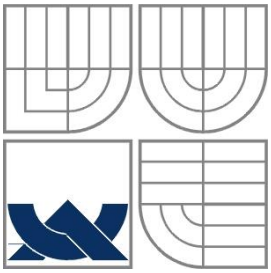

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

THE USE OF MEANS OF ARTIFICIAL INTELLIGENCE FOR THE DECISION MAKING SUPPORT ON FINANCIAL MARKETS

VYUŽITÍ PROSTŘEDKŮ UMĚLÉ INTELIGENCE NA FINANČNÍCH TRZÍCH

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. JIŘÍ MIKLÓSY
AUTHOR

VEDOUCÍ PRÁCE — prof. Ing. PETR DOSTÁL, CSc.
SUPERVISOR

BRNO 2013

Vysoké učení technické v Brně
Fakulta podnikatelská

Akademický rok: 2012/2013
Ústav informatiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Miklósy Jiří, Bc.

Informační management (6209T015)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává diplomovou práci s názvem:

Využití prostředků umělé inteligence na finančních trzích

v anglickém jazyce:

The Use of Means of Artificial Intelligence for the Decision Making Support on Financial Markets

Pokyny pro vypracování:

Úvod
Vymezení problému a cíle práce
Teoretická východiska práce
Analýza problému a současné situace
Vlastní návrhy řešení, přínos návrhů řešení
Závěr
Seznam použité literatury
Přílohy

Seznam odborné literatury:

DOSTÁL, P. Pokročilé metody rozhodování v podnikatelství a veřejné správě. Brno: Akademické nakladatelství CERM, 2012. 718 s. ISBN 978-80-7204-798-7. e-ISBN 978-80-7204-799-4.
DOSTÁL, P. Advanced Decision Making in Business and Public Services. Brno: Akademické nakladatelství CERM, 2011. 168 s. ISBN 978-80-7204-747-5.
HANSELMAN, D. a B. LITTLEFIELD. Mastering MATLAB7. Pearson Education International Ltd., 2005. 852 s. ISBN 0-13-185714-2.
REJNUŠ, O. Finanční trhy. Ostrava: Key Publishing, 2008. 559 s. ISBN 978-80-87071-87-8.

Vedoucí diplomové práce: prof. Ing. Petr Dostál, CSc.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2012/2013.

L.S.

doc. RNDr. Bedřich Půža, CSc.	doc. Ing. et Ing. Stanislav Škapa, Ph.D.
Ředitel ústavu	Děkan fakulty

V Brně, dne 13.05.2013

Abstrakt

Tato práce se zabývá návrhem, realizací a optimalizací systému určenímu k obchodování na finančních trzích, konkrétně s technologickými firmami trhu NASDAQ. K tomuto účelu jsou využívány technické indicatory a hlavně neuronových sítí. Vlastní řešení je pak realizováno v prostředi MATLAB.

Abstract

This master thesis deals with design, realization and optimization of a trading system which proceeds its trades on financial markets. The system focuses on stocks which fall in a technology category on NASDAQ market. In order to build this system, technical indicators and mainly neural networks are used. The solution of automated trading system is built in MATLAB environment.

Klíčová slova

Neuronové sítě, neuron, NARX, technické indikátory, entropie, Hurst, MATLAB, časové řady, predikce, NASDAQ, MACD, VIX

Keywords

Neural network, neuron, NARX, technical indicators, entropy, Hurst, MATLAB, time series, prediction, NASDAQ, MACD, VIX

Citace

MIKLÓSY, J. *The Use of Means of Artificial Intelligence for the Decision Making Support on Financial Markets.* Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2013. 86 s. Vedoucí diplomové práce prof. Ing. Petr Dostál, CSc..

Prohlášení

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).
V Brně dne 1. května 2013

……
Jiří Miklósy

Poděkování

Děkuji vedoucímu mé diplomové práce panu prof. Ing.Petru Dostálovi, CSc., mé rodině, přítelkyni a kamarádům za cenné rady a podporu poskytnuté během studia a při zpracovávání diplomové práce.

Table of contents

1 Introduction

Over the centuries traders have been trying to build more or less successful trading systems. Most of the companies and individuals still think that investing in stock markets brings the highest return on investment. These people need two things – money and system. Money usually comes from companies' good financial situation, while putting money into a bank account does not bring wanting return on investment. The system then may be either analytic, which uses its approach to each stock or some automated trade system.

With the fast evolution of computer science, recent studies in that field has shown that artificial intelligence may help to build this system and provide continuous good predictability. The automated system brings many advantages to the trading like avoiding bias or fast decision making.

This thesis mainly deals with stock market prediction. The output should be a model written in MATLAB which will be able to predict future prices of chosen stocks. The thesis has been separated into three parts. Later on in my thesis I will be examining the mean of each part of created model as much as generated results.

In the first part of my thesis I will introduce the reader to a problem and the basic knowledge required to understand the whole workflow. There are not missing parts devoted to charting, technical analysis or experimental techniques in trading. The second part will be devoted to market research and evaluation of the current situation. And, the final part will describe the model design and provide the reader with generated results.

2 Aim of the work

This work aims at describing and implementing an automated trading system. This system will proceed trades on historical data and produce virtual profit over the trading period. There will be three tests conducted with a final version of the automated trading system and the majority of them must be considered as successful, thus generate profit. Also, the used terms and trading techniques need to be covered in the thesis.

3 Theoretical basis of the work

Working on an automatic trade system requires knowledge which are presented in this chapter. After reading, the reader will be aware of most of the terms and theories needed for the system construction.

3.1 Markets

There are different types of markets and different types of goods to purchase. The skilled trader should be briefly able to describe all of them. In the following text the reader gets to know which goods one can speculate on.

3.1.1 Forex

It is a marketplace for currency exchange which takes place in major financial centers around the world such as Paris, Zurich, London, Tokyo, Frankfurt etc. Forex market is the largest most liquid market in the world with a traded value that exceeds $1.9 trillion per day. Unfortunately for traders, the daily fluctuation is very small and does not exceed 1%. This makes Forex the least volatile financial market. Therefore, the traders rely on the availability of a huge leverage to increase the value of potential movements, which might be very risky.

The availability of high leverage and liquidity has made the market grow. The reason why the traders prefer Forex over other markets is mainly the size of the market, when the price is created by the actual supply and demand. Therefore neither of the big traders (for example banks) can easily move with the price. (investopedia, 2013)

3.1.2 Stock

Stock is a security which gives one a partial ownership in a corporation. The buyer becomes a member of the owners group, thus he may claim for corporation's assets and earnings. The ownership is determined by the number of shares a person owns in comparison to a number of outstanding shares.

There are two types of stock:

- Common stock – entitles the owner to receive dividends and nonetheless vote at stakeholders' meetings.

- Preferred stock – has higher claim on asset earnings than common stock. The preferred stock disadvantage is absence of voting rights a stakeholders' meetings. (investopedia, 2013)

3.1.3 Options

They are a financial contract between seller and buyer which offers the buyer the right, but not the obligation to purchase the asset at an agreed price (also called the strike price). This contract can be closed by the buyer at any time but the latest on the exercise date. There are two types of options.

- Call options – give the right to buy an asset at a certain price, therefore the buyer wants the stock to go up.
- Put options - give the right to sell an asset at a certain price, therefore the buyer wants the stock to go down. (investopedia, 2013)

3.1.4 Futures

Futures are financial contracts which obligate the buyer to purchase the asset and the seller to sell the asset. This transaction has to be fulfilled on the particular date mentioned in the contract. The involved asset is usually well described, therefore the amount of asset to purchase is strictly enforced.

The futures are usually used for market speculations, but also for reducing a risk. For example farmers may reduce risk and lock the good's price with Futures. Last but not least, it is necessary to say that Futures markets are extremely liquid but also very risky and traders should be aware of that. (investopedia, 2013)

3.1.5 Bonds

Bonds are a debt investment. The investor/buyer loans money to the seller/issuer that borrows the fund for a fixed interest rate and an exact time period. Interests on bonds are paid semi-annually thus every six months. Bonds are usually used for companies' expansion or other projects which may pay back in the future. (investopedia, 2013)

3.2 Charts

Marketers often use two main input variables. Volume, which represents a number of trades held on a particular day, and price which can be represented as:

- **Open** – stands for opening price on a particular day.
- **Low** – stands for lowest price on a particular day.
- **High** – stands for highest price on a particular day.
- **Close** – stands for closing price on a particular day.

Usually, charts use all four input prices with different styles of notation. The following text describes the major ones.

3.2.1 Bar chart

The bar chart is the most used one. The following figure represents the daily bar chart. The name "bar chart" comes from the chart's construction. In this example the chart is made by vertical bars which represent every day price range. The bar chart shows *open*, *high*, *low* and closing prices. The notation and common habit shows the opening price as tic at the left of the bar and closing price as a tic at the right of the bar. (MURPHY, 1999)

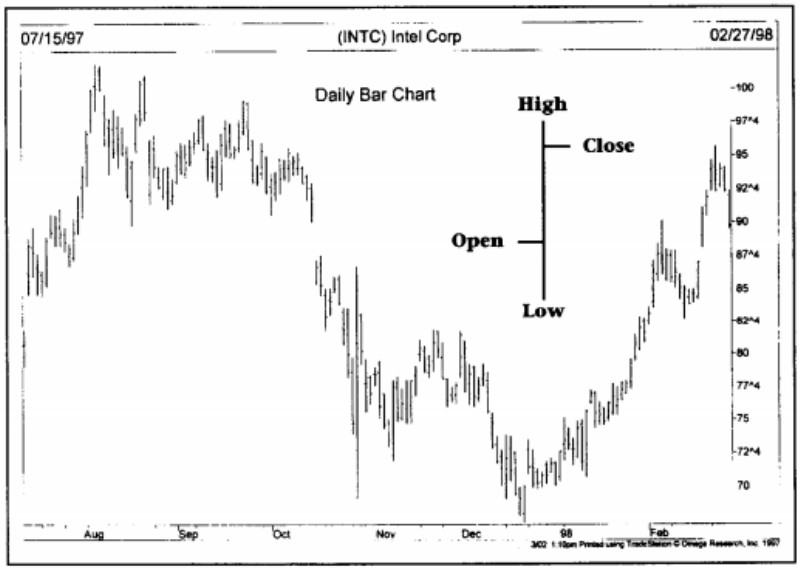

Figure 1 :Daily bar chart of Intel, source: (MURPHY, 1999)

3.2.2 Line chart

In a line chart only the closing price is plotted for each trading day. Most traders believe that the closing price is the most important from the whole trading day. Therefore, the line chart with *close*-only prices is more valid measure of price activity. (MURPHY, 1999)

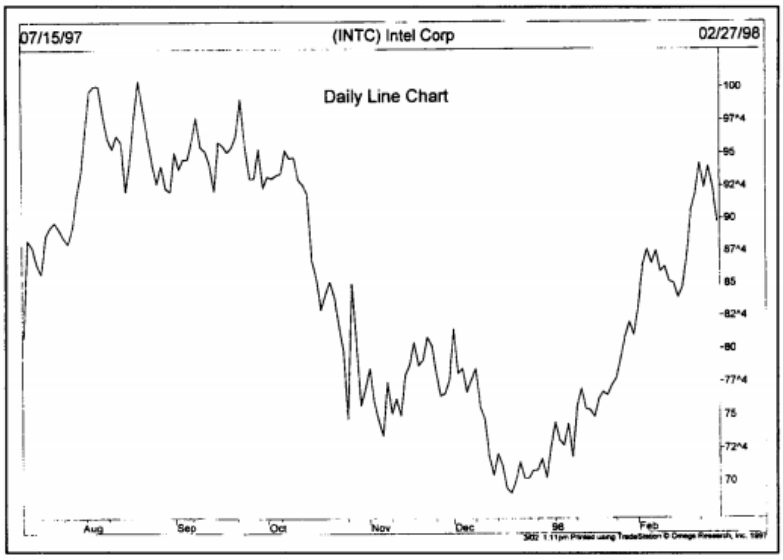

Figure 2: Line chart for Intel, source: (MURPHY, 1999)

3.2.3 Point and figure chart

This chart differs with alternating column of x's and o's. The o column shows declining prices while the x column shows rising ones. In comparison to a bar chart, the buy and sell signals are more precise and easier to spot. (MURPHY, 1999)

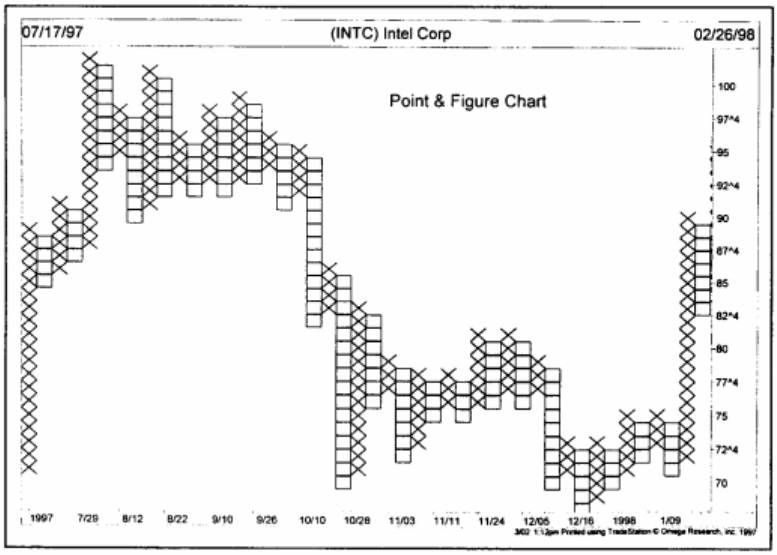

Figure 3: A point and figure chart of Intel, source: (MURPHY, 1999)

3.2.4 Candlesticks

Is a Japanese version of bar chart. The candlestick chart records the same variables as a bar chart does (*open*, *high*, *low*, *close*) the visual presentation differs though. The Figure 4 shows the candlestick chart.

On the candlestick chart a thin line is called *the shadow* and shows the trader day's price range from *high* to *low*. The other wider part is called *real body* and shows the distance between the *open* and the *close*. The real body is green only when the close is higher than the *open*. If the *close* is lower than the *open*, the real body is red.

