Developing a fully automated algo-trading system

PH.D. DISSERTATION OF

KIA FARHAD, M.Sc.

Supervisor:

Dr. Levendovszky János, D. Sc.

Doctor of the Hungarian Academy of Sciences

Department of Networked Systems and Services
Budapest University of Technology and Economics

Budapest, 2016

„You have to learn the rules of the game. And then you have to play better than anyone else." (Albert Einstein)

Acknowledgements

I am very grateful to my supervisor Dr. Levendovszky Janos and my mentor Dr. Jeney Gabor for the patient guidance, encouragement and advice they have provided throughout my work on this thesis and the research work. I would also like to thank, Robert Sipos, Fogarasi Norbert, Attila Ceffer and Thai Hoc Nguyen for useful discussions and collaboration on some of the problems presented in the theses.

Many thanks also belong to my wife and my entire family who supported me while working on this thesis. It would have been very difficult without their support.

Executive summary

The thesis is concerned with developing fast trading algorithms and portfolio optimization for implementing high frequency trading in real-time.

The first thesis group deals with portfolio optimization with extended objective functions over mean reverting processes. Namely, not only the predictability factor is maximized (leading to a generalized eigenvalue problem) but an analytical expression has been derived for the loss probability (the probability of the value of the portfolio hitting negative value). Based on this formula, a minimum risk portfolio can be selected by minimizing the loss probability. In this way, investment risk can be significantly reduced. The corresponding results have also been extended to Levy processes providing a more general framework for portfolio optimization. In order to perform optimization fast methods are worked out for the model-parameter identification, as well. The methods have been extensively tested on the Forex data base. My related results can be found in [1,2,4].

The second thesis group treats the issue of prediction based trading. Neural based prediction is developed to cope with the bid-ask spread. The optimality is proven analytically, while the performance is tested on historical time series of Forex. The second method can not only be used for predicting the single asset but also to predict the value of the optimal portfolios from the first thesis group. My related results can be found in [3, 5].

As a general conclusion, de results developed in the thesis can pave the way toward less-risk trading and a safer financial world in general.

Development of a fully automated algo-trading system

List of Figures

List of Tables

CHAPTER 1

Introduction

In the advent high speed computation and ever increasing computational power, algorithmic trading has been receiving a considerable interest [6-9]. The main focus of recent research has been to develop real- time algorithms which can cope with portfolio optimization and price estimation within a very small time interval, enabling high frequency, intraday trading. In this way, fast identification of favorable patterns on time series becomes feasible on small time scales, as well, which can give rise to profitable trading where asset prices follow each other in a *sec* or *msec* range.

As a result, the present work deals with the problem of finding the best portfolio for successful trading based on different objective functions, e.g. minimizing the risk or maximizing the predictability by using different stochastic models (e.g. mean revering or Levy processes).

However there is an open issue in this kind of trading which is an entry point and trigger to open a position which I then use prediction-based trading to solve and improve this issue.

1.1 Models and methods used in the research

In order to develop efficient prediction algorithms for financial time series, I used several models and methods which are outlined in this chapter.

For solving the optimal portfolio selection problem I proposed a novel method which aims at minimizing the loss probability based on two different underlying models; (i) mean reverting model[10] (ii) Levy model[11]. The traditional strategies are concerned with optimizing the mean reverting parameter of predictability which lends itself to analytical tractability by solving an eigenvalue problem [12-17]. In this dissertation I assume that the multidimensional asset price vector out of which the portfolio is to be constructed can be modeled by a VAR (1) process. Then portfolio selection can be broken down in two steps: (i) fast model identification based on the previously observed samples of the corresponding asset prices and; (ii) choosing a portfolio vector which minimizes the probability of negative return.

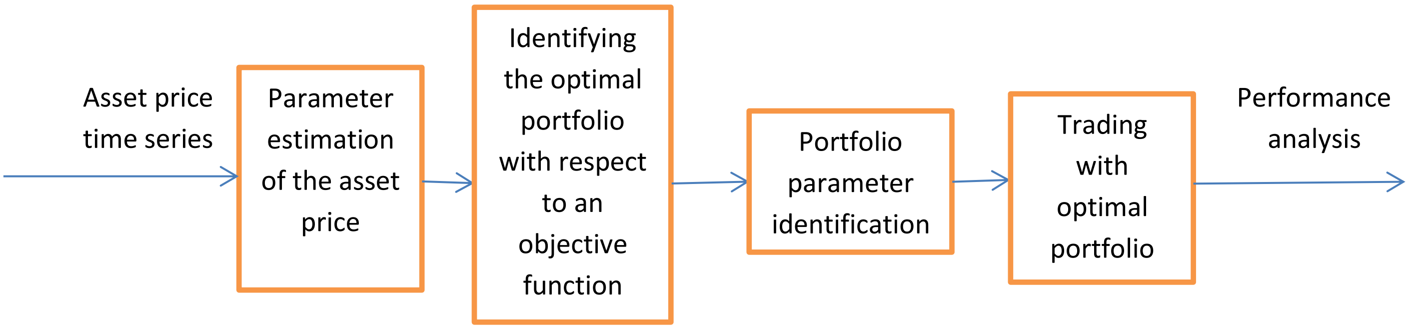

Fig. 1: Overall framework for identifying and trading optimal min-loss portfolios

To apply a prediction based trading strategy I used a suitable nonlinear estimator for predicting the future values of a financial time series provided by a properly trained Feed Forward Neural Network (FFNN) which can capture the characteristics of the conditional expected value. This predicted value can be used directly or it can be used as an estimation of a short term mean of mean reverting model.

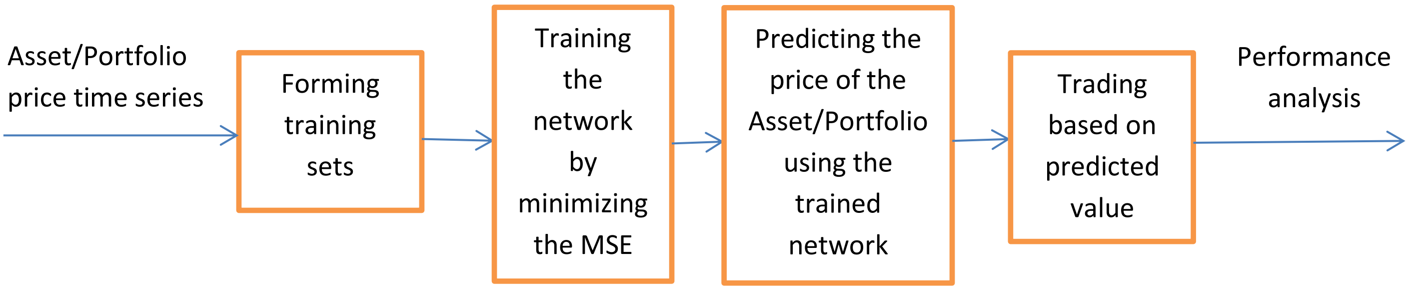

Fig. 2: Overall framework of Prediction-based trading

The implementations are performed on MATLAB and on Metatrader which could give us the opportunity to perform both back-testing and forward-testing and obtaining more promising and reliable results; however these algorithms can also be implemented on faster architectures.

Models used	• VAR(1) model • Ornstein-Uhlenbeck model • Levy Model • Stationary time series
Methods applied	• Modeling(Feed forward neural network) • Identification algorithms(Maximum likelihood estimation, Moore-Penrose pseudo inverse) • Optimization methods (Exhaustive search, greedy search, Simulated annealing) • Learning Algorithms(Back - propagation (BP) which minimizes the mean squared error(MSE))
Validation	Numerical simulations on real historical data from Forex in MATLAB and Mt4

Table 1. Models and methods examined in the dissertation

1.2 Structure of the dissertation

The dissertation has been organized as follows:

- In Chapter 2 I try to find the best portfolio based on the same objective function (minimizing the probability of loss) but I extended the stochastic models of the underlying time series assuming, Levy model, as well.
- In Chapter 3 I focus on estimating the future price of the time series by using an appropriate nonlinear predictor. This predicted value can be used directly or it can be used as an estimation of a short term mean of mean reverting model.

- In Chapter 4 I summarize the main results of the dissertation and show some comparison results regarding the models which have been used and presented in this dissertation.

- Finally in Chapter 5 I present a list of my own publications and a full bibliography for this work.

In the following figure those chapters are listed where novel results (detailed in the thesis booklet) can be found.

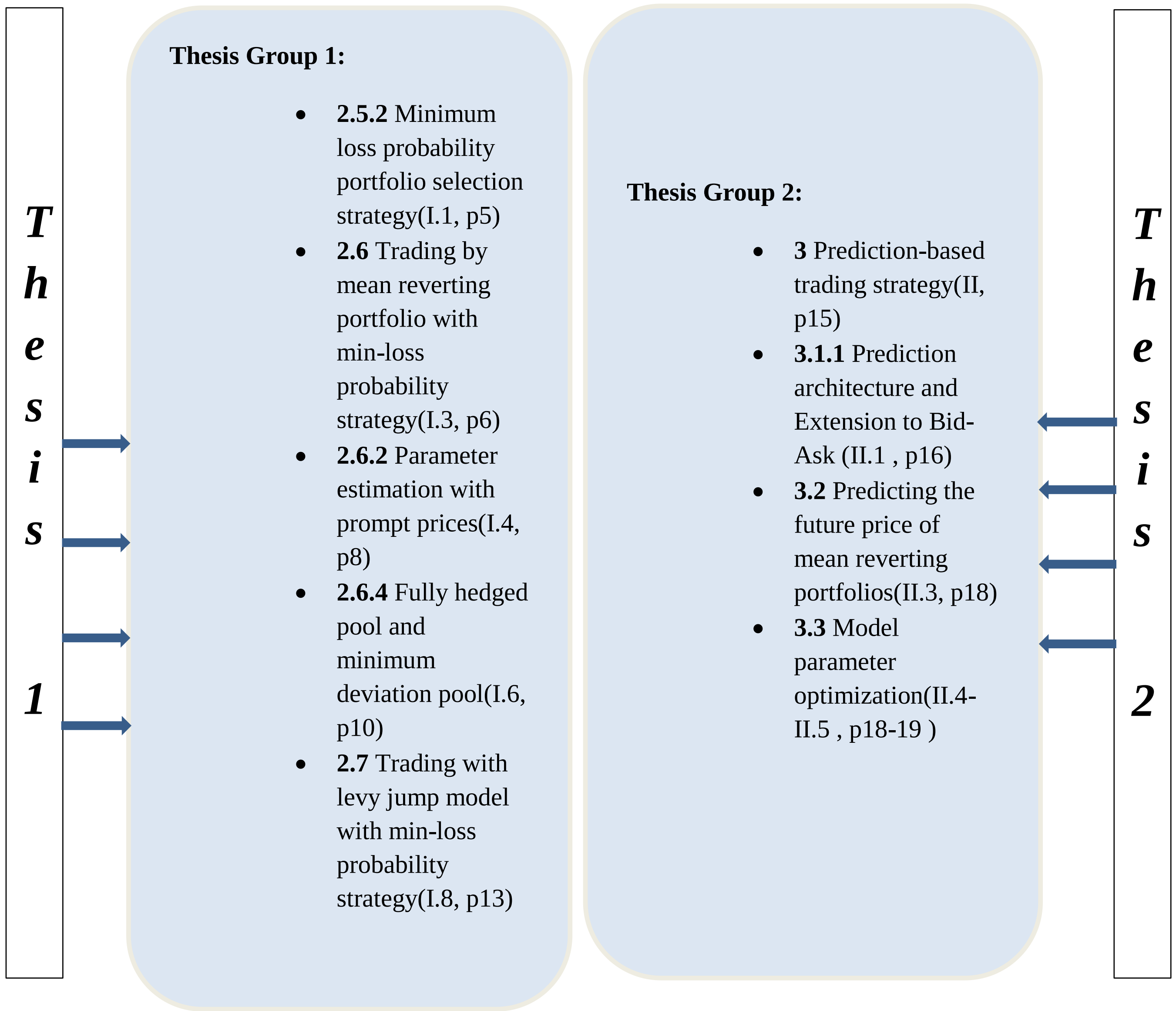

Fig. 3: The structure of the dissertation

CHAPTER 2

Optimal min loss portfolio selection

In this chapter the novel trading strategy and portfolio selection will be introduced which aims at minimizing the loss probability based on identifying mean reverting portfolios or Levy jump model portfolios. After observing historical data for parameter identification, the portfolio selection is performed by minimizing the probability of negative return (loss). Furthermore I developed a new method for calculating the mean and mean reversion parameter as an alternative for traditional way of calculating these parameters.

2.1 Introduction

Portfolio optimization was first investigated by Markowitz in the context of diversification to minimize the associated risk and maximize predictability[18]. Since the first results, many papers have been dealing with portfolio optimization [12, 13, 17, 19], e.g. one of the usual approaches is finding the portfolio which exhibits predictability and minimal risk [18, 20-22]. Another approach is to identify mean reverting portfolios, where trading actions (e.g. buying or selling) are launched when being out of the mean and complementary actions are taken (e.g. selling or buying) after reverting to the mean. The traditional strategies are concerned with optimizing the mean reverting parameter of predictability which lends itself to analytical tractability by solving an eigenvalue problem[12]. However, these approaches fail to provide high profits in the presence of bid ask spread or using a large number of assets due to the transactional cost. Therefore, one has to use more sophisticated objective functions, models and trading strategies.

My contribution to this topic is to find the best portfolio based on the same objective function (minimizing the probability of loss) but I extended the stochastic models of the underlying time series assuming, Levy model, as well. So I generalize the previous idea by assuming that the portfolio price follows a Lévy process [23] which is the more general model than the traditional Ornstein-Uhlenbeck process[10]. Based on this assumption, after identifying the parameters of the underlying Lévy process, such portfolio is selected which minimize the probability of loss.

2.2 System model

In this section we describe the system model, which is used in this dissertation.

$\mathbf{x}$ is always a column vector, and $\mathbf{x}'$ is the transposed version of $\mathbf{x}$, i.e. $\mathbf{x}'$ is a row vector. Vector $\mathbf{0} = [00...0]'$ obviously refers to the zero vector, $\mathbf{1}$ is a column vector with full of ones. While min and max functions are piecewise minimization/maximization functions. We will usually denote the empirical average by μ. Note that the average is computed over a time window with width T and looking backward (over the interval $(-T, 0)$) defined as follows.

$$\mu = \frac{1}{T}\sum_{t=1}^{T} P_t \tag{1}$$

In stock markets the price of the assets is the function of the time t. At a given time t, the i_{th} asset can be sold at a price $s_i(t)$ and the same asset can be bought at a price $b_i(t)$, which is for sure $b_i(t) > s_i(t)$. The difference between $b_i(t)$ and $s_i(t)$ is usually referred to as bid-ask spread. Thus, each asset can be described as a time varying two-dimensional vector $[b_i(t), s_i(t)]$ .Assuming a great number of assets, we can describe them as

$$(\mathbf{b}(t), \mathbf{s}(t)) \;,$$

where $\mathbf{b}(t) = [b_1(t)b_2(t)...b_K(t)]'$ and $\mathbf{s}(t) = [s_1(t)s_2(t)...s_K(t)]'$. The value of the assets owned by an investor depends on the number of assets $\mathbf{n}$ they hold (which is a sparse vector with discrete usually integer – components). Vector $\mathbf{n}$ represents the portfolio. It is important to mention that it is possible to have negative components in this vector which refers to short position. Additionally, the cash they have, which is a scalar, is denoted by $c(t)$. The investor is described by these two parameters as

$$(\mathbf{n}, c(t)).$$

The prompt value of assets depends on the sell price $\mathbf{s}(t)$ if the amount of assets are positive (so-called long position), or the buy price $\mathbf{b}(t)$ if the amount of assets are negative (so-called short position). That is, the value of the portfolio held by the investor is given as

$$P(t) = max(\mathbf{0}', \mathbf{n}')\mathbf{s}(t) + min(\mathbf{0}', \mathbf{n}')\mathbf{b}(t)\,. \tag{2}$$

There are several other ways to describe the same equation. One is to introduce $\mathbf{n}^{+} = max(\mathbf{0},\mathbf{n})$, which is the copy of vector $\mathbf{n}$, but only with the positive elements are kept, the negative components are replaced by zeros. This new vector helps in describing the right part of the equation as $\mathbf{n}-\mathbf{n}^{+}$,thus (2) finally becomes

$$P(t) \;=\; (\mathbf{n}^{+})'(\mathbf{s}(t)-\mathbf{b}(t))+\mathbf{n}'\mathbf{b}(t) \tag{3}$$

The wealth (the value of all properties) of the investor comprises the value of the portfolio and the cash, that is

$$w(t) \;= P(t)+c(t). \tag{4}$$

This expression describes the amount of money the investor could get immediately, if they decide to quit – assuming that the number of assets is small enough to have the same buy/sell price.

Now that we have described the static nature, the dynamic nature of wealth is also investigated. At time instance t , the investor make a decision to change the number of assets they hold by $\mathbf{v}$, where this vector can contain positive and negative integers, based on the type of transaction. Positive value means that the investor buys the corresponding asset, negative value refers to selling. For technical reasons, in the following we assume that at a given time t the investor either buys or sells, but not both. That is, $\mathbf{v}(t)$ contains either positive, or negative components, but it cannot have negative and positive numbers at the same time. The number of assets owned by the investor in the next time instant $t+dt$ is given as

$$\mathbf{n}(t+dt) \;= \mathbf{n}+\mathbf{v} \tag{5}$$

and the cash of the inventor is described as

$$c(t+dt) \;= \begin{Bmatrix} c(t)-\mathbf{vs}(t) if \;\; the\ investor\ sells \\ c(t)-\mathbf{vb}(t) if \;\; the\ investor\ buys \end{Bmatrix}. \tag{6}$$

Note that although in the last term there is a subtraction, the negative components of $\mathbf{v}$ yields an increase. Here, we have neglected the transaction cost (commission or fee). However, it can be represented in the bid-ask spread: sell and buy prices should be shifted with the corresponding transaction cost. The investor follows a good strategy, if their wealth increases over time, that is, for most of the purchases $v_i(\tau)$, there exist a T_i , where $w_i(\tau+T_i) > w_i(\tau)$. The aim of all computational finance is to find a strategy which assures $w_i(\tau+T_i) > w_i(\tau)$ with a relatively high probability.

2.3 Asset price and portfolio models based on different market hypotheses

In financial markets, the price of a given entity is hard to describe. However, there are some models in the literature which assumes some properties lying in the background. With these properties it is possible to see some clear rules of the price movements.

Four models are introduced here: the Samuelson Model from the 1950's [24], the Mean Reverting Model [16]from the 1990's, its linear discrete counterpart and the Lévy Jump Model [23]from the 1990's. All of them try to describe (2).

2.3.1 Samuelson model

Samuelson [25] proposed a very simple model for the modeling of asset's price movements. His model is given as

$$dP(t) = P(t)(\rho dt + \sigma dW(t)) \tag{7}$$

where $\rho, \sigma \in \mathbb{R}, \sigma > 0$, so ρ could be positive or negative, and $W(t)$ is a Wiener process. Please note that negative σ yields the same equation due to the symmetric nature of the Wiener process. Positive ρ represents exponentially growing price $P(t)$, negative ρ shows exponentially diminishing value $P(t) \to 0$. The differential equation of (7) has the solution of

$$P(t) = P(0) exp\left(\rho t + \frac{\sigma^2}{2} t + \sigma W(t) \right) \tag{8}$$

which contains many terms, all of them in the exponent. The model depends on t and since the time is also in the exponent, it shows exponential growth ($exp(\rho + \frac{\sigma^2}{2})$) is powered to t). If $\rho + \frac{\sigma^2}{2}$ is close to zero, then this assumption is quite acceptable in the case of traditional assets (bonds, stocks, etc.). In other cases, if $\rho = -\frac{\sigma^2}{2}$, the "no drift case" could be also described. Finally, $W(t)$ represents the stochastic nature in the model, however since it is also located in the exponent, the motion is distorted: positive movements are amplified, negative movements are suppressed. This effect might invite some criticism of the model.

It is important to emphasize that the Samuelson model introduces a drift term in (8) which seems to be realistic, if one investigates the statistical behavior of stock markets. The mean of the price tends to

$$\mathbb{E}_W\{P(t)\} = P(0)\left(exp\left(\rho + \frac{\sigma^2}{2} \right) \right)^t . \tag{9}$$

That is, as time t tends to infinity, if $\rho > -\frac{\sigma^2}{2}$, then the price of the asset tends to infinity, if $\rho < -\frac{\sigma^2}{2}$, then the price of the asset tends to zero. If $\rho = -\frac{\sigma^2}{2}$, $P(t)$ keeps it's starting value $P(0)$ on average. However, the Wiener motion $W(t)$ heavily influences the real values of $P(t)$.

2.3.2 Mean reverting portfolio model

Here, mean reverting portfolios are introduced, especially sparse portfolios (where only a few assets are used inside the portfolio). Mean reverting portfolios can be described by the Ornstein-Uhlenbeck [10]stochastic differential equation as

$$dP(t) = \lambda(\mu - P(t))dt + \sigma dW(t) . \tag{10}$$

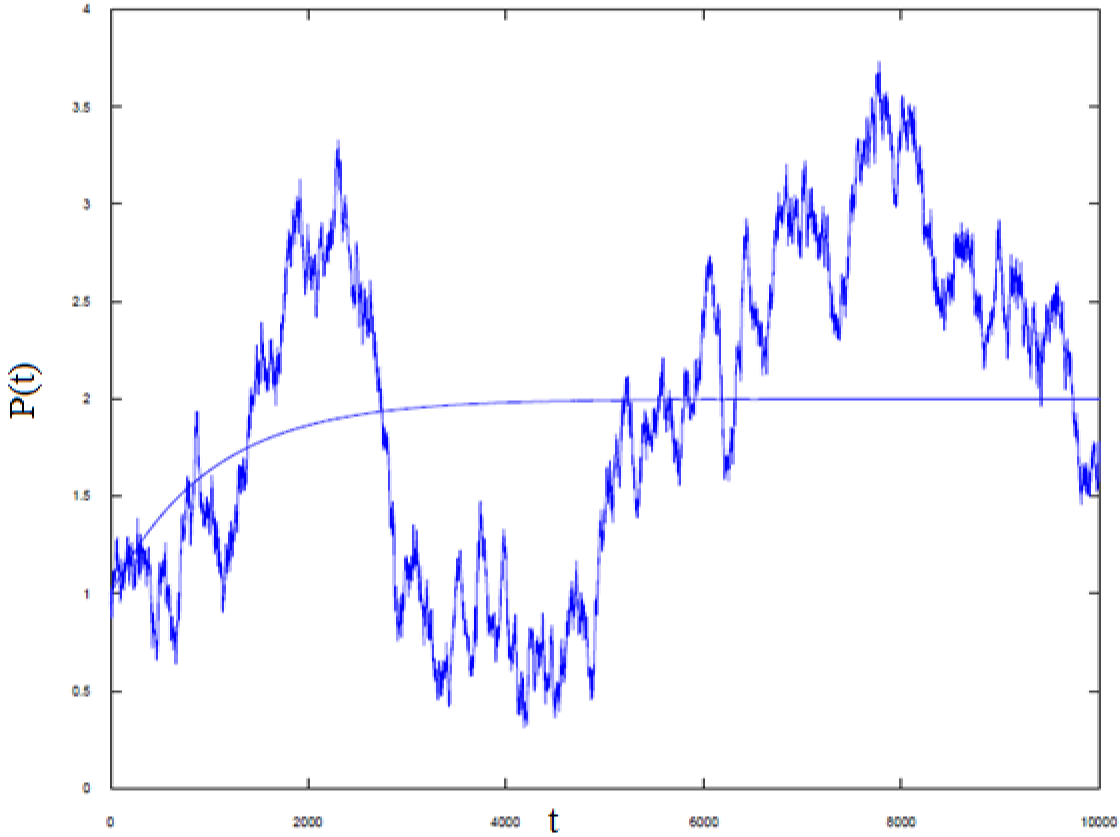

Fig. 4: An example of mean reverting portfolios: $P(0)=1, \mu=2, \lambda=1$ and $\sigma=1$.

