Utilizing artificial neural networks and genetic algorithms to build an algo-trading model for intra-day foreign exchange speculation

Cain Evans [a,*], Konstantinos Pappas [a], Fatos Xhafa [b]

[a] Faculty of Technology, Engineering and the Environment School of Computing, Telecommunications and Networks Birmingham City University, UK

[b] Dept de Lienguatges I Sistemes Informatics, Universitat Politécnica de Catalunya, Spain

ARTICLE INFO

ABSTRACT

The Foreign Exchange Market is the biggest and one of the most liquid markets in the world. This market has always been one of the most challenging markets as far as short term prediction is concerned. Due to the chaotic, noisy, and non-stationary nature of the data, the majority of the research has been focused on daily, weekly, or even monthly prediction. The literature review revealed that there is a gap for intra-day market prediction. Identifying this gap, this paper introduces a prediction and decision making model based on Artificial Neural Networks (ANN) and Genetic Algorithms. The dataset utilized for this research comprises of 70 weeks of past currency rates of the 3 most traded currency pairs: GBP\USD, EUR\GBP, and EUR\USD. The initial statistical tests confirmed with a significance of more than 95% that the daily FOREX currency rates time series are not randomly distributed. Another important result is that the proposed model achieved 72.5% prediction accuracy. Furthermore, implementing the optimal trading strategy, this model produced 23.3% Annualized Net Return.

1. Introduction

Cognitive computing is focusing on how information is represented, processed, and transformed [1]. Artificial intelligence is a branch of Cognitive Science that involves the study of cognitive phenomena in machines. This involves three level of analysis: the computational theory, the algorithmic representation, and the hardware or software implementation. Artificial intelligence has been applied in almost every discipline. The last decade it has been noticed an increase in the utilization of such technique in Business and Finance. One of the applications of models such as Neural Networks is time series prediction in different markets such as the Foreign Exchange (FOREX) market.

According to the Bank of International Settlements, FOREX is a fast growing market that at the moment is estimated at $3.98 trillion. The time series of different currency rates are described as chaotic, extremely noisy, and non-stationary [2]. Almost every research paper that presents a prediction model starts with a reference to the Efficient Market Hypothesis (EMH). According to Fama [3], a random walk-efficient market is big enough to be manipulated by an existing large number of profit-maximizers who are competing with each other trying to predict future market values. This statement implies that

---

* Corresponding author. Tel.: +44 01213315000.
E-mail address: cain.evans@bcu.ac.uk (C. Evans).

the price of an asset reflects all the information that can be obtained at that time and that due to conflicting incentives of the participants the price will reach its equilibrium despite temporal disturbances due to market noise. While the theory seems to be correct for the time it was published, with the passing of time and due to future developments, especially in the area of communication and its impact on a globalized market, there is always the possibility that some areas or instances of the markets may be inefficient.

Today the world is highly interconnected and it takes a fraction of a second to transmit financial news around the globe. The reaction by the markets to market news tends to have an impact on the financial time series. The sequence of those reactions creates patterns that some practitioners believe that, as history repeats itself, so the market reaction does [4]. For instance, Azzof [5] provided evidence that there are cases where markets appear to be inefficient. Furthermore, Scabar [6] developed a hybrid prediction model based on an ANN and a Genetic Algorithm (GA) which gives evidence that financial time series are not entirely random.

Future market forecasting techniques can be classified into two major categories: fundamental analysis, and technical analysis. Fundamental analysis is based on macro-economic data such as Purchasing Power Parity (PPP), Gross Domestic Product (GDP), Balance of Payments (BOP), Purchasing Manager Index (PMI), Central Bank outcomes, etc. It is obvious that this kind of analysis has a more long term prediction spectrum and it is not the case for this paper. On the other hand, technical analysis focuses on past data and potential repeated patterns within those data. The major point here is that the history tends to repeat itself. As opposed to fundamental analysis, technical analysis makes short term predictions such as weekly, daily, or even hourly predictions.

There is a conglomeration of available tools suitable for technical analysis such as ANN, GA, Genetic Programming (GP), Econometrics tools, technical indicators, etc. Hirabayashi [7] introduces a forecasting optimization model that is based on a genetic algorithm that automatically generates trading rules based on technical indexes such as Moving Average (MA), and Relative Strength Index (RSI). Tulson [8] utilizes wavelet analysis to feed an ANN that predicts FTSE 100 index time series. Butler [9] developed an Evolutionary ANN (EANN) that makes future predictions based on macro-economic data. Tay [10] proposes a Support Vector Machine (SVM) regression approximation model.

The literature has revealed that there are cases where the combination of two or more techniques offers a better result. Yu [11] introduced an online learning algorithm to accelerate the learning process of the neural network. The results revealed that the algorithm outperformed other algorithms such as batch learning and Levenberg–Marquardt algorithm. Yao [12] use MA5, MA10, MA20, MA60, and MA120 to feed a Feed Forward Neural Network that forecasts several currency rates. The weekly based results are evaluated as positive.

Selecting the proper tool depend on many factors, such as the nature of data and the adopted methodology. Econometrics tools seem to have an adequate performance when data exhibit linear dependency as opposed when data are non-linearly correlated. On the other hand, neural networks provide generalization and mathematical function approximation to reveal or associate relations between input and target data in the case of supervised learning [13].

The main purpose of this paper is to confirm the hypothesis that intra-day FX market prediction is possible. To achieve this goal, four basic objectives should be satisfied. The first objective is to define the trading strategies and to develop the algorithms that Implements those strategies. The second goal is to detect instances of the market that show an inefficient behaviour. The third goal is to develop the forecasting model based on ANN and GA and the last objective is to test the developed model and evaluate the performance of the introduced trading strategies.

The remainder of the paper has the following structure. Section 2 presents the materials and the methods that formulate this research. It is separated into two main parts namely the trading strategy and the forecasting model. Section 3 presents the results of the data analysis as well as the performance of the forecasting model. Finally, Section 4 concludes the paper and presents some future work.

2. Material and methods

2.1. Relative work

There is a number of papers in the literature that propose different methodologies and techniques for prediction trading models in the FOREX market. Yuan [14] introduces a model that forecasts movement direction in exchange rates with a polynomial support vector machine. Matas et al. [15] developed an algorithm that is based on neural networks and GARCH models to make predictions while operating in an heteroskedastic time series environment. Enam [13] experimented with the predictability of ANN on weekly FX data, and concluded that, among other issues, one of the most critical issues to encounter when introducing such models is the structure of the data. Kamruzzaman [16] compared different ANN models feeding them with technical indicators based on past FOREX data and concluded that a Scaled Conjugate Gradient based model achieved closer prediction compared to the other algorithms. Zhang [17] presented a high frequency foreign exchange trading strategy model based on GA. This model utilizes different technical indicators such as MA, Moving Average Convergence Divergence (MACD), Slow Stochastic, RSI, Momentum Oscillator, Price Oscillator, Larry Williams, Bollinger Bands, etc. Several tests showed an annualized return rate of 3.7%. Yu [18] presented a dual model for FX time series prediction based on Generalized Linear Autoregressive and ANN models. The experiments showed that the dual model outperformed the single model approach based on econometrics techniques.

2.2. The FOREX market and the trading algorithm

The modern foreign exchange market started to take shape after the abandonment of the Bretton Woods system of monetary management. Some of the unique features of this particular market include: the huge daily trading volume, the geographical dispersion, and the continuous operation during the weekdays. Imagine the FX market as a concentric system with different circles around the central point. At the epicentre of the system, the interbank market is made of the largest commercial banks and security dealers. The next circle, the second tier, comprises of other smaller participants such as commercial companies, hedge funds, pension funds, foreign exchange traders, etc. The larger the distance from the epicentre, the wider the bid–ask spread.

According to Altridge [19], foreign exchange markets accommodate three types of players: high frequency traders, long term investors, and corporations. This paper proposes a trading model for high frequency traders who speculate on small intra-day price fluctuations. A trading system consists of three major parts: rules for entering and exiting trades, risk control, and money management [20]. Money management refers to the actual size of the trade to be initiated [21]. This paper will concentrate the efforts on return rates, and as such, it will not refer to money management. Just for informative purposes, the lowest limit for joining the FX market is $100,000. With a leverage of 100:1, an individual player can initiate the trade starting with $1000. Of course, a high frequency trading technique is structured on the basis of making profits from small price fluctuations, and as such, a substantial amount of money should be initialized. The second important part of a trading model, risk management or hedging, refers to the method of covering potential losses by trading assets that are negatively correlated to the current position. Due to the fact that the primary objective of this research is to examine the predictability of foreign exchange time series this method is not included in the suggested strategies.