In the past few years, the candlestick chart has become very popular for western chartists. Therefore the traders will find them in many trade programs. Note that a chart bar representation can be always represented as a candlestick one and any indicator may be used as well. (MURPHY, 1999)

Figure 4: Candlestick-chart, source: (TEXAS A&M AGRILIFE RESEARCH & EXTENSION CENTER AT DALLAS, 2001)

3.3 Efficient-market hypothesis

Traders who follow efficient-market hypothesis believe in four premises.

1. Price - Price reflects all information.
2. Investor - Investor dislikes risk, investor is aware of importance of information and knows how to interpret them.
3. Memory - Markets do not have memory. The past happenings do not influence future ones.
4. Distribution - Price is normally distributed. (DOSTÁL, 2012)

3.4 Fractal-market hypothesis

Fractal-market hypothesis is based on the same factors as the efficient-market hypothesis but with different premises.

1. Price - Information may be interpreted differently by different people.
2. Investors - Investors behave irrationally and tend to take risks when they have loss.
3. Memory – Investors are influenced by past happenings, their market-expectations are formed on gained experiences.

4. Distribution – Price is not normally distributed, but price tends to drop faster than to rise. (DOSTÁL, 2012)

3.5 Technical analysis

The reader of this thesis is required to have a basic understanding of what a technical analysis is. He thus needs to be able to differentiate it from a fundamental one.

John J. Murphy defines technical analysis as:

"study of market action, primarily through the use of charts, for the purpose of forecasting future price trends" (MURPHY, 1999)

Traders believe in three premises of technical analysis "market actions discount everything", "history repeats itself" and "prices move in trends". The statement "market actions discount everything" rely on a belief that all the factors influencing the price are fully reflected in actual price on the market. This leads to a belief that the only thing one needs for a stock market prediction is a price. (MURPHY, 1999)

"price moves in trends". This idea comes from Newton's first law of motion. Technicians believe that prices move in trends and most likely continue in the trend without reversing. If we acknowledge Newton's law, we can say that the trend in motion continues its way, until it reverses. The trend forecasting is essential for technical traders, therefore they focus on signs of reversing.

"history repeats itself" Technical analysis and human psychology go hand in hand. Human mind believes that what was found functional in the past will be functional in the future. Therefore the key to understand the future lies in past happenings. (MURPHY, 1999)

3.5.1 Moving Averages

The moving average is a trend following device. It tells a trader when the old trend stopped and new trend begins. Construction can be very simple or difficult. The moving average can be calculated by the addition of close prices altogether, and divided by the number of days. By following this approach, we can count moving averages for every day. As the reader can notice, moving average only reacts on actions on the market. Traders should keep in mind that moving average is only follower not leader. (MURPHY, 1999) Even though many articles were published about moving averages, there are still some questions remaining. For example, how many days should one use? what kind of prices should one use? *Open*? *Close*? *High*? *Low*? Are there models which work for any kind of commodity or stock?

Moving average is a smoothing device, by averaging closing prices, one gets a smoother line chart than in the case of a daily close data chart. Unfortunately, the smoothing process can exclude important market actions, therefore the trader has to choose an appropriate number of days to entail into the process. For instance, a 200 days moving average will be smoother than a 20 days one but it might have a lack of important market events.

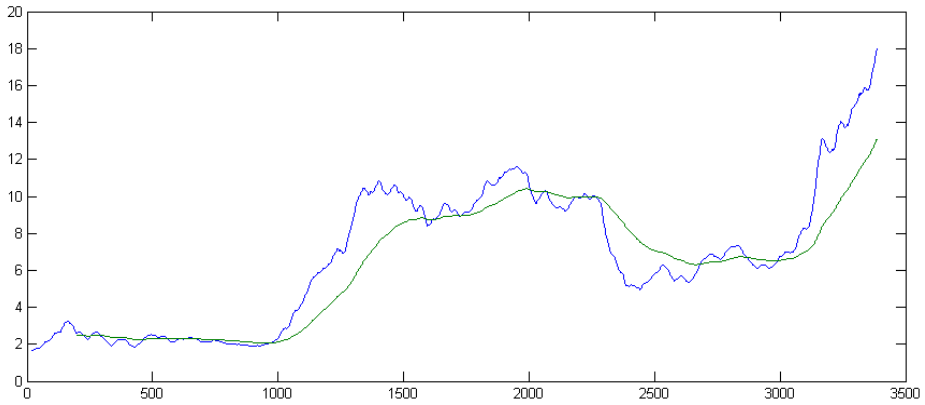

Figure 5: Stock LGF, green line: MACD 200, blue line: MACD 20

The most widely used prices are closed prices. These prices are used through all the thesis if it is not mentioned otherwise.

3.5.1.1 Moving averages convergence/divergence (MACD)

It is a technique which uses two exponential moving averages. It combines oscillator principles with a dual moving average crossover approach. When the trader uses MACD he usually sees two lines included in a chart. The faster of the two lines is called MACD line and represents the difference between two exponentially smoothed moving averages of usually closing prices. The most used and recommended settings are the usage of 12 or 26 days/weeks average. The slower line is called a signal line and usually represents a nine period of exponentially smoothed average of the faster, MACD, line. (MURPHY, 1999)

3.5.2 Rate of Change (ROC)

Is a technique to measure the ratio between the most recent closing price and close price n periods ago. In a literature engaged in stock trading one may find it also under simple momentum term. (MURPHY, 1999)

The equation for computing ROC can be written as follow:

```
ROC = [(Close - Close n periods ago) / (Close n
periods ago)] * 100
```

Figure 6: ROC equation, source: (STOCKCHARTS, 2013)

The equation above uses a midpoint line equal to one hundred, and when the latest closing price is higher than the close price n periods ago the output of rate of change will be above one hundred. The opposite occurs when the latest price is lower. (MURPHY, 1999)

3.5.3 Entropy

For the purpose of time series evaluation one may use entropy. For time series containing random values, higher values than in the case of sinus functions are returned. The following algorithm is used to evaluate entropy. The mathematical definition can be find in publication (DOSTÁL, 2012).

```
function entropy = apen( pre, post, r )

[N,p] = size(pre);
phiM = 0;
phiMplus1 = 0;
foo = zeros(N,p);

for k = 1:N;
    for j = 1:p
        foo(:,j) = pre(k,j);
    end

    goo = (abs(foo - pre) <=r);

    if p==1
        closerpre = goo;
    else
        closerpre = all(goo');
    end

    precount = sum(closerpre);
    phiM = phiM + log(precount);
    inds = find(closerpre);

    postcount = sum(abs(post(closerpre)-post(k)) <
r);
    phiMplus1 = phiMplus1 + log(postcount);
end

entropy = (phiM-phiMplus1)/N;
end
```

Figure 7: Entropy, source: (DOSTÁL, 2012)

3.5.4 Hurst Exponent

Traders use Hurst Exponent in harmonic trading as an estimation of the predictability of a time series. Thus it is very similar to the entropy described in section 3.5.3.

$$H \in \langle 0,1 \rangle$$

For Hurst exponent $H \in \langle 0.5,1 \rangle$ the trader may expect a persistent behavior. In other words, whatever is happening now will be happening in the future as well. If the price increases in the time series from time step $i(t-1)$ to $i(t)$ there will probably be a price increase from $i(t)$ to $i(t+1)$. The same is true of decreases, where a decrease will tend to follow a decrease. The larger the H value is, the stronger the trend. Series involving persistent behavior are easier to predict than series falling in the other two categories.

For Hurst exponent $H \in \langle 0,0.5 \rangle$ the trader may expect anti-persistent behavior. In other words increase will tend to be followed by a decrease, consequently a decrease will be followed by an increase. This behavior is sometimes called *mean reversion* which refers to tendency of time series to return to a longer term mean value. The strength of this mean reversion increases as H approaches 0.

A Hurst Exponent value H close to 0.5 indicates a random walk. In a random walk there is no correlation between any element and a future element (for more information about random walk access part 3.8) . These time series are hard to predict. Thus traders should avoid trading that type of stock. (HENNESSEY, 2010)

3.5.5 VIX index

Is also known as the CBOE (The Chicago Board Options Exchange) Volatility Index. This index calculates and updates the values volatility indexes designed to measure thirty day implied volatility of different securities. These volatility indexes represent key measures of market expectations of near-term volatility. Traders rate this index as one of the best measurement of investor sentiment and market volatility. (CHICAGO BOARD OPTIONS EXCHANGE, 2012)

3.6 Fundamental analysis

Is opposite to technical analysis. Study everything what would possibly influence stock market price. This may involve evaluating of previous companies success, management, position on the market, revenues, earnings, future growth, return on equity etc. (investopedia.com, 2013) As one might see, it is very subjective and one hardly avoid bias. There are many companies which provide this evaluation on

particular stock. These reviews are collected by sites as Google finance or Yahoo finance and one can find it online.

Macroeconomic indicators for fundamental analysis are:

- Interest rates – Interest rate changes are one of the most important for the fundamental traders. The increase of interest rate leads to a strengthening of domestic currency. Consequently the stock price declines. This is due to an increasing price of loans. Reducing interest rates, on the other hand in times of crisis, contributes to the companies' growth. Stock index grows while domestic currency drops.
- Unemployment – The economy growth happens when the unemployment decreases. It also means healthier economy and brighter future.
- Unexpected events – The government resignation, terrorist attacks can significantly influence stock prices.
- Inflation – It is an economy term for the rise in prices. The inflation in the annual range around 3% indicates a healthy economy. High inflation has a negative impact on the economy. The trades often buy commodities as gold, silver or platinum which are not influenced by inflation, but by an amount currently available. (MALÍKOVÁ, 2013)

3.7 Word of Mouth

The word of mouth theory is based on the assumption that people's decision is being influenced by other people's thinking evolving in the same environment, and this leads to similar conclusions. According to Shiller' and Pound's (1989) survey of 131 individual investors whom were asked why they purchased the stock, most of them took into account their families or friends opinion.

The effect of the Word of mouth theory can be also perceived on fund Boston managers holding's. Fund managers tend to hold the same stocks as their colleagues from the city. Fund managers are also more sensitive to the stocks held by managers in the same city than to the stocks held by managers from other cities. (HONG et. al., 2006)

3.7.1 Electronic word of mouth (eWOM)

The Electronic Word of mouth theory stands for the influence received from online recommendations. This topic is still being researched and several interesting facts are being announced. For example a person who is browsing the internet and sees a product recommended by a friend on a social website will not automatically buy the product, nonetheless researchers found there a significant correlation which can lead to the presumption that the person is significantly influenced by a friend's recommendation.

A very commonly used example is the "Movie box-office problem". Researchers try to predict stock market prices while evaluating company's product sales. They pick up the movies category mainly because there is enough content and content information to evaluate. According to a movie database server IMDB.com there are over 200 movies released annually only in the U.S.A. There are also around 100 000 tweets for each movie found on the social site Twitter.com. Last but not least, IMDB.com publishes statistics for the past or upcoming opening weekends. The whole idea of stock movies box-office prediction via eWOM relies on the presumption that people who write about a movie on social websites before their actual release, are surely interested in the movie and will most likely go to watch it. After the release they turn out to be sort of eWOMs with influence to other people. (YU et. al., 2012)

3.7.2 Inside trading

Is a stock trading within the company where the managers have the right to have shares and therefore part of the company's profit. By this approach the owners hope to increase the productivity of their employees. The law distinguishes an illegal form of inside trading when managers use internal data for their own profit. Unfortunately we will never be able to see those information even though it would be a very powerful signal for a market action (buy, hold, sell). The only source of data concerning legal trades are the trading reports filled by corporate insiders, and it is unlikely that managers will willingly report their violations. (BHATTACHARYA et. al., 2002)

But even though the public audience will never be aware of those inside trades, the trading reports may be still valuable for the stock evaluation and will be used in one of the models further in the text.

3.8 Random Walk

The Random walk theory was first published by Karl Pearson in a Nature magazine (PEARSON , 1905). He expressed a problem of a man who is randomly walking, while the probability of finishing at a certain position is unknown. The Random walk theory is a formalization of a path that consists of a succession of random steps. The full mathematical explanation can be found in particular literature as (SPITZER , 1976).

3.8.1 Random walk theory in market environment

Is the theory that stock prices change, have the same distribution, and are independent of each other, therefore past movements and trends of a stock price or market cannot be used to predict its future movement. (investopedia, 2013)

3.9 Social sites, specialized web pages and data mining

Social sites and specialized web pages are very interesting data sources for the stock price prediction.

3.9.1 Social sites

For the past few years, the social sites have been evolving and spreading over our society quite rapidly. We define social sites as :

"Social network sites are web-based services that allow individuals to construct a public or semi-public profile within a bounded system, articulate a list of other users with whom they share a connection, and view and traverse their list of connections and those made by others within the system." (ELLISON et. al., 2007)

Social sites may be very interesting sources of data for any decision maker. Since people post their thoughts and opinions online and the amount of data one posts increases we will see many traders willing to use this data for online trading.

3.9.2 Specialized web pages

There are many web pages suitable as data feeds for a stock market analysis. The mentioned ones are used for the system creation.

3.9.2.1 Yahoo finance

Yahoo finance[1] is a subdomain of the huge server Yahoo inc. providing various services from general to financial news. For our purposes, it is very convenient to take data from Yahoo finance in *csv.* format. The automated trade system can be trained only by historical data which are on the server for free. Yahoo inc. provides free historical data for stocks with one day delay. The following lines show how to retrieve historical data using MATLAB and Java automatically.

The historical data are downloaded by accessed following a URL address in users web browser. Where ***s*** is a stock symbol, ***a*** stands for the start of the month (minus 1), ***b*** stands for the start of the day, ***c*** is the start year. The final ***g*** parameter lets the user choose between getting historical stock information on a daily, weekly, or monthly basis. (MASHBURN, 2010)

```
http://ichart.finance.yahoo.com/table.csv?s=AAPL&a=10&b
=15&c=2005&d=01&e=17&f=2006&g=d&ignore=.csv
```

Figure 8: link example, source: (MASHBURN, 2010)

When there are particular data we want to download, we may proceed to actual download process. We will be using the MATLAB environment throughout the thesis. This example is featured with java applet.

