parameter d and f are located the feet of the rectangle and the parameter e is located the peak.

Rule(antecedent, consequent): Defining the rules in a fuzzy control system by connecting the antecedent to consequent

ControlSystem(rules): provide the base class to fuzzy control system. If the rules are provided as a parameter the system is initialized with the given set of fuzzy rules.

*ControlSystemSimulation(controlSystem):*Calculates the results from fuzzy control system. A fuzzy control system object will be passed as a parameter.

compute(): Simulate fuzzy control system by calculating the value of fuzzy output.

4.2 Algorithm for Exact Matching of Moving Average Using Cross Over Strategy

```
url ← “http://api_url"
json_data ←get(url)
file_path ← “E:/Data.csv”
signals ← DataObject.create()
portfolio ← DataObject.create()
signals[‘signal’] ← 0.0
short_window ←50
long_window ←150
initial_capital ← 100000
price_data_csv ← convert_to_csv(json_data)
price_data ← read_csv(file_path)
signals[‘short_mavg’] ← rolling_mean(price_data[‘Close’], short_window)
signals[‘long_mavg’] ← rolling_mean(price_data[‘Close’], long_window)
signals['signal'] ←generate_signals()
signals['possitions'] ←generate_diff()
portfolio[‘holdings’]←multiply(signals['possitions'],price_data[‘Close’])
portfolio[‘cash’]←
            initial_capital – cumsum (portfolio[‘holdings’])
portfolio[‘total’]←portfolio[‘cash’]+cumsum(portfolio[‘holdings’])* price_data[‘Close’]
portfolio[‘return’]← pct_change(portfolio[‘total’])
```

4.3 Algorithm for Approximate Matching of Moving Average Using Cross Over Strategy

```
url ← "http://api_url"
json_data ←get(url)
file_path ← "E:/Data.csv"
signals ← DataObject.create()
portfolio ← DataObject.create()
signals['signal'] ← 0.0
short_window ←50
long_window ←150
initial_capital ← 100000
price_data_csv ← convert_to_csv(json_data)
price_data ← read_csv(file_path)
signals['short_mavg'] ← rolling_mean(price_data['Close'], short_window)
signals['long_mavg'] ← rolling_mean(price_data['Close'], long_window)
signals['fuzzyInput'] ← 100*((signals ['short_mavg'] - signals['long_mavg']) / signals['short_mavg'])
normalizedInput ← Antecedent(arange(-1, 1, 0.00001)
fuzzyOutput ← Consequent(arange(0,3,0.001), 'fuzzyOutput')
normalizedInput['low'] ← trimf(normalizedInput.universe, [-1, -1, 0])
normalizedInput['medium'] ←trimf(normalizedInput.universe, [-1, 0, 1])
normalizedInput['high'] ← trimf(normalizedInput.universe, [0, 1, 1])
fuzzyOutput['low'] ← trimf(fuzzyOutput.universe, [0, 0.5, 0])
fuzzyOutput['medium'] ← trimf(fuzzyOutput.universe, [0, 0.5, 1])
fuzzyOutput['high'] ← trimf(fuzzyOutput.universe, [2.0, 2.5, 3.0])
rule1 ← Rule(normalizedInput['low'] , fuzzyOutput['low'])
rule2 ← Rule(normalizedInput['medium'], fuzzyOutput['medium'])
rule3 ← Rule(normalizedInput['high'] , fuzzyOutput['high'])
movingAverage_ctrl ← ControlSystem([rule1, rule2, rule3])
movingAverageCrossOver  ← ControlSystemSimulation(movingAverage_ctrl)
  for x in signals['fuzzyInput']:
   movingAverageCrossOver.input['normalizedInput'] ←      round(x,5)
   movingAverageCrossOver.compute()
   signals['signal'][short_window:][i] ← movingAverageCrossOver.output['fuzzyOutput']
    if movingAverageCrossOver.output['fuzzyOutput']< 1.0:
      signals['signal'][short_window:][i] ← -1
    else:
      if movingAverageCrossOver.output['fuzzyOutput']> 1.0 and
movingAverageCrossOver.output['fuzzyOutput']< 2.0:
        signals['signal'][i] ← 0
      else:
      signals['signal'][i] ← 1
signals['possitions'] ←generate_diff()
portfolio['holdings']←multiply(signals['possitions'],            price_data['Close'])
portfolio['cash']←
                      initial_capital – cumsum (portfolio['holdings'])
portfolio['total']←
portfolio['cash']+cumsum(portfolio['holdings'])* price_data['Close']
portfolio['return']← pct_change(portfolio['total'])
```

5 Evaluation

Over a 5 year period starting from 31/1/2013 to 31/1/2018 data was collected with out of sample evaluation over a recent years with sliding window of one year period. The portfolio performance of 5 best strategies was evaluated and compared to benchmark strategies. Sample sizes should be mentioned.

OPTIMIZED-VERIFY-EVALUATE methodology was used to evaluate the proposed solution as in [22]. The first strategy was adjusted or optimized over the in the sample data, then verified over more recent out of sample data and finally evaluated over another set of more recent sample data. Data set were divided into three data sets as shown in Table 3. Each data set is consists of an OPTIMIZE, VERIFY and EVALUATE date ranges. EVALUATE date range remain constant for each data set.