The rules to enter and exit require adequate knowledge of the markets. The definition of a strategy includes the introduction of the entry time, the trading duration and exit conditions, the trading, assets, as well as other parameters such as for instance whether more than one asset will be traded simultaneously, etc. High volume is one of the criteria to decide when to start trading and this happens when two markets overlap. The highest volume is usually observed during the switching between European and American markets. The reason that this parameter is so important is because high volume means narrower bid–ask spread. Therefore, in the proposed trading strategy the trading session starts at 12:36 GMT and is being terminated after six hours.

The next paragraphs introduce three trading strategies that are implemented by the proposed model. The first is the simplest one. There is a base currency and a quote currency. Based on the outcome of the prediction model that will be introduced later on, the model produces a long or short signal. The second strategy involves two currency pairs. Here the decision about the wining trade is based on the best return. The third strategy includes two currency pairs that will be both traded simultaneously.

First FX trading strategy S1

$$base(curr1) \wedge up(curr2) \Rightarrow trade(long(curr2))$$
$$base(curr1) \wedge down(curr2) \Rightarrow trade(short(curr2)).$$

Second FX trading strategy S2

$$base(curr1) \wedge down(curr2) \wedge down(curr3) \wedge (Greater(curr2, curr3)) \Rightarrow trade(short(curr3))$$
$$base(curr1) \wedge down(curr2) \wedge down(curr3) \wedge (Greater(curr3, curr2)) \Rightarrow trade(short(curr2))$$
$$base(curr1) \wedge up(curr2) \wedge down(curr3) \wedge (Greater(abs(curr2), abs(curr3))) \Rightarrow trade(long(curr2))$$
$$base(curr1) \wedge up(curr2) \wedge down(curr3) \wedge (Greater(abs(curr3), abs(curr2))) \Rightarrow trade(short(curr3))$$
$$base(curr1) \wedge down(curr2) \wedge up(curr3) \wedge (Greater(abs(curr2), abs(curr3))) \Rightarrow trade(short(curr2))$$
$$base(curr1) \wedge down(curr2) \wedge up(curr3) \wedge (Greater(abs(curr3), abs(curr2))) \Rightarrow trade(long(curr3))$$
$$base(curr1) \wedge up(curr2) \wedge up(curr3) \wedge (Greater(curr2, curr3)) \Rightarrow trade(long(curr2))$$
$$base(curr1) \wedge up(curr2) \wedge up(curr3) \wedge (Greater(curr3, curr2)) \Rightarrow trade(long(curr3)).$$

Third FX trading strategy S3

$$base(curr1) \wedge down(curr2) \wedge down(curr3) \Rightarrow trade(short(curr2), short(curr3))$$
$$base(curr1) \wedge down(curr2) \wedge up(curr3) \Rightarrow trade(short(curr2), long(curr3))$$
$$base(curr1) \wedge up(curr2) \wedge down(curr3) \Rightarrow trade(long(curr2), short(curr3))$$
$$base(curr1) \wedge up(curr2) \wedge up(curr3) \Rightarrow trade(long(curr2), long(curr3)).$$

Having introduced the strategies, now it is time to define the algorithm that executes those strategies. An algorithm is a sequence of executable commands that has a beginning, a body and an end. In addition, an algorithm is fed with some kind of inputs and produces some outputs. In this particular case the algorithm is given the 16 h previous currency prices in 40 min intervals, and the algorithm makes a trading decision. It is important to mention here that the data will be analysed and processed before being ported to the model.
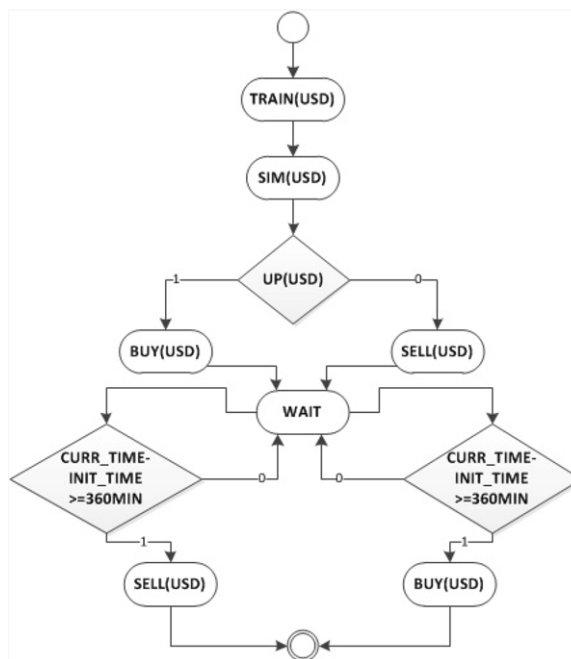
Fig. 1. S1 — Activity diagram.

The first two nodes of the activity diagram in Fig. 1 represent the forecasting part of the algorithm. Given the forecasting results, and depending on whether the prediction shows the currency going up or down, the algorithm sends the appropriate signal. The trading time has been set to six hours. After the pass of this time, the algorithm sends a trading signal to terminate the session and to record possible gains or losses.

Fig. 2 depicts both the first and the second algorithm share the same basic principles. The difference between the first and the second algorithm is in the fact that while the first strategy initiates only one currency pair during the whole session, in the case of the second algorithm two currencies are initiated and the algorithm decides to trade one of the two currencies based on the best performance.

On the other hand, the third algorithm that is depicted in Fig. 3 shows that the third strategy is somewhere between the previous two algorithms. In this case, two currencies are initiated and both currencies are traded simultaneously. This is a kind of modest strategy that tries to minimize the risk.

Fig. 4 presents the overall trading algorithm that gives the opportunity to the investor to choose between a single strategy or to leave the system to choose the optimal trading strategy. Therefore, the algorithm accepts as input the base currency and one of the three trading strategies if the case is to run one of the strategies. This is defined by choosing path A of the trading algorithm. If path B is activated, then there is no need for defining the strategy as the algorithm compares the performance of the three strategies and initiates the most profitable one.

An important property of this algorithm is the scalability which allows additional strategies to be incorporated into the model. Furthermore, the modularity of the proposed model means that the system is able to trade other financial instruments such as securities and more complex products including futures and options.

2.3. Data analysis

FX intra-day rates time series are described as noisy, chaotic, displaying a nonlinear relation, and exhibiting non-stationary behaviour. It is obvious that it is difficult to provide those data for prediction without firstly having some kind of transformation. One of the first issues to deal with is the frequency of the sampling. High sampling frequency means additional useless and sometimes disorienting information. On the other hand, lower frequency means that not all the essential information is included. According to Refenes [22], it is sufficient to sample the market data at intervals between 5 and 60 min depending upon the currency pair. There are different methodologies for defining the appropriate frequency with the most frequent the analysis of the autocorrelation. The noise of the sample is another issue that should be addressed. The literature review revealed that technical indicators e.g. MA is one of the preferred solutions. The last question to be answered before the forecasting model is developed concerns the level of data predictability. This means that the time series should be exposed to statistical tests to confirm or to reject random behaviour. The Wald–Wolfowitz test and Kolmogorov–Smirnov are examples of statistical tests that are utilized for this purpose.

This paper examines and experiments with the predictability of three major currency pairs: GBP \USD, EUR \GBP, and EUR \USD. These data correspond to 70 week spot rates tick observation from 1/10/2010 to 28/2/2012. The selected sampling
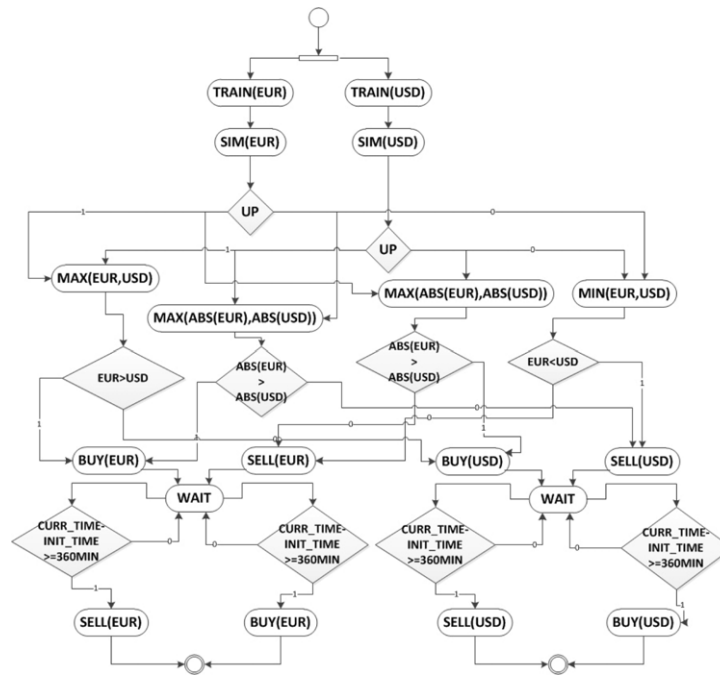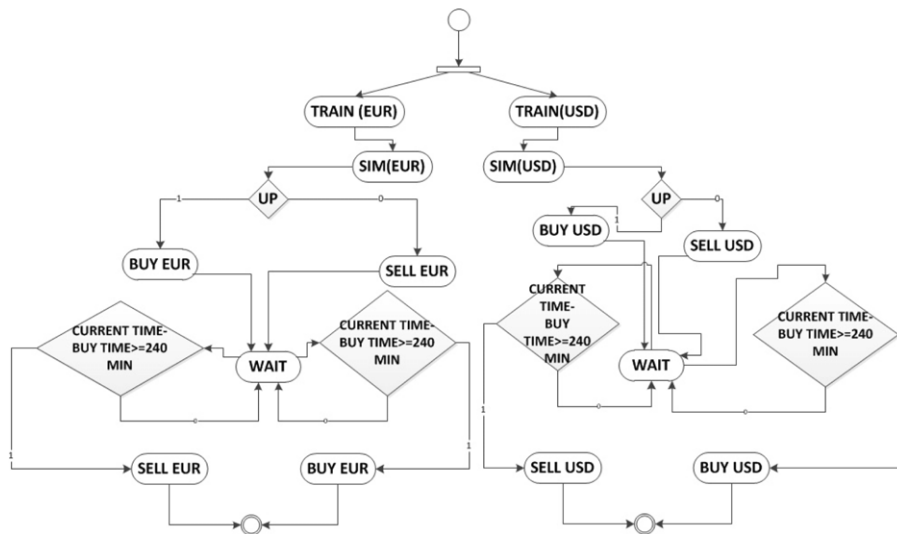
Fig. 2. S2 — Activity diagram.