[1] http://finance.yahoo.com/

```
% Open a connection to the URL and retrieve data into a
buffer

buffer      = java.io.BufferedReader(...
              java.io.InputStreamReader(...
              openStream(...
              java.net.URL(url_string))));
```

Figure 9: Data retrieving using java applet, source: (MASHBURN, 2010)

The code above creates a direct tunnel to a given URL address which was created earlier. Fetched data are stored in a variable called *buffer*.

We managed to download a rough data set of the wanted stock. The next step is a data normalization. The normalization stands for a data customization allowing the data to reflect actual happenings on the market. When it comes to dividend pay day or stock splits the price rapidly decreases and it can violate trading algorithm. Therefore traders should normalize all fetched data. The Yahoo finance historical prices feature includes notations for all splits and dividend distributions.

(YAHOO, 2012) Following example shows adjusted close calculation.

```
2/13/03 Close = 46.99
2/14/03 Close = 48.30
2/18/03 Split = 2:1
2/18/03 Close = 24.96
2/19/03 Cash Dividend = 0.08 (ex-date)
2/19/03 Close = 24.53

Split Multiplier = 0.5
Dividend Multiplier = 1 - (0.08/24.96) = 0.9968

2/13/03 Adj. Close = 0.5 * 0.9968 * 46.99 = 23.42
2/14/03 Adj. Close = 0.5 * 0.9968 * 48.30 = 24.07
2/18/03 Adj. Close = 0.9968 * 24.96 = 24.88
2/19/03 Adj. Close = 24.53
```

Figure 10: Adjusted close calculation example, source: (YAHOO, 2012)

3.9.2.2 Google finance

Google finance[2] is another reliable historical data source. As Yahoo finance, Google services provide developers with daily, weekly or month data set, free of charges. It will be interesting to see how Google evolves this young service and includes it into the portfolio.

3.9.2.3 MarketWatch[3]

Is a site which is fully specialized in a stock markets, as Google finance and Yahoo finance, MarketWatch offers companies overviews, news, historical data and many other information traders can use to evaluate stock values. This site differentiate from others by its category *insider*. This data set provides user with inside trading information within a company and they are presented into a chart and a detailed table (Figure 11) unfortunately, MarketWatch does not have any API[4] and users have to parse its data by themselves.

Therefore following lines explains how to get valuable data from MarketWatch insiders.

Date	Name	Shares	Transaction	Value
01/26/2013	Robert A. Iger Chairman & Chief Executive Officer	11,671	Derivative/Non-derivative trans. at $54.09 per share.	631,284
01/26/2013	Alan N. Braverman Secretary, Senior Executive VP & General Counsel	2,334	Derivative/Non-derivative trans. at $54.09 per share.	126,246

Figure 11: Inside trading information from MarketWatch

The chart above represents a part of the inside trading table for the Walt Disney Co. and it shows that Chairman & Chief Executive Officer sold his 11,671 shares. This information is represented in a html code as Figure 12. The html code looks almost the same for every entry and one has to find the pattern and parse data accordingly.

[2] http://www.google.com/finance
[3] http://www.marketwatch.com
[4] Application Programming Interface

```
<tr>
  <td class="date">01/26/2013</td>
  <td class="name">
          <a
href="http://www.marketwatch.com/investing/stock/DIS/insiders?pid=38166">Robert A. Iger</a>
<br>
          <span>Chairman &amp; Chief Executive
Officer</span>
  </td>
  <td class="shares">11,671</td>
  <td class="transaction">
      <span class="transactionArea">
          <span class="transPosNeg neg"> </span>
          <span class="transactionDesc">Derivative/Non-
derivative trans. at $54.09 per share.</span>
      </span>
  </td>
  <td class="value">631,284</td>
</tr>
```

Figure 12: Html code of inside trade transactions

Each field from the table (Figure 12) has to be parsed using regular expressions, which in computer science stands for a way to describe a set of strings based on common characteristics shared by each string in the set. (ORACLE, 2013) For the purposes of code understanding and the data extraction from MarketWatch we will need to know only few predefined characters:

.	Any character (may or may not match line terminators)
?	Matches the preceding element zero or one time
*	Matches the preceding element zero or more times

Table 1: regular expresion symbols

With the combination of predefined characters and the html structure shown in the example above, one is able to construct code which will parse inside trades. The code below represents the whole flow from the html request using regular expresions.

The reader should focus on the part of the code with *(.*?)* which represents the extracted string, in this case we extract the date of transaction.

```
%% download data for a particul company
url =
strcat('http://www.marketwatch.com/investing/stock/',co
mpany,'/insideractions');

html = urlread(url);

    .
    .
    .

regexpi(transactions{i}, '<td class="date">(.*?)</td>',
'tokens');
```

Figure 13: An usage of regular expression

3.9.2.4 Google trends

Google trends[5] is an interesting tool from Google for marketers and market researchers. The position of the most popular search engine these days, gave Google the ability to find trending searches. It can tell one when a term, for example a product, was popular over the time. There is also the possibility to compare two or more terms. Thus the tool can metaphorically provide one with information about people's mind. (YOSSI , 2012)

The following picture shows a trend in an entertainment industry. The chosen examples are recent movies from different production companies. The task is to figure out which movie was the most searched over the other, during the past few months and thus has generated a bigger revenue.

[5] http://www.google.com/trends/

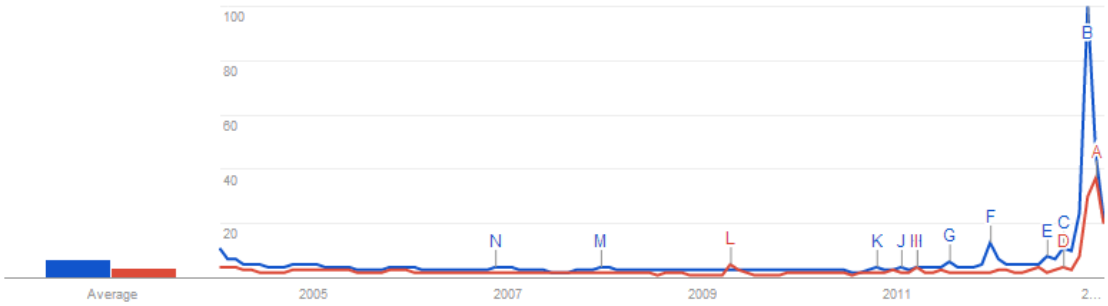

Figure 14: An example of Google trends for terms “The Hobbit” as a blue curve, and “Les Miserables” as a red curve.

As shown in the picture above, the movie represented with a blue line was more popular over the internet searches and supposedly generated bigger revenues. Google trends is more a tool which can help researchers to evaluate a present state, than a future one. (VARIAN et. al., 2009) A research which will prove the correlation between Google trends output and stock market prices is still needed. This might be challenge for future researchers.

3.10 MATLAB environment

MATLAB is a numerical computing environment which involves a programing language. MATLAB over the years, has evolved in multifunction environments suitable for various researches. Through tool boxes the scientists are able to easily simulate electrical circuits or to create a neural network easily without coding. Even though for complex problems coding is still required.

From the perspective of this thesis, the most interesting is the Neural Network Time Series toolbox, which allows one to create various customized neural networks, plot testing, result validation or print mean square error.

3.11 Neural networks

The aim of this text is to provide a better understanding of neural networks and their implementation, which are necessary to understand the model creation.

Neural networks were built to simulate the human brain. The reader can picture it as a “black box”, a system we do not see into. It is therefore not possible to see inside

the system, the user just make some suppositions about the inner structure and specifies a transfer function. Neural networks are especially used in situations when one cannot describe all the processes, the influences on searched phenomena is random or deterministic relationships are very complicated. Last but not least we use neural networks in the case of decision making. (DOSTÁL, 2011)

This concept was inspired by biological neurons, which can be described in simple way, as a structure made of inputs (dendrites), body (soma) and one output (axon). (DOSTÁL, 2011) The following picture shows the neuron structure.

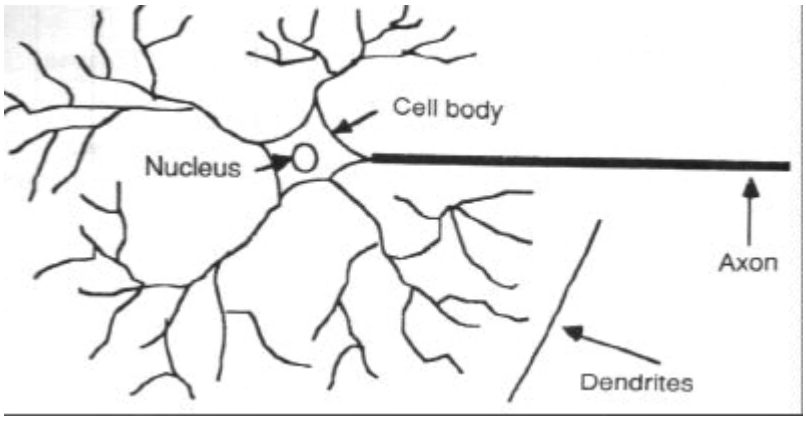

Figure 15: Biological neuron, source: (INDIANA UNIVERSITY SOUTH BEND, 2008)

3.11.1 History

The first publication about neural networks was submitted in the second half of the twentieth century by McCulloch. Later, Rosenblatt created the first functional perceptron which was able to solve problems involving areas that are linearly separable. After this discovery Rumelhalt discovered multilayer networks, followed by Hinton and Williams who developed the back-propagation method for multilayer networks. (DOSTÁL, 2011)

3.11.2 Concept

A neural network works in two phases. In the first one the network presents a model of a complicated system and tries to set up parameters in order to tune them to fit to the neural network topology.

In the second phase the network becomes an expert. The system consumes the knowledge given in the first phase, and produces an output accordingly. During the building phase, it is necessary to have all input, hidden, output layers defined. (DOSTÁL, 2011)

The following example shows the simplest neural network called perceptron. The structure of perceptron involves N input variables p_1, p_1 … p_N which are multiplied by its weight w_1, w_1 … w_N. The threshold value influences the output; it increases the value just about this value. (DOSTÁL, 2011)

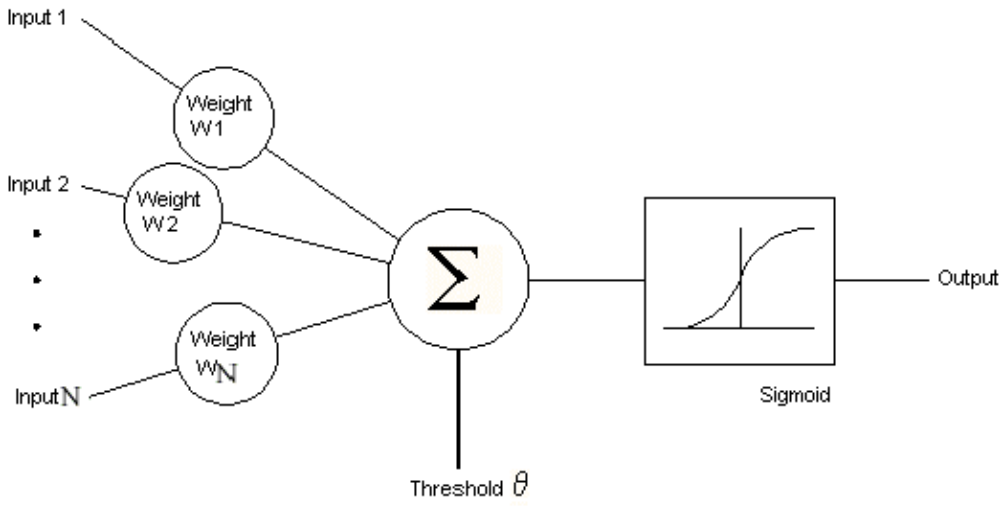

Figure 16: Perceptron, source: (NIKOLAEV, 2012)

$$a = w_1 * p_1 + w_2 * p_2 + w_3 * p_3 + .. + w_N * p_N + \boldsymbol{\Theta} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} w_i * p_i + \boldsymbol{\Theta}$$

Figure 17: equation, source: (DOSTÁL, 2011)

The next step is the usage of transfer function f to get output n.

$$n = f(a)$$

The most used transfer functions in MATLAB are *logsig, hardlim, purelin, tansig*.

3.11.2.1 Hardlim transfer function

The hard-limit transfer function limits the output of the neuron to either 0 or 1. The function returns 0 if the net input argument n is less than 0, accordingly when the net input argument n is greater than or equal to 0 the return value is equal 1. (DEMUTH et. al., 2000)

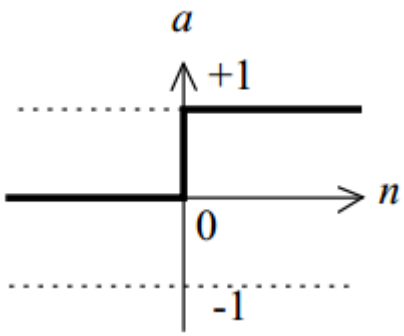

Figure 18: Hardlim, source: (DEMUTH et. al., 2000)

Hardlim equation in MATLAB environment:

$$a = hardlim(n)$$

3.11.2.2 Purelin transfer function

Is a linear function which is commonly used in the last layer of multilayer networks dedicated for approximation. (DEMUTH et. al., 2000)

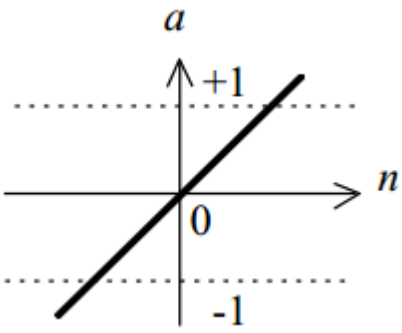

Figure 19: Purelin, source: (DEMUTH et. al., 2000)

Purelin equation in MATLAB environment:

$$a = purelin(n)$$

3.11.2.3 Sigmoid transfer function

This function gets on its input a value which may be in a range from minus infinity to plus infinity and squeeze it into a range 0 to 1. This function is often used in hidden layers in multilayer networks. (DEMUTH et. al., 2000)

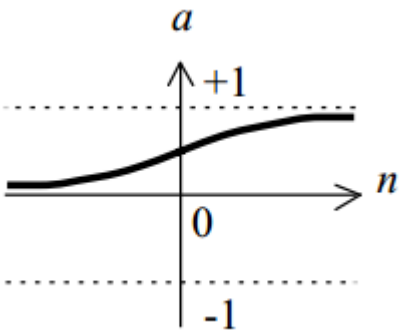

Figure 20: Log-sigmoid transfer function, source: (DEMUTH et. al., 2000)

Log-sigmoid transfer function equation in MATLAB environment:

$$a = logsig(n)$$

3.11.2.4 Tan-sigmoid transfer function

Tan-sigmoid transfer function is similar to sigmoid funcion, but it differs in range which variable a can be placed into.