Table 3 Data set used for evaluation

Data set name	Optimize years	Verify years	Evaluate years
401	4	0	1
311	3	1	1
302	3	0	2

For the evaluation of the trading strategies following matrixes were calculated for each and every strategy.

5.1 Sharpe Ratio

This ratio defines how much excess return traders are receiving for the extra volatility that bear for holding a riskier asset. In 1966, William Sharp derived a formula to calculate the Sharpe ratio [23]. Nowadays it has become a most referenced risk/return measure used in finance. The formula used for calculating shape ratio can be defined as follows [24].

$$SR(m) = (r - R)/Std(m) \tag{21}$$

Where m is the investment, r is the average rate of return of m, R is the best available rate of return of a risk-free security and Std(m) is the standard deviation of r.

The greater the Sharpe ratio means the better ratio between the return and additional risk. Usually the a strategy which is having a Sharpe ratio greater than 1 is acceptable by the investors and when the Shape ratio getting increase the acceptance of the trading strategy gets increased.

5.2 Maximum Drawdown

This is an indicator of the risk of a portfolio chosen based on a specific strategy. It gives a measurement of the largest drop from peak to bottom in the value of a portfolio before a new peak is achieved [25]. Here for calculating the maximum drawdown and Eq. 22 is used [26].

$$MDD = \frac{(PV - LV)}{PV} \tag{22}$$

MDD is Maximum Drawdown of a given strategy, PV is peak value before largest drop and LV is the lowest value before new high established [27].

5.3 Compound Annual Growth Rate (CAGR)

This provides the annual growth rate of an investment over a specified period of time longer than one year. This rate tells what the traders have at the end of the investment period [28]. The calculation of CAGR can be done by using the formula mentioned as Eq. 23 [29].

$$CAGR = (E/B)^{1/n} - 1 \tag{23}$$

Here E is the value at the investment beginning and B is the value at the investment end and n is the number of periods.

As shown in the Fig. 4, the system generates the graphical view of the closing price with the corresponding trading signals (buy = up arrow, sell = down arrow), and a graphical view of the account equity curve.

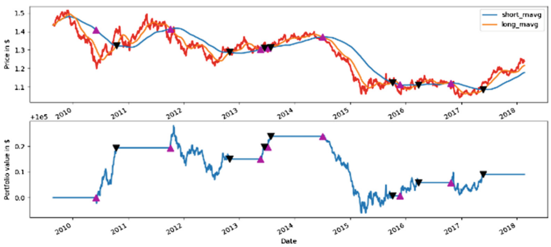

Fig. 4. Graphical view is provided by Exact Matching Module

Evaluation results of the system for the standard and fuzzy trading strategies with respect to the above mentioned matrixes are shown in Tables 4 and 5.

Table 4 Evaluation results

Data set: 2016-01-01/2017-01-01 Currency Pair: EUR/USD			
Strategy	Sharp ratio	CARG	MDD
Moving average cross over	− 0.36365	−0.02992	−16.9011
Fuzzy moving average cross over	1.154825	0.02992	−16.9011
MACD	0.019602	0.02881	15.9910
Fuzzy MACD	0.132493	0.07891	17.8984
Stochastic	0.123907	−0.03239	−8.78734
Fuzzy stochastic	0.234295	0.05673	−8.78713
RSI	0.342323	−0.06831	12.3454
Fuzzy RSI	0.345381	−0.10075	9.09849
Bollinger band	0.435668	0.12467	−4.99878
Fuzzy bollinger band	0.689839	0.124600	5.67648

Table 5 Evaluation results

Data set: 2017-01-01/2018-01-01 Currency Pair: EUR/USD			
Strategy	Sharp ratio	CARG	MDD
Moving average cross over	0.213237	0.25641	12.34447
Fuzzy moving average cross over	−0.484823	0.25778	12.55436
MACD	0.154310	−0.36414	10.45812
Fuzzy MACD	0.167682	−0.12894	12.43695
Stochastic	−0.564892	0.78262	−8.546433
Fuzzy stochastic	−0.349485	0.78231	−7.215457
RSI	0.213468	0.12596	14.5766
Fuzzy RSI	0.487940	−0.22535	23.9878
Bollinger band	0.697578	0.12425	−23.190
Fuzzy Bollinger band	0.697937	0.23155	24.2452

6 Conclusion

The results of the system show that the fuzzy logic based trading strategy developed under the approximate matching module perform better than the standard trading strategies and thus significantly improve the overall performance of the system and this confirms that the fuzzy logic based approximate matching module have a positive contribution to a successful trading system. Once the successful trading strategy has been developed and verified, traders can be successfully applied it real time to make the trading decisions. Selecting the right currency pair is important as selecting the trading strategy because the different currency pair has different price pattern and a specific strategy doesn't work same for all currency pairs. As well as the trading strategies backtested over historical data are no guarantee that it will perform well in another time period, therefore the traders should always remember this when they applied the backetested trading strategies in real time. For the Evaluation of the system only five years price data has taken into the account, but in order to increase the confidence in the trading system, it can be tested over more than five years and verify whether it maintain the consistent performance. On the other hand the evaluation and analyzing processes are manual processes in this system, but it is possible to significantly reduce the time and effort by automating more of the evaluation process. In the future this research work will further extend to backtest additional trading strategies and their combinations.