The fluctuating curve depicts a realization of the stochastic process, the smooth curve shows the expected value, i. e. (12) where $P(t)$ is the portfolio, defined in (2), $\lambda, \mu, \sigma \in \mathbb{R}, \sigma > 0$ are parameters, which can be estimated based on real transaction, $W(t)$ is a Wiener process. Without the loss of generality we assume that σ is also positive (note that negative σ can be replaced by $-\sigma$). The solution of (10) by using the ito-doeblin [26] formula is given as

$$P(t) = P(0)e^{-\lambda t} + \mu(1-e^{-\lambda t}) + \int_{s=0}^{t} \sigma e^{-\lambda(t-s)} dW(s) . \tag{11}$$

An example for the mean reverting portfolios is given in Figure 4. Note that $P(0), \mu$ and $P(t)$ could be either positive, or negative. Usually λ is positive. Negative λ is also possible; however it has no practical importance. We will restrict ourselves to the positive λ case. (11) has very nice properties. For example, its mean is given as

$$\mathbb{E}_W\{P(t)\} = P(0)e^{-\lambda t} + \mu(1-e^{-\lambda t}) \tag{12}$$

and as time tends to infinity, $P(t)$ yields a Gaussian random variable

$$\lim_{t\to\infty} P(t) \sim N\left(\mu, \frac{\sigma}{\sqrt{2\lambda}}\right). \tag{13}$$

That is, if λ is large, μ of (12) is quickly amplified and the deviation is small, see (13). The portfolio quickly tends to μ , and thus, the portfolio can be easily predicted. That is, technical investors look for mean reverting portfolios with high λ value. Vector $\mathbf{n}_{opt}$ should be optimized such that the corresponding λ value is maximal.

However, maximizing the λ parameter is not sufficient under most circumstances. Let us assume that (λ, μ) parameter pairs could be estimated and they are known for the investor. If $P(0) \approx \mu$, it means that λ could be arbitrary large, the expected profit equals zero. Thus, purely maximizing the λ parameter is not sufficient. In section **2.5.2** we provide some alternative trading strategies.

For the sake of later investigation we describe the infinitesimal step (Δt) behavior for (11):

$$P(t+\Delta t) \;= P(t)e^{-\lambda\Delta t} + \mu(1-e^{-\lambda\Delta t}) + \int_{s=t}^{t+\Delta t} \sigma e^{-\lambda(t+\Delta t-s)} dW(s), \tag{14}$$

which could be simply written as

$$P(t+\Delta t) \;= a.P(t) + b + z \tag{15}$$

where

$$a = e^{-\lambda\Delta t}, \tag{16}$$

$$b = \mu(1-e^{-\lambda\Delta t}), \tag{17}$$

and

$$z = \int_{s=t}^{t+\Delta t} \sigma e^{-\lambda(t+\Delta t-s)} dW(s). \tag{18}$$

This latter term is a zero mean Gaussian noise with variance $\sigma^{2\frac{1-e^{-2\lambda\Delta t}}{2\lambda}}$. As shown in (15), the original exponential function is transformed to the simple linear model.

The only difference between (11) and (15) is that continuous t could be substituted into the first one: the latter one could handle only discrete time series. The linear model will help us to use simple equations later.

2.3.3 Lévy jump model

As a generalization of the Samuelson model[25], one can describe financial markets 'portfolio (and also security) prices as Lévy processes [23]. Lévy process $X(t)$ is characterized by the following properties:

• The paths of $X(t)$ are right continuous, with left limits, i.e.

$$\forall\gamma,\ \lim_{\varepsilon\to 0+} \mathbb{P}\left\{X(t+\varepsilon)-X(t)< \ \gamma\right\}=0\ .$$

• The increments have identical distribution, i. e., for $0\le s\le t, X(t)-X(s)$ is equal in distribution to $X(t-s)$.

• $X(t)$ has independent increments, that is, for $0\le s\le t, X(t)-X(s)$ is independent of $\{X(u): u\le s\}$.

The simplest Lévy process is the linear drift, which is a deterministic process, of course. Brownian motion is the only (non-deterministic) Lévy process with continuous sample paths. Other examples of Lévy processes are the Poisson and compound Poisson processes [27]. Notice that the sum of two Lévy processes is again a Lévy process; thus a Brownian motion with linear drift is also a Lévy Process.

We will focus on the Lévy jump process, which is constructed from the linear drift and a Poisson point process, thus it can be given as

$$P(t) \ = P(0)+\delta t+\sum_{k=1}^{N_t} J_k \tag{19}$$

where δ is the drift parameter, $\{N_t\}$ is a Poisson point process, with mean $E\{N_t\} = \Lambda t$, and J_k is a sequence of independent, identically distributed discrete random numbers with zero mean, i. e. $\mathbb{E}\{J_k\} = 0$. In stock markets, J_k represents the jumps between different ticks in subsequent transactions. As long as ticks are taken from a discrete set, the J_k jumps must be discrete too. Unbalanced portfolios (tending down or up) are described by the linear drift term. In balanced portfolios $\delta = 0$.

2.4 Discrete time model of asset price movements

In the previous discussion we have investigated the most known portfolio models with respect to continuous time; However in present section we take into account that the asset price vector is a discrete time series modeled by a VAR (1) process.

Here we will focus on the model of Box and Tiao [28], which assumes that sell and buy prices, can be described in the following way

$$\mathbf{s}(t+\Delta t) = \mathbf{A}\mathbf{s}(t) + \mathbf{z}(t) \tag{20}$$

where $\mathbf{z}(t)$ is Gaussian ; the covariance matrix of which

$$\mathbf{K} = \mathbb{E}_t\left\{\mathbf{z}(t)\mathbf{z}(t)'\right\}. \tag{21}$$

Matrix $\mathbf{A}$ is the transition matrix of the sell prices. Δt is a pre-defined interval which later gets optimized(in day-trading, it could be one hour, in other cases it could be one day, etc.).

In the following discussion we will restrict our investigation to sell prices for the sake of obtaining simpler equations. In other words we assume zero bid-ask spread $\mathbf{s}(t) = \mathbf{b}(t)$ .Now we will further analyze (20). Since we have $\mathbf{n}$ number of assets, the value of the portfolio can be computed by multiplying (20) with $\mathbf{n}$ from the left, it yields

$$\mathbf{n}'\mathbf{s}(t+\Delta t) = \mathbf{n}'\mathbf{A}\mathbf{s}(t) + \mathbf{n}'\mathbf{z}(t) \tag{22}$$

which is three scalar terms. On the left, we have the future value of the portfolio ($P(t+\Delta t)$) assuming that there is no trade in the interval $(t, t+\Delta t)$ (that is, $\mathbf{n}(t+\Delta t) = \mathbf{n}(t)$). On the right, the first term looks very similar to the portfolio value at time $t (P(t) = \mathbf{n}'\mathbf{s}(t))$, however it is not the same (please note that there is matrix $\mathbf{A}$ between the terms).

2.5 Portfolio selection strategies

In this section we sum up some of the trading strategies which could be followed if one of the models of the previous section is valid.

2.5.1 Maximum λ portfolio selection

We will use (22), assuming mean reverting portfolio model in the background. If the variance of both the left hand side, and the first term on the right hand side is computed in (22), and these two are compared, we can make some assumptions on the mean reverting nature of the portfolio. Let us do it first and then describe this property.

As long as portfolio is kept constant during the investigated time window the following two variance parameters are introduced

$$\sigma^2(t) = \mathbb{E}\left\{\mathbf{n}'\mathbf{A}\mathbf{s}(t)\mathbf{s}'(t)\mathbf{A}'\mathbf{n}\right\} - \mathbb{E}\left\{\mathbf{n}'\mathbf{A}\mathbf{s}(t)\right\}^2, \tag{23}$$

$$\sigma^2(t+\Delta t) = \mathbb{E}\left\{\mathbf{n}'\mathbf{s}(t+\Delta t)\mathbf{s}'(t+\Delta t)\mathbf{n}\right\} - E\left\{\mathbf{n}'\mathbf{s}(t+\Delta t)\right\}^2, \tag{24}$$

$$\mathbf{G} = \mathbb{E}\left\{\mathbf{s}(t)\mathbf{s}'(t)\right\}. \tag{25}$$

If $\sigma^2(t)$ is smaller than $\sigma^2(t+\Delta t)$, then the portfolio has smaller variance at time t than at time $t+\Delta t$.

This effect is quite natural since the noise always adds some variance to the price. However, we can say in general that the greater $\sigma^2(t)/\sigma^2(t+\Delta t)$, the smaller the effect of the noise in $\mathbf{s}(t+\Delta t)$. The smaller the effect of the noise, the more accurately we can estimate the portfolio's value. Let us derive this fraction in more details:

$$\frac{\sigma^2(t)}{\sigma^2(t+\Delta t)} = \frac{\mathbb{E}\left\{\mathbf{n}'\mathbf{A}\mathbf{s}(t)\mathbf{s}'(t)\mathbf{A}'\mathbf{n}\right\} - \mathbb{E}\left\{\mathbf{n}'\mathbf{A}\mathbf{s}(t)\right\}^2}{\mathbb{E}\left\{\mathbf{n}'\mathbf{s}(t+\Delta t)\mathbf{s}'(t+\Delta t)\mathbf{n}\right\} - E\left\{\mathbf{n}'\mathbf{s}(t+\Delta t)\right\}^2}. \tag{26}$$

Getting the expectation operator into the appropriate places, and substituting $\mathbb{E}\left\{\mathbf{s}(t)\mathbf{s}'(t)\right\}$ and $\mathbb{E}\left\{\mathbf{s}(t+\Delta t)\mathbf{s}'(t+\Delta t)\right\}$ by $\mathbf{G}$ one gets

$$\frac{\sigma^2(t)}{\sigma^2(t+\Delta t)} = \frac{\mathbf{n}'\mathbf{AGA}'\mathbf{n} - \left(\mathbf{n}'\mathbf{A}\mathbb{E}\left\{\mathbf{s}(t)\right\}\right)^2}{\mathbf{n}'\mathbf{Gn} - \left(\mathbf{n}'E\left\{\mathbf{s}(t+\Delta t)\right\}\right)^2}. \tag{27}$$

For the sake of simplicity and without losing generality, we assume in the sequel that both $\mathbb{E}\{\mathbf{s}(t)\}$ and $\mathbb{E}\{\mathbf{s}(t+\Delta t)\}$ equals zero.

$$\frac{\sigma^2(t)}{\sigma^2(t+\Delta t)} = \frac{\mathbf{n}'\mathbf{AGA}'\mathbf{n}}{\mathbf{n}'\mathbf{Gn}}. \tag{28}$$

As stated above, the optimal portfolio (which is mean reverting) is described by a vector $\mathbf{n}$ which maximizes this expression the so-called predictability, so the optimal portfolio can be described as

$$\mathbf{n}_{opt} = \arg\max_{\mathbf{x}} \frac{\sigma^2(t)}{\sigma^2(t+\Delta t)} = \arg\max_{\mathbf{x}} \frac{\mathbf{x}'\mathbf{AGA}'\mathbf{x}}{\mathbf{x}'\mathbf{Gx}}. \tag{29}$$

Solving (29) is equivalent with finding the largest eigenvalue of $det(\mathbf{AGA}' - \lambda\mathbf{G}) = 0$ .In (29) we substitute $x = \mathbf{G}^{-1/2}\mathbf{z}$ and thus it turns into

$$\mathbf{n}_{opt} = \mathbf{G}^{-1/2} \arg\max_{\mathbf{z}} \frac{\mathbf{z}'\left(\mathbf{G}^{-1/2}\right)'\mathbf{AGA}'\mathbf{G}^{-1/2}\mathbf{z}}{\mathbf{z}'\mathbf{z}}. \tag{30}$$

It can be solved e. g. with Oja's algorithm [29].

2.5.2 Minimum loss probability portfolio selection strategy

The investor wants to find a portfolio which minimizes the probability of loss. That is, the event of loss should have minimum probability.

We try to find a good explanation for the loss. If the investor decided to quit (close the portfolio) at time $t=0$, then (s) he would earn $P(0)$ at $t=0$. Let us assume that $P(0)$ is positive (there is no need to spend money for the close). The opposite case could be easily constructed. It would be possible to put this cash into a bank account and earn interest rate r for unit time without risk. After time t the investor could withdraw $P(0)(1+r)^t$ from the bank account. It is natural that the investor wants to see at least the same value for the portfolio (s) he has at time t , i.e.

$$P(t) > P(0)(1+r)^t, \tag{31}$$

should hold for all t . When the inequality does not hold, the investor realizes that a bank account would have been a better choice – this is the event, when we can speak about loss.

Obviously (31) inequality holds with a given probability. The aim for the investors is to maximize this probability or – similarly – minimize the probability of loss, i. e. the probability of the event that the price of the portfolio is under the minimum expected profit

$$\mathbb{P}\left\{P(t) < P(0)(1+r)^t\right\}. \tag{32}$$

Accordingly choose the best strategy such that (32) is minimized. The loss region is shown in Figure 5 with color blue. Although many people regard the profit as $P(t)-P(0)$, it is not right. In Figure 5, it is also depicted that the real profit is the difference between the actual price $P(t)$ and the exponential curve $P(0)(1+r)^t$. The reader should note that the negative $P(0)$ (having with a short position) is more attractive, since in the long run the loss region diminishes (tends to $-\infty$). Independently of the underlying portfolio, if the investor starts a short position and waits a sufficiently long time, the loss region becomes so small that it is almost impossible to have loss. However, please note that short positions are admitted only for limited time. That is, after a given period the investor is required to close the short positions. For instance, in stock markets, the rules usually admit short positions only inside the same day, i. e. at the end of the day they are automatically closed.

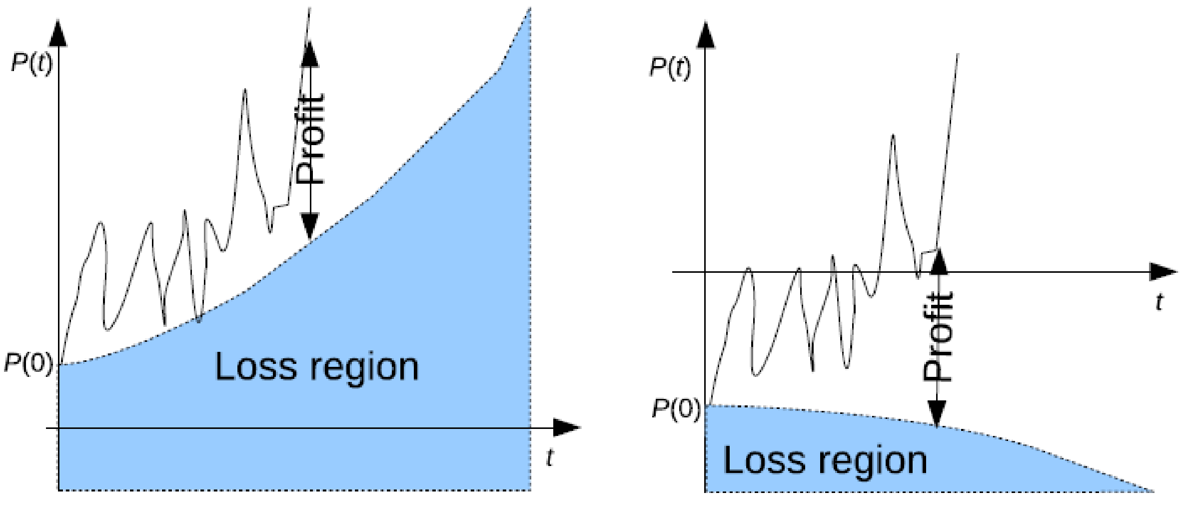

Fig. 5: The loss region for positive and negative P (0) values

Independently of the underlying model, the investor must define (or obtain) a time frame t which limits the portfolio's holding time and an expected return r .These two parameters play essential roles in (32), as will be highlighted later.

2.6 Trading by mean reverting portfolio with min-loss probability strategy

In this section we assume that the portfolio price constructed from VAR (1) and asset price vector will follow a mean reverting process described by the stochastic differential equation (11).

We start from (11), where the Wiener process $W(s)$ is included and we want to find the parameters which yield the minimum probability for the loss (compared to the minimum expected profit). We assume that parameters μ, λ could be derived or estimated for all possible portfolios.

Without the loss of generality we assume that the investor opens (either short, or long) positions at $t = 0$ ($\mathbf{n}$ is set) at the price of $P(0)$) and listens the price movements, i. e. $P(t)$. Note that for the sake of generality, the μ and $P(0)$ parameters could be either positive, or negative, they could also have opposite signs. Negative $P(0)$ means: opening a long position yields cash, opening short position costs money. Similarly, negative $P(t)$ means: closing the long position needs cash, closing a short position results in cash.

It is clear that if $\mu > P(0)$, then the investor should take a long position, independently of the sign of $P(0)$: it is expected that the price of the portfolio will go up, and only thus long position yields profit. Otherwise, if $\mu < P(0)$, the investor should take a short position, expecting positive profit independently of the sign of $P(0)$. If $\mu \approx P(0)$ then the investor should not take any position. If there is an open position, the investor can either win, or loose, depending on the $P(t)$ value at the time of closing its position.

Depending on the signs of μ, $P(0)$ and their relation, six different cases could happen. These are depicted in Figure 6. In the odd cases (left hand side column), μ is greater than $P(0)$, long position should be taken. On the contrary, even cases (right hand side column), μ is smaller than $P(0)$, thus short positions should be opened at $t = 0$. Please note that Case 1 can be considered as minus one times the portfolio of Case 6 (taking a long position in a positive portfolio is the same as taking a short position in the negative portfolio). Similarly, Case 2 can be transformed into Case 5. Case 3 and Case 4 could be also replaced by each other.

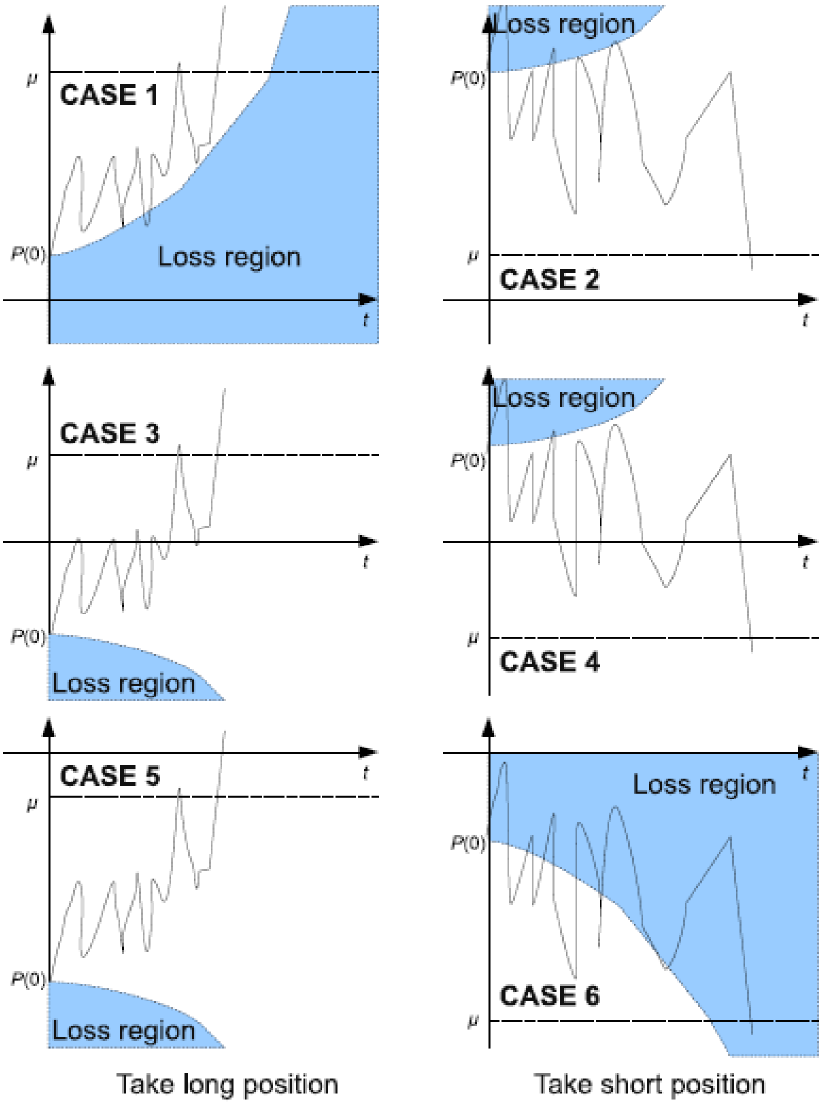

Fig. 6: Six different cases for $P(0)$ and μ values.

Finally, we can say that one column describes all possible events. First of all, $P(t)$ must be compared against $P(0)(1+r)^t$, that is, holding the portfolio for a t period is expected to yield interest also, which is represented in this formula. The larger r , the larger the interest and the interest grows exponentially with time t . When long position was opened at $t=0$ $(\mu > P(0))$, and then $P(t) < P(0)(1+r)^t$ describes the situation of loss, i. e. the prompt value of the portfolio is smaller than the expected price. On the other hand, when short position was opened at time $t=0$ $(\mu < P(0))$

, and then $P(t) > P(0)(1+r)^t$ describes the event of loss. Figure 6 shows the region of loss in blue color.

The probability of loss (P_L) could be expressed as the probability that the portfolio, which we have bought is worth less than a financial instrument with the expected return r for the given time t , i.e.

$$P_L = \begin{cases} \mathbb{P}\left\{P(t) < (1+r)^t P(0)\right\}, & \text{if } \mu > \text{P(0), long position is taken,} \\ \mathbb{P}\left\{P(t) > (1+r)^t P(0)\right\}, & \text{if } \mu < \text{P(0), short position is taken.} \end{cases} \tag{33}$$

For the sake of simplicity, we continue with one equation only, long position ($\mu > P(0)$), and later we alter it to include the short positions as well. Thus we start with

$$P_L = \mathbb{P}\left\{P(t) < (1+r)^t P(0)\right\}. \tag{34}$$

In mean reverting portfolios, $P(t)$ must be substituted from (11) into (34). Thus one gets:

$$P_L = \mathbb{P}\left\{(\mu - P(0))\left(P(0)e^{-\lambda t} + \mu(1-e^{-\lambda t}) + \frac{\sigma}{\sqrt{2\lambda}}\nu - P(0)(1+r)^t\right) < 0\right\}, \tag{35}$$

where ν is a standard Gaussian process $\nu \sim N(0,1)$. Rewriting it one gets:

$$P_L = \mathbb{P}\left\{\nu < \frac{\sqrt{2\lambda}}{\sigma\left(1-e^{-2\lambda t}\right)}\left(P(0)\left((1+r)^t - e^{-\lambda t}\right) - \mu\left(1-e^{-\lambda t}\right)\right)\right\}, \tag{36}$$

which is easily computable, since ν follows a standard normal distribution. That is, using the error function (erf) we can express the above probability. However, we should be careful: $P(0)$ could be either positive, or negative. In the latter case, multiplying by it changes the inequality. However, in both cases, the probability can be measured by the same error function as

$$P_L = \frac{1}{2} + \frac{1}{2} erf\left(\frac{P(0)\sqrt{\lambda}}{\sigma\left(1-e^{-2\lambda t}\right)}\left((1+r)^t - e^{-\lambda t} - \frac{\mu}{P(0)}(1-e^{-\lambda t})\right)\right) \tag{37}$$

where

$$erf(x) = \frac{2}{\sqrt{\Pi}} \int_0^x e^{-t^2} dt \; . \qquad (38)$$

Covering the short position, one gets

$$P_L = \frac{1}{2} + \frac{1}{2}\operatorname{sgn}(\mu - P(0)) erf\left(\frac{P(0)\sqrt{\lambda}}{\sigma\left(1-e^{-2\lambda t}\right)} \left((1+r)^t - e^{-\lambda t} - \frac{\mu}{P(0)}(1-e^{-\lambda t}) \right) \right). \qquad (39)$$

The results show us how to derive the probability of loss, if all the parameters of (39) is known ($\lambda, \sigma, r, t, \mu, P(0)$ must be determined for the calculation). Note that for $t = 0$ the loss probability (39) equals exactly 50% on a very tiny time scale:

$$\lim_{t\to 0+} P_L = \frac{1}{2} + \frac{1}{2}\operatorname{sgn}(\mu - P(0)) erf\left(\frac{P(0)\sqrt{\lambda}}{\sigma\left(1-e^{-2\lambda t}\right)} \left(\overbrace{(1+r)^t}^{1} - e^{-\lambda t} - \frac{\mu}{P(0)}(1-e^{-\lambda t}) \right) \right)$$

$$= \frac{1}{2} + \frac{1}{2}\operatorname{sgn}(\mu - P(0)) erf\left(\frac{\sqrt{\lambda}}{\sigma} \overbrace{\sqrt{\frac{1-e^{-\lambda t}}{1+e^{-\lambda t}}}}^{0} (P(0) - \mu) \right) = \frac{1}{2}.$$

It can easily be accepted, since we have Brownian motion in the system and the probabilities of moving up, or down are equally 50% in a very tiny time scale. As t tends to infinity, the probability of loss becomes one, or zero, depending on the relation between $P(0)$ and μ :

$$\lim_{t\to\infty} P_L = \frac{1}{2} + \frac{1}{2}\operatorname{sgn}(\mu - P(0)) erf\left(\frac{P(0)\sqrt{\lambda}\left(\overbrace{(1+r)^t}^{\infty} - \overbrace{e^{-\lambda t}}^{0} - \frac{\mu}{P(0)}(1-\overbrace{e^{-\lambda t}}^{0}) \right)}{\sigma\left(\sqrt{1-e^{-2\lambda t}}\right)} \right)$$

$$= \begin{cases} 1 & \text{if } \mu > \mathrm{P}(0) > 0 \quad \mathit{Or} \quad \mu < \mathrm{P}(0) < 0, \\ 0 & \mathit{otherwise.} \end{cases}$$

As $t \to \infty, PL \to 1$, if μ and $P(0)$ has the same sign, and $|P(0)| < |\mu|$, and $P_L \to 0$, otherwise. That is, in the far future, we have one probability of loss, if $P(0)$ is closer to zero, than μ and they are in the same half of the plane (Case 1 and Case 6 in Figure 6. The explanation is simple: we have exponential expectation in the profit ($\sim (1+r)^t$), while the price of the portfolio tends to its mean (μ). That is, after a sufficiently long time period, the expectation will exceed the mean with one probability, even though there is a Gaussian effect in the system.