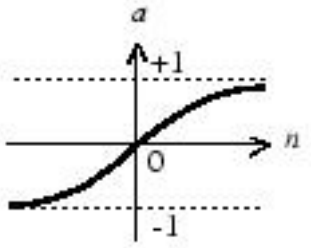

Figure 21: tansig, source: (DEMUTH et. al., 2000)

Tan-sigmoid transfer function equation in MATLAB environment:

$$a = tansig(n)$$

3.11.3 Network architecture

A neural network may have several layers. Each layer has a bias vector b, a weight matrix w, and an output vector a. These multilayers networks usually contain one or more *hidden layers*. Thus a layer which produces a network output is called *output layer* and the remaining ones are called *hidden layers*. The figure below shows a multilayer network which has R^1 inputs and different numbers of neurons for each layer. The first layer consists of S^1 neurons. The second layer contains of S^2 neurons and the third layer contains of S^3 neurons. Note it is common to use different numbers of neurons for each layer. The figure also shows the connections between layers where the outputs of intermediate layer are inputs to the following layer. Consequently the second layer can be analyzed as a one-layer network with S^1 inputs, S^2 neurons, an S^2 x S1 weight matrix w^2. The input to the second layer is a^1; the output is a^2. After this, one can approach the second layer as a single-layer network. (DEMUTH et. al., 2000)

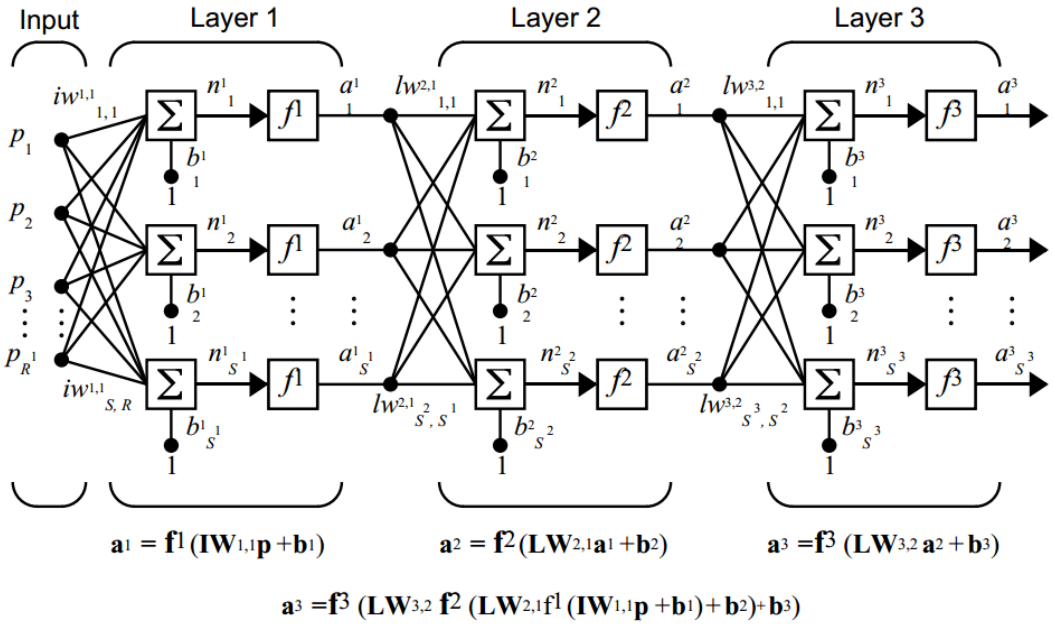

Figure 22: Multilayer network, source: (DEMUTH et. al., 2000)

Multiple-layer networks are very powerful, they are thus, often used. For example, a network of two layers, where the first layer has a sigmoid transfer function and the second layer is linear, can be trained to approximate any function arbitrarily well. (DEMUTH et. al., 2000)

3.11.4 Neuron with a vector input

In many cases one needs to have a vector input instead of a single value one. For this purpose the neuron with a vector input was developed. An example of the usage of vector input may be that of a network which has open, high, low, close prices on its input. If there were not any vector input possibility the user would have to use only one of these prices.

In the figure below a neuron with a single R-element input is shown. Where the inputs p_1, p_2.. p_R, are multiplied by their weights $w_{1,1}$, $w_{1,2}$.. $w_{1,R.}$ (DEMUTH et. al., 2000)

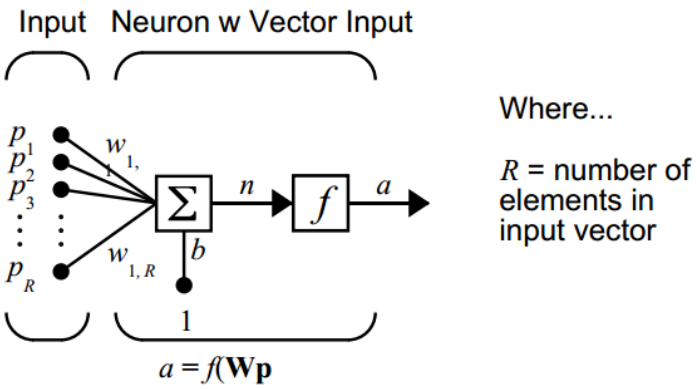

Figure 23: Neuron with a vector input, source: (DEMUTH et. al., 2000)

3.11.5 Learning

The ability of neural network to learn is the most powerful among all others. Through an iterative process of adjustments applied to its synaptic weights and thresholds the neural network learns from its environment. (HAJEK, 2005)

Hajek defines the process of learning in context of neural networks as:

"*Learning is a process by which the free parameters of a neural network are adapted through a continuing process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place."* (HAJEK, 2005)

Hajek also specifies various types of learning such as error-correction learning, hebbian learning, supervised learning, unsupervised learning. Following text is focused on the most famous supervised learning and unsupervised learning.

3.11.5.1 Supervised learning

Is a variant of an error-correction approach. The learning process is featured by "teacher" – a sample data set, which the neural network uses to learn and tries to become a teacher. The typical workflow looks like following: As soon as the sample inputs are applied to the network, the network outputs are compared to the given targets. After that the learning rule is used to adjust the weights w and biases $\boldsymbol{\theta}$ of the network

in order to minimalize error distance between network outputs and the targets. (HAJEK, 2005) (DEMUTH et. al., 2000)

Because the learning process stays on teacher's quality, It is necessary to obtain the best dataset for our system in order the system to generate appropriate output.

In some cases, after the learning process, the mean square error of the neural network may be of a high number, this is an unwanted effect of overlearning. Overlearning occurs when the neural network extracts too much information from the individual cases forgetting the relevant information of the general case. In this case the neural network should be reinitialized and trained again. (DEMUTH et. al., 2000) (HOLLMEN , 1996)

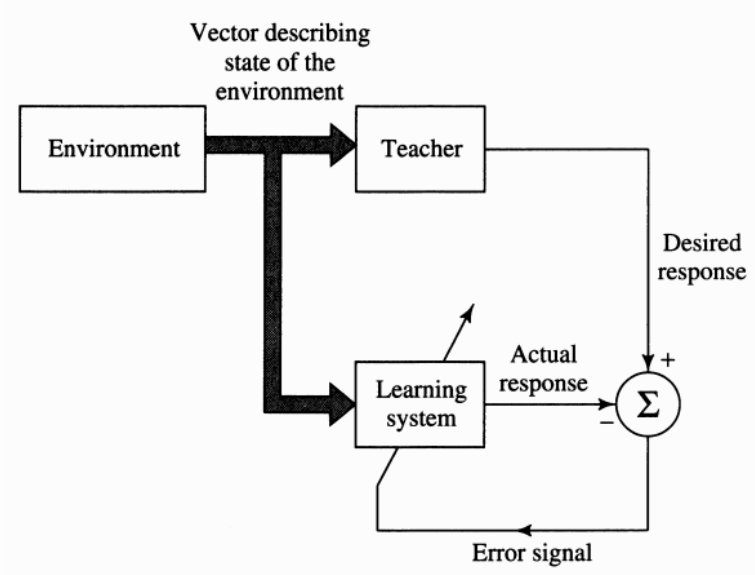

Figure 24: Learning with a teacher, source: (HAJEK, 2005)

3.11.5.2 Unsupervised learning

Unsupervised learning requires a neural network to form all useful neural wiring without any feedback from the teacher. In other words, there are no samples of the function to be learned by the network. (NTNU, 2010) (HAJEK, 2005) In this thesis unsupervised learning method will not be used.

3.11.6 Dynamic neural network

Dynamic neural networks, which include tapped delay lines are commonly used for non-linear filtering and predictions. (DEMUTH et. al., 2000)

3.11.7 Nonlinear Autoregressive Network with exogenous inputs

This network is mostly known for its shortcut NARX. This network is recurrent and dynamic with feedback connections enclosing several layers of the network. The NARX model is based on the linear ARX model, which is not covered in this thesis but is often used in time-series modeling. (MATHWORKS, 2012) According to (DIACONESCU, 2008) the NARX model for approximation function can be implemented in various ways, but the simplest appeared to be a usage of *feedforward* neural network with a tapped delay in the first layer and a delayed connection from the output of the second layer to the input (a second tapped delay line).

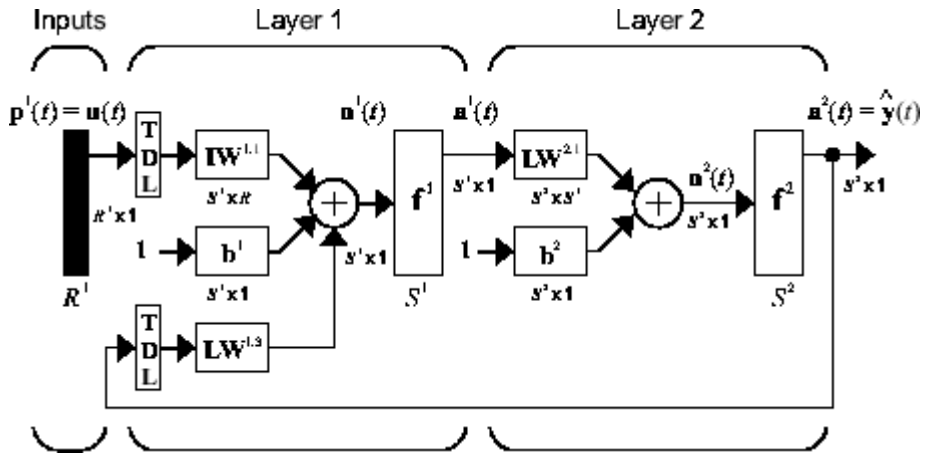

Figure 25: Two layer *feedforward* network, , source: (MATHWORKS, 2012)

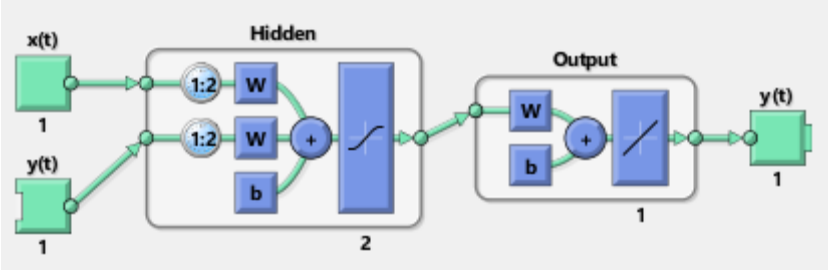

Figure 26: NARX network in MATLAB

In terms of building a NARX network we distinguish two types of architecture. As one can see, in the figure above the output is fed back to the input of the *feedforward* neural network, thus the input value is estimated, but in many cases the next value is known. Therefore NARX neural network can be implemented with a parallel architecture or series-parallel architecture, see Figure 27.

The main advantage of series-parallel architecture is mainly in its accuracy. The second advantage is that the network has a purely *feedforward* architecture, nonetheless the static *backpropagation* can be used for training, thus the training is faster. (MATHWORKS, 2012)

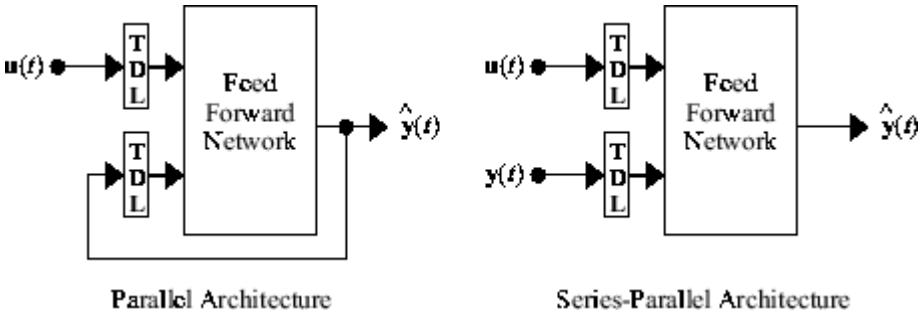

Figure 27: Parallel and Series-Parallel Architecture, source: (MATHWORKS, 2012)

3.11.7.1 Network usage in MATLAB environment

MATLAB provides one with a straight forward environment with prepared function for convenient coding NARX systems. The following lines describe the typical workflow.

First of all, load a necessary data set into the *workspace*. After that create series-parallel NARX network using the function *narxnet*. The following example contains ten neurons in the hidden layer, which is the third argument in *narxnet* function. Consequently the *trainlm* function is used for the training process and then function *preparents* for a data prepare. Final step is the function *train* which trains the network (MATHWORKS, 2012)

```
% NARX example
.
.
narx_net = narxnet(d1,d2,10);
.
.
[p,Pi,Ai,t] = preparets(narx_net,u,{},y);
.
.
narx_net = train(narx_net,p,t,Pi);
```

Figure 28: *Narxnet*, *preparets*, *train* example, source: (MATHWORKS, 2012)

3.11.8 Focused Time-Delay network

Is a *feedforward* network with a tapped delay line at the input. Shortly the network is usually called FTDNN. This network belongs to dynamic networks where the dynamic part appears at the input layer of a multilayer *feedforward* network.