Fig. 3. S3 — Activity diagram.

frequency is 40 min and the noise has been mitigated by taking the average of the 40 min time intervals. Fig. 5 shows the behaviour of GBP \USD exchange rate during the sampling period.

The middle compartment of the graph displays the observations of 2011. As the variance is much greater than the data of 2010 and 2012, it is clear that the volatility of this year is greater than observations coming from the previous or the next year. It is important here to mention that there are two rates in the forex market: the ask or offer, and the bid or sell rate. The presence of both ask and bid rates is described as redundant. There is a high correlation between those rates due to the fact that on average, the bid–ask spread seems to be the same. The proof of this claim is presented in the Results and Discussion section.

Data representation is the next step of the process. According to Vastone [21], it is important not to let the ANN have visibility of the market prices, or currency rates in this case. If raw prices are being provided as input, then two identical patterns that defer by a constant will be treated as two different patterns and make very difficult the generalization process. To overcome this issue, the datasets are being transformed to return rates which is the log difference of the exchange rates. Additionally, the log difference transformation does contribute in eliminating the presence of heteroskedasticity in the

Fig. 4. Activity diagram of the trading algorithm.
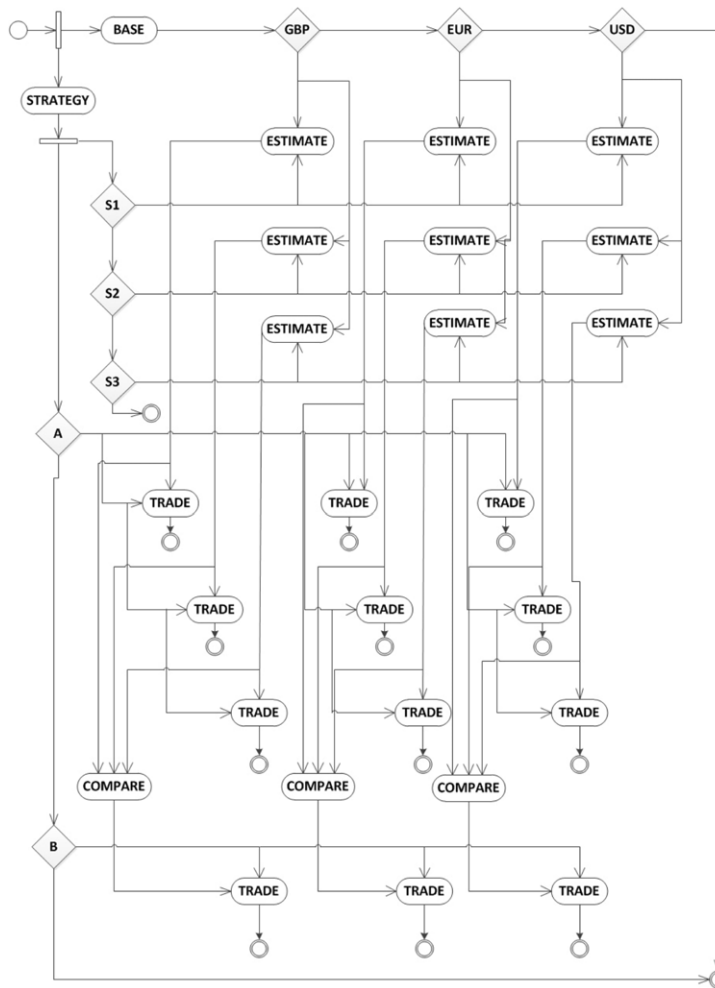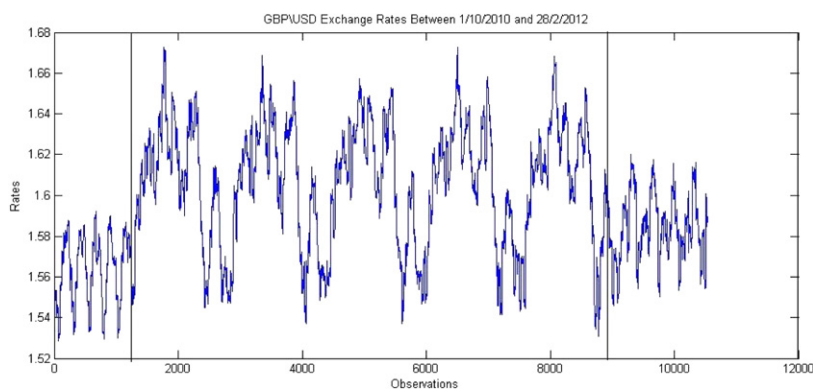

Fig. 5. GBP\USD exchange rates 1-10-2010–28-2-2012.

examined time series. Eq. (1) shows the log difference function:

$$r = \ln\left(\frac{P_t}{P_{t-1}}\right). \tag{1}$$

ln is the natural logarithm, P_t is the price at time t, and P_{t-1} is the price at time $t-1$. The graphs included in Figs. 6 and 7 respectively show the data before and after the transformation.
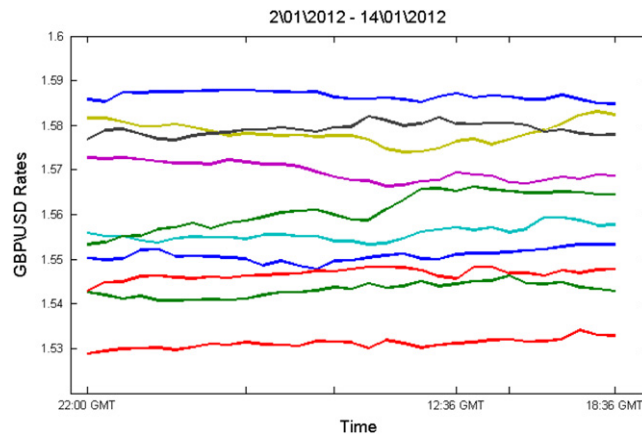
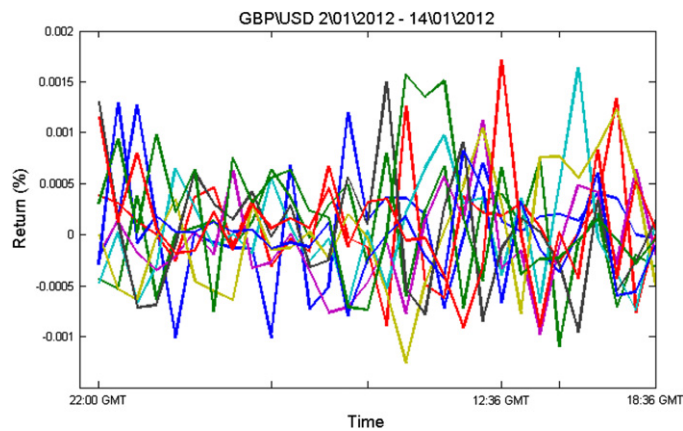**Fig. 6.** The data before the transformation.

Fig. 7. The data after the transformation.

While this transformation provide better generalization, on the other hand raises new issues such as convergence (generalization and convergence will be covered in more detail when referring to neural networks). One frequent way of dealing with this issue is by introducing some kind of technical indicators to smooth the data. Prediction systems that intend to make short term predictions expose relatively adequate positive performance when fed with technical indicators. Martinez [23] utilizes Exponential Moving Average (EMA) and Bollinger Bands (BB) to build a day-trading system based on NN that attempts to indicate the optimal enter and exit strategy. Furthermore, Tsang [24] feeds the proposed model with a basket of technical and fundamental data both processed with technical indicators such as MA, MACD, and RSI. The experiments showed that the particular system achieved a 70% success in predicting the right direction of the market.