On the other hand, in the far future, we have zero probability of loss, if $P(0)$ and μ have different signs, or if they do not, μ is closer to zero than $P(0)$ (Case 2 . . . Case 5 in Figure 6). This is more interesting from the investor point of view. The reason for having such lucky constellation is that the price of the portfolio tends to a constant value μ, while the money we got for the long/short position (note that in all four cases we receive cash) yields an exponentially growing profit (e. g. we put it into a bank account). The more time we wait the more profit we have. In the meantime, the price of the portfolio tends to a finite value with less and less variance. That is, with one probability we can pay the price of closing our (either long or short) position. The reader might be surprised having zero probability in the long run, however, we should not forget that

• short positions are admitted for only a limited time frame (the short position must be closed after this period),

• the limitation of our model yields strange situations. Since the exponential growth of portfolio prices is not included, the price tends to a constant value with lower variance.

Extending the model with getting rid of the constant nature is a future work. One might choose a trading strategy that (39) is minimized. As explained earlier, it makes sense only when μ and $P(0)$ have the same signs, and $|P(0)|<|\mu|$, i.e. (37) holds with the plus sign:

$$P_L = \frac{1}{2} + \frac{1}{2} erf\left(\frac{P(0)\sqrt{\lambda}}{\sigma\left(1-e^{-2\lambda t}\right)} \left((1+r)^t - e^{-\lambda t} - \frac{\mu}{P(0)}(1-e^{-\lambda t}) \right) \right). \tag{40}$$

For the sake of minimization at least one of the parameters must be released. We will investigate two cases: (1) the interest rate (*r)* is variable, and (2) the portfolio holding time (*t)* is variable. In both cases we make use of the fact that the error function is a monotonous increasing function that is, minimizing (39) is the same as minimizing the argument of the erf function, which will be denoted by $f(P(0),\lambda,\sigma,r,t,\mu)$ in the sequel:

$$f(P(0),\lambda,\sigma,r,t,\mu) = \frac{P(0)\sqrt{\lambda}}{\sigma\left(1-e^{-2\lambda t}\right)} \left((1+r)^t - e^{-\lambda t} - \frac{\mu}{P(0)}(1-e^{-\lambda t}) \right). \tag{41}$$

The first case is easy. If the interest rate (r) is an open parameter, it should be set to zero, since $r=0$ yields the lowest possible value in (37). As r deviates more from zero, the probability of loss becomes larger. There is no sense to optimize the value of r. Thus, only fixed r is considered in the following. If r is fixed, the investor might be interested to find an optimal time frame t_{opt} for which the portfolio should be held. For the sake of minimization

$$\begin{aligned} t_{opt} &= \arg\min_t f(P(0),\lambda,\sigma,r,t,\mu) \\ &= \arg\min_t \left(\frac{P(0)\sqrt{\lambda}}{\sigma\left(1-e^{-2\lambda t}\right)} \left((1+r)^t - e^{-\lambda t} - \frac{\mu}{P(0)}(1-e^{-\lambda t}) \right) \right), \end{aligned} \tag{42}$$

must be solved. The $f(.)$ function must be differentiated subject to t and the result must be equal to zero. First the derivative of $f(P(0),\lambda,\sigma,r,t,\mu)$ is computed.

$$\begin{aligned} &\frac{\partial}{\partial t} f(P(0),\lambda,\sigma,r,t,\mu) = \frac{P(0)\sqrt{\lambda}}{\sigma} \cdot \\ &\frac{\left(\ln(1+r)(1+r)^t + \lambda e^{-\lambda t}\left(1-\frac{\mu}{P(0)}\right)\right)\left(1-e^{-2\lambda t}\right) - \left((1+r)^t - e^{-\lambda t} - \frac{\mu}{P(0)}\left(1-e^{-\lambda t}\right)\right)}{\left(1-e^{-2\lambda t}\right)^2} \cdot \\ &2\lambda e^{-t}. \end{aligned} \tag{43}$$

The former statement can be proven here: if $1-\frac{\mu}{P(0)}$ is positive, then (43) is positive for all t values. In other words, there is no super value of $f(.)$ function. Vice versa, if $1-\frac{\mu}{P(0)}$ is negative, we can find the optimum by finding the time t which yields zero in (43).

Due to the fact that t appears in several places, we introduce a simplification: in the sequel, $(1+r)^t$ will be substituted by one. The reasoning behind is very simple; r is very close to zero, thus $(1+r)$ is very close to one. The power of this “almost one” number will not fall far from one, since t (according to the results we obtained in different situations) is always a small number. Thus this simplification yields a small error, which is negligible. After this step and some calculations, t_{opt} finally yields:

$$t_{opt} = \frac{1}{\lambda}\ln\left(-1-\lambda\frac{1-\frac{\mu}{P(0)}}{\ln(1+r)} + \sqrt{1+6\lambda\frac{1-\frac{\mu}{P(0)}}{\ln(1+r)} + \lambda^2\left(\frac{1-\frac{\mu}{P(0)}}{\ln(1+r)}\right)^2} \right) \quad (44)$$

Note the argument of the second logarithm: $\mu / P(0)$ must be greater than 1 (in words: the mean value of the long portfolio should be larger than the starting value– the same has been supposed at the start of this part and in the previous paragraph), otherwise the logarithm becomes intractable. The optimal timeframe for the portfolio thus can be computed if all the parameters $(\lambda, \mu, P(0), r)$ are available. Using this t_{opt} , (39) can be evaluated.

As an example, Figure 7 shows the case, where $P(0) = 1, \mu = 2, \lambda = 10, r = 0.1, \sigma = 1$. Note that both axes are in logarithmic scale.

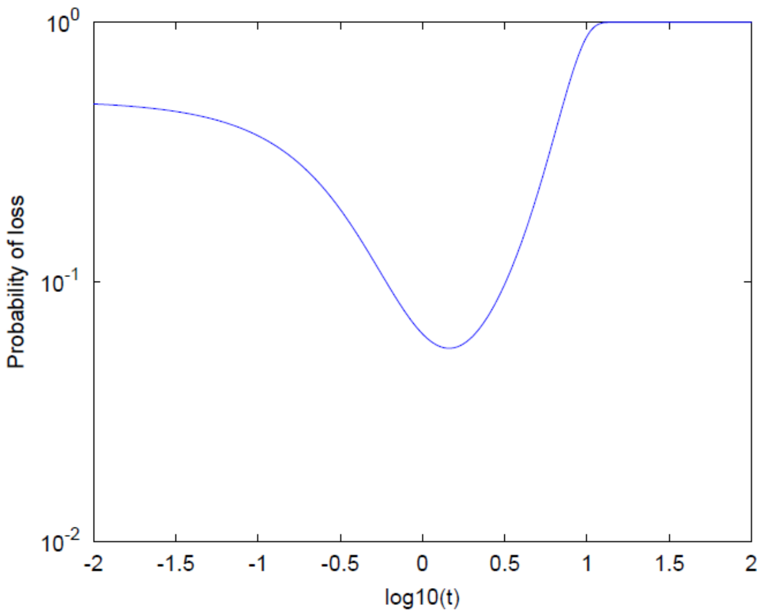

Fig. 7: An example with the parameter set $P(0) = 1, \mu = 2, \lambda = 2, r = 0.1, \sigma = 1$.

As one can compute (44) yields $topt = 1.4527$. The figure shows well how the curve behaves, starting from 0.5 and tending to 1 yielding a global minimum at the described point. Please note how small the minimum probability is: the curve goes under 10^{-1} ! If one decreases the interest rate expectation, for example $r = 0.01$, and all other parameters are left untouched, the loss probability decreases and the minimum is also shifted. The result is depicted in Figure 8. The corresponding optimal timeframe equals $t_{opt} = 2.6385$ in this case. Here the probability almost reaches the 10^{-2} limit. The reader is suggested not to forget that the minimum exists only when μ and $P(0)$ has the same sign, and $|P(0)|<|\mu|$. In all other cases, the longer t is, the lower the probability of loss P_L. Usually this is limited by some external factors; e. g. short positions must be closed after a given amount of time (denoted by t_{close}).

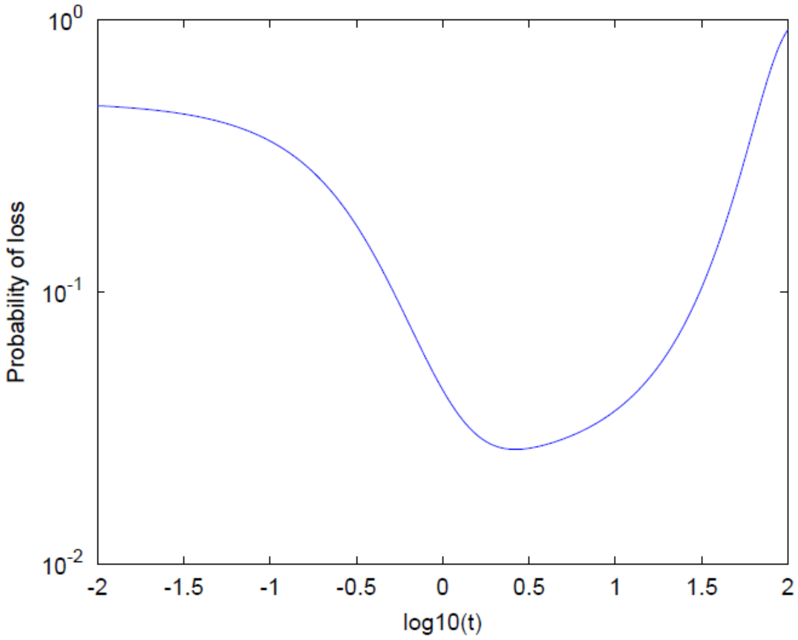

Fig. 8: Another example with smaller interest rate expectation; $P(0)=1, \mu=2, \lambda=2, r=0.01, \sigma=1$

2.6.1 Linear model parameter estimation

The parameters of (29) are not available, but they can be estimated. The estimates of parameters are given here:

$$\hat{\mathbf{G}}=\frac{1}{\mathrm{T}}\int_{t=0}^{T}\mathbf{s}(\mathrm{t})\mathbf{s}'(\mathrm{t})\mathrm{dt}\ , \tag{45}$$

$$\hat{\mathbf{A}}=\frac{1}{\mathrm{T}}\int_{t=0}^{T}\mathbf{s}(\mathrm{t}+\Delta\mathrm{t})\mathbf{s}'(\mathrm{t})\left(\mathbf{s}(\mathrm{t})\mathbf{s}'(\mathrm{t})\right)^{-1}\mathrm{dt}, \tag{46}$$

$$\hat{\mathbf{K}}=\frac{1}{\mathrm{T}}\int_{t=0}^{T}\left(\mathbf{s}(\mathrm{t}+\Delta\mathrm{t})-\hat{\mathbf{A}}\mathbf{s}'(\mathrm{t})\right)\left(\mathbf{s}(\mathrm{t}+\Delta\mathrm{t})-\hat{\mathbf{A}}\mathbf{s}'(\mathrm{t})\right)'\mathrm{dt} \tag{47}$$

or, if we have discrete samples

$$\hat{\mathbf{G}}=\frac{1}{K}\sum_{t=1}^{K}\mathbf{s}(t)\mathbf{s}'(t)dt\,, \tag{48}$$

$$\hat{\mathbf{A}}=\frac{1}{K}\sum_{t=1}^{K}\mathbf{s}(t+1)\mathbf{s}'(t)\left(\mathbf{s}(t)\mathbf{s}'(t)\right)^{-1}dt\,, \tag{49}$$

$$\hat{\mathbf{K}}=\frac{1}{K}\sum_{t=1}^{K}\left(\mathbf{s}(t+1)-\hat{\mathbf{A}}\mathbf{s}'(t)\right)\left(\mathbf{s}(t+1)-\hat{\mathbf{A}}\mathbf{s}'(t)\right)'dt\,. \tag{50}$$

In both continuous and discrete cases the standard deviation of the noise in a portfolio with a portfolio vector $\underline{n}$ (e. g. in (2.7)) can be estimated as

$$\hat{\sigma}(t)=\sqrt{\mathbf{n}'\hat{\mathbf{K}}\mathbf{n}}\,. \tag{51}$$

2.6.2 Estimating λ and μ of the mean-reverting portfolio with prompt prices

Now assume that prompt prices are available, so we can measure $P(t)$ values. If one wants to have good estimates on $P(t+\Delta t)$ with the estimated λ parameter, different methods yield different solutions. As a starting point, we use (15) with all the corresponding substitutes:

$$P(t+\Delta t)=e^{-\lambda\Delta t}P(t)+\mu\left(1-e^{-\lambda\Delta t}\right)+\sigma\frac{1-e^{-2\lambda\Delta t}}{2\lambda}\upsilon\,, \tag{52}$$

where ν ~ N (0, 1) is a zero-mean white Gaussian noise with unit variance. That is, we can measure the price of the asset (or portfolio), and based on the above equation we can also estimate the next value $(P(t+\Delta t))$ of the asset as

$$\hat{P}(t+\Delta t)=P(t)e^{-\lambda\Delta t}+\mu\left(1-e^{-\lambda\Delta t}\right). \tag{53}$$

Using advanced list square estimate one gets

$$\hat{x}=\arg\min_{x}\mathbb{E}\{g(x)\}=\arg\min_{x}\mathbb{E}\left\{\left\|P(t+\Delta t)-P(t+\Delta t)\right\|^{2}\right\} \tag{54}$$

where x could be any parameter which we would like to estimate, e. g. λ, or μ. Computing the square and substituting the expectation with average, one gets:

$$g(x) = S_2^+ - 2e^{-\lambda\Delta t} S^{\pm} + 2\mu(1-e^{-\lambda\Delta t}) S_1^+ + -2e^{-\lambda\Delta t}\mu(1-e^{-\lambda\Delta t}) S_1 + 2e^{-2\lambda\Delta t} S_2^+ + \mu^2 (1-2e^{-\lambda\Delta t} + e^{-2\lambda\Delta t}) \quad , \tag{55}$$

where

$$S_1 = \frac{1}{K}\sum_{i=0}^{K-1} P(i.\Delta t) \ , \tag{56}$$

$$S_1^+ = \frac{1}{K}\sum_{i=1}^{K} P(i.\Delta t) \ , \tag{57}$$

$$S_2 = \frac{1}{K}\sum_{i=0}^{K-1} (P(i.\Delta t))^2 \ , \tag{58}$$

$$S_2^+ = \frac{1}{K}\sum_{i=1}^{K} (P(i.\Delta t))^2 \ , \tag{59}$$

$$S^{\pm} = \frac{1}{K}\sum_{i=0}^{K-1} P(i.\Delta t) P((i+1).\Delta t) \ , \tag{60}$$

where K is the number of points available for the calculation. In other words, the length of the time window, where measured results are available, equals $K * \Delta t$.

With the above parameters (after a few pages of analysis) one can get

$$\hat{\mu} = \frac{S_1^+ S_2 - S_1 S^{\pm}}{S_2 + S_1 S_1^+ - (S_1)^2 - S^{\pm}} \tag{61}$$

and

$$\lambda = \frac{1}{\Delta t} \ln\left(\frac{\mu^2 - \mu S_1^+ - \mu S_1 + S^{\pm}}{S_2 - 2\mu S_1 + \mu^2} \right). \tag{62}$$

2.6.3 Mean reverting portfolio selection

The process of portfolio selection is depicted in Figure 9. First, all possible portfolios must be identified. As a next step, for each portfolio, the parameters of the portfolio must be estimated. Since (44) could be computed for portfolios, where $\mu / P(0)$ is greater $1.$ than only those portfolios are considered in Step 4 where this assumption holds. This is clearly only a subset of all possible portfolios. This subset is optimized with an exhaustive search. The best loss probability and t_{opt}, value must be remembered. The rest of the portfolios are considered in Step 5. The probability of loss values are computed based on the t_{opt}, value which is related to the best portfolio from the subset of Step 4. The computed probability of loss value is compared against the best probability of loss of the best portfolio from the subset of Step 4. If a better portfolio is found, then it must be remembered.

Finally, the best portfolio must be chosen, where the „best" means the one with the lowest probability of loss.

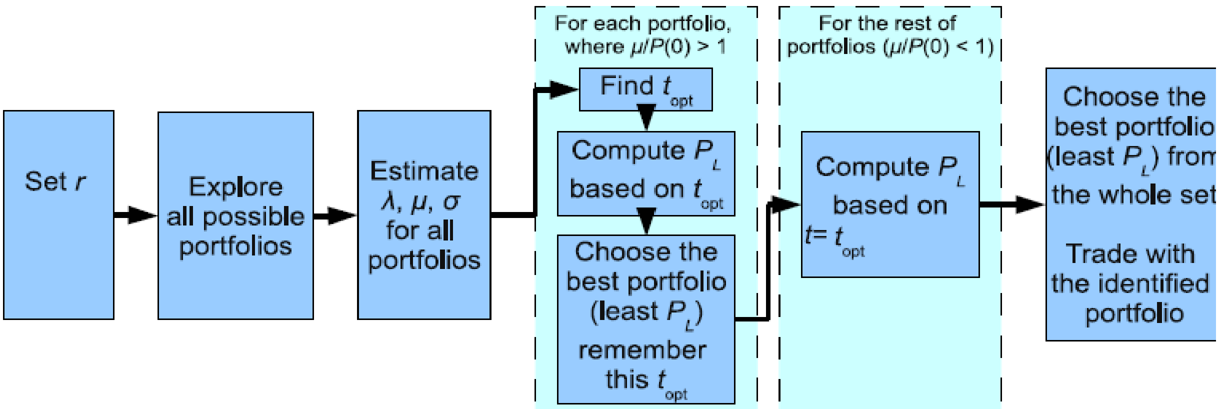

Fig. 9: The process of portfolio selection based on loss probability minimization

2.6.4 Performance analysis of mean reverting model

Since in our performance analysis of the minimum probability of loss trading strategy we used exhaustive search, we introduced two kinds of portfolio pools in order to have smaller searching path and also to distribute the risk in the case of having correlation between our assets.

Fully hedged pool

Assume that we have the G7 FOREX assets (EURUSD, GBPUSD, USDJPY, USDCHF, AUDUSD, NZDUSD, USDCAD), and the investor is interested in three asset portfolios which are fully hedged. Now translate every asset in to a common base for the sake of simplicity: let us use USD now. The above assets are transformed into EURUSD, GBPUSD, JPYUSD, CHFUSD, AUDUSD, NZDUSD, and CADUSD: although exchanging the names yields completely different assets, in the following we will assume that this modification has no effect on the portfolio[1]. It is quite easy to see that if all the money is put into such fully-hedged portfolio, exactly one asset must have a unit measure: either $+1$, or -1 must be the multiplier of the asset. As a proof, one can imagine that it is not possible to construct a fully hedged portfolio without having $+1$ or -1 value somewhere the possible portfolio combinations could be computed as

$$\binom{7}{1}.2.\binom{6}{2}.9.\frac{1}{2}=945,$$

where the first term refers to the asset having $+1$ or -1 as a multiplier, the second term (2) explodes the, state space due to the sign of this ± 1. The third term chooses two symbols from the remaining six, while the fourth term represents the nine different multiplier we could have (0.1, 0.2 . . . 0.9). Finally, the last term halves the state space due to the fact that all combinations have been counted twice (the last two assets have been selected twice, e.g. $-1A + 0.1B + 0.9C = -1A + 0.9C + 0.1B$).

Minimum deviation Pool

Suppose the number of assets is denoted by n=1…..N. we have cardinality constraint denoted by k which should be small in compare to n (we want to have small transaction cost therefor we would like to choose only few of them).

The solution is first we pick one asset and then we try to find the best linear combination of (k-1) assets which gives us the best estimate of the chosen asset in a given period. With this asset and the linear combination which we achieved we can construct our portfolio.

The size of our new pool would be $\frac{n!}{(k-1)!(n-k)!}$.

In this case we have much smaller pool of portfolios in comparison to the pool of all possible portfolios and also because we have only special portfolios which had smallest deviation in the past we can expect that they might be safer than the other portfolios.

This new „loss-probability-minimization" strategy has been tested on FOREX data series with fully-hedged pool. For the sake of comparative performance analysis, we tested the lambda maximization strategy against the optimal min loss method. In our comparisons we tried to be fair: we did not apply any sophistication, or plug-in that to optimize the models.

We have one model, the mean reverting portfolio model but developed another objective function beside maximizing parameter lambda, i.e. minimizing the probability of loss. The two algorithms are given in Table 1. The decision limits are given in Table 2.

Nick	Model	Objective
MaxLambda	Mean Reversion	Maximizing the lambda
MinPrLoss	Mean Reversion	Minimizing the Probability of Loss

Table 2. The algorithms used for obtaining numerical results

Here we used a very simple trading system. First we take the best portfolio, based on the objective function (42). We immediately initiate the trading position based on the portfolio's weights (positive weight means buy – long position, negative weight means sell – short position), if the price is closer to zero than $P(0)$ $(|P(t)|\leq|P(0)|)$. If it is not, we wait for getting closer to zero (hitting $P(0)$), meanwhile the portfolio selection algorithm is rerun. If another portfolio is chosen, a new $P(0)$ is defined (according to the actual price of the selected portfolio) and we try to open the position at a price closer to zero than the new $P(0)$. This process is run until the position is opened.

The position is closed based on the mean (μ) and t_{opt}. If the mean is hit $(|P(t)|>|P(0)|)$. then the position is closed after the first positive peak for both methods. For the loss probability minimization algorithm, if t_{opt} has elapsed, and we have not yet hit the mean, than $P(t_{opt})$ must be hit with the same rule (after the first peak, we close the position). After 10 days of the position opening, if the price is still in a valley, the position will be closed. For the StopLoss versions, the StopLoss limit is set as $P(0)-10(\mu-P(0))=11P(0)-10\mu$ .This setting yields a Risk/Reward ratio equal to 10.

As soon as the position is closed, another portfolio is going to be selected, and the process starts from all over again.

	MinPrLoss		MaxLambda	
	No SL	StopLoss	No SL	StopLoss
Entry Point	$\|P(t)\|\leq\|P(0)\|$			
Take Profit	Mean (μ), or $P(topt)$		Mean (μ)	
StopLoss (SL)	$P(topt)$, 10 days	$11P(0)-10\mu$	–	$11P(0)-10\mu$

Table 3. Decision limits for the two algorithms

We used four assets (EURUSD, GBPUSD, AUDUSD, and NZDUSD) for the FOREX operations. The sparsity constraint was set to three (maximum 3 of 4 assets were chosen for each portfolio), the trading frequency was one day (operations were taken once per day). The testing period of the algorithms was between 2009 and 2012. The initial (virtual) deposit was 1000 USD and no leverage was used (the leverage ratio was 1:1).