FTDNN has one unique feature that does not require dynamic *backpropagation* to compute the network gradient. It is caused by the fact that a tapped delay line appears only at the input of the network, and does not contain any feedback loops or adjustable parameters. It also makes network train process faster than within other dynamic networks. (MATHWORKS, 2012)

The following example shows a two-layer FTDNN, which is commonly used for time-series prediction.

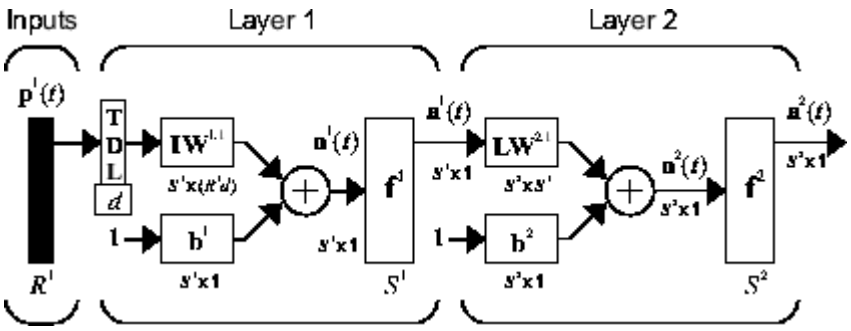

Figure 29: two-layer FTDNN, source: (MATHWORKS, 2012)

3.11.8.1 Network usage in MATLAB environment

The process of the FTDNN creation requires several steps. The following text leads the reader step by step throughout the whole process with the code examples .

First of all, it is necessary to load the data set, which we have previously created. The data need to be normalized and converted into time sequence represented by a cell array.

```
% Data load example

y = dataset;
y = y(1:600);
```

Figure 30: Data load example, source: (MATHWORKS, 2012)

The next step is the network creation. In MATLAB environment we use the *timedelaynet* command. This command, without the additional input of the tapped delay line vector, is basically *feedforwardnet* command which is used in the following examples. In this particular example of *timedelaynet* the command usage works with one to eight delays and ten neurons in hidden layer. (MATHWORKS, 2012)

```
% timedelaynet command example

ftdnn_net = timedelaynet([1:8],10);
ftdnn_net.trainParam.epochs = 1000;
ftdnn_net.divideFcn = '';
```

Figure 31: *timedelaynet* command example, source: (MATHWORKS, 2012)

Because we set the network for a maximum delay equal to eight the trained network will begin to product the prediction for the ninth value of the time series. The following code shows how to load tapped delay lines with the eight initial values of time series represented as a variable *Pi*. The user should be aware that the model generates one step prediction.

```
% Tapped delay line example

p = y(9:end);
t = y(9:end);
Pi=y(1:8);
ftdnn_net = train(ftdnn_net,p,t,Pi);
```

Figure 32: Tapped delay line example, source: (MATHWORKS, 2012)

The network above is now fully trained and prepared to make predictions. Readers might find useful to compute the overall mean square error of a created network which is presented in the code below.

```
% Mean square error example

yp = ftdnn_net(p,Pi);
e = gsubtract(yp,t);
rmse = sqrt(mse(e))
```

Figure 33: Mean square error example, source: (MATHWORKS, 2012)

According to MATLAB Help Focused Time Delay network almost in any situation produces better mean square error than linear predictor. (MATHWORKS, 2012)

3.11.9 Distributed Time-Delay Network (TDNN)

Another type of network is the Focused Time Delay network. If the reader compares the Figure 34 below of Distributer Time-delay network and the Figure 29 of Focused Time Delay network one will find out that the tapped delay within Distributer Time-delay network is involved in the first layer and also in the second one. In other words the creator of the network can distribute the tapped delay lines throughout the network. The following figure shows distributed TDNN with two layers. (MATHWORKS, 2012)

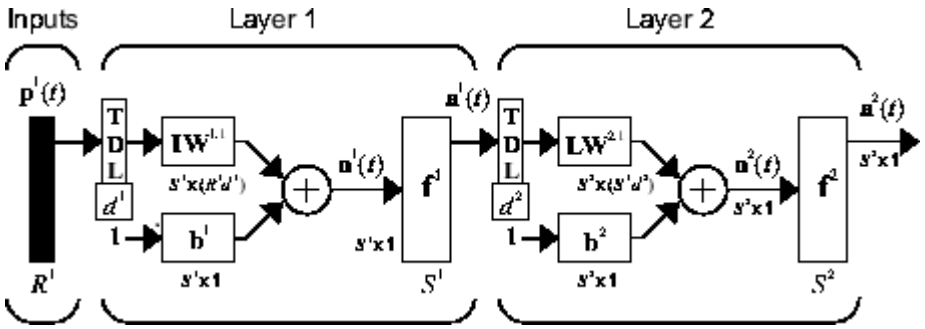

Figure 34: TDNN with two layers, source: (MATHWORKS, 2012)

3.11.9.1 Network usage in MATLAB environment

To create TDNN network the creator may use the same approach as in the FTDNN example mentioned above with *timedelaynet* function replaced by *distdelaynet* function. The difference between those two functions lies in the first argument. *distdelaynet* function has as a first argument cell array that contains the tapped delays to be used in each layers. (MATHWORKS, 2012)

```
% distdelaynet function

dtdnn_net = distdelaynet({d1,d2},5);
```

Figure 35: Example of *distdelaynet* function, source: (MATHWORKS, 2012)

4 Thesis Scope

This part of the thesis is devoted to the market examination involving technology companies. The text covers questions such as why to trade technology sector or what should the trader expect? Note that the trading period contains data from 1.1.2010 to 25.3.2013 and more recent market situations are not covered.

4.1 Technology industry

The technology sector is a huge investment opportunity. It is the largest single segment of the market involving other markets as industrial or financial sector. The technology companies are usually involved in innovations. This is also expected from investors that these companies make progress in their researches and comes to the market with new innovative products. These innovative products are connected with other businesses throughout all sectors. Modern companies almost every time applies the fact that if they want to improve their processes or increase profit they need to involve an innovative technology.

Technology companies' behavior always includes constant drive to adapt and overcome competitors with new products, it leads to faster and faster cycles in new products production. Due to a short cycle the leader company does not rule the industry for a long time, but is often overtaken by faster element.

Another interesting fact about the technology sector is that a significant number of its companies do not produce any profit or cash flow, therefore there is not much space for fundamental analysis based upon written facts and traders incline to guesswork. (SIMPSON, 2012)

Despite higher volatility and some disadvantages the technology sector is a very interesting place where investors often focus their assets. The technology sector is also very close to the author of this thesis, thus the promoted automated trader system performs on stocks from the sector.

4.2 NASDAQ

NASDAQ is a global electronic marketplace which purpose is to buy and sell stocks. The service replaced ineffective and slow in-person stock transactions starting over forty years ago. Today, NASDAQ spans six continents, owns and operates twenty four markets, five central security depositories and three clearing houses. It is also the largest single liquidity pool for the United States of America. The NASDAQ website offers many indexes and statistics for companies traded within their site. (NASDAQ, 2013)

5 Model building

The information, the reader has gained in the previous parts of my thesis, should help to understand the model with all its components.

The automated trade system is built with components/classes. These components are merged together, tested and results are recorded and evaluated. The aim is to find a combination of components which produces the best result. In the text below the total workflow is described.

5.1 Model design

The goal of this thesis is to build automated trading system for a chosen stock market. In other words the build system acts without human interaction or influence. The system should mainly be able to:

1. Handle unexpected occasions as price drops or market crisis.
2. Should be able to automatically choose the stock.
3. Perform minimalistic wrong direction ratio.
4. Produce revenue.

To achieve these sub goals the two system approaches was designed. These two systems – inner and outer trading systems represent the automatic trade system. The outer system mainly communicates with data sources and separates important data. These data then serves to inner system which communicates with outer system through an interface. Inner system then receives the data and replies with particular response. The inner system's output involves trained network and statistics which the outer system uses to evaluate trained network. The figure below illustrates the inner and outer system concept.

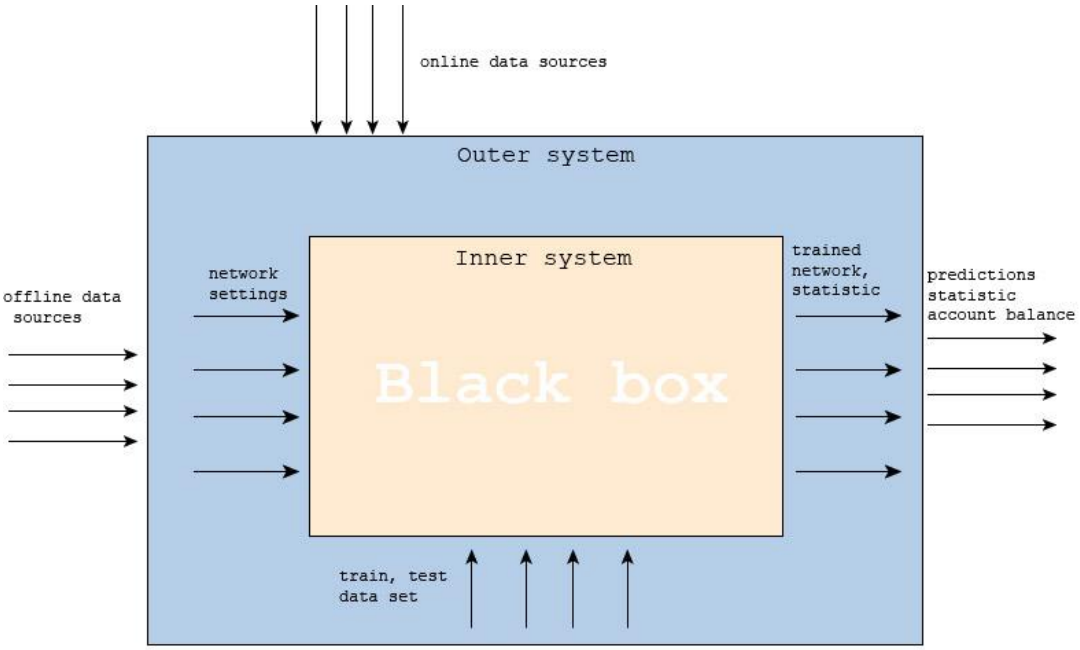

Figure 36: inner and outer system, representing automatic trading system.

For a trader who is not up to interfere with system's structures and wants to use it as it is distributed, these two systems remain as black boxes with an interface and hidden inner structure.

5.2 Outer system

The Outer system is the main part of the automatic trading system. Its features are downloading historical data from data sources, parsing data, choosing the best-fit stock, communicating with inner system and finally proceeding the trades. The following text is focused on important parts of the outer system.

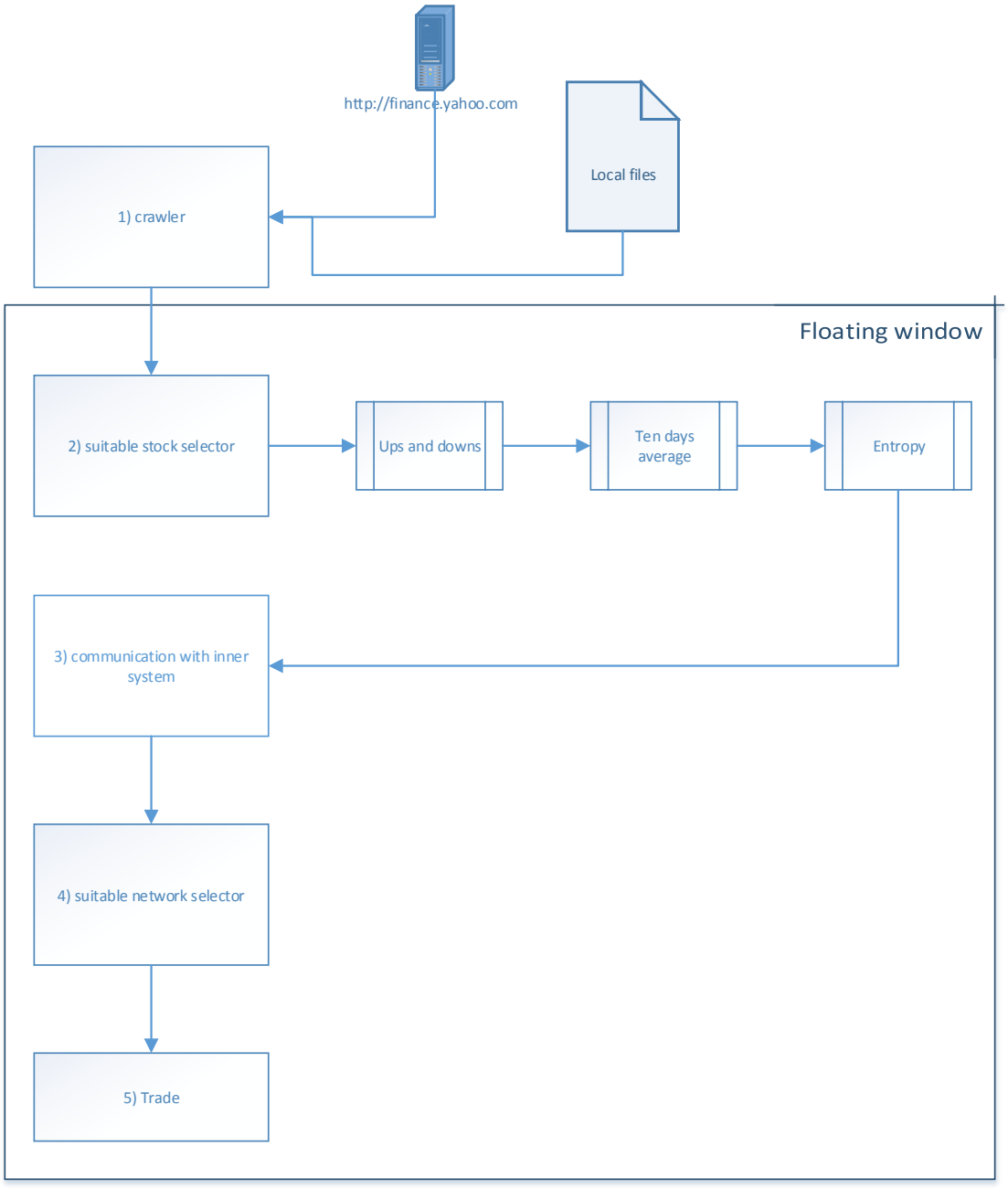

Figure 37: outer system workflow

5.2.1 Crawler

The crawler is a component which takes care about all data for the trading system. It obtains data from Yahoo finance and MarketWatch.com plus accesses local files. In order to create a database with all the stocks participating in technology category on NASDAQ market, the crawler reads *.csv* file with stock symbols and

choose only stocks which have been traded before 1.1.2010. NASDAQ contains over six hundred companies traded under the technology category and over five hundred stocks meeting the mentioned condition. These stocks historical data from Yahoo finance are downloaded and parsed. The detailed workflow regarding accessing historical data on Yahoo finance was explained in part 3.9.2.1 and will not be further examined.