Moving Average, in one or other form, is a technical indicator found in almost every paper describing financial time series forecasting models. The MA technique performs pretty well when the market follows a trend. However, this indicator performs rather poorly when the index changes direction. To tackle this issue, this paper proposes an alternative version that is shown in Eq. (2):

$$y_i = \frac{\sum_{j=1}^{i} x_i}{i}. \tag{2}$$

This version is named Incremental Window Moving Average (IWMA) and the intention is that the indicator sleeks the data points gradually so that the system can react better to the market turning points. The graph shown in Fig. 8 presents the data structure after the deployment of the proposed transformation.

Once detrending and normalization have been completed, the dataset should be exposed to statistical tests to examine for the presence of random behaviour. If the null hypothesis of the data being randomly distributed is rejected, the dataset is ready to be imported into the developed forecasting model. On the contrary, the presence of random behaviour makes the forecasting process impossible and the whole attempt should be abandoned. The execution and the results of such tests applied on the examining dataset will be presented in the Results section.
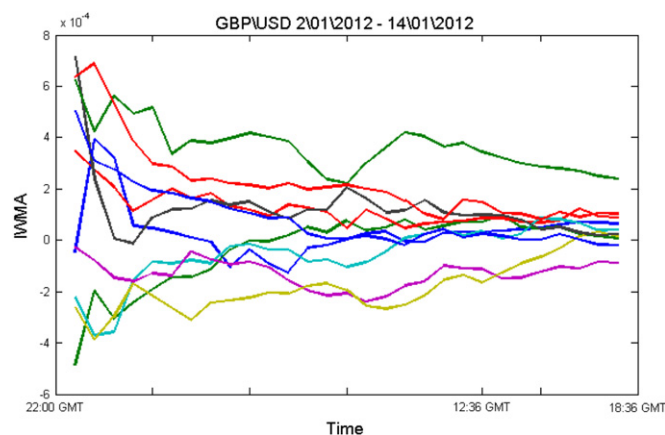
Fig. 8. The data after the IWMA transformation.

There is a last issue to solve before presenting the neural networks. Data consistency is very important. If the prediction model is fed with irrelevant data, then the results are going to be poor. To tackle the market evolution, a good practice is to keep the input data consistent. One way to achieve this is to periodically replace past data with more recent data. In other words, imagine the sample test as a long queue where the data is being placed in reverse chronological order and where for each new trading day, the queue is being pushed by a portion of a trading day data so that the oldest trading day is discarded from the front of the queue, while the most recent data are placed at the back end.

2.4. Neural networks

Neural networks are described as the processing methodology that maps the input values to the target values. Non-linear modelling, generalization, and universal approximation are some of the advantages of neural networks [25]. These tools are classified according to the learning techniques in three main categories: supervised learning, reinforcement learning, and unsupervised learning. Feed Forward Multi-Layered Perceptron belongs to the first category. The network has a layered structure where each neuron's synapses are connected only to the output of the previous layer and the output of the same neuron is connected only to units of the following layer. There are several important factors to consider when developing a neural network including: the input and output vectors, the activation function, the training function, and the structure of the network.

2.4.1. The dataset

As mentioned earlier, supervised learning requires some input and the corresponding target data. Given the input data, the network is responsible for being able to produce outputs similar or approximately similar to the target data. The whole dataset is usually separated into three parts, the training set (70%), the validation set (15%) and the testing set (15%). The validation set indicates when the network has been trained. This happens when the validation error is becoming greater than the training error. This process guaranties that the network is not being over-fitted. On the other hand, the testing set measures the forecasting performance of the network. The training and the validation set is shuffled to avoid time dependent learning. The literature review revealed that the testing or the validation set should be approximately from one fourth to one eight of the training set [26]. Additionally, Kaufman [27] suggests that a balanced split is 70-15-15 for training, validation, and testing sets.

Back to the proposed model, the dataset comprises the exchange rates of the three major currency pairs (GBP \USD, EUR \GBP, EUR \USD) for the period from 1-10-2010 to 28-2-2012. The previous section described the analysis and the preparation of the data. As such, the input dataset contains vectors of 20 elements of detrended and normalized daily currency rates that correspond to 14.6 h of daily trading between 22:00 and 12:36 GMT. The target dataset comprises of single point values that correspond to the detrended and normalized return rate experienced 6 h (at 18:36 GMT) after the last value of the input dataset. This means that the prediction horizon is 6 h. The dataset is separated into four parts. From the first 300 time series, 70% of the data is the training set, 15% the validation set, and 15% the in-sample testing set. The remaining 40 time series serve as the out-of sample testing set.

2.4.2. Activation function and the back-propagation algorithm

The activation function is a way that the output of an individual neuron is scaled to the desired value range. There are different activation functions such as the linear activation function, the logistic function, the hyperbolic tangent function, etc. The hyperbolic tangent function takes values from the range $[-\infty, \infty]$ and squashes them to the range $[-1, 1]$. Eq. (3)

shows the hyperbolic tangent function:

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^{xg} - e^{-xg}}{e^{xg} + e^{-xg}} = \frac{e^{2xg} - 1}{e^{2xg} + 1}. \tag{3}$$

Since the log difference of the exchange rates lies in the region $[-1, 1]$, the developing neural network will be incorporating the hyperbolic tangent function into the structure of each neuron.

The back-propagation method is a technique for adjusting network weights in order to minimize the cost or the energy function. The back-propagation method has two parts: the error calculation, and the learning of the network. There are several training algorithms that follow the general principals of the back-propagation methodology. The Levenberg–Marquardt (LM) algorithm is a standard technique used to solve nonlinear least square minimization problems. LM was designed to approach second-order training speed without having to compute the Hessian matrix which represents the second derivative of the energy function matrix.

The LM curve-fitting method is actually a combination of the gradient descent and the Gauss–Newton method. In the gradient descent method, the sum of the squared errors is reduced by updating the parameters in the direction of the greatest reduction of the least squares objective. In the Gauss–Newton method, the sum of the squared errors is reduced by assuming the least squares function is locally quadratic, and finding the minimum of the quadratic [28]. The LM method acts more like a gradient-descent method when the parameters are far from their optimal value and acts more like the Gauss–Newton method when the parameters are close to their optimal value [29]. For more information about the evolution of the Levenberg–Marquardt algorithm the reader may search the original papers written by Levenberg [30] and Marquardt [31].

2.4.3. ANN topology

The choice of the number of layers and the neurons inside each layer is a very crucial factor to be considered when designing neural networks. The number of units in the input and the output layer is dictated by the solution itself. The proposed model has 20 input nodes and 1 output node. While the definition of the input and output layer is quite straightforward, things are very different when trying to define the number of hidden layers and the units for each such layer.

According to Kaastra [32], one or two hidden layers are enough to approximate any smooth bounded function. The introduction of additional hidden layers to the network makes the training more difficult due to the fact that the training process of large networks is more complex and time consuming. It is also increasing the possibility of the network being over-fitted.

The literature review has revealed that while there is no standard procedure in defining the optimal number of hidden neurons, practitioners usually adopt between three different approaches. The direct search is a trial and error problem solving approach that dictates the structuring of several different topologies which are executed and compared so that the optimal topology is selected.

A second solution is the so called rule of thumb where there are some basic rules saying for instance that the number of hidden layers is somewhere between the number of inputs and the number of outputs, or the hidden neurons are equal to 75% of the number of neurons in the input layer [33]. It is obvious that this is an approximation of the solution that needs more refinement such as combining it with the third option. The third option is based on genetic algorithms and their ability to find the global minima by searching at the same time in many directions.

2.5. Genetic algorithms

The Genetic Algorithm (GA), developed by Holland [34], is an optimization and search technique based on the principles of genetics and natural selection. A GA allows a population composed of many individuals or chromosomes to evolve under predefined selection rules to a state that maximizes the fitness [35].

The algorithm mimics natural selection in an iterative process where a population of potential solutions is evolving so that the optimal solution is found. Once the iteration has run the next generation is produced by selecting and processing the most-fit individuals based on their fitness. Each member of the population represents knowledge by a number of genes that form the chromosome. The next sub-section presents the definition of the population as well as the fitness function.

2.5.1. Chromosome, genes, and fitness function

To give some mathematical notations, let n be the number of the population where each individual represents a chromosome for $i = 1 \ldots n$ and let m be the number of genes in each chromosome for $j = 1 \ldots m$ where $x_i = \{z_1 \ldots z_m\}$ Additionally, consider $z_j \in N = \{a \ldots b\}$

Each $x_i \in \bigcup_{j=1}^{m} z_j$ represents a candidate solution to the problem $\min f(x_i); x \in \bigcup_{j=1}^{m} z_j \, \forall z_j \in N = \{a, \ldots, b\}$ where the function f is defined over the range of natural numbers $N = \{a, \ldots, b\}$. Therefore, the task of the genetic algorithm is to minimize the f looking at the space $N = \{a, \ldots, b\}$ The f is the fitness function and the outcome of the function is the fitness value of the input argument that represents each individual member of the population.