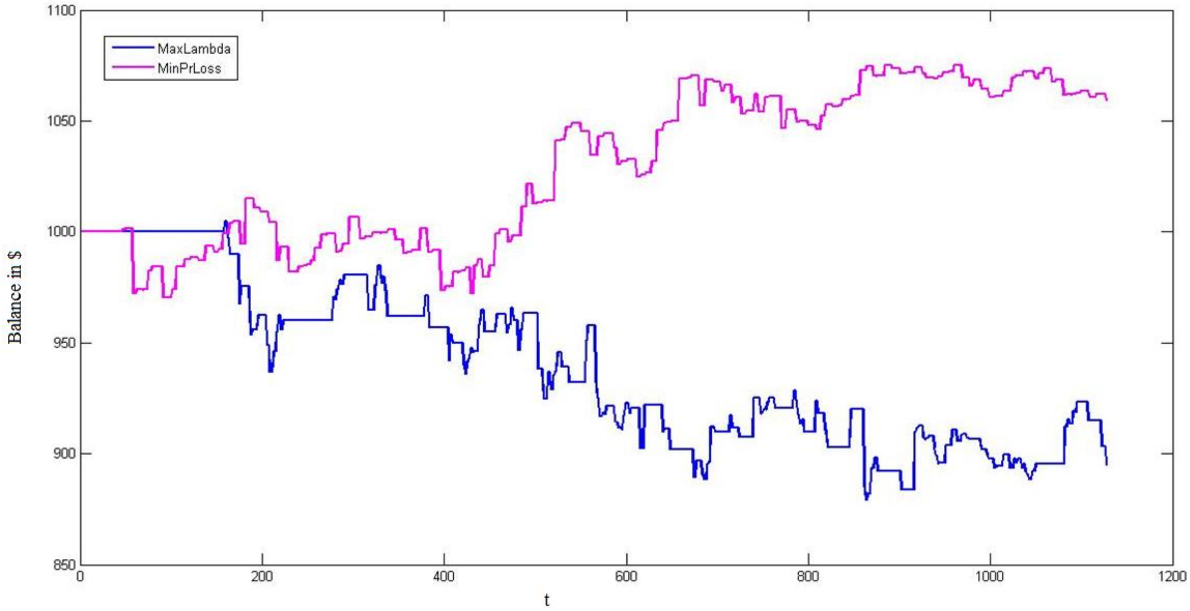

Fig. 10: The performance of the two algorithms – Balance in case of existing stop loss

Figure 10 shows the performance of the two algorithms. One can see that the strategy "minimizing probability of loss" strikingly outperforms the traditional strategy which focuses on maximizing lambda (i.e. maximizing predictability). The poor performance of the latter one is due to the fact that there are many positions closed in loss, since the stop loss limit is hit. For fair comparison we also show the performance, when stop loss limit is switched off.

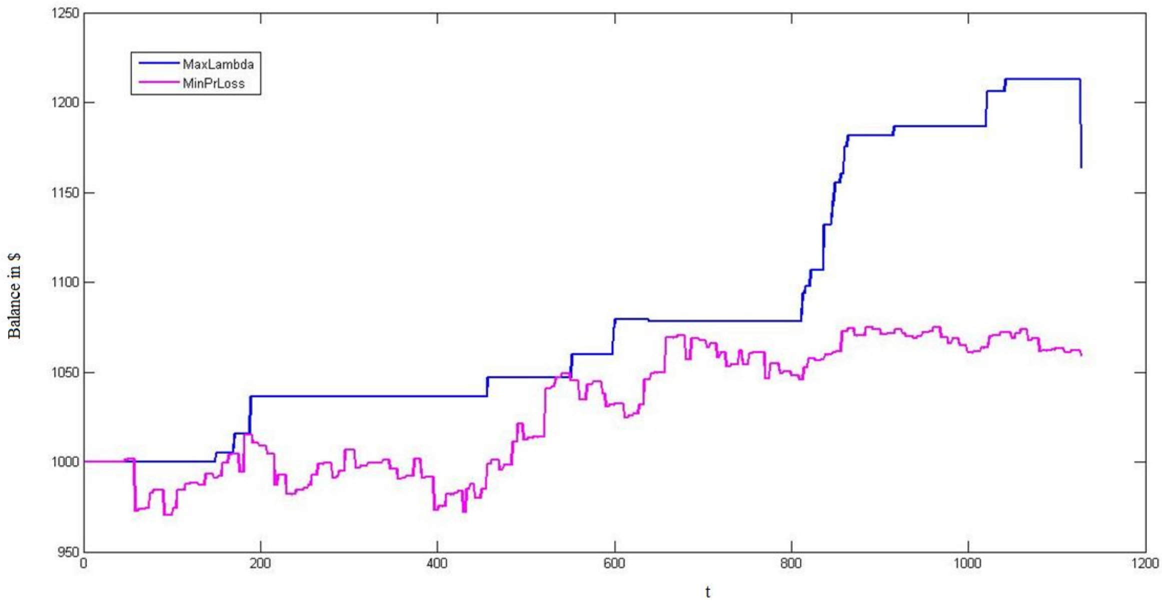

Fig. 11: The performance of the two algorithms – Balance in case of absence of stop loss

Although the strategy “maximizing lambda” is better without stop loss in Figure 11, one should not forget that the lambda maximization algorithm yields worse balance in the flat periods, when the equity remains constant. That is, if there is a leverage applied, this strategy could easily yield bankruptcy. Meanwhile, the algorithm based on the minimization of the loss probability presented the maximum drawdown of 4.3%. That is, even with a leverage ratio of *20*:1, it will avoid bankruptcy.

Performance of different pools

So far we saw a performance of the min-loss portfolio selection strategy using fully hedged pool. Now we use minimum deviation pool to find the best portfolio based on minimizing the loss probability under the assumption of mean reverting property. Our new pool has been tested on FOREX data series. For the sake of comparable performance, we tested the complete fully-hedged pool against the proposed pool. In our comparisons we tried to be fair: we did not apply any sophistication, or plug-in that to optimize the models.

We used four assets (EURUSD, GBPUSD, AUDUSD, and NZDUSD) for the FOREX operations. The sparsity constraint was set to three (maximum 3 of 4 assets were chosen for each portfolio), the trading frequency was one day (operations were taken once per day). The testing period of the algorithms was between 2009 and 2012. The initial (virtual) deposit was 1000 USD and no leverage was used (the leverage ratio was 1:1).

In this case the n is equal to 4 and k is equal to 3 therefor our pool size is $\frac{4!}{(3-1)!(4-3)!} = 12$.

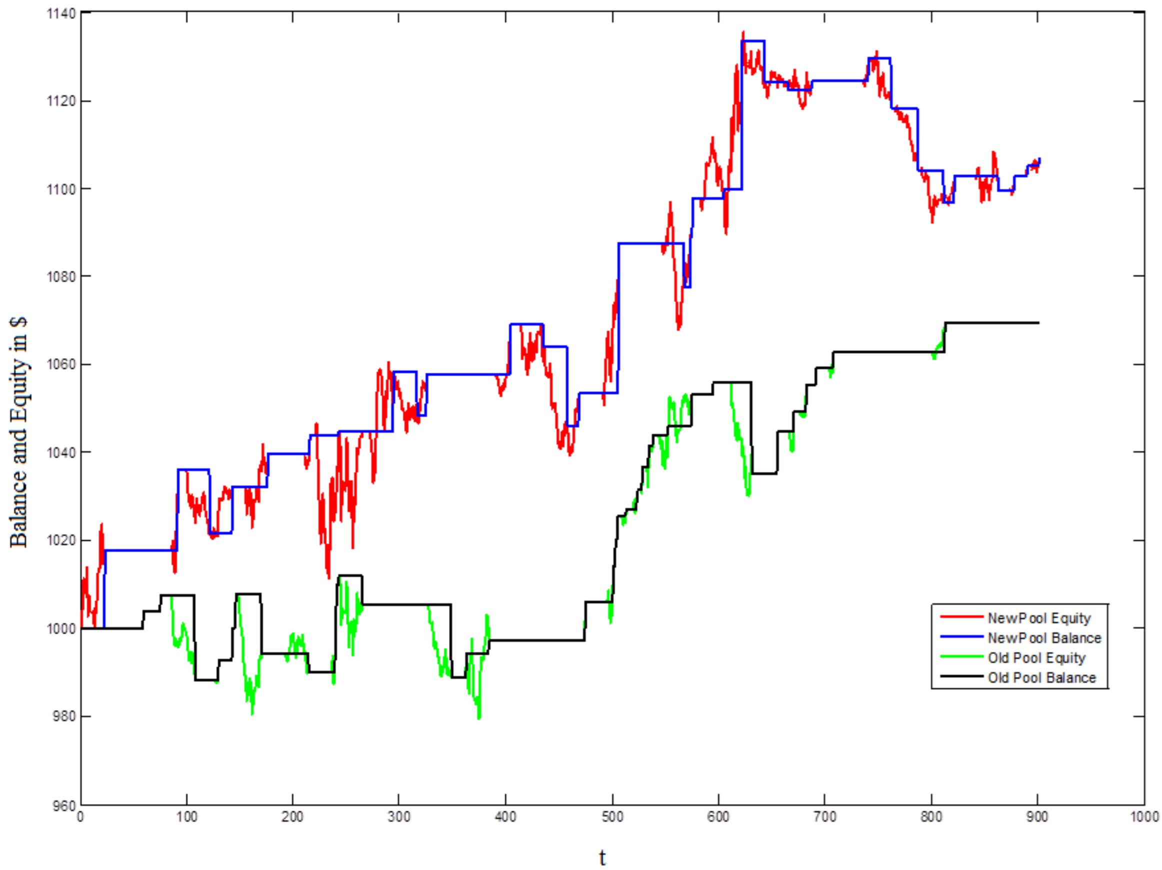

Fig. 12: The performance of minimum deviation pool vs fully-hedged pool

The numerical results obtained on FOREX data in figure 12 have been demonstrated that the minimum deviation pool can achieve higher profit than fully-hedged pool.

Performance of new parameter estimation method

For the sake of comparing the performance, we tested our old and common way of λ and μ calculation against the method introduced in 2.6.2. In our comparisons we tried to be fair: we did not apply any sophistication, or plug-in that to optimize the models.

We used four assets (EURUSD, GBPUSD, AUDUSD, and NZDUSD) for the FOREX operations. The sparsity constraint was set to three (maximum 3 of 4 assets were chosen for each portfolio), the trading frequency was one day (operations were taken once per day). The testing period of the algorithms was between 2009 and 2012. The initial (virtual) deposit was 1000 USD and no leverage was used (the leverage ratio was 1:1).

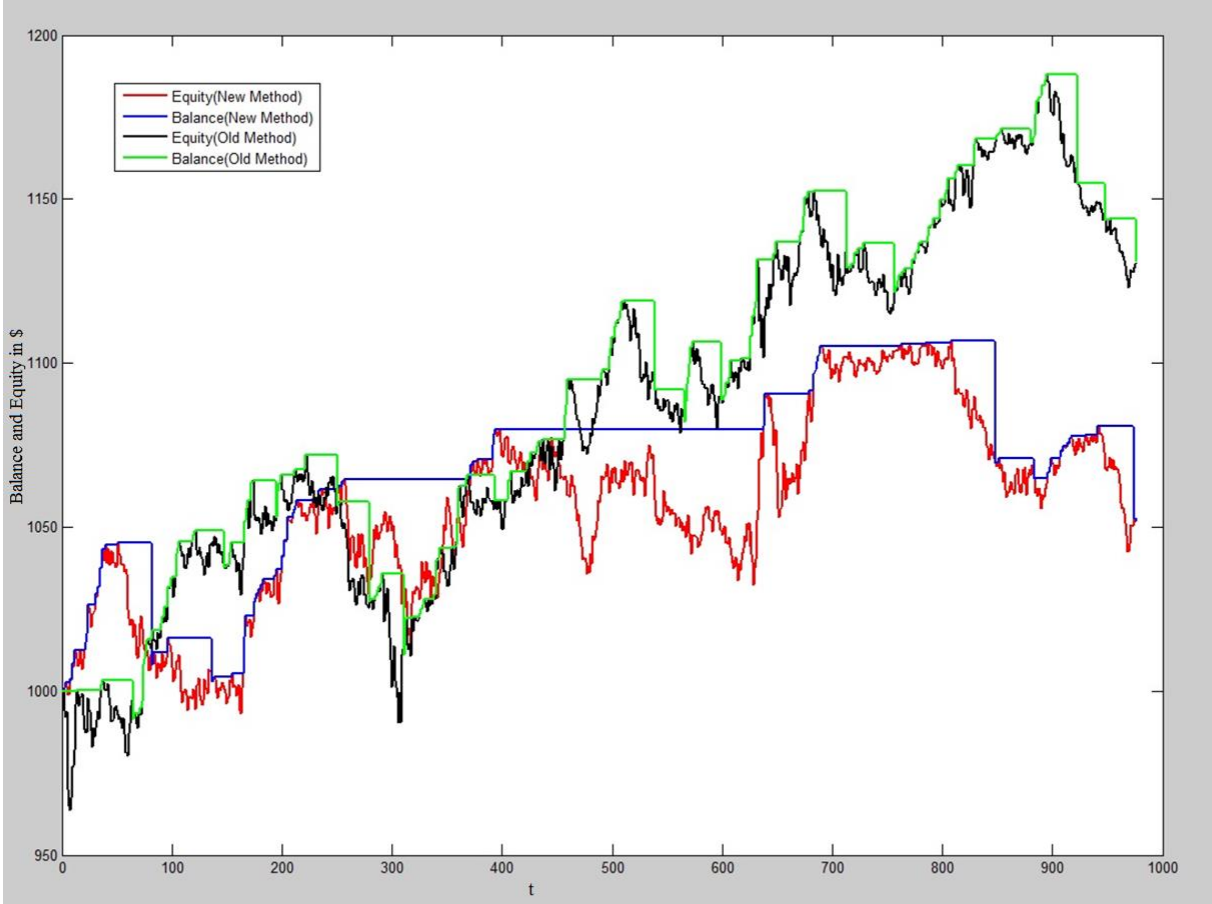

Fig. 13: Performance of traditional parameter estimation vs new parameter estimation

The numerical results obtained on FOREX data have been demonstrated that higher profit can be achieved by the traditional parameter estimation than the proposed parameter estimation. However one shouldn't forget that the complexity of the proposed method is much lower than the traditional method which is demonstrated in figure 14, 15.One can see that when the number of asset increase the calculation time which is demonstrated in figure 14 and the calculation time ratio of old-method/new-method increase dramatically.

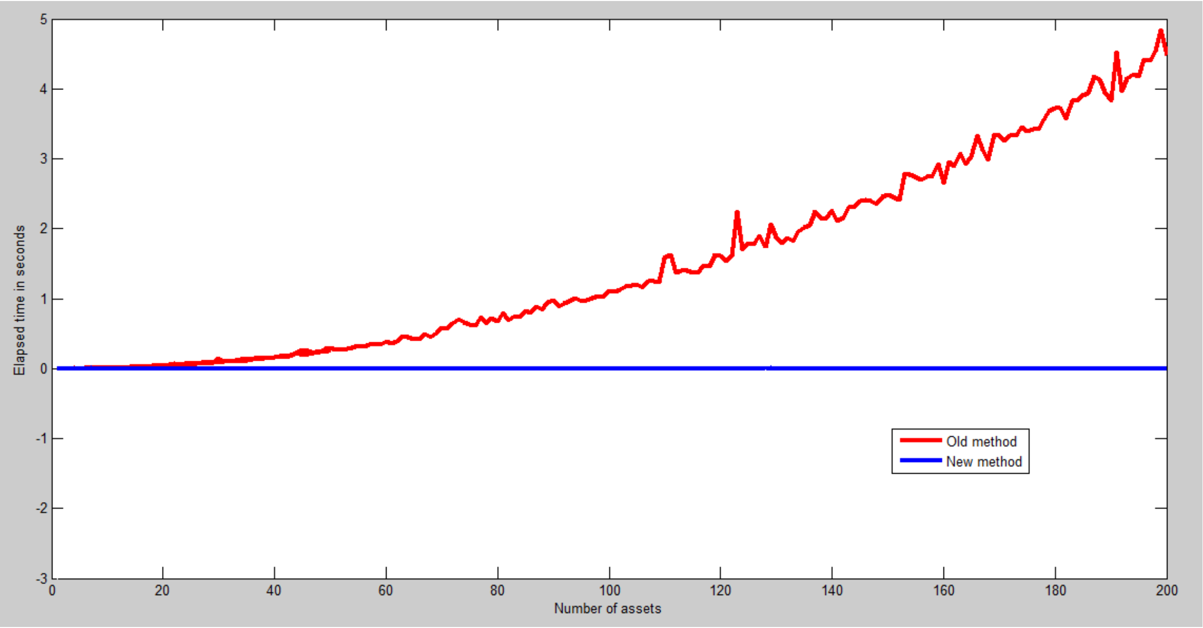

Fig. 14: Calculation time of new and traditional method over different asset numbers

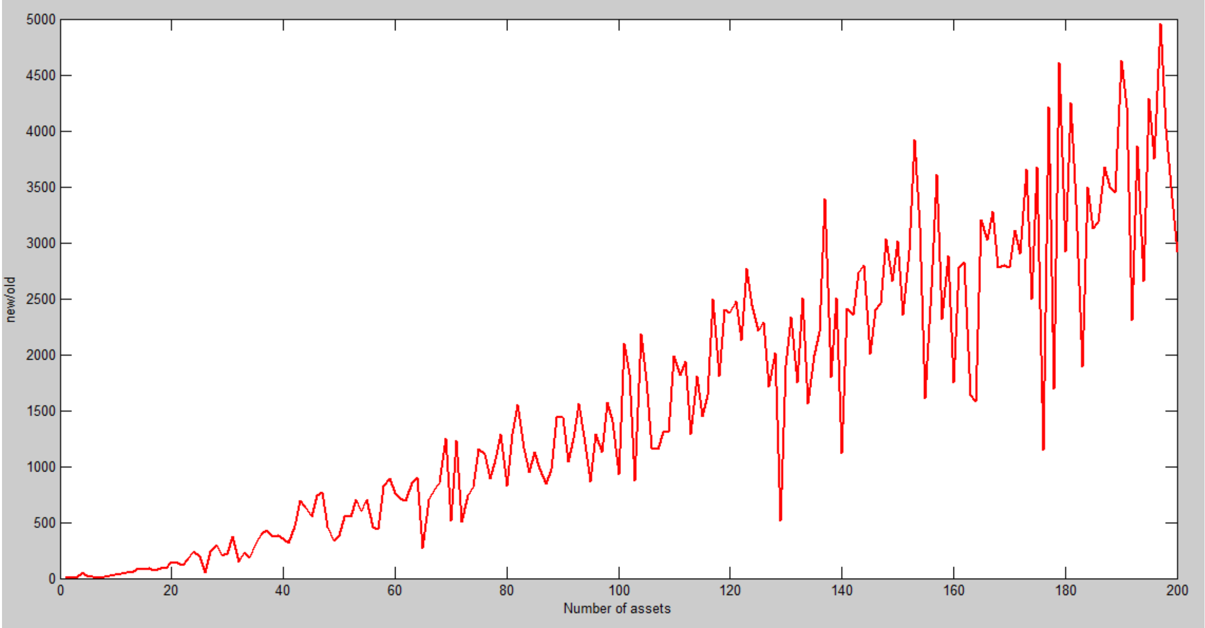

Fig. 15: Traditional method/new method calculation time ratio over different asset numbers

Performance of GPGPU implementation of the new parameter estimation method

In our portfolio selection strategy we used exhaustive search, therefore implementing this strategy on parallel architecture may let us making our portfolio pool bigger and as a result pave the way towards a more sophisticated high frequency trading. However there are some algorithmic challenges to be resolved when implementing this on multi-core architectures.

Regarding the portfolio selection, we perform parallel and independent calculation over each portfolio in our pool therefor the portfolio selection part in performed in GPU and the rest of the tasks such as Data collection, Portfolio parameter identification, Risk/Money management, trading and monitoring the open positions is performed on CPU. Accordingly, the computational framework is shown by the block diagram Fig. 16.

Depending on the properties of the actual GPU we know the maximum number of wavefront slots (from vendor documentation) per computing units (CU) and then based on our implementation we will know how many wavefronts will actually run. Afterwards we can calculate the occupancy to design significantly faster kernels. Another challenge is global memory channel conflicts. (If single channel gets multiple memory requests they are handled serially) so we should be very careful how we access memory.

Fortunately the nature of algorithms and methods used in our strategy allows parallel implementation and also because of enough ALU and much bigger portfolio pool we can hide latency (ALU efficiency), all these properties make parallel solution feasible and practical in our case.

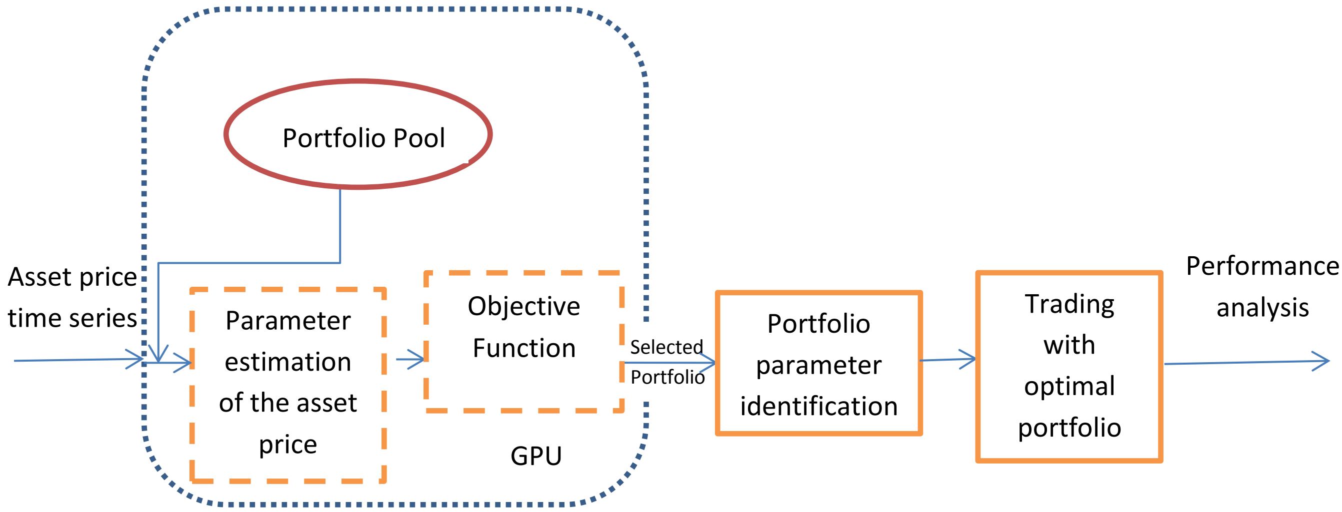

Fig. 16: GPGPU computational approach

For the sake of comparing the performance, we tested CPU implementation against this new implementation. In our comparisons we tried to be fair: we did not apply any sophistication, or plug-in that to optimize the models. The forward testing period is 3 days on five minute chart (M5) and the leverage was 1:100. FOREX rates (EURUSD, GBPUSD, AUDUSD, NZDUSD, USDCAD, USDCHF, USDJPY) from the year of 2015 in 5 minutes resolution). Regarding the sparsity constraint, 3 assets were selected out of 7 in each transaction. The portfolio pool size in CPU-mode is 945 versus 10395 in the GPU-mode.

As in GPGPU implementations we can explore bigger portfolio pool, therefore the better value of our objective function has been achieved. The next figure indicate the result of this implementation vs the result of CPU implementation. The result in Fig. 17 shows that the final profit is almost the same but we have much smaller Drawdown and more stable equity in the whole trading period in compare to CPU mode. There is not a big difference in a running time because in GPU mode we have much bigger portfolio-pool to search.

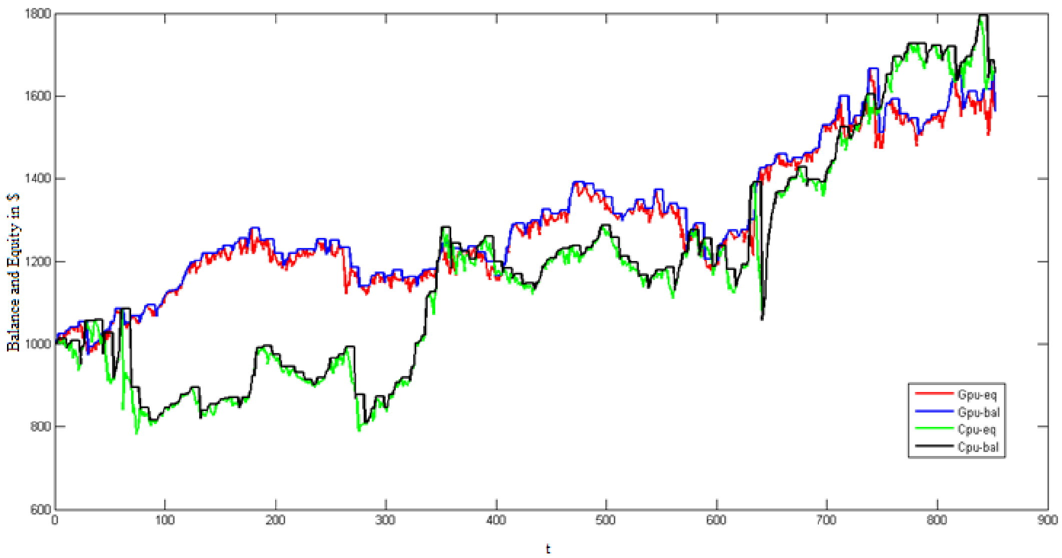

Fig. 17: The performance of CPU-based implementation and GPU-based implementation.