The resulting database *StockData2.mat* may be loaded into a *workspace* using *load* command. Loaded database is then accessible through *DataStockIDArrayFiltered* variable.

```
% loading

load('StockData2.mat');
```

Figure 38: Code sample, loading database into *workspace*

attribute	type	description
xDataSet	array	indicator of price rise or drop
yDataSet	array	hist_close price, *depreciated*
ID	integer	stock ID
numberOfdata	integer	amount of trading days
Stock	string	stock
hist_open	array	historical open price
hist_high	array	historical high price
hist_low	array	historical low price
hist_close	array	historical close price
hist_date	cell	dates

Table 2: *DataStockIDArrayFiltered* inner structure

5.2.2 Floating window

Floating window is not a component unlike the crawler, but is a control system over the data set. It allows us to create trading periods. It can be pictured as a frame in which the whole process of training and trading happens. Figure 39 shows the whole idea of what a floating window is. Red quads represent floating windows, while Blue

quads represent historical data. Floating windows move towards the end of data set with respect to variable *daysfortrading.* When the neural network is chosen and predictions applied to *daysfortrading* period floating windows move to the right which causes loading new data to *daysfortrading* and old ones push into *daysfortrain*.

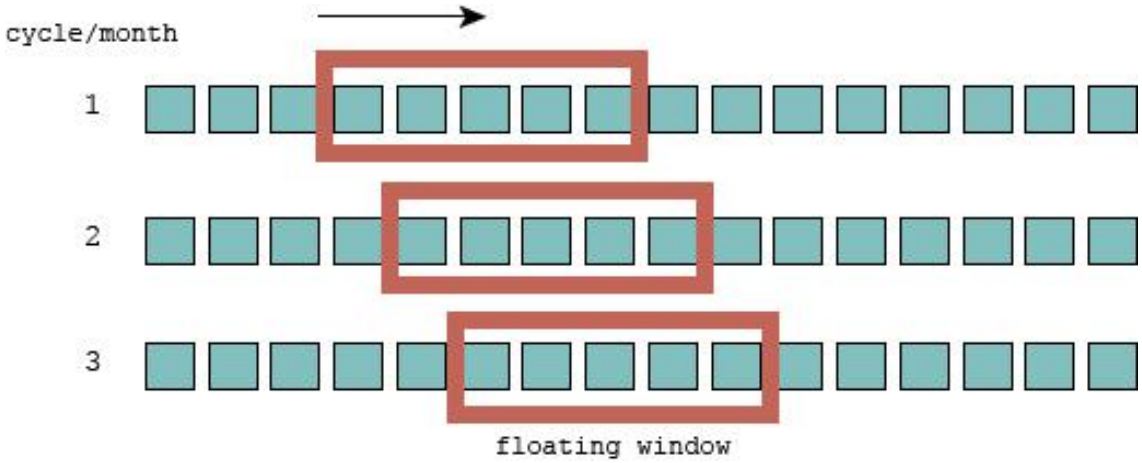

Figure 39: Floating window

The size of the window is determined by *daysfortrain* variable and *daysfortrading*.

$$FW = D_{train} \cup D_{trade}$$

$$\emptyset = D_{train} \cap D_{trade}$$

Executed tests show that an appropriate window for technology stocks should be 371, where *daysfortrain* were set to 351 and *daysfortrading* to 20. Twenty trading days represent an average number of trading days in one month. Similarly the *daysfortrain* was inspired by cycles. The number is a result of joining trains, validations and test data size. The train and validation data set size follow one year cycle with 251 trading days. The test data set is then five months, thus 100 days.

$$D_{train} = D_{tr} \cup D_{te} \cup D_{va}$$

$$\emptyset = D_{tr} \cap D_{te} \cap D_{va}$$

Details regarding training, validation and test data for neural network will be covered in further chapters.

5.2.3 Suitable stock selector

This component takes care about choosing suitable stocks for trading. In the beginning of this chapter ground goals were jot down. One of them told us that the trading system should be able to choose a stock for trading without human interaction. This is important in order to not bring bias into the system. Traders often tend to include into the trading emotions, which can lead to a situation when the trading system is tested only on stocks with a majority of moves in uptrend. Then when the trader decides to jump into real trade and the stock suddenly changes to downtrend, the system cannot react. Therefore the designed system presented in this work chooses stock automatically with the help of three blocks. These blocks are called ups and downs, two averages and entropy.

5.2.3.1 Ups and downs

In order to teach the neural network how to react with a certain type of input it is necessary to give the network proper data. Proper data, in terms of this thesis, means to ensure that the number of cases when the stock goes down is close or is equal to the number of cases when the stock goes up. Thus the stock which contains 50% of cases when the price goes up and 50% of cases when the stock's price goes down will be evaluated as the best fit.

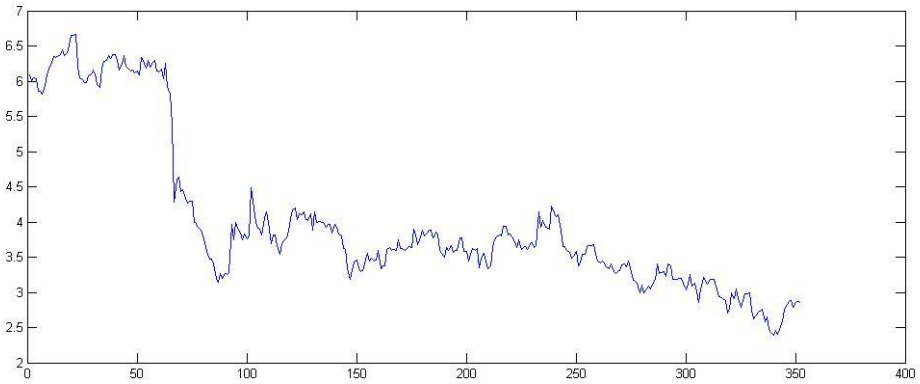

Figure 40: example of a stock ERII with 46.87% of cases going up

5.2.3.2 two averages

The simplest trade system can be based upon two averages, a short one and a long one. These averages create signals leading to buy, sell or hold. In the outer system the indicator helps select the best fitting stock.

```
% short and long averages

averageShort = tsmovavg(hist_close(startPoint :
startPoint+daysfortrain, 1), 'm', 7, 1);

averageLong = tsmovavg(hist_close(startPoint :
startPoint+daysfortrain, 1), 'm', 14, 1);
```

Figure 41: code sample

averageShort is the one average which reacts fast on market happenings, consequently *averageLong* reacts slower. Usually, traders buy when the fastest average crosses the slowest from the bottom. In this system the stock is favored when the last sample in the train data set shows *averageShort* above *averageLong*.

5.2.3.3 Entropy

As it was already mentioned in chapter 3.5.3 by mean of use entropy or Hurst Exponent 3.5.4 the system tries to estimate the predictability of a price data stream. In order to find which one of these indicators perform better on chosen time series, several tests were conducted to compare both of the indicators.

The tests were conducted on ten randomly chosen stocks, where the indicator picks one. Consequently a neural network is trained and tested. If the ratio of wrong direction predictions is in range of $\langle 0,0.5)$ then the indicator succeeded. Because each of the indicators are based on different methodologies different pick-up methods must be used.

1. Entropy – the stock with the lowest entropy is chosen as a best fit.
2. Hurst Exponent – Hurst exponent is calculated for each stock. The chosen stock must have number of cases going up bigger than cases going down and Hurst exponent in range of $\langle 0.5,1\rangle$

	Success of	
test	Entropy	Hurst Exponent
1	70%	50%
2	50%	40%
3	60%	30%
4	60%	60%
5	30%	30%
6	70%	40%

Table 3: shows success ratio for entropy versus Hurst Exponent test

The test reveals the entropy as being a more accurate indicator, for stocks participating in technology category in NASDAQ market. Researches on the internet have shown that Hurst Exponent performs better on commodities rather than on stocks. Therefore the entropy is favored over Hurst Exponent in the outer system.

5.2.4 Communication with inner system

After the suitable stocks are selected the outer system starts to communicate with the inner system. For now the inner system may be pictured as a black box whose inner construction is not known, but the interface is exposed, thus one can determine inputs and outputs from the system. Further in the text the reader will find chapters devoted to the inner system, which is represented by training algorithm of neural network.

When the outer system is in a point when it has its suitable stocks prepared, provides the stocks one by one to the inner system with information about numbers of neurons and delays. Thus the inner system replies for each configuration of delays and neurons with a trained neural network.

```
for neurons = 1: maxNeurons
    for delays = 1:maxDelays
      array(i,1) = innersystem(stock, neurons, delays);
    end
end
```

Figure 42: pseudocode of communication with inner system

The response from the inner system is recorded into an array. This array holds all trained networks and because the response is an object, instance of a class, the outer system can examine the attributes. It should be mentioned here that for one floating window there are 840 networks created. The *maxNeurons* variable is set up to fifteen neurons and the *maxDelays* to seven. Tests proved that even small network with up to fifteen neurons and seven delays can produce similar results such as huge networks with hundreds of neurons and delays. Patrik Vrba mentioned in his work (VRBA, 2011) that bigger networks do not guarantee better results and adding an additional neuron must be reasonable. There are also limitations, especially coming from the hardware which is not powerful enough to proceed calculations for bigger networks. The complexity of the calculation grows exponentially with a number of neurons and delays, therefore the calculation time grows as well. Thus the mentioned values were estimated as the best settings for the outer system and computer on which the tests were being run.

5.2.5 Suitable network selector

At this point the outer system gained necessary array containing for each stock its trained networks. The system now has to choose from over one thousand networks. It is necessary to note that neural networks which do not produce any buy signal for test data set are not included in the array thus the networks are discounted from future possible usage in the trade stage.

attributes	type	Description
performance	double	MSE of the trained network
trained_net	net	trained network structure
valMaskRatio	double	bad prediction ratio of buy signal for validation dataset
trainMaskRatio	double	bad prediction ratio of buy signal for train dataset
neurons	integer	number of neurons
delays	integer	number of delays
extratestRatio	double	bad prediction ratio of buy signal for test dataset
testvaltranRatio	double	bad prediction ratio of buy signal for validation and train dataset

Table 4: *NarxBuilder* attributes

The outer system then looks for a network which is trained with the ability to correctly predict buy signals. Note that the system looks for a minimal error in predictions, thus the closer the buy prediction ratio is to zero, the better it is. The successful networks are required to have the following attributes passing the conditions, as below:

$$\text{testvaltranRatio} \in \langle 0,0.5) \cap \text{extratestRatio} \in \langle 0,0.5)$$

Consequently the outer system counts a number of successful networks for one particular stock and chooses the stock with the highest cases. At this point the outer system knows which stock will be traded for the particular floating window. The next step before approaching the trading part is to pick one network out of all the successful networks.

testvaltranRatio	extratestRatio	trade ratio
0.375	0.3469	0.6363
0.25	0.3768	0.3333
0.4	0.3947	0.5384
0.5102	0.4153	0.25
0.2823	0.4179	0.5

Table 5: example of networks' statistics for one stock in one run

In early stages of testing, best performing networks on test data set were picked (network with lowest *extratestRatio* variable). But this approach was not successful in real trading. The picked networks seemed to get tired of performing on test data. And in real trading it produced high ratio of wrong signals.

Conducted experiments showed that over 65% of networks performing well on train, validation and test data are able to repeat that on trade set, but only 24% of best performing networks on test set are able to repeat the same on trade set. Thus the system does not pick the best performing network, but a pseudo-random number generator is taking care of the selection process.

5.2.6 Trade

When the outer system reaches the point where the stock is selected and an appropriate network as well. The system can approach the trade part which performs *buy*, *sell* and *hold* for particular floating window's trade data set.

SimulatedTrade class was created for the purpose of an easy network evaluation. It allows one to simulate the chosen trading period, in most cases twenty days, with the initial account balance. The output is then the account balance after the trading period has passed, which result in a loss or profit by subtracting the initial value from the final one.

The final account balance is resulted by buy, sell, hold signals. These signals are produced when the price direction is estimated. If the estimation shows that the price will raise, the system buys,(in case the account balance is experiencing an sufficient amount of money to stock purchases) or holds, otherwise the system sells. The figure below illustrates the signal creation process

$$signal = \begin{cases} buy & hist\_data < prediction - (0.8 * performance) \\ hold & hist\_data = prediction \\ sell & hist\_data > prediction \end{cases}$$

Figure 43: Buy, hold, sell signals

5.2.6.1 VIX usage

The trade component has two helpers which protect the system against losses in trades. The first one is the VIX index which measures traders sentiments. It simply tells the system when the traders are afraid to invest money and when they are fearless.

The system interrupts the buying process when the days average is above the current VIX value. In the average trade there are more than 50% of trading days interrupted by the index.

5.2.6.2 Stop loss

Is another helper of trade components. The Stop loss value prevents from money losing. When the automated system loses more than 3% of its initial account balance within a particular floating window, stop loss is executed and other buy signals are prohibited. Thus, the automated trade system risks only 3% at each floating window (trading month). Stop loss generates significantly less interruptions than the VIX index, but is still a very valuable and powerful tool in the automated trading system.