An important factor is the representation of each individual or chromosome. The encoding of the chromosome is dictated by the solution. As far as the topology optimization problem is concerned, the chromosome can be encoded in integer or in binary (or bit-string) format. Therefore, a potential structure of the chromosome could be formatted by concatenating two

genes each consisting of an integer which represents the number of neurons in a hidden layer. Setting $m = 2$, and for $a = 0$ and $b = 15$, the fitness function will have the following structure:

Fitness_Function(chromosome)
hidden_layer_1 := chromosome(0)
hidden_layer_2 := chromosome(1)
input_layer := 20
output_layer := 1
ann := **CreateAnn**(hidden_layer_1, hidden_layer_2, input_layer, output_layer)
ann := **TRAIN**(ann)
prediction := **SIMULATE**(ann)
results := **EVALUATE**(prediction)
wrong_predictions := **GET_WRONG_PREDICTIONS**(results)
mae := **mae**(results)
aof := a*mae + b*wrong_predictions
return aof **end**

Meanwhile, the integer encoding causes difficulties in the process of crossover and mutation. It is clear that a better solution would be for the population to be encoded in binary format. With $a = 0$ and $b = 15$, this means that each gene has ($2^4 = 16$) 4 bits and with $m = 2$, the chromosome comprises of 8 bits. Therefore, the population is encoded in binary format and is transformed in integer format inside the fitness function.

Having defined the structure of the chromosome, the next step is to define the initial population. According to Man [36], despite the required processing power, a large initial population may provide a faster convergence. Of course this relies on the size of the search area. Here the initial population is set to 30 individuals and the generation number is set to 40.

Another important factor is the way the initial population is produced. A fully randomly generated initial population may get trapped to local minima while an ad hoc method may become too biased and direct the solution to a specific area. However a combination of those solutions seems to be a reasonable compromise.

2.5.2. Selection

Selection can be best described as the process that provides the fitness of each individual number and decides which individual survives for the next generation. There are several algorithms that execute this process. One such example is the roulette-wheel selection [37]. Provided the initial population n the algorithm calculates the cumulative fitness $F = \sum_{i=1}^{n} f(x_i)$ for each individual x_i Then each chromosome is assigned the selection probability $f(x_i)/F$. A portion of the individuals with the highest probability passes directly to the next generation without any changes as an elitism policy which in this solution is set to 1 individual. The rest of the selected population is applied a stochastic process to identify the population that is going to be applied in the next stage which is the crossover process.

2.5.3. Crossover

Crossover is the process of choosing a portion of the population with fitness value probably better than the average, and creating pairs that exchange information [38]. The basic idea here is that the parents with a high score may produce children with even higher scores. It is like a couple where both the man and woman are tall and where the obvious answer is that their child is going to be at least as tall as their parents. Crossover is the central feature of chromosomes. This is the reason why in most of the implementations the crossover parameter is quite high, say between 50% and 80% of the population. Another important parameter as far as the crossover process is concerned is the crossover point. Depending on the length of the chromosome as well as the nature of the solution, the chromosome can be split on one or more points. The point where the chromosome is split can be a predefined value or can be produced by a stochastic process. This solution adopts the two point crossover operator and the crossover rate is set to 80%.

2.5.4. Mutation

The previous process was based on past information. The new individual combines information from both parents. In nature, apart from the genes that organism has inherited from their parents, there is also something else that affects the process of the evolution and this is the radiation coming from the Sun or from the environment. It is obvious that this is a complete stochastic process. Mutation directs a portion of the population to an area close to the existing one but never visited before. The hope is that solutions around an individual with high fitness value may prove to have a better performance. The idea is that if a solution is near to global minima or global maxima, then a small step may set the solution a step closer to the final solution. This process is stochastic and as such a mutation rate should be defined which in this case is set to 20%.

2.5.5. Termination criteria

There are two basic termination criteria namely the number of generations and the number of stall generations. In this experiment, the number of generations is set to 40 while the number of stall generations has been set to 15. Therefore,

the algorithm cannot create more than 40 generations and if the optimal fitness value does not change for 15 consecutive generations, then the algorithm terminates even if the generation number is less than 40.

2.6. Performance metrics

Choosing the appropriate performance metrics is crucial for both the development of the forecasting model, and for evaluating different trading strategies as well. There are two basic categories of performance metrics: the traditional performance metrics based on statistics outcomes such as Mean Absolute Error (MAE), Mean Square Error (MSE), or Theils Inequality Coefficient (Theils U), and those based on direct measurement of the forecasting model objectives such as Annualized Return, Sharpe Ratio, Cumulative Investment Return, etc. Castiglione [25] assess the efficiency of the learning and discard bad trained nets by utilizing the MSE while Trujillo [39] developed a genetic algorithm that evaluate net topology using RMSE as a cost function. Similarly, Gao [40] introduced an algorithm that evaluates ANN performance by adopting a technique that is based on the sum of squared error (SSE). On the other hand, according to Dunis [41], traditional standard statistical measures are often inappropriate to assess financial performance. Furthermore, in some cases, different trading strategies cannot be compared with these standard measures for the simple reason that they are not based on forecasting the same nature of output.

When developing a prediction model the first and more important metric is the right direction of the market. Another important issue is the prediction error. Meanwhile, the classic optimization problem focuses on a single objective approach. When more than one objective ought to be optimized, then the problem is called multi-objective optimization. This approach requires the introduction of a single aggregate objective function (AOF). There are different techniques for creating an AOF. One intuitive approach to creating such a function is the weighted linear sum of the objectives. Therefore, the adopted approach includes the construction of an AOF that consists of the weighted sum of the missing trades and the mean absolute error. Eq. (4) depicts the aforementioned function:

$$aof(x_1, x_2) = b_1 x_1 + b_2 x_2. \tag{4}$$

The previous function acts as the objective function of the developed GA that optimizes the topology of the forecasting model. However, evaluation of the model as far as the trading strategy is concerned dictates the adoption of those metrics that express the performance of the system in terms of the profitability.

Mean absolute error

$$mae(r) = \frac{\sum_{i=1}^{n} |r_i - \bar{r}|}{n}. \tag{5}$$

Annualized return

$$r^A = 252 \frac{1}{n} \sum_{i=1}^{n} r_i. \tag{6}$$

Sharpe ratio

$$sr = \frac{\frac{\sum_{i=1}^{n} r_i}{n} - \frac{r_{fix}}{252}}{\sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (r - \bar{r})^2}}. \tag{7}$$

Correlation coefficient

$$corr = \frac{\sum_{i=1}^{n} ((x_i - \bar{x})(y_i - \bar{y}))}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}}. \tag{8}$$

Annualized return is the return an investment provides over a period of time, expressed as a time-weighted annual percentage. The Sharpe Ratio shows how appealing an investment is. Eq. (8) presents the correlation coefficient that provide information about how correlated the predicted and the actual values are.

3. Results and discussion

The performance of the financial forecasting model depends primarily upon three general factors: the appropriate data processing and presentation, the optimal trading strategies, and the structure of the forecasting model.

Table 1
Correlation between bid and ask rates.

Currency pair	Correlation coefficient	P-value	t-value
GBP\USD	0.99999974	0	123,925
EUR\GBP	0.99999643	0	33,286.7
EUR\USD	0.99999944	0	84,040.9

Table 2
Descriptive statistics.

	GBP\USD	EUR\GBP	EUR\USD
Observations	8700	8700	8700
Mean	−0.000026	0.000032	0.000038
Median	−0.000028	−0.00003	0.000028
Std. div.	0.000305	0.00029	0.0004
Skewness	−0.2151	−0.5848	0.0798
Kurtosis	13.6738	15.519	11.9587
Minimum	−0.003	−0.0041	−0.0034
Maximum	0.0026	0.0023	0.0045
Kolmogorov H	1	1	1
Kolmogorov P	0	0	0
Kolmogorov ST	0.499	0.4991	0.4988
Wald-W Test H	1	1	1
Wald-W Test P	0	0	0

3.1. Testing for correlation

The previous section defined the data model that will be imported into the developed forecasting model. The structure of the dataset is as important as the information of the dataset itself. It is crucial that the dataset includes only the necessary information required to serve its purpose. Redundant information increases the error and makes the forecasting process slower. Including both bid and ask rates simultaneously into the model looks like a redundancy because most of the time, those variables show similar behaviour.