In this section we have examined a novel portfolio selection, based on minimizing the loss probability under the assumption of mean reverting property. After identifying the corresponding parameters of the mean reverting process, one can compute probability of making negative profit which the so-called loss probability. Since this computation can be carried out for a large number of portfolios, we can choose the one which has the lowest loss probability. The numerical results obtained on FOREX data have been demonstrated that higher profit can be achieved by the new strategy than simply maximizing the mean reverting parameter λ .

However, the complexity of the method can become overwhelming due to the exhaustive search in the space of all possible portfolios. As a future work, we are interested in finding better strategies to reduce the complexity.

2.7 Trading with Lévy jump model

In Lévy jump process portfolios, $P(t)$ must be substituted from (19) into (34).Thus one gets

$$P_L = \begin{cases} \mathbb{P}\left\{ P(0) + \delta t + \sum_{k=1}^{N_t} J_k < (1+r)^t P(0) \right\}, & \text{if } \mu > \mathrm{P}(0), \text{long position is taken,} \\ \mathbb{P}\left\{ P(0) + \delta t + \sum_{k=1}^{N_t} J_k > (1+r)^t P(0) \right\}, & \text{if } \mu < \mathrm{P}(0), \text{short position is taken.} \end{cases} , \qquad (63)$$

In (63), for technical reasons, we have considered only the first case, the other case can be similarly constructed (the limits of the sum will change). Which can be rewritten as

$$P_L = \mathbb{P}\left\{ \sum_{k=1}^{N_t} J_k < P(0)\left((1+r)^t - 1\right) - \delta t \right\}. \qquad (64)$$

The left hand side of the inequality in the probability depends on several independent random variables. The number of points in N_t is a Poisson random variable. Since J_k follows the discrete distribution of ρ_i, their sum is another discrete random variable. The distribution of n independent J_k random variables (we will denote it as J_k^n in the following) has an important role here. Thus, we should first construct their distribution based on (77). Since the question is the probability of remaining under a certain limit, the discrete probability values must be summed. Thus, (64) yields

$$P_L = \sum_{n=0}^{\infty} \mathbb{P}\left\{ N_t = n \right\} \mathbb{P}\left\{ J_k^n < P(0)\left((1+r)^t - 1\right) - \delta t \right\}, \qquad (65)$$

which could be rewritten as

$$P_L = \sum_{n=0}^{\infty} \frac{e^{-\Lambda t} (\Lambda t)^n}{n!} \cdot \sum_{i=-\infty}^{\left\lfloor P(0)\left((1+r)^t - 1\right) - \delta t \right\rfloor} \rho_i^{(n)}. \qquad (66)$$

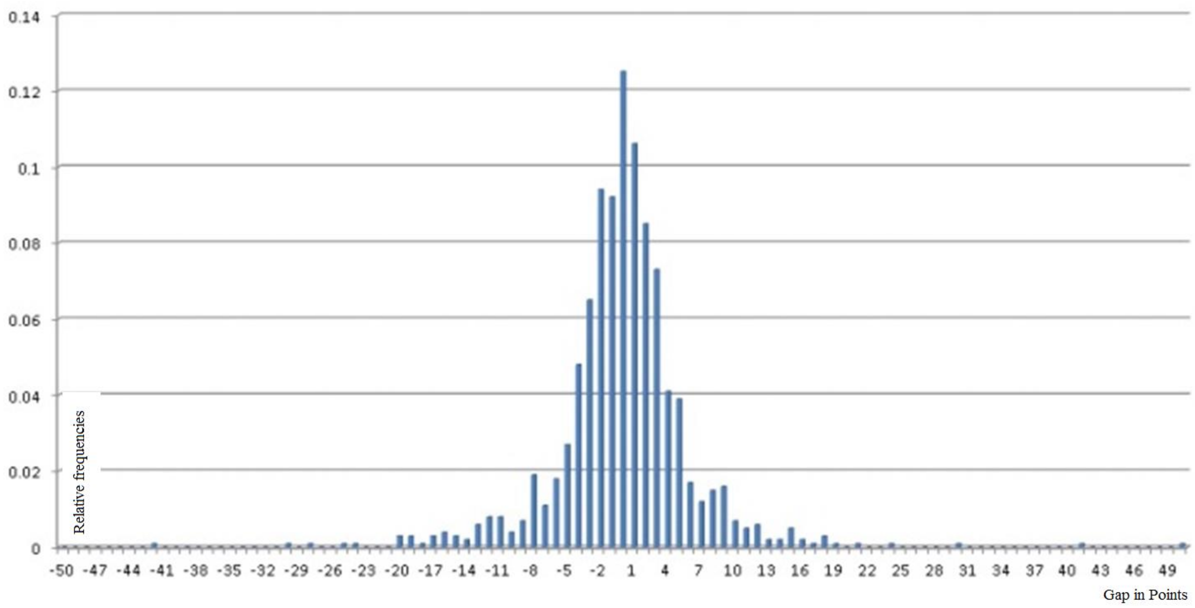

Fig. 18: Relative frequencies of different hop sizes for the FOREX EURUSD charts

For $n = 0$ (there is no transaction in the time period t), $\rho_i^{(0)} = I(\mathrm{i}=0)$. Note that although the second sum starts from $-\infty$, its real starting value depends on the measured properties of subsequent transactions. If $J_k = -l$ is the lowest value which happens in the stock market, $J_{nk} < -l$ n has zero probability ($\rho_i^{(\mathrm{n})} = 0, \forall i < -l\ n$), thus the sum can be started from $-l\ n$,

$$P_L = \sum_{n=0}^{\infty} \frac{e^{-\Lambda t}(\Lambda t)^n}{n!} \cdot \sum_{i=-\ln}^{\left\lfloor P(0)\left((1+r)^t-1\right)-\delta t \right\rfloor} p_i^{(n)} . \tag{67}$$

As an example, the EURUSD jumps are analyzed from the FOREX market in Figure 18. On the horizontal axis one sees the size of the jump; on the vertical axis their relative frequency is given. It is acceptable to choose $l = 20$ in the FOREX, as long as we consider the EURUSD asset.

Also, the first sum can be limited. The nice property of the Poisson distribution is that there is an upper (Chernoff) bound for the tail probability (the sum of the probabilities of the subsequent elements after the k th element), i.e., for any $k \in R+, k > \lambda$, i.e. k must be greater than the mean:

$$\frac{e^{-\lambda}(\mathrm{e}\lambda)^k}{k^k} \geq \sum_{i=k}^{\infty} \frac{e^{-\lambda}\lambda^i}{i!} \tag{68}$$

Note that the multiplier of the Poisson probabilities in (66) is smaller than one. Thus, (66) could be estimated as

$$\widehat{P}_L = \sum_{n=0}^{k-1} \frac{e^{-\Lambda t}(\Lambda t)^n}{n!} \cdot \sum_{i=-\ln}^{\lfloor P(0)((1+r)^t-1)-\delta t \rfloor} \rho_i^{(n)}, \tag{69}$$

where $k > \Lambda t$ and $P_L - \widehat{P}_L \le \frac{e^{-\Lambda t}(\mathrm{e}\,\Lambda\,\mathrm{t})^k}{k^k}$.

If there is an acceptable error, the first sum of (66) could be limited with the condition

$$e^{-\Lambda t}\left(\frac{(\mathrm{e}\,\Lambda\,\mathrm{t})}{k}\right)^k < 10^{-\varepsilon}, \tag{70}$$

that is, (69) with (70) could be used to calculate the probability of loss with arbitrary decimal precision ε . If ε is given, first the upper bound k must be found based on (70), then (69) must be computed. The error of the computation is smaller than $10^{-\varepsilon}$ for sure.

Example 1

Let us have $\underline{\rho}^{(1)} = [0.1, 0.1, 0.55, 0.2, 0.05]$, for $\underline{J}^{(1)} = [-2, -1, 0, 1, 2]$ then

$\underline{\rho}^{(2)} = [0.01, 0.02, 0.12, 0.15, 0.3525, 0.23, 0.095, 0.02, 0.0025]$ $\underline{J}^{(2)} = [-4, -3, -2, -1, 0, 1, 2, 3, 4]$

$\underline{\rho}^{(3)} = [0.001, 0.003, 0.0195, 0.04, 0.12075, 0.16575, 0.2623750.216, 0.118125,$
$0.04175, 0.010125, 0.0015, 0.000125]$

$\underline{J}^{(3)} = [-6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6]$

...

Assume that $\Lambda = 0.1, t = 1, r = 0.1\ (10\%), P(0) = 10, \delta = 0.1$ and $\varepsilon = 0.001$, then the limit for the first sum (*k*) equals three: for $k = 2$ (70) gives 0.016613, for $k = 3$ it yields 0.000667, which is smaller than ε .

The upper limit of the second sum yields $\lfloor 10 \cdot ((1+0.1)^1 - 1) - 0.1 \cdot 1 \rfloor = \lfloor 0.9 \rfloor = 0$. The $\underline{\rho}$ vectors must be summed up to the middle. Thus, (69) yields

$$\widehat{P}_L = \sum_{n=0}^{3} \frac{e^{-0.1}(0.1)^n}{n!} \cdot \rho_i^{(n)},$$

$$\widehat{P}_L = e^{-0.1}\left(\frac{0.1^0 1}{0!} + \frac{0.1^1 0.75}{1!} + \frac{0.1^2 0.6525}{2!} + \frac{0.1^3 0.612375}{3!}\right) = 0.975745.$$

The probability of loss for this portfolio is in the interval of 97.5745% . . . 97.6412% (this is where the effect of ε appears).

2.7.1 Identification of Lévy jump process parameters

The estimation becomes more involved when the model contains random time changes. Since the activity rates are not observable, some filtering technique is often necessary to be implemented to determine the current level of the activity rates. Some researchers propose to estimate the dynamics using a Bayesian approach involving Markov Chain Monte Carlo (MCMC) simulation[30, 31]. They use MCMC to Bayesian update the distribution of both the state variables and model parameters. Another researcher proposes a maximum likelihood method in estimating time-changed Levy processes[32]. Under this method, the distributions of the activity rates are predicted and updated according to Bayesian rules and using Markov Chain Monte Carlo simulation. Then, the model parameters are estimated by maximizing the likelihood of the time-series returns; however in our case we assumed that we have access to activity rates (number of transactions) therefore we calibrated the model in a different manner.

Under the conditions described above, the mean can be calculated as

$$\begin{aligned} \mathbb{E}\left\{P(t)-P(0)\right\} &= \mathbb{E}\left\{\delta t + \sum_{k=1}^{N_t} J_k\right\} \\ &= \delta t + \mathbb{E}\left\{\sum_{k=1}^{N_t} J_k\right\} = \delta t + \Lambda t \mathbb{E}\left\{J_k\right\} = \delta t. \end{aligned} \tag{71}$$

Thus, the statistical mean of $\mathbb{E}\left\{P(t+\tau)-P(t)\right\}/\tau$ yields parameter δ, for any T

$$\hat{\delta} = \frac{1}{K}\sum_{i=1}^{K}\left(\frac{P(iT)-P((i-1)T)}{T}\right) = \frac{P(KT)-P(0)}{KT}. \tag{72}$$

In most cases δ proved to be a very small number, hence we assume that its effect can be neglected in subsequent transactions i.e. $\delta\Delta t \approx 0$.

Since $\{N_t\}$ is a Poisson point process, its mean, Λ can be estimated by the average number of transactions,

$$\hat{\Lambda} = \frac{1}{KT} \sum_{i=1}^{K} \left(N_{(i+1)T} - N_{iT} \right) = \frac{N_{(K+1)T} - N_T}{KT}, \tag{73}$$

for any T.

Finally, the statistics of J_k must be simply counted by relative frequencies,

$$\hat{\rho}_i = \frac{1}{K} \sum_{k=1}^{K} I(J_k = i), \tag{74}$$

where $I(.)$ is the indicator function and $\hat{\rho}_i$ estimates the probability $\mathbb{P}\{J_k = i\}$. Since the convolution of such random numbers will play a major role in the following, the 2-step probabilities can be calculated as:

$$\hat{\rho}_n^{(2)} = (\hat{\rho} * \hat{\rho})_n = \sum_i \hat{\rho}_i \hat{\rho}_{n-i}. \tag{75}$$

Note that the limit of the sum depends on the number of non-zero elements in $\hat{\rho}$. That is, if $\hat{\rho}_i = 0$ for all $|\mathrm{i}| > 1$, the sum contains three elements only ($i \in \{-1, 0, 1\}$).

As a generalization, the n-step probabilities can be easily constructed; n such numbers generate a process with distribution $\hat{\rho}^{(n)}$, such that

$$\hat{\rho}^{(n)} = \underbrace{\hat{\rho} * \hat{\rho} * \hat{\rho} * \ldots * \hat{\rho}}_{n}, \tag{76}$$

or equivalently

$$\hat{\rho}_i^{(n)} = \sum_j \hat{\rho}_j^{(n-1)} \hat{\rho}_{i-j}. \tag{77}$$

2.7.2 Performance analysis of Levy model

We analyzed the performance of trading based on the Lévy model in two different cases:

In the first case, we did the back testing for a long period (almost 3 years) on a daily timeframe to reveal the capability of our model in selecting the best portfolio out of the pool.

In the second case, we did forward testing on shorter periods, such as 5 minute timeframe, for one month and investigated the performance of the results.

In both cases we have a sparse portfolio, i.e. a combination of maximum 3 assets out of four (EURUSD, GBPUSD, AUDUSD, NZDUSD) in the FOREX market was chosen by the Lévy based trading algorithm. Our pool, which we call it fully-hedged pool, contains those kinds of portfolios which has equal volume of short and long positions.

The trading strategy applied the following decisions:

- Entry point -- Due to different time scales, the two cases differ in the position-opening schedule. The following two subsections give details about the entry point decisions.
- No Stop Loss point -- we do not have any stop loss, instead we have maximum waiting time which is one of our predefined critical parameters. If this period has elapsed, the positions are closed independently of the actual prices. The two cases have different waiting times, which are given in the following two subsections.
- Take Profit point -- Due to different time scales, the two cases differ in the portfolio closing decision. The following two subsections give details about the taken profit points.
- Risk/Money management -- as we do not have an explicit stop loss point, we trade with full margin and thus we are not able to really manage the risk. Risk management is limited to setting the leverage ratio. The two cases applied different leverages which are given in the following two subsections.
- The initial deposit was 1000$ in both cases.

Back-testing

Back-testing was performed on daily time frames. That is, all positions were opened at the beginning of the day and were closed at the end of the day.

The trade was immediately started right after the portfolio was identified.

In our test we used 10 days of waiting time. We closed the trades at the end of each day if our profit was positive. If not, at the end of the next day the same comparison was done ....etc. After 10 days, the portfolio is closed if still remained in the negative profit region.

We did not use any leverage for back-testing. The results are indicated in figure 17.

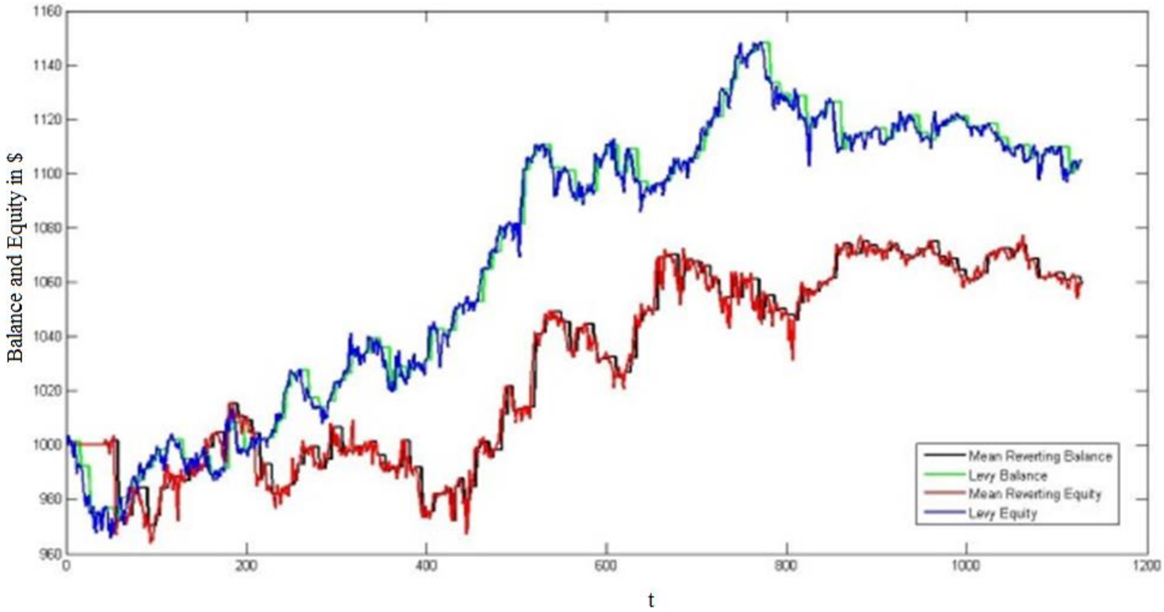

Fig. 19: The performance of the proposed method in back-testing mode: balance and equity of the proposed method (Lévy) and the mean reverting algorithm

As one can see from the Fig. 19, the profit was approximately 11%; the maximum drawdown was less than 5% in this three year period.

Forward-testing

In forward-testing, we switched from the daily timeframe to five-minute timeframe. The process of searching for the best portfolio proved to be a time consuming task. During the identification of the best portfolio, the price usually changed. Therefore -- despite of the strategy in back-testing -- we waited till we get the same price, or lower price, and the position was opened after the criteria was met.

Here, we used 50 minutes of waiting time. We had implicit take profit point which is equal to our expected interest rate (r), in our case 10$ per 1 standard lot was the profit point.

We used leverage of 100 for forward-testing. The results are indicated in figure 18.On the horizontal axis, the transaction steps are given.

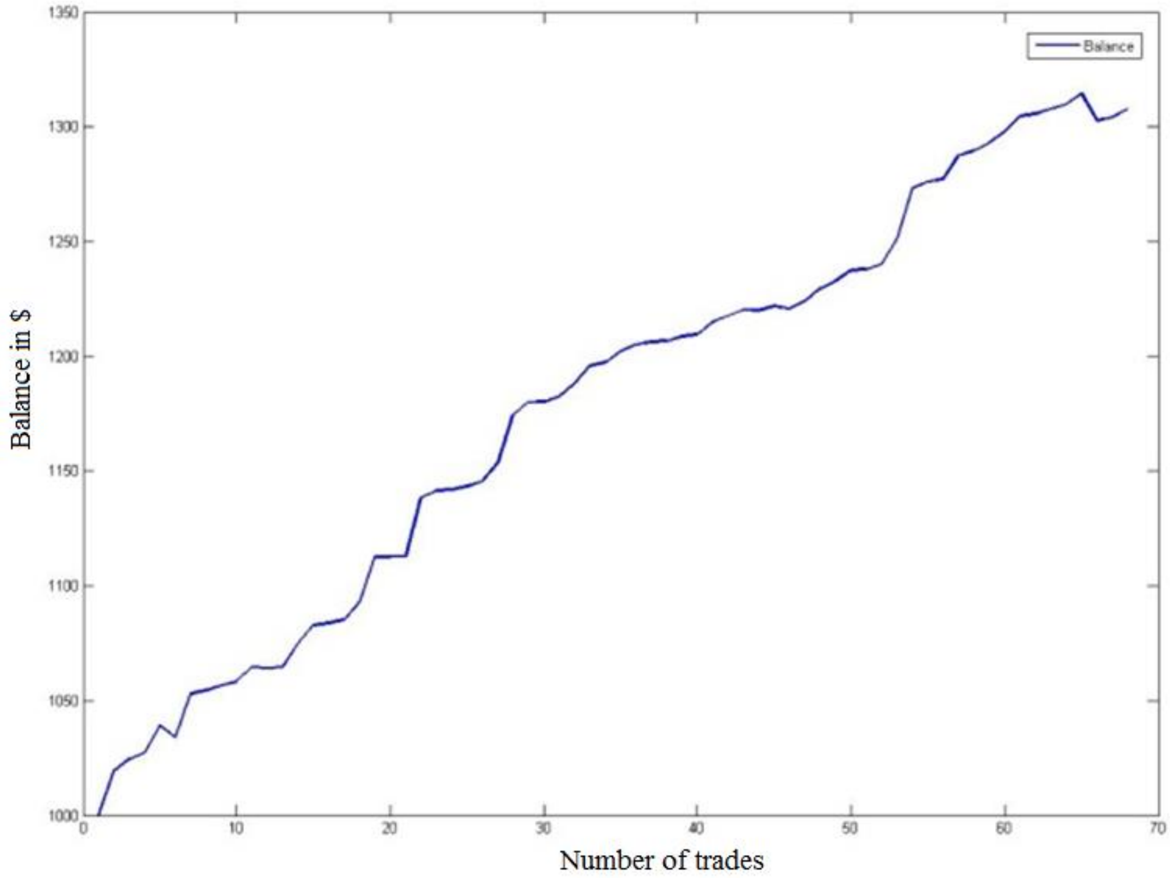

Fig. 20: The performance of the proposed method in forward-testing mode: balance

As visible from the Figure 20, the profit was 31%; the maximum drawdown was approximately 1% in the three week period.

In this section, we have examined a novel portfolio optimization method and trading algorithm based on the Lévy model. By using the Lévy model, one can capture a wild class of random portfolio value sequence. Estimates for the identification of model parameters were also given. Based on the Lévy process one can analytically calculate the probability of positive returns and the optimal portfolio holding time. As a result, the optimal portfolio could be selected which maximizes the probability of positive return. The method was tested by back- and forward testing and its performance outdid the traditional mean reverting trading.

2.8 Conclusion

In this chapter we traded based on our novel proposed objective function assuming different underlying models (e.g. mean reverting and levy-jump model).However the strategy was profitable one shouldn’t forget that profits are not everything – as important as profits (in absolute numbers) are the projected losses. Because these temporary losses – the drawdown – determine how much capital is required. We haven’t had any trigger for entry points which we believe that can improve the performance of our trade and also reduce the drawdown. In the next chapter we try to construct a model which can be used as a trigger for our entry points.

CHAPTER 3

Prediction-based Trading

In this chapter we implement the trading strategy using feed-forward neural network. Furthermore we used feed-forward neural network to predict the future values of mean reverting portfolio identified by our previous method. I will demonstrate the performance of this approach.

In the previous chapter our trading strategy was always based on stochastic model (Mean reverting or Levy). However now we provide “Model-free” trading methods by prediction. As a result in this chapter we focus on developing fast prediction of the forward distribution. Once the predictor has been implemented, these algorithms can trade on any type of random time series. Thus there is no need for model identification.

3.1 Introduction

The prediction of financial market indicators is a topic of considerable practical interest [33-35] and, if successful, may involve substantial pecuniary rewards. Neural networks have been used for several years in the selection of investments because of their ability to identify patterns of behavior that are not readily observable. Much of this work has been proprietary for the obvious reason that the users want to take advantage of the insight into the market they gained through the use of neural network technology. From the analyses of various statistical models, Artificial Neural Networks are analogous to nonparametric, nonlinear, regression models. So, Artificial Neural Networks (ANN) certainly has the potential to distinguish unknown and hidden patterns in data which can be very effective for share market prediction. If successful, this can be beneficial for investors and financers and that can positively contribute to the economy.

There have been many attempts and plenty of research to predict the market behavior using neural networks because of their ability to deal with uncertain and fuzzy data which changes rapidly in very short periods of time. Some researchers tried to classify stocks into the classes such as stocks with either positive or negative return [36-38], another group of researchers used NNs for stock price predictions [34, 39-42], they tried to predict stock prices for one or more days in advance and finally the last group of researchers tried to modeling stock performance [43, 44]. These authors are not only focused on the prediction of future values, but also on the factor significance estimation, sensitivity analysis among the variables that could impact the result and other analyses of mutual dependencies(including portfolio model and arbitrage pricing models). However according to Wong, Bodnovich and Selvi the share of using NNs in finance is half of the share of NNs in production/operation (Production/operations (53.5% and finance (25.4%).

Learning algorithms have a significant impact on the performance of algorithmic trading. Therefore, the choice of a suitable learning algorithm play a key role in the success of any system since they behave differently on different data sets.

Many different learning algorithms have been proposed and evaluated experimentally in various real-world applications domains, including speech recognition, handwritten character recognition, image classification and algorithmic trading [45-48]. In this work we demonstrate BP method [49] for training the Multilayer Feed forward Neural Network in order to forecast the simple asset price or the price of any given portfolio.