References

1. Ren, C.: The Investor's Guide to Technical Analysis. McGraw-Hill, New York (2004)
2. Schwager, J.: Getting Started in Technical Analysis. Wiley, New York (1999)
3. Colby, R., Meyers, T.: The Encyclopedia of Technical Market Indicators. Irwin Professional Publishing, New York (1998)
4. Achelis, S.: Technical Analysis from A to Z. McGraw Hill, New York (2001)
5. Brock, W., Lakonishok, J., Lebaron, B.: Simple technical trading rules and the stochastic properties of stock returns. J. Finance **47**(5), 1731–1764 (1992)
6. Sullivan, R., Timmermann, A., White, H.: Data-snooping, technical trading rule performance, and the bootstrap. J. Finance **54**(5), 1647–1691 (1999)
7. Lo, A., Harry, M., Wang, J.: Foundations of technical analysis: computational algorithms, statistical inference, and empirical implementation. J. Finance **55**, 1705–1765 (2000)
8. Park, C.-H., Irwin, S.H.: What do we know about the profitability of technical analysis? J. Econ. Surv. **21**(4), 786–826 (2007)
9. Metatrader4: MetaTrader 4 Official Release. https://www.metatrader4.com/en/company/140. Accessed 20 July 2017
10. VonAltrock, C.: Fuzzy Logic and Neurofuzzy Applications in Business and Finance. Prentice-Hall, Inc., Upper Saddle River (1997)
11. Weissman, R.: Mechanical Trading Systems: Pairing Trader Psychology with Technical Analysis. Wiley, Hoboken (2005)
12. Achelis, B.S.: Technical Analysis from A to Z, pp. 199–200. McGraw Hill, New York (2001)
13. Naranjo, R., Meco, A., Arroyo, J., Santos, M.: An intelligent trading system with fuzzy rules and fuzzy capital management. Int. J. Intell. Syst. **30**(8), 963–983 (2015)
14. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Pearson Educational Inc., Upper Saddle River (2003)
15. Gamil, A., El-fouly, R., Darwish, N.: Stock technical analysis using multi agent and fuzzy logic. In: Proceedings of the World Congress on Engineering, vol. 1, pp. 142–147 (2007)
16. Yager, R.R., Lotfi, A.Z. (eds.): An Introduction to Fuzzy Logic Applications in Intelligent Systems, vol. 165. Springer, Berlin (2012)
17. Castillo, O., Melin, P., Kacprzyk, J., Pedrycz, W.: Type-2 fuzzy logic: theory and applications. In: IEEE International Conference on Granular Computing, 2007. GRC 2007, pp. 145–145 (2007)
18. Iancu, I.: A Mamdani type fuzzy logic controller. In: Fuzzy Logic-Controls, Concepts, Theories and Applications. InTech (2012)

19. Zimmermann, H.-J.: Fuzzy control. In: Fuzzy Set Theory—And Its Applications, pp. 203–240. Springer, Dordrecht, (1996). http://doi.org/10.1007/978-94-010-0646-0
20. Patyra, M.J., Grantner, J.L., Koster, K.: Digital fuzzy logic controller: design and implementation. IEEE Trans. Fuzzy Syst. **4**(4), 439–459 (1996)
21. Christoffersen, P., Pelletier, D.: Backtesting value-at-risk: a duration-based approach. J. Financ. Econom. **2**(1), 84–108 (2004)
22. Zeigenbein, S.: A Fuzzy Logic Stock Trading System Based on Technical Analysis (2011)
23. Lo, A.W.: The statistics of sharpe ratios. Financ. Anal. J. **58**(4), 36–52 (2002)
24. Zakamouline, V., Koekebakker, S.: Portfolio performance evaluation with generalized sharpe ratios: beyond the mean and variance. J. Bank. Finance **33**(7), 1242–1254 (2009)
25. Pospisil, L., Vecer, J.: Portfolio sensitivity to changes in the maximum and the maximum drawdown. Quant. Finance **10**(6), 617–627 (2010)
26. DataCamp Community: Python For Finance: Algorithmic Trading. https://www.datacamp.com/community/tutorials/finance-python-trading. Accessed 8 Apr 2018
27. Perlin, M.S.: Evaluation of pairs-trading strategy at the Brazilian financial market. J. Deriv. Hedge Funds **15**(2), 122–136 (2009)
28. Barth, M.E., Elliott, J.A., Finn, M.W.: Market rewards associated with patterns of increasing earnings. J. Account. Res. **37**, 387–413 (1999)
29. Soliman, M.T.: The use of DuPont analysis by market participants. Account. Rev. **83**(3), 823–853 (2008)