The bivariate correlation test for the bid–ask rates of the three currency pairs is presented in Table 1. The column of the p-value shows that with a significance level of less than 1% the correlation coefficient in all cases is extremely high. Additionally, the t-value is well above the critical value of Student's Distribution which is 2.576 at 1% two tail significance level with degree of freedom equal to 8698. Therefore, in each of the three tests, the null hypothesis of the correlation coefficient being insignificant is rejected indicating that bid and ask rates are highly correlated. For that reason, only the ask rates are proceeded for being utilized to train the developed neural network.

3.2. Testing for randomness

At this stage, the dataset should be tested for random behaviour. There is a conglomeration of statistical tests for this purpose. Table 2 shows the descriptive statistics of the dataset as well as the results after the execution of the Kolmogorov–Smirnov and the Wald–Wolfowitz statistical tests.

Table 2 shows that both tests have rejected the null hypothesis that the three time series follow the normal distribution. Additionally, the kurtosis of those time series is well above the value 3 which is what kurtosis would be if the time series were normally distributed. Therefore, the time series are not normally distributed.

3.3. Optimal network topology

Before presenting the results taken after executing the GA, it is important to define the optimal topology. One topology is optimal when it satisfies three basic features namely convergence, generalization, and stability. An ANN is meant to converge when all the input patterns have been assimilated. The perfect convergence is when the minimum error of each pattern is almost equal. Generalization means that the network exhibits a good performance when out-of sample data is introduced. Finally, stability means that even when the network is being fed with slightly different datasets, the performance remains satisfactory. For instance, the network has been trained with GBP \USD rates, and then it simulates EUR \GBP rates.

As the GA is a stochastic process by its nature, the algorithm has to be executed several times. Therefore, the algorithm has been executed 5 times for each currency pair. Fig. 9 depicts the execution of the developed genetic algorithm and particularly the optimization of the network for the GBP \USD currency pair.

The results taken after the execution of the developed GA revealed that topology 20-9-8-1 was the winning topology in four out of the five executions when applied to EUR\GBP. In the case of the EUR\USD pair the same topology won three times. On the other hand, GBP\USD optimization indicated topology 20-1-8-1 as the winning topology for three times while for the other two times the winning topology was 20-9-8-1. Table 3 shows the results after the fifth execution of the algorithm. The
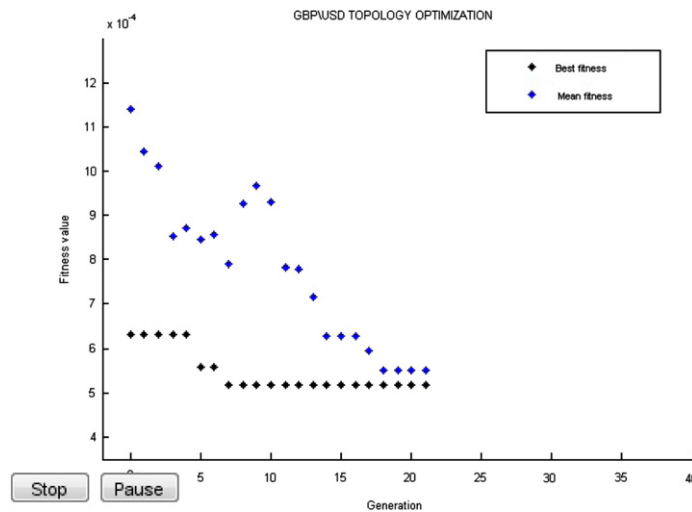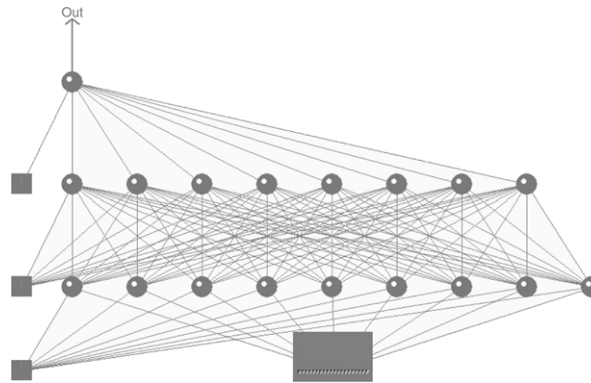
Fig. 9. Topology optimization.

Fig. 10. The winning topology 20-9-8-1.

topology column shows an eight digit binary number that represents the chromosome. The first four digits show the size of the first hidden layer of the network while the other four digits show the size of the second layer. If the first four digits or the last four digits are zero, this means that the network has only one hidden layer. The Fitness column shows the value of the fitness function for the corresponding topology implementation. The lower the fitness value, the most efficiently the topology is performing.

Therefore, provided the results from the execution of the algorithm, the topologies under investigation are the topologies 20-9-8-1 and 20-1-8-1. Table 4 presents the fitness value and the correlation of each of the testing topologies applied both in in-sample and out-of sample data.

The testing of both topologies with respect to the datasets of the other currency pairs both in in-sample and out-of-sample datasets revealed that the topology 20-9-8-1 is the most stable and efficient, and also outperforms in the case of the out-of-sample dataset apart from the EUR\GBP currency pair as in this case the error is higher than when applied to the in-sample dataset. Fig. 10 depicts the structure of the aforementioned topology.

Fig. 11 shows the performance of this particular topology during the training, testing and validation stage using in-sample data corresponding to EUR\USD currency rates.

Fig. 12 shows the innovations after the execution of the EUR\USD ANN. It is important to mention here that the error terms have been tested for autocorrelation as well as for correlation with the independent variables. In both case the statistical results were negative meaning that there is no dependency and the model is free of misspecification error.

As the performance graphs in Fig. 13 show, the R^2 in training, validation, and testing stages are all above 0.79 indicating the improved accuracy of this particular topology. Therefore, as the only topology to have satisfied all the predefined conditions, the topology 20-9-8-1 is the optimal topology for the given forecasting model.

Table 3
Winning topologies.

GBP\USD topology	GBP\USD fitness	EUR\GBP topology	EUR\GBP fitness	EUR\USD topology	EUR\USD fitness
10011000	0.0004	10011000	0.0003	00111100	0.0005
00011000	0.0005	10000100	0.0003	10000100	0.0007
01010011	0.0007	10011000	0.0003	00110100	0.0008
01100001	0.0007	10011000	0.0003	01100001	0.0008
00011110	0.0008	10011000	0.0003	00110100	0.0008
10010001	0.0009	00111100	0.0006	01101100	0.001
10110010	0.0009	00111100	0.0006	00100101	0.001
00110000	0.0009	01010011	0.0006	00100000	0.0011
01101001	0.0009	00110100	0.0008	00110101	0.0011
01110001	0.0009	00100000	0.0009	00010000	0.0012
00110010	0.001	00100001	0.0009	00000001	0.0012
01010101	0.001	10010011	0.001	00010100	0.0012
01000001	0.001	00110000	0.001	00010100	0.0012
10001010	0.0011	10111101	0.001	00000001	0.0012
00001011	0.0011	00011000	0.001	00100100	0.0013
11010000	0.0011	10011010	0.0011	00100100	0.0013
00001101	0.0011	00000001	0.0012	01100101	0.0013
10000001	0.0011	00010000	0.0012	00100100	0.0013
00011101	0.0012	01010000	0.0013	01101101	0.0013
11000000	0.0012	10001000	0.0013	01110010	0.0014
00001111	0.0012	01110000	0.0013	11110101	0.0015
11000000	0.0012	10011101	0.0014	10000101	0.0015

Table 4
Fitness function results.

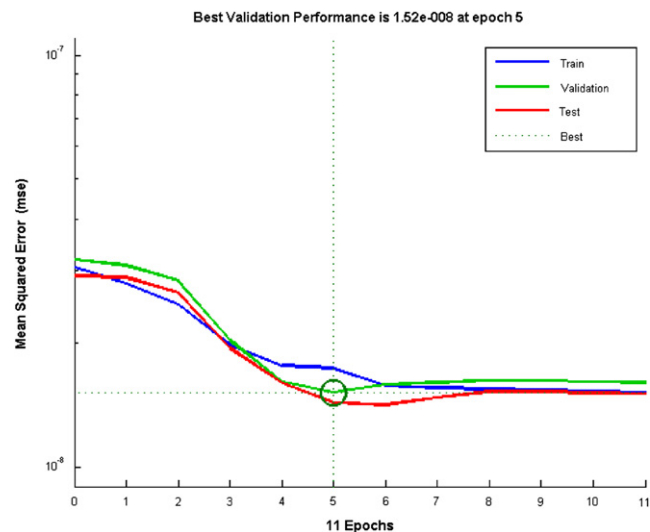| Currency pair	20-9-8-1		20-1-8-1	
	Fitness	Correlation	Fitness	Correlation
In-sample test				
GBP\USD	0.00038	0.8212	0.0034	0.7348
EUR\GBP	0.00026	0.7302	0.00054	0.5413
EUR\USD	0.00039	0.7647	0.00077	0.2776
Out-of-sample test				
GBP\USD	0.00029	0.8479	0.0029	0.8351
EUR\GBP	0.0005	0.8307	0.00047	0.8444
EUR\USD	0.00026	0.88	0.00027	0.8786

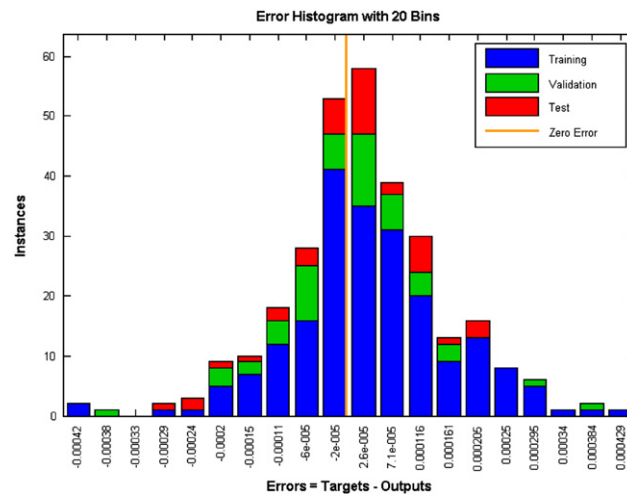**Fig. 11.** The performance of the optimal topology.