3.1.1 Prediction architecture

Let us assume that we trade on the mid prices, the corresponding asset price time series is denoted by x_n and follows a nonlinear AR(J) process

$$x_n = F\left(x_{n-1,\ldots,}x_{n-J}\right) + \nu_n, \tag{78}$$

where F is a Borel measurable function and $\nu_n \sim N\left(0,\sigma\right)$ i.i.d.r.v.-s, being independent of x_n. For trading, we construct an estimator

$$\tilde{x}_n = Net\left(x_{n-1},\ldots,x_{n-J}, w_1,\ldots,w_M\right) = Net\left(\mathbf{x},\mathbf{w}\right), \tag{79}$$

where

$$\tilde{x}_n = Net\left(\mathbf{x},\mathbf{w}\right) = \varphi\left(\sum_i w_i^{(L)}\varphi\left(\sum_j w_{ij}^{(L-1)}\ldots\varphi\left(\sum_m w_{nm}^{(1)}x_{n-m}\right)\ldots\right)\right), \tag{80}$$

is a Feed-forward neural Network (FFNN) depicted by Figure 21 and vector **w** denotes the free parameters subject to training.

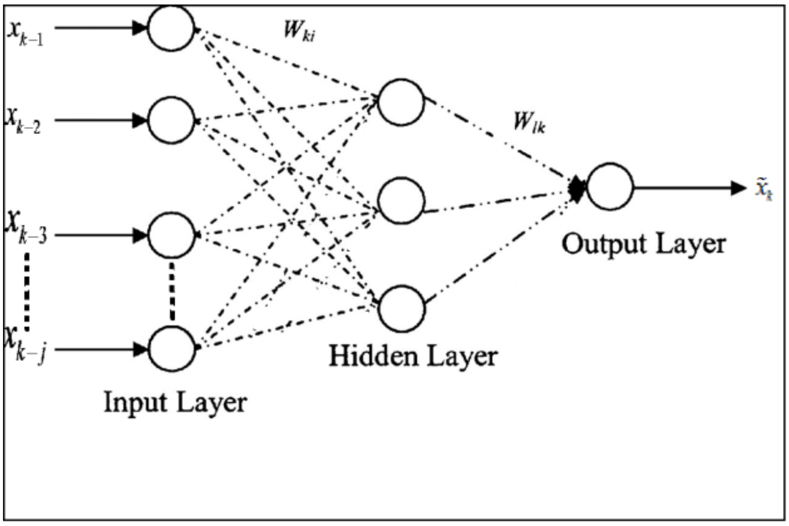

Fig. 21: The structure of feed forward neural network (FFNN).

The output layer has a target value which is used to compare with calculated value. As the errors are back propagated through the nodes, the connection weights are continuously updated. The method calculates the gradient of a loss function with respect to all the weights in the network. The gradient is fed to the optimization method which in turn uses it to update the weights, in an attempt to minimize the objective function. In our case the objective function is MSE. If $\hat{\mathbf{Y}}$ is a vector of n predictions, and $\mathbf{Y}$ is the vector of observed values corresponding to the inputs to the function which generated the predictions, then the MSE of the predictor can be estimated by

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \| \hat{\mathbf{Y}}_i - \mathbf{Y}_i \|^2 . \qquad (81)$$

Trading is performed as follows:

Stage 1. Observing a historical time series and forming training set $\tau^{(n)} = \left\{ \left(\mathbf{x}^{(n)}, x_n \right), n = 1, ..., N \right\}$ where $\mathbf{x}^{(n)} = \left(x_{n-J}, ..., x_{n-1} \right)$.

Stage 2. Training the weights by minimizing the objective function $\min_{\mathbf{w}} \frac{1}{N}\left(x_n - Net\left(\mathbf{x}^{(n)}, \mathbf{w} \right) \right)^2$ by the BP algorithm.

Stage 3. Trading on the real data as follows:

Calculate $\tilde{x}_n = Net\left(\mathbf{x}^{(n)}, \mathbf{w} \right)$, where $\mathbf{x}^{(n)} = \left(x_{n-J}, ..., x_{n-1} \right)$ and we are in time instant $n-1$.

- If $\tilde{x}_n > x_{n-1}$ then buy at time instant $n-1$ and sell at time instant n.
- If $\tilde{x}_n < x_{n-1}$ then sell at time instant $n-1$ and buy at time instant n.

It can be easily proven that this is the optimal strategy, as far as the expected profit maximization is concerned.

In the Simple Model our training set is constructed only by mid-price which is the only option in some conditions but in this Model, we use two separate networks(82,83) to include the bid-price which denoted by $\mathbf{x}$ and ask-price denoted by $\mathbf{y}$ to construct our new training sets where

$$\tilde{x}_k = Net^{Bid}(\mathbf{y}, \mathbf{x}, \mathbf{w}) \tag{82}$$

and

$$\tilde{y}_k = Net^{Ask}(\mathbf{y}, \mathbf{x}, \mathbf{u}) \tag{83}$$

are Feed-forward neural Networks (FFNN and vector **u** and **w** denotes the free parameters subject to training.

$$\begin{aligned} &\tilde{x}_k > y_{k-1} -> BUY \\ &\tilde{y}_k < x_{k-1} -> SELL \end{aligned} . \tag{84}$$

$\tilde{x}_k$ and $\tilde{y}_k$ are predicted Bid and Ask values which we compare them to the current price, so if the predicted Bid-price is bigger than current Ask-price then we buy the asset or if the predicted Ask-price is less than current Bid-price then we sell that asset. In this case there might be some times that neither of these conditions is met so we might have no trade in some periods which may decrease the number of trades in compare to the simple model.

3.2 Predicting the future prices of mean reverting portfolios

In this strategy first we select the best mean reverting portfolio and then try to predict the value of portfolio with this method to filter out the bad entry points!

By using FFNNs, which exhibit universal representation capabilities, one can model the nonlinear AR (J) process (the current value of the time series depends on *J* previous values and corrupted by additive Gaussian noise). Assuming the price series to be a nonlinear AR (J) process, we first develop the optimal trading strategy and then approximate the parameters of nonlinear AR (J) by an FFNN.

One can introduce a portfolio vector $\mathbf{r}^T = (r_1,...,r_n)$ which can be any optimal portfolio from Chapter 3, where component r_i denotes the amount of asset i held. In practice, assets are traded in discrete units, so $r_i \in \{0,1,2,...\}$ but for the purposes of our analysis we allow r_i to be any real number, including negative ones which denote the ability to short sell assets. We then plug this portfolio to our simple or advanced model in order to predict the value of the portfolio.

The values of portfolio $\mathbf{p}(t)$ at a given time t is,

$$\mathbf{p}_{Bid}(t) = max(\mathbf{0}',\mathbf{r})\mathbf{s}(t) + min(\mathbf{0}',\mathbf{r})\mathbf{b}(t), \tag{85}$$

$$\mathbf{p}_{Ask}(t) = max(\mathbf{0}',-\mathbf{r})\mathbf{s}(t) + min(\mathbf{0}',-\mathbf{r})\mathbf{b}(t), \tag{86}$$

where $\mathbf{s}(t)$ and $\mathbf{b}(t)$ vectors are Ask and Bid prices of each asset at a given time t.

Now we have this virtual asset $\mathbf{p}(\mathrm{t})$ which we can use it as a simple asset as before to predict its value and use it for trading. Trading is performed as follows:

- Find the best mean reverting Portfolio vector $\mathbf{r}$.
- Calculate the Bid and Ask price of the portfolio using (85, 86).
- Observing a historical portfolio time series and forming training sets.
- Training the weights by minimizing the objective function (MSE).
- Predict the value of the portfolio with our trained networks.
- Compare the predicted value with the mean of the portfolio; if they are in the same direction then we do trade.

Accordingly the process of using prediction-based filter for mean reverting portfolios is shown by the block diagram Fig. 22.

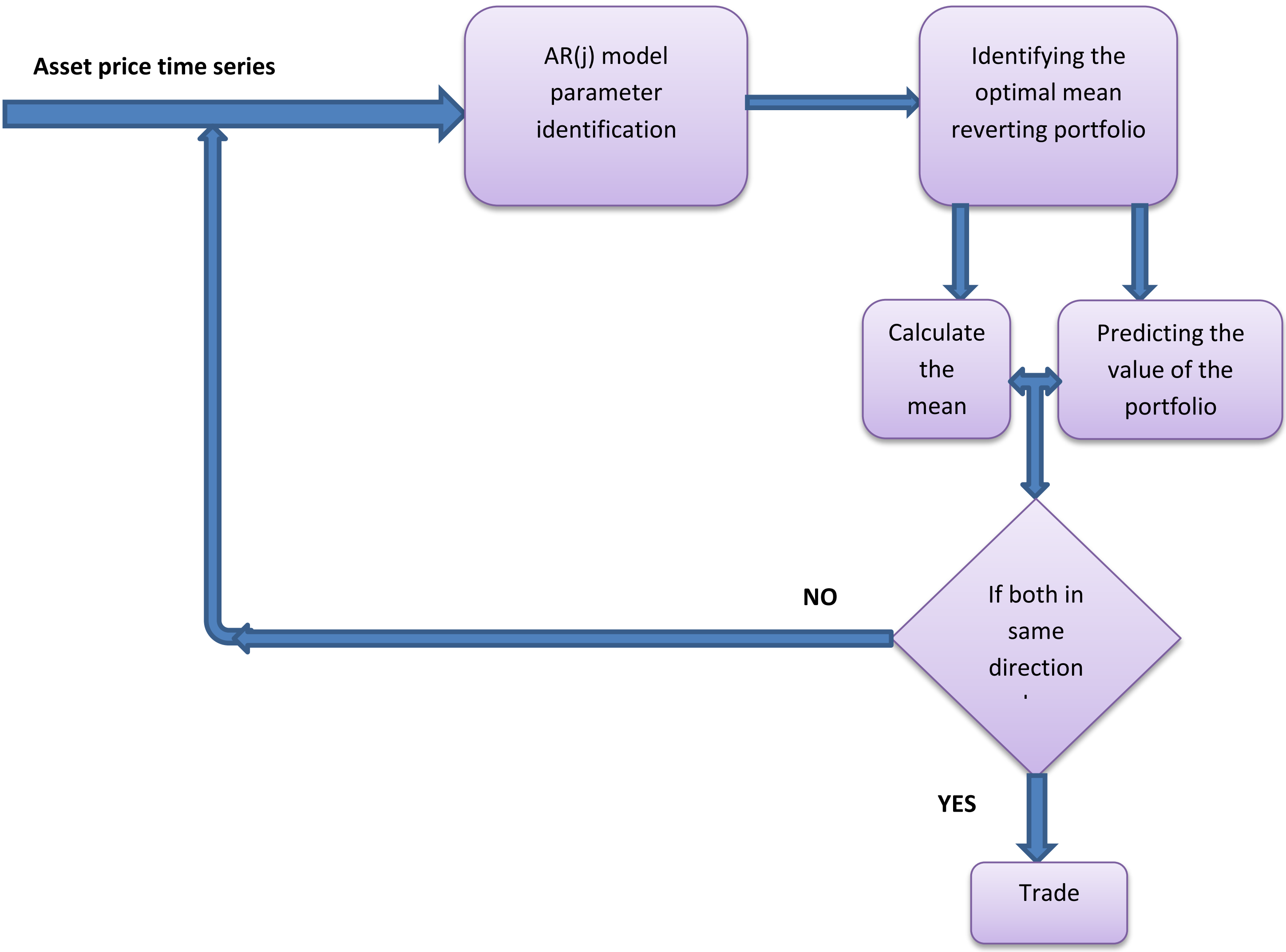

Fig. 22: The process of using prediction-based filter for mean reverting portfolios

3.3 Model parameter optimization

In our model there are some free parameters which can be subject to optimizations e.g. step, observation period, number of layers.

3.3.1 Optimizing the step

So far in our main model we only predicted the next candle (time instant), but we can also predict more than one candle. It can help us to better cover the spread and possibly extend our profit, as we let the price series change more dominantly to get out of the spread and materializing more profit. Here our goal is to find the optimal step parameter, where the step is denoted by l where $l = \{1..L\}$ and defined as how many candles in the future we predict.

$$x(k)...x(k-j) \rightarrow x(k+l)$$

The next figure shows the result regarding the step parameter, i.e. the account profit in percentage (optimization period is 6 months) is plotted as a function of the step parameter.

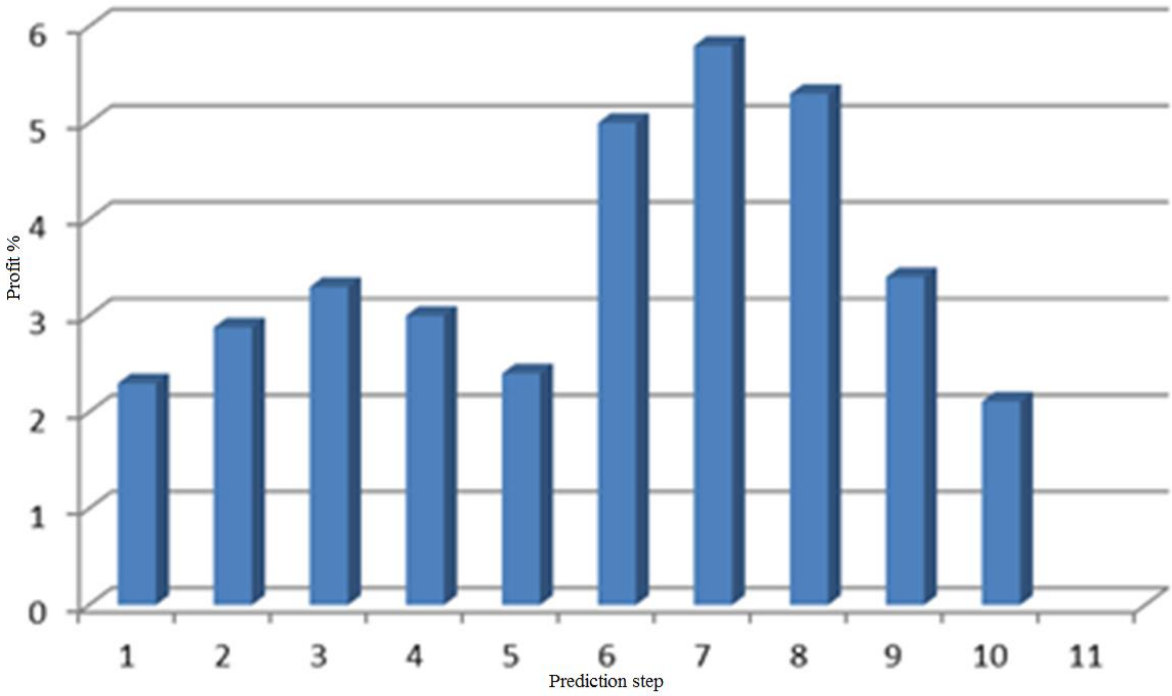

Fig. 23: Profit in percentage as a function of prediction step.

From Figure 23, one can conclude that the optimal step parameter is 7.

3.3.2 Dynamic observation period

I introduced a novel method to have better logical length for observation period and to have more important and meaningful points which can help us to avoid range market noises.

To find important points (IPs) I used Moving Average Convergence/Divergence oscillator (MACD) and with the help of it I found the maximum value and minimum value in each MACD phase change. The procedure is indicated in Figure 22. In methods proposed in 3.1.1 the observation period contains J consecutive candles which J is the number of the inputs of our network whereas in this new method still J is the number of the inputs but it is not anymore the observation period but the number of IPs. This approach makes our observation period dynamic since J number of IPs will be obtained from different period lengths which are strongly dependent to the market condition.

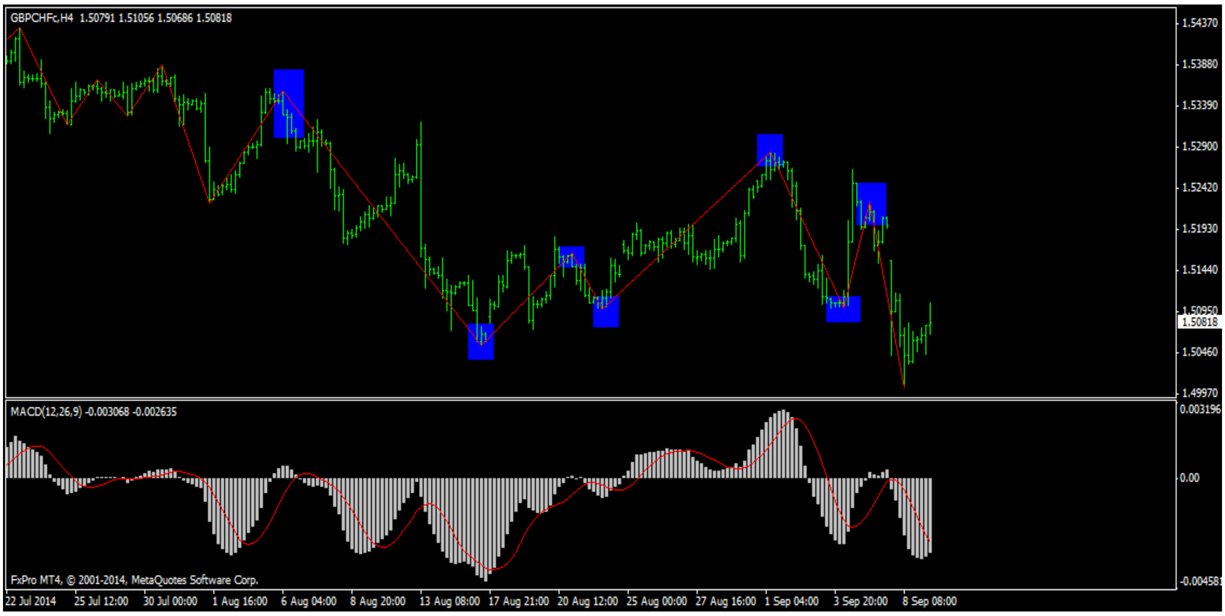

Fig. 24: The Process of finding the important points (IPs).

3.4 Performance analysis

We have tested the proposed method for predicting the price of a single asset and then the value of a selected mean reverting portfolio in three different cases:

- In the first case we predict the future price based on mid-price and we also trade on mid-price;
- In the second case we still predict by using the mid-price but we trade in the presence of Bid/Ask spread.
- In third case we predict by using Bid/Ask and also trade in presence of Bid/Ask spread.
- In forth case we show the result of using this model as a filter for mean reversion portfolios.
- In last case we show the performance of the dynamic observation period.

The testing parameters (period lengths, timeframe, J , initial deposit,...) are the same in all 3 cases and taken to be: Testing period length = 1 Month (2012.06.01-2012.07.02); J=5; Training period length=12 Months ; retraining period=20, timeframe=M15; single asset(EURUSD); Portfolio(EURUSD,GBPUSD,AUDUSD,NZDUSD; Data source Forex Metatrader[50] ; Initial deposit=1000. The neural network had three hidden layers, which can approximate any continuous function on according to the universal approximation theorem [51].

In the figures the number of trades is shown in the horizontal axis, while the account balance is indicated on the vertical axis.

3.4.1 Prediction and trading on mid-price

As was mentioned before, in the first case we only predict and trade on the mid-price. The results are indicated by figures 25 and 26, respectively.

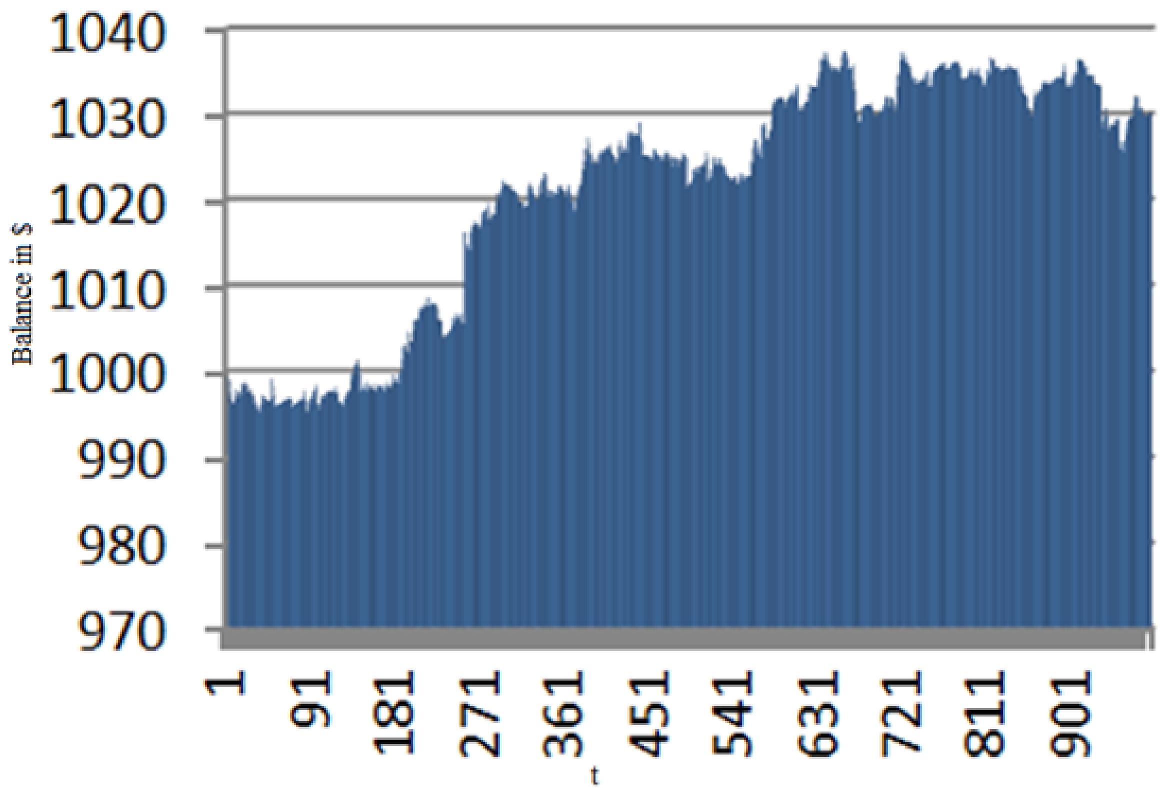

Fig. 25: Balance with respect to time (single asset).

One can see that we can achieve a 3 % or in quantity profit= $30.29; Profit=3%; MAX Drawdown=0.5%.

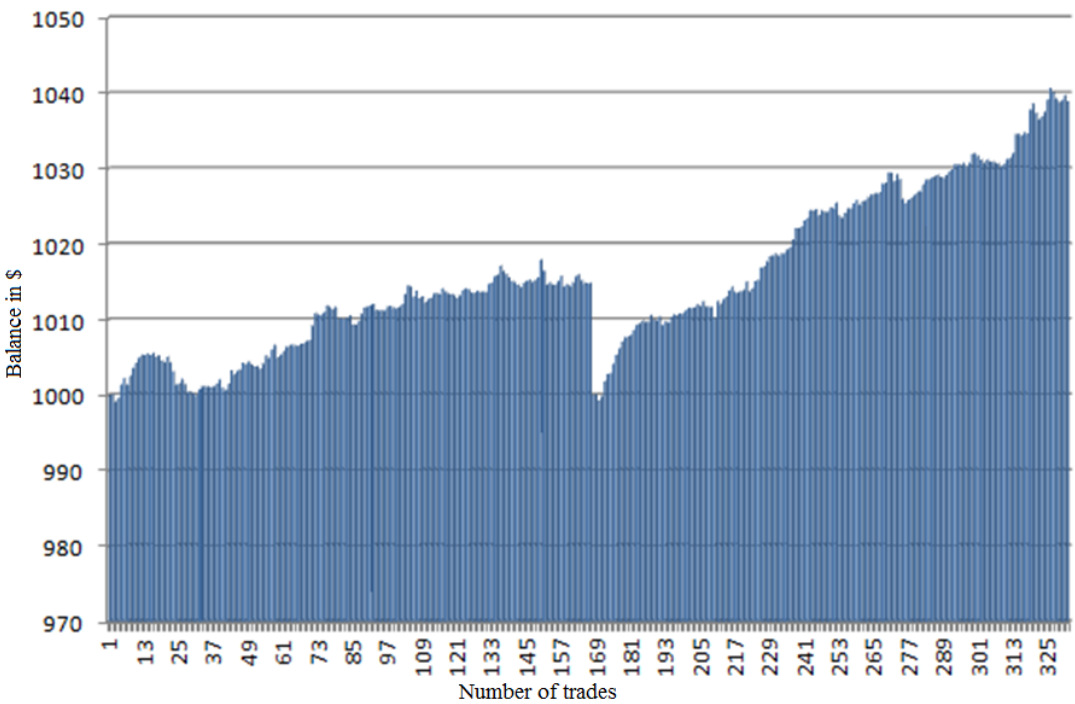

Fig. 26: Balance with respect to time (Portfolio).