5.3 Inner system

The following text involves several neural networks which varies mainly in input arguments. The text does not include every models, but only the ones which were interesting and produced satisfying results. In the part 5.3.3, one can find the chosen network which is in comparison to others, deeply described.

In order to find the best settings for a chosen neural network a huge computer power is required. For that purpose a personal computer was not enough, thus the solution of remote MATLAB desktop provided by the Faculty of Business and Management appeared to be a very practical solution. Figure 62, located in appendices, shows large CPU usage required for testing.

5.3.1 NARX with Inside data

This model involves NARX neural network, Inside data and yahoo historical data as described in previous chapters.

Inside data are taken from MarketWatch and are normalized to fit historical trading data set. The download and parse workflow was explained in the past chapters

and will not be examined again. For further details regarding information one should access the thesis attachments with a full source code.

The part which has not been examined yet is the one concerning data normalization, which is required in order to have a data set fitting historical data dates. After a closer WatchMarket insider platform inspection, one may realize that the data sometimes do not follow official trading dates, but in some cases proceeds market actions of the official trading dates. This fact may result in a wrong data set applied as an input to the created system and thus cause a bad influence on the trading results. Therefore at first, the data are read and compared with the official trading dates. After that every anomaly found is modified to fit the closest trading day.

When the important data are obtained the neural network is trained for various neurons and delays. The network produces results which are recorded and the network is trained again, as the final result is the best neurons and delay setting for particular stock.

neurons	delay	MSE	wrong direction
18	2	0.4299	0.3571
19	2	0.1143	0.3928
11	3	0.0913	0.4074
14	3	0.1474	0.4074
17	3	0.5341	0.4074
19	3	0.8684	0.4074
15	3	0.0577	0.4444
16	3	0.2321	0.4444
18	5	0.1640	0.52
12	3	0.1515	0.5555
18	3	0.1051	0.5555

Table 6: MSFT and inside data.

The table above represents the experiment and its best results. The wrong direction column is the most important indicator of successful network. The experiment proves the ability of neural network to predict next movements with 35% chance to predict wrong ones. The wrong prediction range fluctuates around 35% – 65% when the median is 56%.

It is important to note that minimization of MSE did not give better results, quite the opposite. The models might be over fitted and product excellent results for training data, but they fail on trading ones.

neurons	delay	MSE	wrong direction
9	5	0.0511	0.56
16	4	1.0399	0.3846

Table 7: MSFT and inside data.

The table above represents an example when the best MSE result in the test case produces 56% of bad predictions while the network with the worst MSE in the test case produces only 38% of wrong predictions.

5.3.2 NARX with indicators

Many indicators were tested as an input argument of NARX. These indicators should lead neural networks to better results. Figure 44 shows the model with eleven input arguments $x(t)$, one target variable $y(t)$. The target variable stands for close price. The input variables which were recommended by (KARAA et. al., 2011) (Simple moving average, Weighted moving average, Momentum, Stochastic K%, Stochastic, RSI, MACD, LW R%, A/D, CCI) are accompanied with ROC and Force indicators.

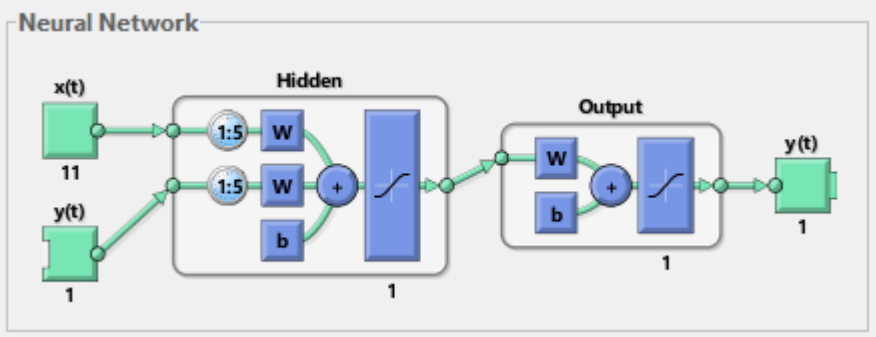

Figure 44: Neural network with more than one input variable

The researchers were able to achieve a stable predictability with a 30% wrong direction prediction. In my model, the similar results were not reproduced. The minimum value for wrong direction prediction is 39%. The wrong prediction range fluctuates around 39% – 80% with median around 60%.

5.3.3 NARX with buy-sell vector

NARX with sell-buy vector seems to be like one of the best performing systems, therefore unlike other systems it is deeply described.

5.3.3.1 Input variable

As many indicators were tested and the results were not satisfying the simple buy-sell indicator was created. Literally, the indicator says if the current price is higher or lower than the previous one. The indicator's range is {1, 0, -1}; 1 for price rise, 0 for the price holding the position and -1 for the price decrease.

$$x(t) = \begin{cases} 1 & y(t-1) < y(t) \\ 0 & y(t-1) = y(t) \\ -1 & y(t-1) > y(t) \end{cases}$$

Figure 45: indicator

Note that the indicator does not have the ability to see the future values, but only the previous one.

t	1	2	3	4	5	6	7	8
x(t)	-1	1	1	-1	0	0	0	1
y(t)	2.35	2.51	2.53	2.49	2.49	2.49	2.49	2.52

Figure 46: input and target example

- **t** – time
- **x(t)** – input of the neural network
- **y(t)** – targets of the neural network

5.3.3.2 Network's Layers

At this point the input and target variables were determined. The next important part which needs to be explained is the inner settings for NARX network. MATLAB Neural Network Toolbox provides whole readymade solutions, thus the creator does not have to code the inner structure on his own, but just tailor appropriate setting.

First of all, it is essential to mention here, that the network has only one hidden layer. This decision comes from experiments conducted over the sites containing more than one layer. These sites required significantly more time to train than networks containing only one layer, plus they did not resolve with better predictions. One layer solution is the best fit for the trading system while taking into account hardware resources and time for the calculation. Adding more layers may be considered by the reader only when better resources are available.

```
% Create a Nonlinear Autoregressive Network with
% External Input

inputDelays = 1:delays;
feedbackDelays = 1:delays;
hiddenLayerSize = neurons;

net = narxnet(inputDelays, feedbackDelays,
hiddenLayerSize);
```

Figure 47: network creation

The figure above shows a code used to create one hidden and one output layer for a particular size. Note that delays and neurons come from the outer system. When the instance of *narnex* class is created, transfer functions for each layer may be specified. The chapter devoted to transfer functions can be found in part 3.11.2.

```
% transfer functions

net.layers{1}.transferFcn= 'tansig';
net.layers{2}.transferFcn= 'purelin';
```

Figure 48: transfer functions

The chosen inner system uses *Tan-sigmoid* transfer function for its hidden layer and *purelin* transfer function for the output layer.

5.3.3.3 Data sets

After creating an instance of *narnex* class the network requires data decomposition setting up. In other words the network needs to know which samples to select as train, validation and test data. The inner setting offers a random selection, which leads to different train, validation and test set at every test running sessions. The MATLAB environment also provides a solution when the pseudo-random number generator produces the same set of numbers and thus the results of training is repeatable, but in the presented solution is not used.

It is also required to mention that random data set selection for test data does not produce wanted results. The reason of the failure lies in the whole concept. In a real trading environment one trades in a certain trading period. The trader does not proceed random days selection, therefore the inner system excludes the test data set from the initial data set (gained from the outer system). The test is then conducted on continuous series.

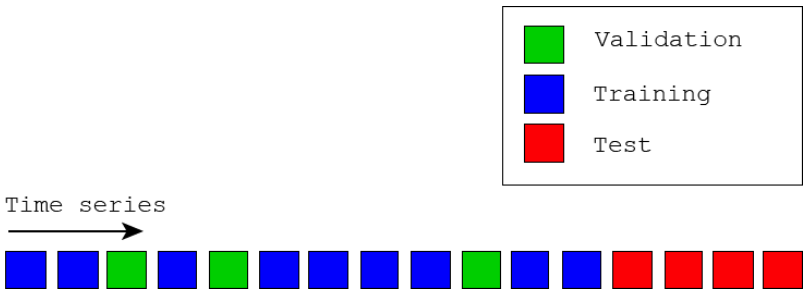

Figure 49: Data selection for neural network.

One of the most important factors which influences the ability of neural network to accurately predict future happenings in the time series is number of cases going to train, validate and test data set. In the previous chapter it was mentioned that the test data set is put to a five month cycles, one hundred trading days, thus only validation and train data size need to be determined.

```
% Data Division

net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'value';  % Divide up every value

% Setup Division of Data for Training, Validation,
Testing
net.divideParam.trainRatio = 85/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 0/100;
```

Figure 51: Data division

For this system, mainly for the purpose of avoiding *underfitting* and *overfitting*, the validation holds 15% and train 85% of data set which goes to the neural network. The following table shows the ratio of each part from the floating window perspective.

data set	size [%]
train	54.7170
validation	12.9380
test	26.9542
trade	5.3908

Figure 52: sizes in respect to floating window size

5.3.3.4 Performance function

The MATLAB Neural Network Toolbox offers a four performance functions. These functions tell the system how big is the error in the predictions. Functions differ from each other by their approach of counting the performance.

1. MAE - Mean absolute error performance function.
2. MSE - Mean squared error performance function.
3. SAE - Sum absolute error performance function.
4. SSE - Sum squared error performance function.

The inner system is built on MSE. The following code sample shows how to set the MSE performance function to be used by the network.

```
% Choose a Performance Function

net.performFcn = 'mse';
```

Figure 53: Mean square error

5.3.3.5 Training process

When the network is set up there is a need to train it. In this case MATLAB Neural Network Toolbox uses supervised learning, which was described in the chapter 3.11.5.1, and trains itself on train data set. Thus the train data set acts as a teacher.

There are many training functions offered by the toolbox. They can be divided into two categories. The first one uses Jacobien derivatives and is not supported by hardware's GPU[6] and the second one does not use Jacobien derivatives, but is supported by GPU and the MATLAB Parallel Computing toolbox.

The fastest solution for a hardware the testing is conducted on is the use of the functions with Jacobien derivatives. The toolbox offers two functions under the Jacobien category.

1. trainlm - Levenberg-Marquardt backpropagation.
2. trainbr - Bayesian Regulation backpropagation.

```
% Customizing training function

net.trainFcn = 'trainlm';  % Levenberg-Marquardt
```

Figure 54: setting the training function up

[6] Graphics processing unit

The functions uses gradient to find the smallest distance between predictions and targets. The training ends when the gradient reaches the point where it cannot be lower. It is important to mention that the Jacobien makes the learning process much faster. The following images show the training process, mainly the gradient and performance over eleven epochs.

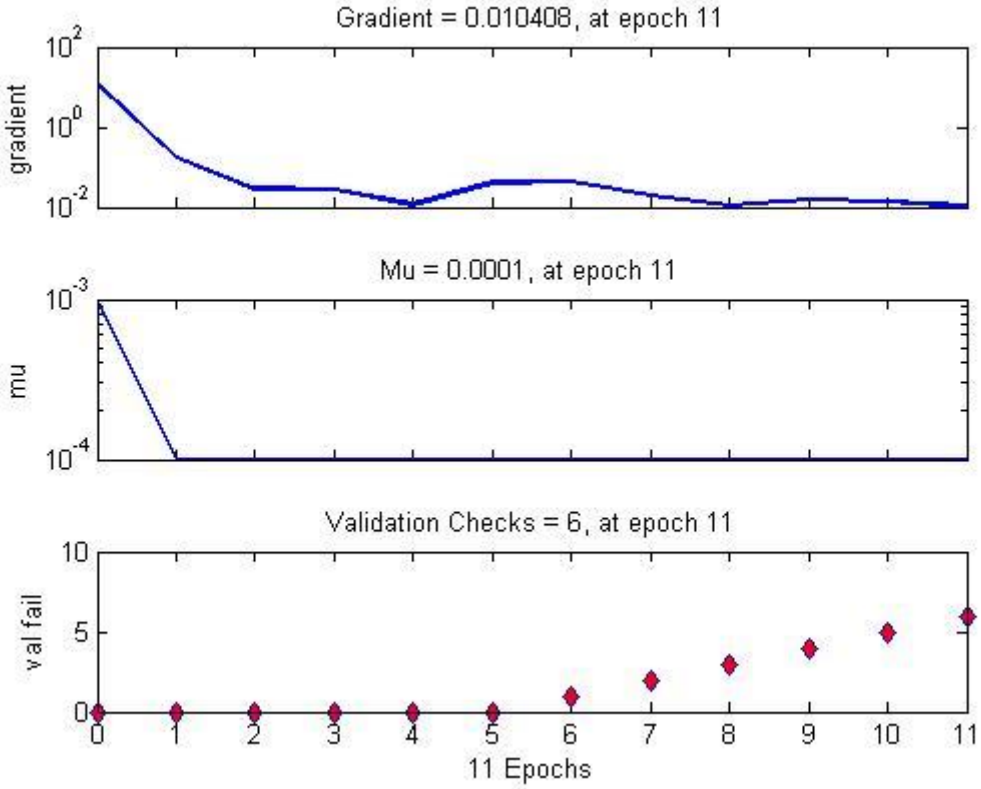

Figure 55: learning process

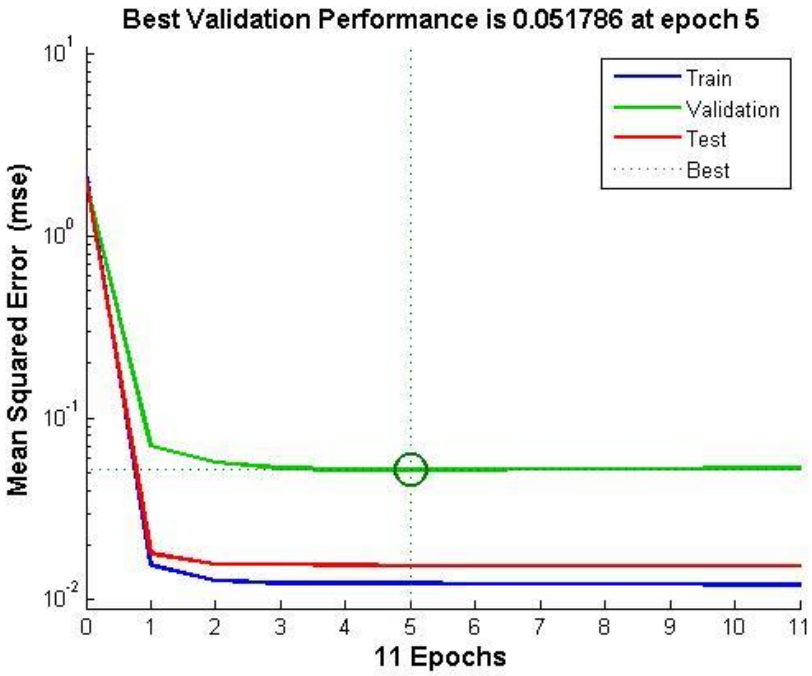

Figure 56: chart of performance

5.3.3.6 Network evaluation

As mentioned earlier the solution of minimization over network's performance on particular data set was not producing appropriate predictions and the ratio of bad predictions was very high. Therefore the presenting system is based on minimization over a wrong buy prediction ratio while taking into account the network's performance.