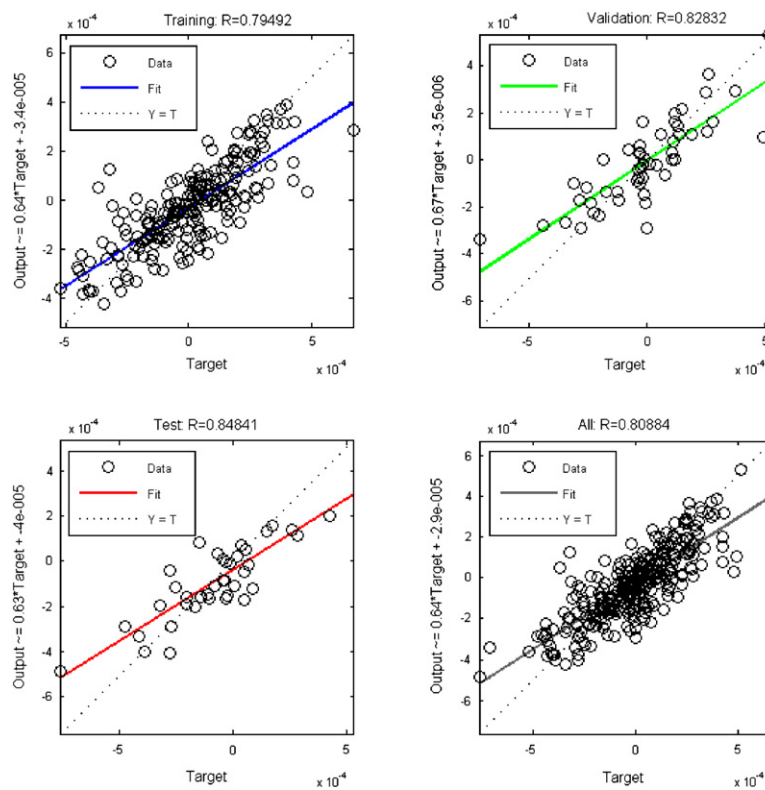**Fig. 12.** The error histogram of the EUR\USD ANN.

Fig. 13. The regression of the EUR\USD ANN.

3.4. Trading results

As commented earlier, while mean square error is an acceptable measure for performance, in practice, the ultimate goal of any testing strategy is to confirm that the forecasting system is to producing positive figures. There are several metrics available for this purpose. The following table contains the average results of the strategies corresponding to the three base currencies.

Table 5 shows that the proposed model produces a quite promising profit with an average annualized gross profit at 27.8%. Meanwhile, an important issue that has not been mentioned so far is the trading cost. While there is no direct transaction fee, the bid–ask spread is a kind of indirect charge. The dataset has shown that the average bid–ask spread during the proposed trading timespan is 3 pips, and the average profit is 36 pips. As the spread affects both trading directions, the average trading

Table 5
Trading results.

Strategy	Winning rate (%)	Hit–miss rate	Annualized gross return (%)	Sharp ratio
GBP as the base currency				
S1	72.5	2.6	26.9	0.7
S2	72.5	2.6	24.24	0.62
S3	72.5	2.6	26.9	0.7
EUR as the base currency				
S1	71.25	2.48	27.7	0.6
S2	75	3	32.33	0.7
S3	71.25	2.48	27.7	0.6
USD as the base currency				
s1	71.25	2.48	27.9	0.54
s2	70	2.33	28.9	0.56
s3	71.25	2.48	27.9	0.54

Table 6
Monte Carlo simulation.

	GBP\USD (%)	EUR\GBP (%)	EUR\USD (%)
Proposed model	72.50	72.50	70
Random walk	42.50	49	51.60
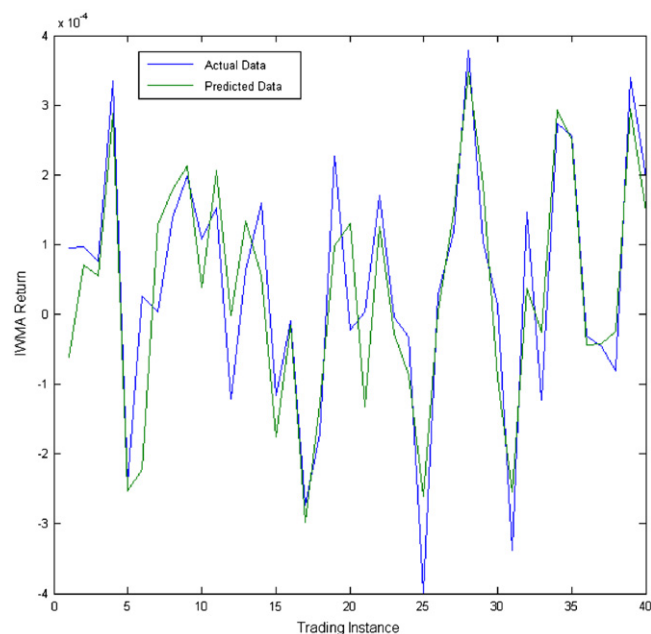
Fig. 14. EUR\USD actual and predicted data between 1-1-2012 and 28-2-2012.

cost is 16%. This gives an annualized net profit at 23.3% which is in many cases better than that of the co-operate earnings although the risk in FOREX trading is substantially higher.

Another reason that makes the FOREX trading more appealing this period is the very low level of interest rates which has a great influence in fixed income products such as government bonds. This is one of the reasons that makes the investment in more risky products more appealing.

A rather reasonable question at this point is what the results of the proposed model would be if conducting Monte Carlo simulation. The next table show exactly those results in comparison with the real data.

As presented in Table 6, the proposed model has a very good performance in comparison with the random walk model. Fig. 14 shows the performance of the model by plotting the actual returns and the predicted returns. The graph depicts the accuracy of the prediction model as the two lines follow the same trend.
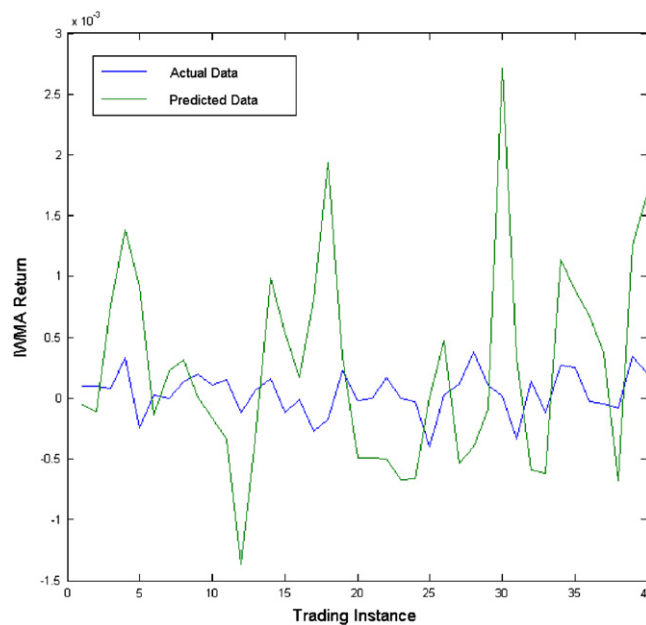
Fig. 15. Monte Carlo simulation.

On the other hand, Fig. 15 shows a completely different image. Here the processed real rates of return are compared with what the model predicted while being trained with random time series that have the same mean value and standard deviation as the actual data. It is obvious that the two lines deviate substantially from each other.

In addition, the model appears to have also a good performance in comparison with other proposed models. Dunis [42] conducted a comparison between technical analysis, econometrics, and an ANN Regression technique as FX forecasting models. The MACD model gave an annualized return at about 4.54% and a winning trade at 30.85% while the ARMA model gave 12.65% and 50.24% respectively. The NNR model outperforms both models by producing an annualized return rate of 19.02% and 48.14% winning rate.