One can see that we can achieve a 3.9 % or in quantity profit= $39.70; Profit=3.9%; MAX Drawdown=0.24%.

3.4.2 Training on the mid-price and trading on Bid/Ask

In this case we use simple model for prediction but we trade on Bid/Ask. Here on horizontal axis we have the same number of trades as in the previous figure.

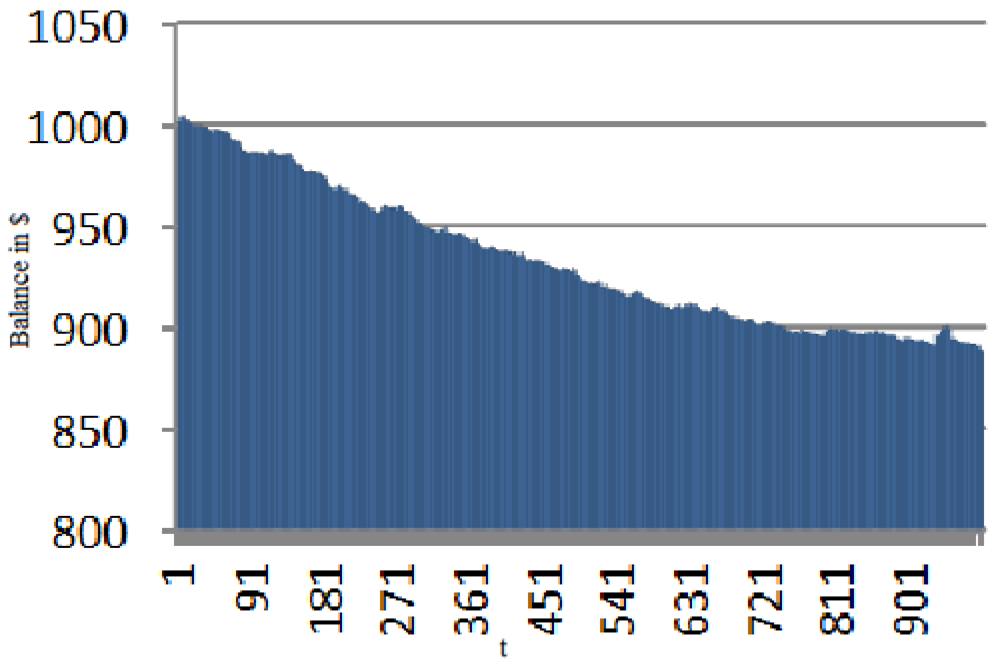

Fig. 27: Balance with respect to time (single asset).

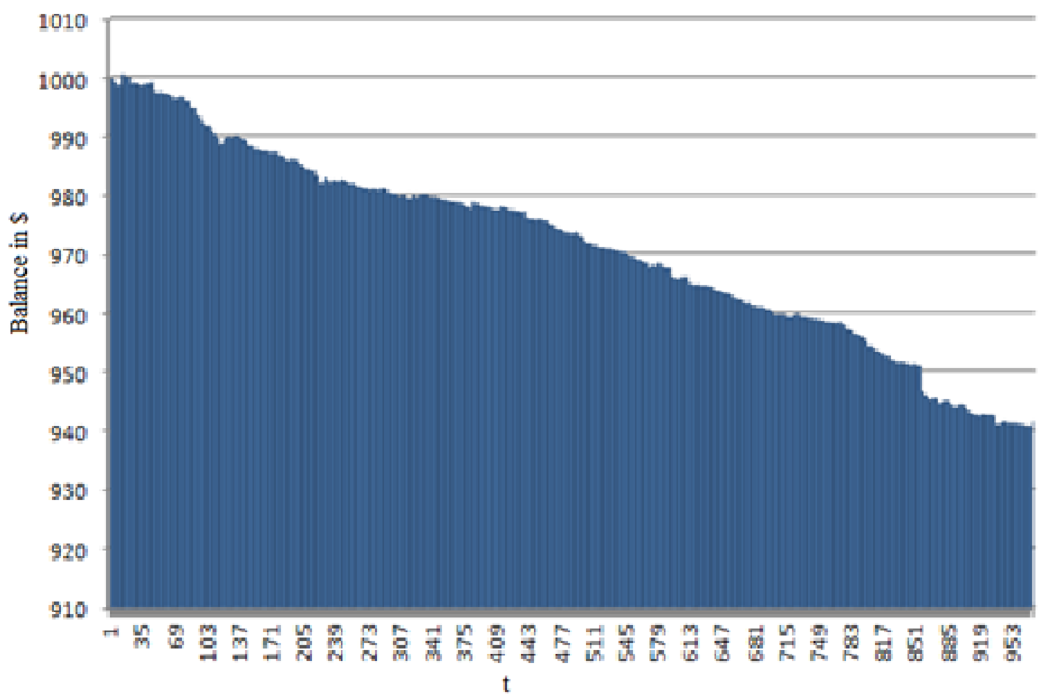

Fig. 28: Balance with respect to time (Portfolio).

The achieved profits are negative -10.70%,-5.8% (Profit: -110.70$, -58$) MAX Drawdown=3.4%, 2.1%.

The result is not so good as we have negative balance growth on vertical axis. However, as expected this is due to the fact we have not exploited the information given in the bid and ask series.

3.4.3 Training on the Bid/Ask and trading on the Bid/Ask

In the Fig. 29 we have smaller number of trades on horizontal axis in comparison to previous cases because it might happen that the predicted value is not greater than Ask Price or it is not even less than Bid Price, therefore in some cases we do not trade. Again on the vertical axis we have account balance but this time we have some positive growth.

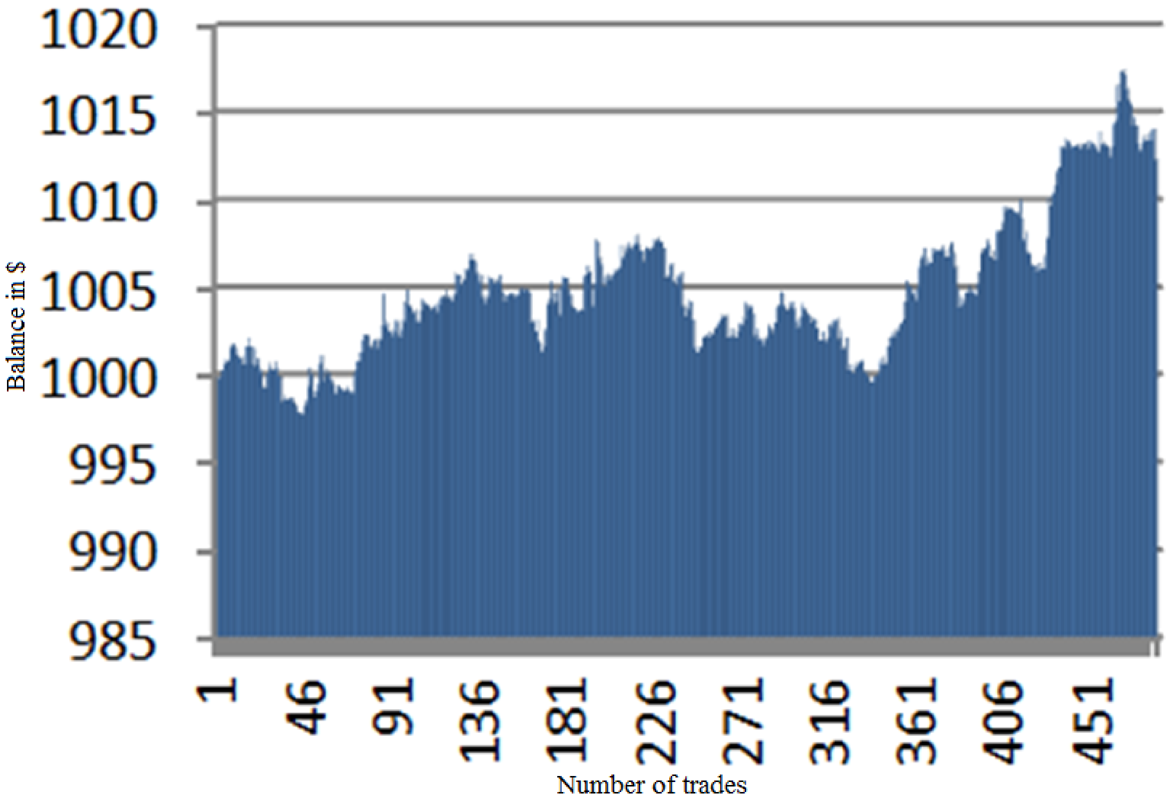

Fig. 29: Balance with respect to time (single asset).

The achieved profit 1.23 %(Profit: $12.30) which is good in the presence of bid-ask spread. MAX Drawdown=0.85%. One can see that even in the presence of bid-ask spread the method can materialize profit.

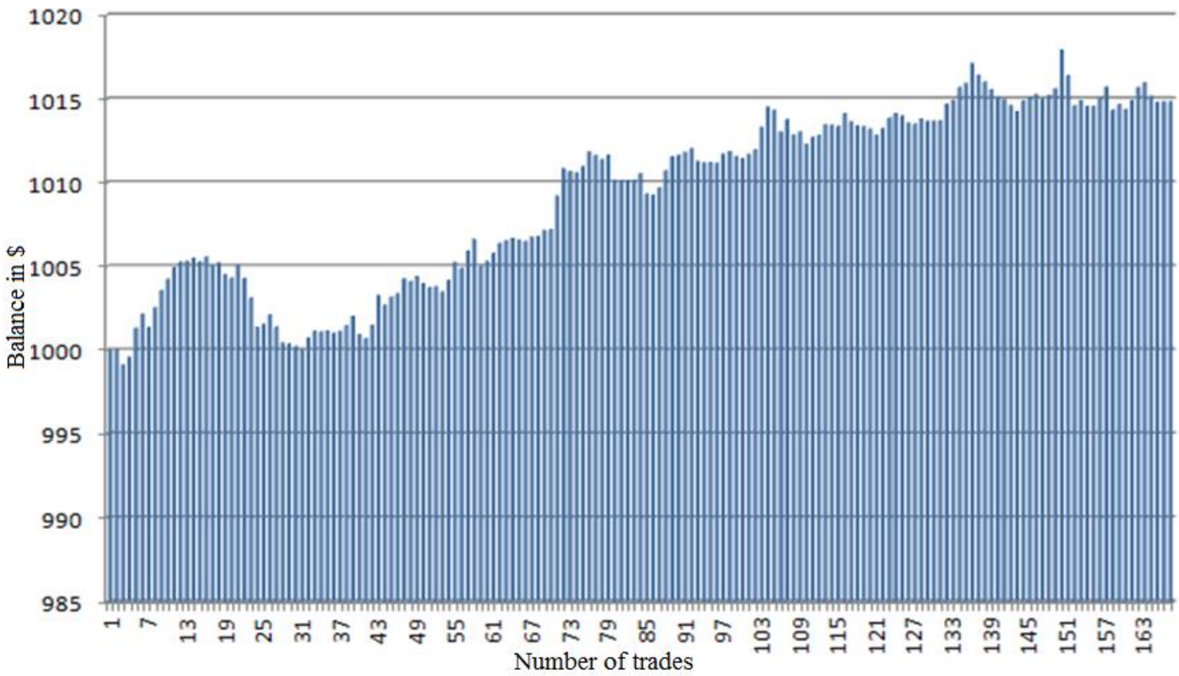

Fig. 30: Balance with respect to time (Portfolio).

The achieved profit is 1.48 %(Profit: $14.8) which is good in the presence of bid-ask spread. Max Drawdown=0.29%.

3.4.4 Using Prediction-Based model as an effective filter

We developed an effective filter based on proposed method which can be used with other trading strategies e.g. Mean reverting strategy to improve the performance of those strategies.

In figure 31 you can see how good this method improved the result of traditional mean reverting portfolio selection model with using this prediction-based filter. It is also clear from the figure that we can use this filter to reduce the drawdown.

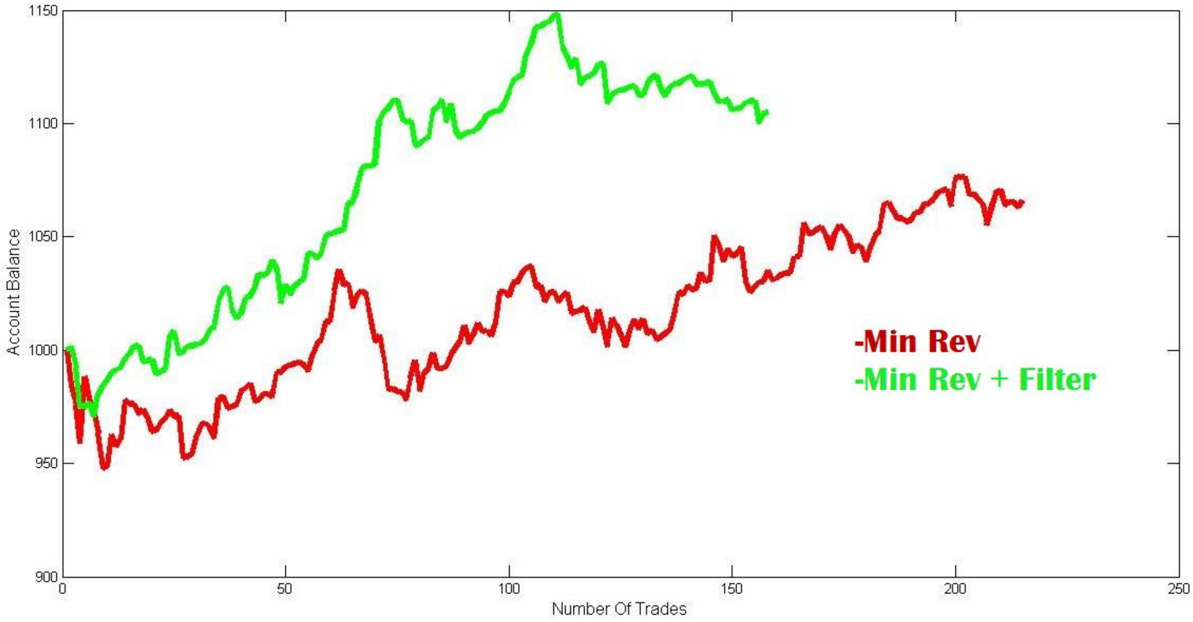

Fig. 31: Performance of Mean reverting portfolio selection method with and without using the prediction-based filter.

3.4.5 Performance of dynamic observation period

For the sake of comparing the performance, we tested the performance of dynamic observation period which is proposed in **3.3.2** against the performance of static observation period. Since the dynamic observation period is predicting the next swing and not the next time instant price, to have the comparison fair we use stop-loss instead of closing the position in next time instant. The stop-loss has been set in a way that the Risk/Reward ratio is 1.

As one can see from the figure 32, with dynamic observation period we can achieve higher and more stable results than using the static observation period.

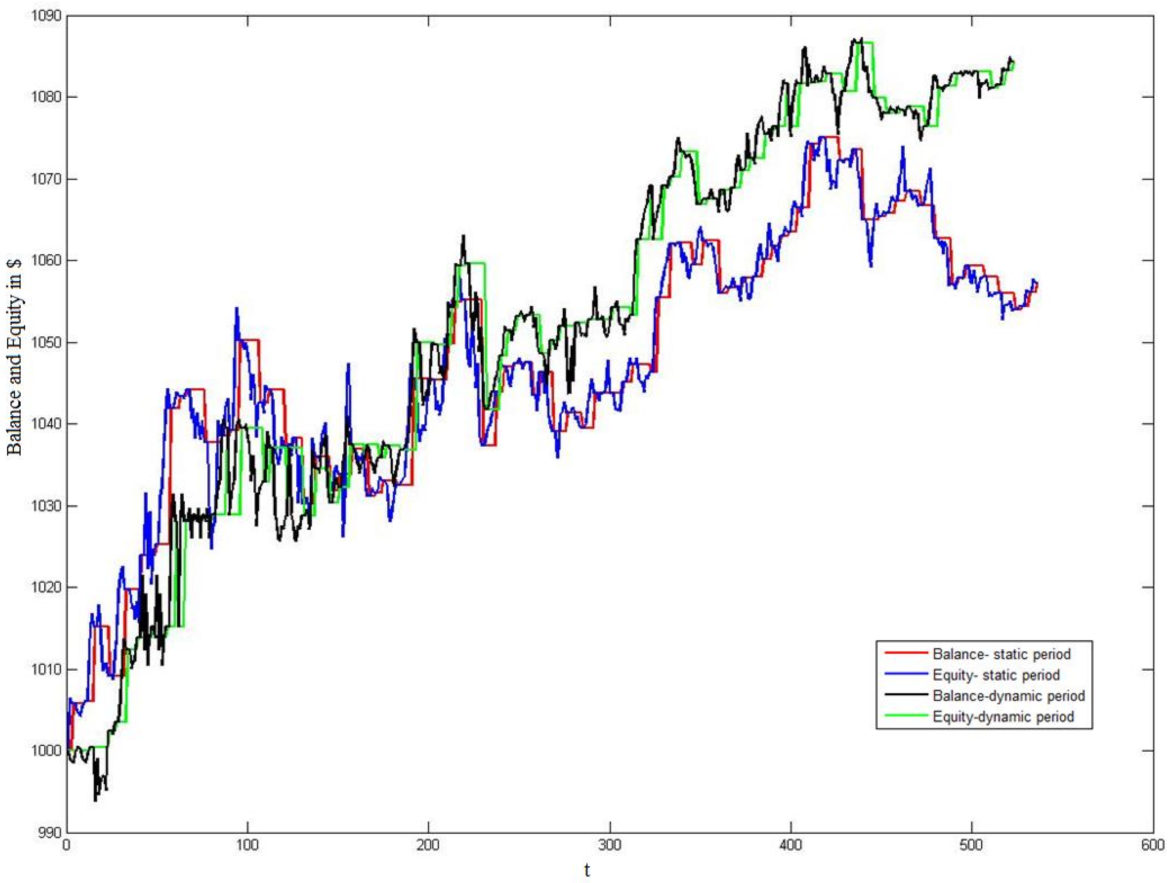

Fig. 32: Performance of dynamic vs. static observation period

3.5 Conclusion

In this chapter I used FFNN based prediction for trading on financial time series. The optimal trading strategy has been derived by using the fact that FFNN can represent the conditional expected value. Furthermore, I have optimized the prediction step parameter numerically. In the case of trading on the mid-price a considerable amount of profit can be accumulated. In the case of trading in the presence of bid-ask spread the method is still profitable but the achieved profit is more modest.

The methods presented here can pave the way towards high frequency, intraday trading. Furthermore, in our tests we did not use leverage, but with these low drawdowns which we had, we can easily use bigger leverages to magnify our profit. Although the ability to earn significant profits by using leverage is substantial, leverage can also work against investors. To avoid such a catastrophe, Forex traders usually use money management techniques.

CHAPTER 4

Summary of the Results

In this dissertation I have explored and examined two different aspects of computational finance: optimal portfolio selection based on min loss probability and prediction based trading. These methods were tested by back- and forward testing by using Matlab and Meta trader and their performance has indeed far outdone other traditional methods. In each case, I have managed to come up with novel approaches which

- can capture a wild class of random portfolio value sequence;
- can pave the way towards high frequency, intraday trading;
- can easily use bigger leverages to magnify our profit with low drawdowns.

Furthermore, I managed to improve the model identification methods as well. In the case of the sparse, mean reverting portfolio selection I have made significant improvements to the parameter estimation of the μ and λ . In the prediction based trading I introduced a novel method which make the observation period dynamic and more logical.

Considering the above results, I have achieved the aims of the dissertation stated in chapter1.

Finally, in each case I have implemented a proof of concept and have run extensive simulations on real world data. The results on real world data are convincing in each case.

4.1 Comparison of the strategies

Throughout the comparison the algorithms were tested on the same grounds (i.e. none of them was optimized).

We have three different main Models; Mean reversion, Levy and Prediction Based. In the first model we have 3 different objectives therefore we have comparison of five.

CASE	MODEL	OBJECTIVE	CONSTRAINTS
1	Mean Reversion	Maximizing the lambda	Price Movement
2	Mean Reversion	Maximizing the Profit	Price Movement
3	Mean Reversion	Minimizing the Probability of Loss	Time
4	Levy	Minimizing the Probability of Loss	Time
5	Prediction-Based	Predict the Next Candle	Time

Table 4. List of different trading strategies for different underlying models which is examined in this comparison

As we don't have any explicit constraint in first two objectives, therefore to keep the comparison fair we added the price movement constraint to them beside with this constraint we also control the risk and maximum drawdown of them. To make this movement-constraint dynamic and efficient, we used one of the most common parameters in trading which is called Risk/Reward Ratio.

In our model the reward is known and it is the profit which we can achieve if we reach the mean, therefore we can define our movement constraint using this Reward and our desired Risk/Reward Ratio.

Risk = Reward*Risk/Reward

In our simulation we Used R/R=1

As we don't have any explicit stop loss in the last three cases therefore to control the risk of these cases we used special pool of portfolios which has only the fully hedged portfolios.

In our test we have 4 different assets which have strong correlation so with having fully hedged portfolios we help our models to control the risk and also having these kind of pool help us to speed up the process of finding the best portfolio because our searching space is smaller.

CASE	Portfolio Pool	Stop Loss	Take Profit
1	Complete Pool	Based on Risk/Reward Ratio	Mean
2	Complete Pool	Based on Risk/Reward Ratio	Mean
3	Fully Hedged Pool	Based on waiting time	Based on waiting time
4	Fully Hedged Pool	Based on waiting time	Based on waiting time
5	Fully Hedged Pool	Based on waiting time(1)	Based on waiting time(1)

Table 5. List of different test-cases

Here we have a simple trading system:

- we take the best portfolio from each case.
- we put trades.
- we exit based on explicit stop loss and take profit point in first two cases or we close our trades in profit or in loss in last three cases.

Testing criteria

Number Of assets : 4 (EURUSD,GBPUSD,AUDUSD,NZDUSD)

Sparsity : 3

Time Frame : Daily

Testing Period : 2009-2012

Initial Deposit : 1000 USD

Leverage : 1-1

4.2 Comparison of results

In Figures 33 and 34 we have equity of the account on the vertical axis with respect to time on the horizontal axis. In figures 35 and 36 we have the balance of the account with respect to time .In figure 38 we have profit and maximum drawdown in percentage and in the last figure we tried to optimized the R/R value so we can see the profit in percentage on vertical axis with respect to different R/R value on horizontal axis. In figures 35 and 36 we skipped the Price Movement constraint.