Tilakarante [43] compared the performance of Feed-forward Neural Networks (FNN) and Probabilistic Neural Networks (PNN) models in classifying trade signals into three classes: buy, hold, and sell. Both models produced a correct classification rate of approximately 20% with the PNN model to have shown a slightly better performance. Finally, Subramanian [44] present an approach to autonomous agent design that utilizes the genetic algorithm and genetic programming in order to identify optimal trading strategies. The competing agents recorded an average sharp ratio between 0.33 and 0.85.

4. Conclusions and future work

The main purpose of this paper was to introduce a prediction and decision making model that produces profitable intra-day FOREX transactions. The paper has shown that, despite the highly noisy nature of the tick FOREX data, proper analysis and pre-processing can identify repeatable patterns that provide a source for developing a forecasting model. The developed forecasting model is based on Feed Forward Neural Networks with Back-Propagation architecture. Furthermore, a GA was developed to search for the optimal network topology. Several experiments identified the topology 20-9-8-1 as the one that provides the network with better approximation and generalization. Additionally, the proposed trading strategies were able to produce a promising annualized net profit of 23.3% which makes FOREX algorithmic trading an appealing choice.

It is important though to mention that in general, the developed financial markets are efficient due to the fact that the large number of participants as well as the huge number of transactions tend to push the prices towards their equilibrium. However, although the overall market seems to be efficient at least to its semi-strong form, this particular research, as well as other research mentioned throughout this paper has shown that even in mature and well developed financial markets there are pockets of inefficiencies that can be exploited. Furthermore, once those inefficiencies have been identified, and the deployed methodology is being made available publicly, those pockets of inefficiencies will cease to exist as the exploitation of those inefficiencies will push the prices towards their equilibrium.

Last but not least, Back-propagation is a well implemented technique that has dominated the prediction models over the previous many years. Since the introduction by Vapnik in 1995, many argue that Support Vector Machines (SVM) are more robust and tend to provide more accurate models. Considering the market prediction problem as a classification problem where the question is whether the market goes up or down, an interesting question is what the performance of SVM would be in comparison with a back-propagation model both operating in a highly noisy data environment such as the FX market.

References

[1] L. Ogiela, Advances in cognitive information systems, Cognitive Systems Monographs 17.
[2] C. Giles, S. Lawrence, S. Tsoi, Noisy time series prediction using a recurrent ann, Journal of Machine Learning 44 (2001) 161–183.
[3] E. Fama, Efficient capital markets: a review of theory and empirical work, Journal of Finance 25 (1970) 383–417.
[4] S. Makridakis, C. Wheekwright, R. Hyndman, Forecasting: Method and Applications, John Wiley & Sons, NY, 1998.
[5] M. Azzof, Neural Network Time Series Forecasting of Financial Markets, John Wiley & Sons, NY, 1994.
[6] A. Scabar (Ed.), Financial Trading and the Efficient Market Hypothesis, Vol. 4, ACM, Darlinghurst, Australia, 2002.
[7] A. Hirabayashi, C. Arahna, H. Iba, Optimization of the trading rules in forex using genetic algorithms, online (2009). URL www.dollar.biz.uiovwa.edu.
[8] D. Tulson, S. Tulson, Intelligent financial systems, online (2007). URL www.if5.com.
[9] M. Butler, A. Daniyal (Eds.), Multi-objective Optimization with an Evolutionary Artificial Neural Network for Financial Forecasting, ACM, Mondreal Quebec Canada, 2009.
[10] H. Tay, L. Cao, Application of svm in financial time series, online (2001). URL www.zenithlib.googlecode.com.
[11] L. Yu, S. Wang, K. Lai, An online learning algorithm with adaptive forgetting factors for forward neural networks in financial time series forecasting, online (2007). URL http://citeseerx.ist.psu.edu.
[12] J. Yao, C. Tan, A case study on using neural networks to perform technical forecasting of forex, online (2000). URL http://citeseerx.ist.psu.edu.
[13] A. Enam (Ed.), Optima Artificial Neural Network Topology for Foreign Exchange Forecasting, ACM, NY, 2008.
[14] Y. Yuan, Forecasting the movement direction of exchange rate with polynomial smooth support vector machine, Mathematical and Computer Modeling 57 (2013) 9320944.
[15] J. Matas, Boosting garch and neural networks for the prediction of heteroskedastic time series, Mathematical and Computer Modeling 51 (2010) 256–271.
[16] J. Kamruzzaman, Ann-based Forecasting of Foreign Currency Exchange Rates, ACM, Canberra, Australia, 2004.
[17] H. Zhang, R. Ren, High frequency foreign exchange trading strategies based on genetic algorithms, in: Proc. Second Int Networks Security Wireless Communications and Trusted Computing (NSWCTC) Conf, Vol. 2, 2010, pp. 426–429.
[18] L. Yu, S. Wang, K. Lai, A novel nonlinear ensemble forecasting model incorporating glar & ann for foreign exchange rates, Computers and Operations Research 32 (2004) 2523–2541.
[19] I. Altridge, High Frequency Trading, John Wiley & Sons, NY, 2010.
[20] S. Chante, Beyond Technical Analysis: How to Develop and Implement a Winning Trading System, Wiley, NY, 1997.
[21] B. Vastone, G. Finnie, An empirical methodology for developing stockmarket trading systems using artificial neural networks, Expert Systems and Applications 35.
[22] P. Refenes, Neural Networks in the Capital Markets, Wiley, 1995.
[23] L. Martinez (Ed.), From an Artificial Neural Network to a Stock Market Day Trading System: A Case Study on BM&F Bovespera, IEEE, Atlanta, Georgia, USA, 2009.
[24] P. Tsang, Design and implementation of ann5 for hong kong stock price forecasting, online (2006). URL www.sciencedirect.com.
[25] F. Castiglione, Forecasting price increments using an artificial neural network, Advances in Complex Systems 4 (2001) 45–56.
[26] R. Pardo, Ddesign, Testing, and Optimization of Trading Systems, Wiley, NY, 1992.
[27] J. Kaufman, Trading Systems and Methods, Wiley, NY, 1998.
[28] A. Ranganathan, The levenberg-marquardt algorithm, online (2004). URL www.citeseerx.ist.psu.edu.
[29] H. Gavin, The levenberg-marquardt method for nonlinear least squares curve fitting problems, online (2011). URL www.duke.edu.
[30] K. Levenberg, A method for the solution of certain problems in least squares, Quarterly of Applied Mathematics 2 (1944) 164–168.
[31] D. Marquardt, An algorithm for least squares estimation of nonlinear parameters, SIAM Journal of Applied Mathematics 11 (1962) 431–441.
[32] I. Kaastra, M. Boyd, Designing a neural network for forecasting financial and economic time series, online (1995). URL www.seas.harvard.edu.
[33] D. Baily, D. Thompson, Developing neural network application, AI Experts (1990) 33–44.
[34] J. Holland, Adaptation of Natural and Artificial Systems, University of Michigan Press, Michigan, 1975.
[35] R. Haupt, S. Haupt, Practical Genetic Algorithm, Wiley, New Jersey, 2004.
[36] K. Man, K. Tang, S. Kwong, Genetic Algorithms: Concept and design, Springer-Verlag, Heidelberg, 1999.
[37] E. Kalyvas, Using neural networks and genetic algorithms to predict stock market return, online (2001). URL www.citeseerx.ist.psu.edu.
[38] L. Barolli, E. Spaho, T. Oda, A. Barolli, F. Xhafa, M. Takizawa (Eds.), Performance Evaluation for Different Settings of Crossover and Mutation Rates Considering the Number of Covered Users: A Case Study, ACM, NY, 2011.
[39] L. Trujillo (Ed.), How Many Neurons? A Genetic Programming Answer, ACM, Dublin, Ireland, 2011.
[40] P. Gao, C. Chen, S. Qin, An optimization method of hidden nodes for neural network, in: Proc. Second Int Education Technology and Computer Science (ETCS) Workshop, Vol. 2, 2010, pp. 53–56.
[41] C. Dunis, J. Jalilov, Neural network regression and alternative forecasting techniques for predicting financial variables, online (February 2001). URL www.livjm.ac.uk.
[42] C. Dunis, L. Laws, P. Nai, Applied Quantitative Methods for Trading and Investment, Wiley, NY, 2003.
[43] C. Tilakarante, S. Morris, M. Mammadov, C. Hurst (Eds.), Predicting Stock Market Index Trading Signals Using Neural Networks, Springer-Verlag, Berlin, 2008.
[44] H. Subramanian, S. Ramamoorthy, P. Stone, L. Kuipers (Eds.), Designing Safe, Profitable Automated Stock Trading Agents Using Evolutionary Algorithms, ACM, NY, 2006.