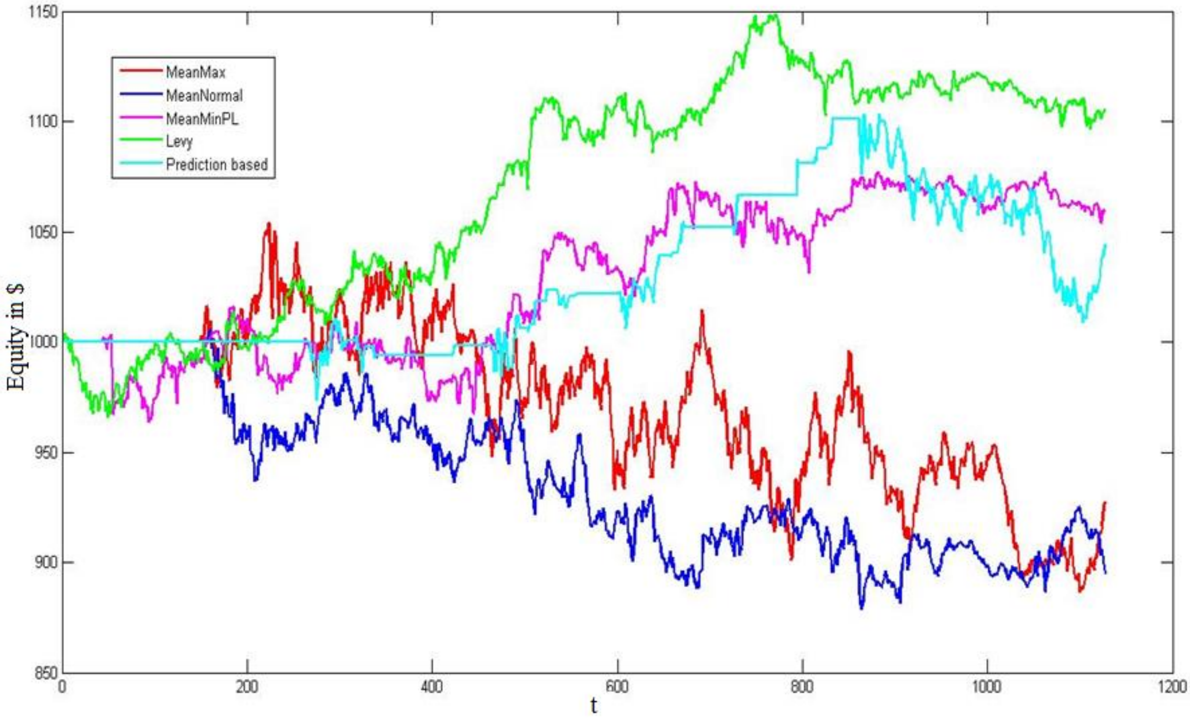

Fig. 33: Equity in Presence of Stop loss

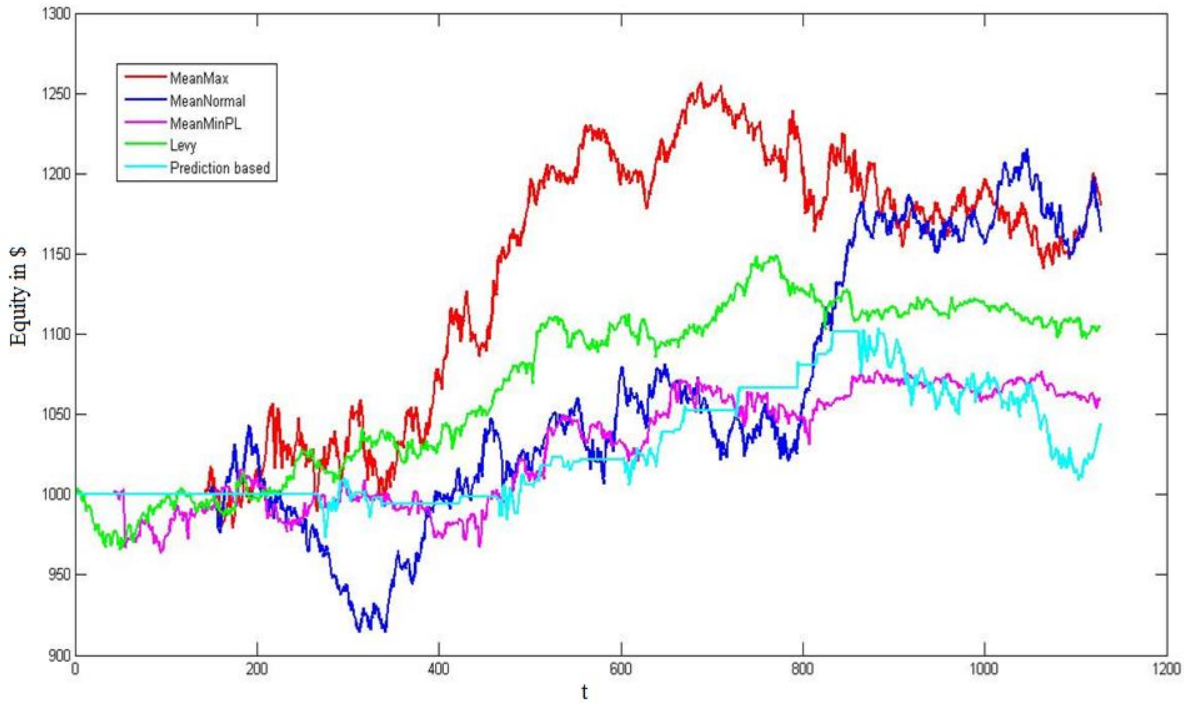

Fig. 34: Equity in the absence of Stop loss

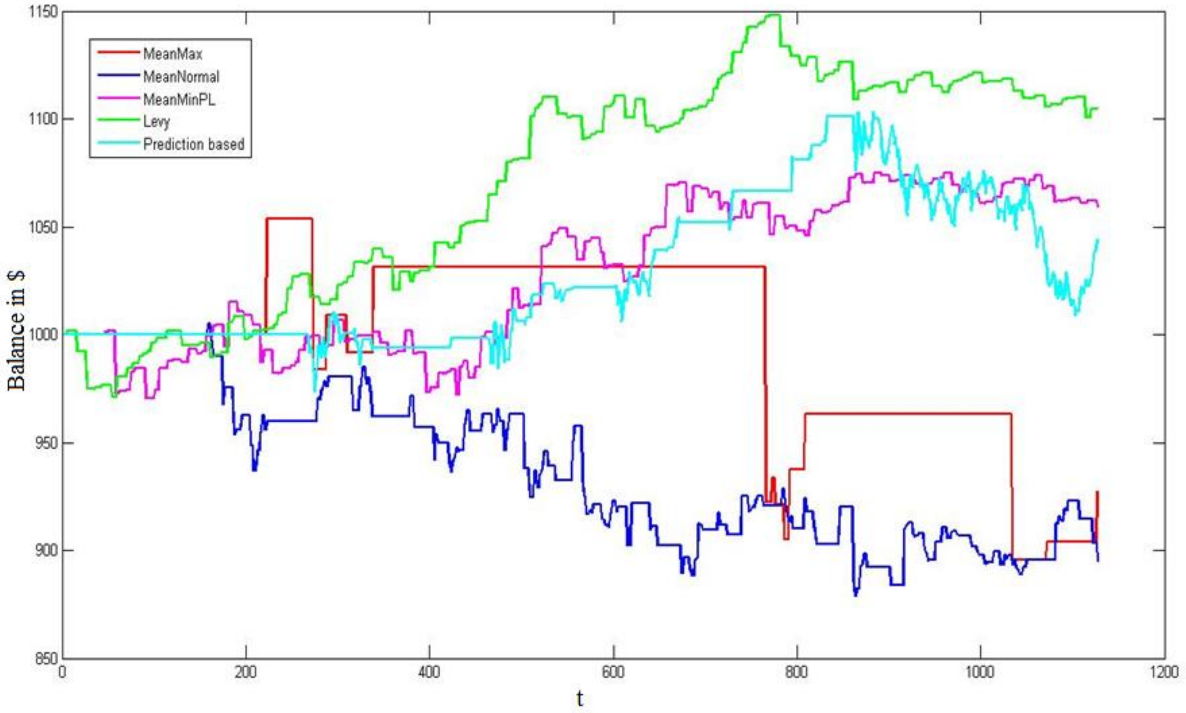

Fig. 35: Balance in Presence of Stop loss

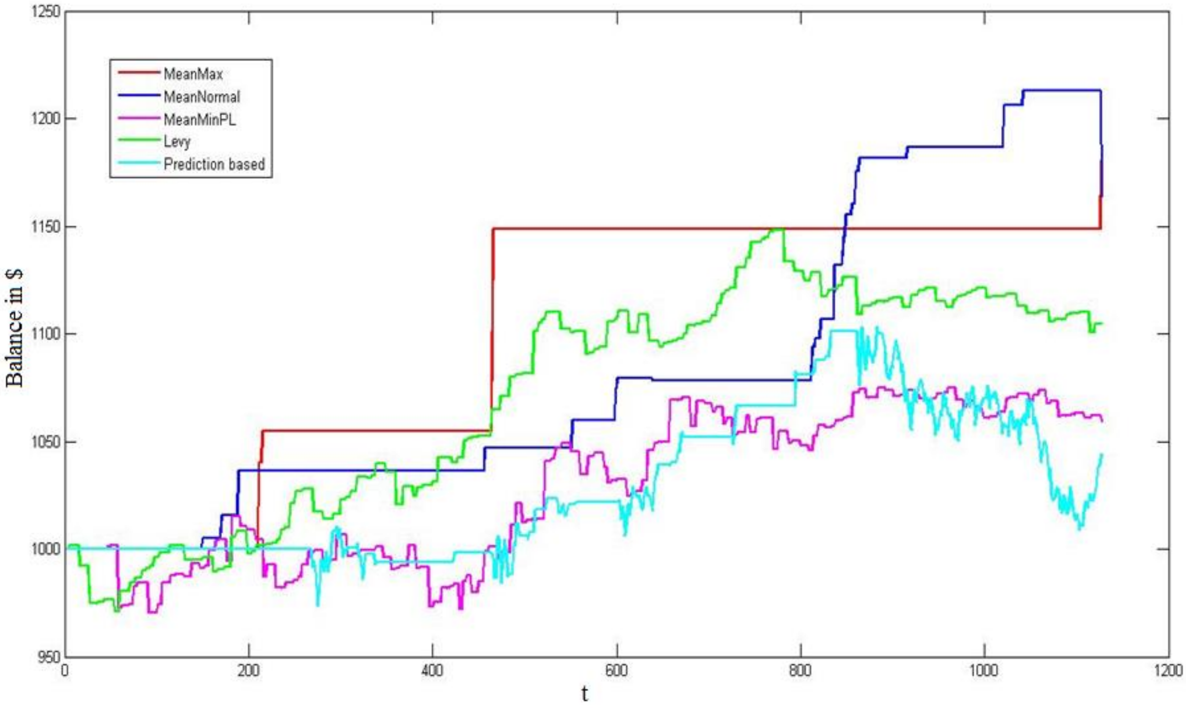

Fig. 36: Balance in the absence of Stop Loss

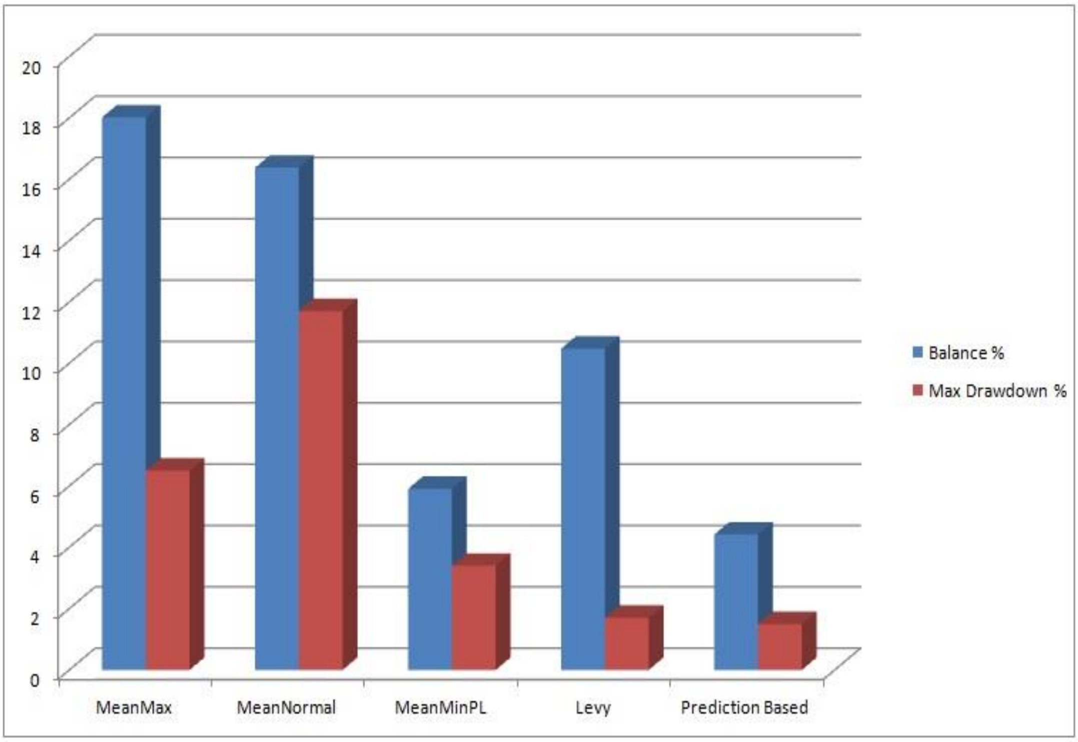

Fig. 37: Profit & Max Drawdown in absence of Stop Loss

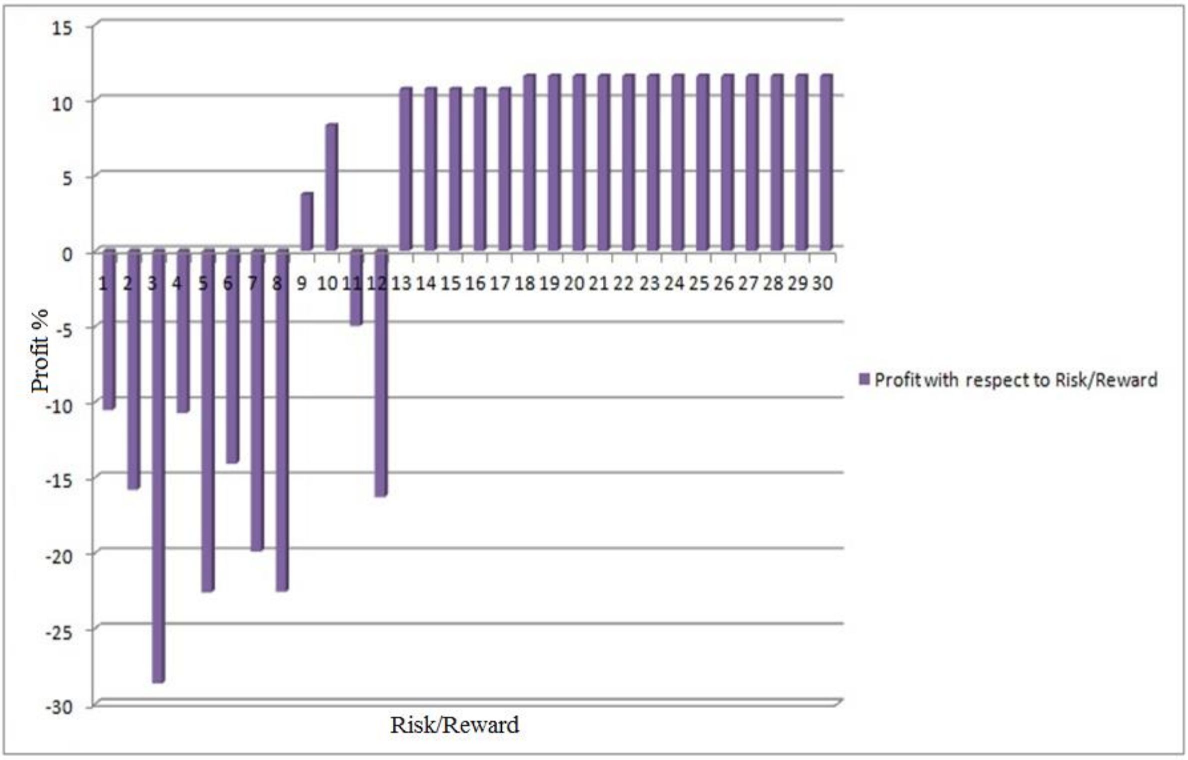

Fig. 38: Profit With respect to R/R

As we can get from the figures, the maximum drawdown of first two cases in the presence of stop loss is too high .although they have the biggest profit but they might be too risky. However from the last figure we can come to the conclusion that we should skip the stop loss to make them profitable.

From figure 37 we can see that the best model considering both parameters together (Profit and Risk) is levy model. With having that small Drawdown we can use bigger leverages and magnifying our profit.

4.3 Final conclusions

This dissertation provides a detailed examination of novel portfolio selection methods. The main motivation is that in most cases a risk is an important aspect which might be more interesting than the profit itself for traders; i.e. the highest priority is not to lose money rather than making money.

When examining the different trading strategies the results show that the most significant drawback of traditional portfolio selection methods is a very bad and high Risk/Reward ratio. The higher R/R is the higher risk becomes, thus in contrast to the traditional methods mentioned in this dissertation, the main goal of our novel methods is to minimize the risk rather than maximize the profit.

In the real world, risk-to-reward ratios are not set in stone. They must be adjusted depending on the time frame, trading environment, and your entry/exit points. Since the entry point has an important role in R/R, we tried to improve traditional strategies by employing the power of neural networks in order to achieve better entry points, to filter-out some bad signals and finally to have better R/R.

In each trading decision we need to answer the following questions:

- What is the direction (Buy/Sell/no Signal)?
- Which is the good entry point?
- When should we close our position?
- What is the Risk of this position? (Money management)

After making a system which can answer all of these questions we should also have fast implementation which does not miss the entry point and fast enough to monitor the open position in case of having dynamic exit point.

As a final conclusion , we can say that with the new strategies proposed here we can implement a system which has answers for all those questions and eventually can make algo-trading safer and more practical.

CHAPTER 5

LIST OF REFERENCE PUBLICATIONS

5.1 Publications of the author

Journal Publications

1. Kia, F., G. Jeney, and J. Levendovszky, *Minimum Probability of Loss Trading Strategy for Mean Reverting Portfolios.* Interdisciplinary Journal of Scientific Research 2014. **1**(3): p. 53-63.
2. Kia, F., G. Jeney, and J. Levendovszky, *Loss-minimal Algorithmic Trading Based on Levy Processes.* TEMJOURNAL, 2014. **3**(3): p. 210-215.
3. Kia, F., Levendovszky, J. (2012) Prediction based – high frequency trading on financial time series. Periodica Polytechnica 56/1 (2012) 29–34 doi: 10.3311/PPee.7165.

Conference Presentations

4. Kia, F., G. Jeney, and J. Levendovszky. *Minimum Probability Of Loss Trading Strategy for Mean Reverting Portfolios*. in *10th International Symposium on Business Information Systems*. November 2013. Győr, Hungary.
5. Kia, F. and J. Levendovszky. *Prediction Based-High Frequency Trading On Financial Time Series*. in *5th International Conference on Neural Computation Theory and Applications*. September 2013. Algrave, Portugal.

5.2 Publications connected to the dissertation

6. Pole, A., *Statistical arbitrage: Algorithmic trading insights and techniques*. Vol. 411. 2011: John Wiley & Sons.
7. Kissell, R.L., *Algorithmic trading strategies.* 2006.
8. Bergan, P. and C. Devine, *Algorithmic Trading: What Should You Be Doing?* Trading, 2005. **2005**(1): p. 14-18.
9. Hanif, A. and R.E. Smith, *Algorithmic, Electronic, and Automated Trading.* The Journal of Trading, 2012. **7**(4): p. 78-86.
10. Blackwell, P.G., *Ornstein–Uhlenbeck process.* Encyclopedia of Biostatistics, 1998.
11. Wim, S., *Lëvy Processes in Finance Pricing Financial Derivatives.* Chapter, 2003. **5**: p. 82.
12. d'Aspremont, A., *Identifying small mean-reverting portfolios.* Quantitative Finance, 2011. **11**(3): p. 351-364.
13. Balvers, R., Y. Wu, and E. Gilliland, *Mean reversion across national stock markets and parametric contrarian investment strategies.* Journal of Finance, 2000: p. 745-772.
14. Fouque, J.-P., G. Papanicolaou, and K.R. Sircar, *Mean-reverting stochastic volatility.* International Journal of theoretical and applied finance, 2000. **3**(01): p. 101-142.
15. Hillebrand, E., *Mean reversion models of financial markets*. 2003: Universitätsbibliothek Bremen.
16. Poterba, J.M. and L.H. Summers, *Mean reversion in stock prices: Evidence and implications.* Journal of financial economics, 1988. **22**(1): p. 27-59.

17. Wachter, J.A., *Portfolio and consumption decisions under mean-reverting returns: An exact solution for complete markets.* Journal of financial and quantitative analysis, 2002. **37**(01): p. 63-91.
18. Markowitz, H., *Portfolio selection*.* The journal of finance, 1952. **7**(1): p. 77-91.
19. Cai, L., *Predictability and Hedge Fund Investing.* Journal of Investment Consulting, 2012. **13**(1): p. 30-39.
20. Grant, J., *High-frequency trading: Up against a bandsaw.* Financial Times. Sept, 2010. **10**: p. 2010.
21. Brigo, D., et al., *A stochastic processes toolkit for risk management.* Available at SSRN 1109160, 2007.
22. Rachev, S.T. and S.V. Stoyanov, *Advanced Stochastic Models, Risk Assessment, and Portfolio Optimization: The Ideal Risk, Uncertainty, and Performance Measures (Frank J. Fabozzi Series).* 2008.
23. Matacz, A., *Financial modeling and option theory with the truncated Lévy process.* International Journal of Theoretical and Applied Finance, 2000. **3**(01): p. 143-160.
24. Samuelson, P., *The Foundations of Economic Analysis.* Bull. Amer. Math. Soc. 56 (1950), 266-267 doi. org/10.1090/S0002-9904-1950-09379-3 PII, 1950: p. 0002-9904.
25. Samuelson, P.A., *Rational theory of warrant pricing.* IMR; Industrial Management Review (pre-1986), 1965. **6**(2): p. 13.
26. Gardiner, C.W., *Handbook of stochastic methods*. Vol. 4. 1985: Springer Berlin.
27. Tankov, P., *Financial modelling with jump processes*. Vol. 2. 2003: CRC press.
28. Box, G.E. and G.C. Tiao, *A canonical analysis of multiple time series.* Biometrika, 1977. **64**(2): p. 355-365.
29. Oja, E., *Principal components, minor components, and linear neural networks.* Neural Networks, 1992. **5**(6): p. 927-935.
30. Eraker, B., M. Johannes, and N. Polson, *The impact of jumps in volatility and returns.* The Journal of Finance, 2003. **58**(3): p. 1269-1300.
31. Li, H., M.T. Wells, and L.Y. Cindy, *A Bayesian analysis of return dynamics with Lévy jumps.* Review of Financial Studies, 2008. **21**(5): p. 2345-2378.
32. Javaheri, A., *The volatility process: a study of stock market dynamics via parametric stochastic volatility models and a comparaison to the information embedded in the option price*. 2004, École Nationale Supérieure des Mines de Paris.
33. Pantazopoulos, K.N., et al., *Financial prediction and trading strategies using neurofuzzy approaches.* Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 1998. **28**(4): p. 520-531.
34. Schöneburg, E., *Stock price prediction using neural networks: A project report.* Neurocomputing, 1990. **2**(1): p. 17-27.
35. Kim, K.-j., *Financial time series forecasting using support vector machines.* Neurocomputing, 2003. **55**(1): p. 307-319.
36. Kryzanowski, L., M. Galler, and D.W. Wright, *Using artificial neural networks to pick stocks.* Financial Analysts Journal, 1993. **49**(4): p. 21-27.
37. SwalesGeorge Jr, S., *Applying artificial neural networks to investment analysis.* Financial Analysts Journal, 1992. **48**(5).
38. Trippi, R.R. and D. DeSieno, *Trading equity index futures with a neural network.* The Journal of Portfolio Management, 1992. **19**(1): p. 27-33.
39. Grudnitski, G. and L. Osburn, *Forecasting S&P and gold futures prices: An application of neural networks.* Journal of Futures Markets, 1993. **13**(6): p. 631-643.
40. Dunis, C.L., J. Laws, and G. Sermpinis, *Modelling and trading the EUR/USD exchange rate at the ECB fixing.* The European Journal of Finance, 2010. **16**(6): p. 541-560.
41. Yao, J. and C.L. Tan, *A case study on using neural networks to perform technical forecasting of forex.* Neurocomputing, 2000. **34**(1): p. 79-98.

42. Kaastra, I. and M. Boyd, *Designing a neural network for forecasting financial and economic time series*. Neurocomputing, 1996. **10**(3): p. 215-236.
43. Refenes, A.N., A. Zapranis, and G. Francis, *Stock performance modeling using neural networks: a comparative study with regression models.* Neural networks, 1994. **7**(2): p. 375-388.
44. Yoon, Y., T. Guimaraes, and G. Swales, *Integrating artificial neural networks with rule-based expert systems.* Decision Support Systems, 1994. **11**(5): p. 497-507.
45. Waibel, A., *Modular construction of time-delay neural networks for speech recognition.* Neural computation, 1989. **1**(1): p. 39-46.
46. Ciresan, D.C., et al., *Deep, big, simple neural nets for handwritten digit recognition.* Neural computation, 2010. **22**(12): p. 3207-3220.
47. Ciresan, D., U. Meier, and J. Schmidhuber. *Multi-column deep neural networks for image classification*. in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. 2012. IEEE.
48. Cartea, A. and S. Jaimungal, *Modelling asset prices for algorithmic and high-frequency trading.* Applied Mathematical Finance, 2013. **20**(6): p. 512-547.
49. Chauvin, Y. and D.E. Rumelhart, *Backpropagation: theory, architectures, and applications*. 1995: Psychology Press.
50. Corp., M.S., *METATRADER FOREX time series* [WWW]. Available from:http://www.metaquotes.net/en/metatrader4, 2014.
51. Gybenko, G., *Approximation by superposition of sigmoidal functions.* Mathematics of Control, Signals and Systems, 1989. **2**(4): p. 303-314.