The performance counted by the MSE function helps the system identifying best buy opportunities in the data set. Buy signals must then follow this condition: the prediction subtracted by 80% of the performance must be higher than the current price (Figure 43). This solution mainly reduces the number of wrong buy signals and the cost as well. It also brings into the system such a situation as within one floating window there is any buy signal produced. Conducted tests have shown that over 50% of floating windows resulted in any buy signal generated.

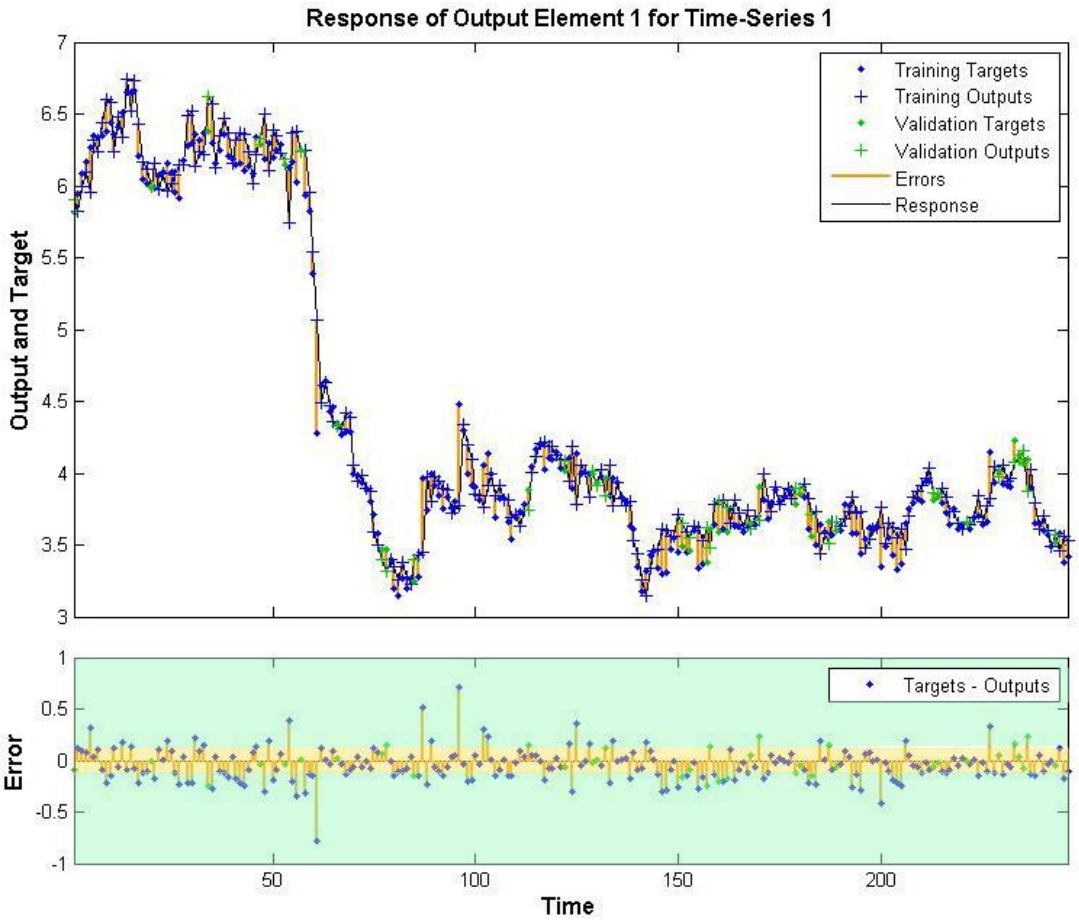

Figure 57: yellow area shows 80% of the performance

5.4 Trading system testing

Many tests were conducted to find appropriate settings for the inner and outer system. These tests are not presented in the following part, but only tests performed with the final version of automated trade system. The tests have the same initial settings, it is only up to the system to choose which settings work best.

5.4.1 Output

Automated trading system contains one output string for each floating window. It tells one how the trading system performed, important happenings and interruptions, and last but not least the account balance.

- Company – tells one which company was traded for a particular floating window
- Cycle/month – number of floating windows
- VIX interruption – how many trading days in the floating window were interrupted
- Stoploss value – stop loss [%]
- Stoploss interruption – tells if trades within the floating window were interrupted

```
trending stocks : 8
company : KOPN
cycle/month : 1
VIX interuption: 10
stoploss value: 0.03
stoploss interuption: NO
1)wrong BUY signal for validation, train data: 0.37113
2)wrong BUY signal for last values in dataset: 0.18182
4)wrong direction prediction for all days(interupted days included): NaN
5)wrong direction prediction for trading days(sell,hold,buy): 0.6
6)wrong BUY signal prediction for trading days: NaN
final account : 10000
******************************
trending stocks : 8
company : LPSN
cycle/month : 2
VIX interuption: 0
stoploss value: 0.03
stoploss interuption: NO
1)wrong BUY signal for validation, train data: 0.39286
2)wrong BUY signal for last values in dataset: 0.38889
4)wrong direction prediction for all days(interupted days included): 0.5
5)wrong direction prediction for trading days(sell,hold,buy): 0.3
6)wrong BUY signal prediction for trading days: 0.5
final account : 10389
******************************
trending stocks : 8
company : CIMT
cycle/month : 3
VIX interuption: 10
stoploss value: 0.03
```

```
stoploss interuption: NO
1)wrong BUY signal for validation, train data: 0.38261
2)wrong BUY signal for last values in dataset: 0.4881
4)wrong direction prediction for all days(interupted days included): 0.4
5)wrong direction prediction for trading days(sell,hold,buy): 0.4
6)wrong BUY signal prediction for trading days: 0.33333
final account : 12949
******************************
```

Figure 58: output example

Note that value Nan in MATLAB environment comes from operation $Nan = \frac{0}{0}$ which in the system means that does not proceed any market action over particular data set. The most important measurement of successful network in the output lies at line ***6***, which represents wrong buy direction mentioned earlier in the thesis.

5.4.2 Test 1

floating window/month	6) wrong BUY signal prediction for trading days	VIX	final account	company
1	NaN	10	10000	KOPN
2	0.5	0	10389	LPSN
3	0.3333	10	12949	CIMT
4	0	10	13141	DWCH
5	0	15	13212	MSFT
6	NaN	13	13212	MSFT
7	NaN	20	13212	KOPN
8	0	19	13212	KOPN
9	0	16	13804	AMKR
10	NaN	16	13804	CYOU
11	NaN	7	13804	INFA
12	NaN	1	13804	CIMT
13	NaN	18	13804	RCMT
14	0	14	14233	RCMT
15	1	15	14131	RCMT
16	NaN	12	14131	ASUR
17	NaN	7	14131	BCOR
18	NaN	11	14131	BCOR
19	NaN	7	14131	RCMT
20	NaN	18	14131	AFOP
21	0.75	12	13676	HHS

Table 8: output for test1

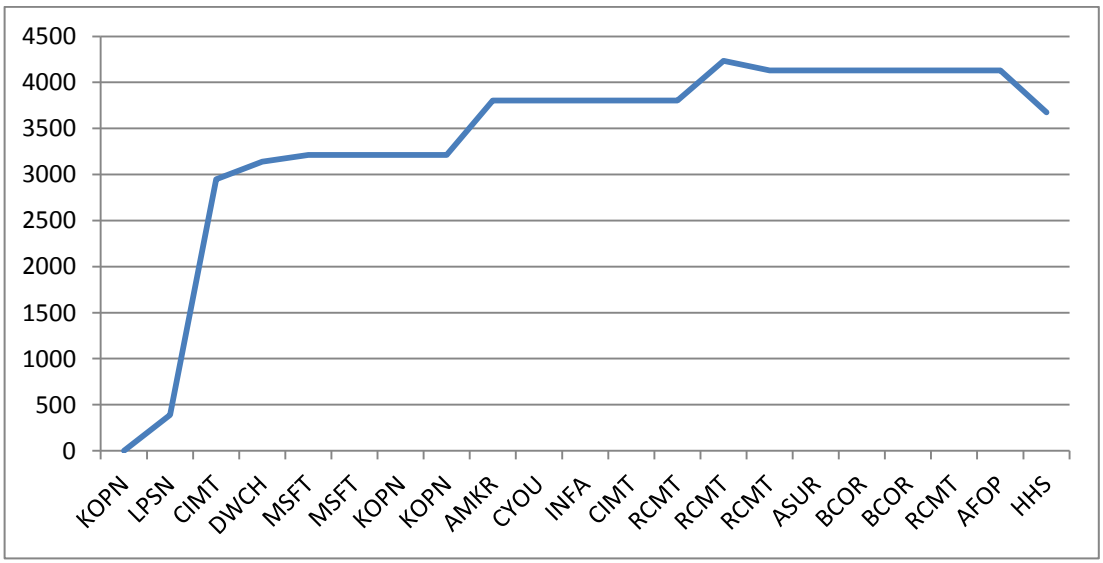

Figure 59: Profit for test 1

- Starting account balance: $10000
- Final account balance: $13676
- Average of wrong buy signal prediction for trading days: 28.7%

5.4.2.1 Test 1 Conclusion

As mentioned earlier, the most important sign of a successful network mainly resides in *wrong buy direction* which is in the output as a number six. The number says how many times we executed buy signals and that the price dropped on the next day. Ratio 28.7% means very good probability. This also supports $3676 in profit. It also means 36.7% return on investment. For the negative part, one can notice that the networks started with high profitability and slowed down over the time as it even dropped in the end.

5.4.3 Test 2

floating window/month	6) wrong BUY signal prediction for trading days	VIX	final account	company
1	NaN	10	10000	KOPN
2	1	0	9288	LPSN
3	0.3333	10	10574	CIMT
4	NaN	10	10574	DWCH
5	NaN	15	10574	MSFT
6	1	13	10617	MSFT
7	NaN	20	10617	CVR
8	0	19	10617	KOPN
9	0	16	10919	SAI
10	NaN	16	10919	CYOU
11	NaN	7	10919	INFA
12	NaN	1	10919	CIMT
13	0	18	11060	RCMT
14	0	14	11287	RCMT
15	1	15	11070	RCMT
16	NaN	12	11070	ASUR
17	NaN	7	11070	BCOR
18	NaN	11	11070	BCOR
19	0	7	12577	RCMT
20	NaN	18	12577	AFOP
21	1	12	12219	HHS

Table 9: output for test 2

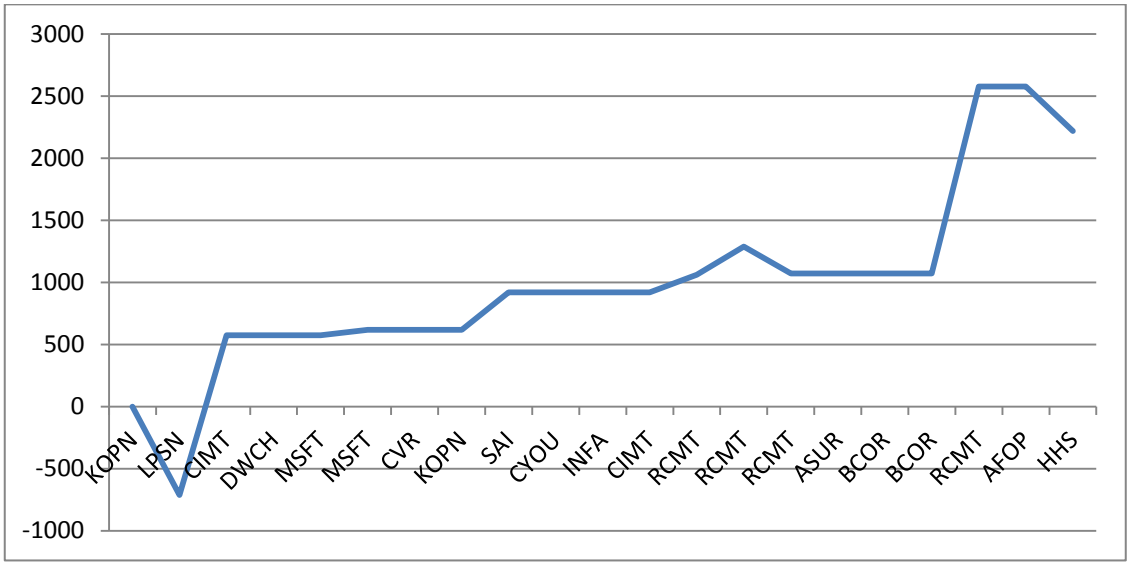

Figure 60: Profit for test 2

- Starting account balance: $10000
- Final account balance: $12219
- Average of wrong buy signal prediction for trading days: 43.3%

5.4.3.1 Test 2 Conclusion

The test was successful, the automated trading system was able to generate a significant profit, with return on investment equal to 22.2%. Ratio of wrong buy signal predictions increased to 43% in comparison to the first test. The profitability also slightly decreased due to a higher ratio of wrong buy signal predictions, and ended up on $2219 in profit. The biggest disadvantage is the fact that the account balance went below the initial one. The test is evaluated as being less successful than the first one, but still suitable for trading.

5.4.4 Test 3

floating window/month	6)wrong BUY signal prediction for trading days	VIX	final account	company
1	0.5	10	10221	KOPN
2	0.5	0	10619	LPSN
3	0.3	10	12553	CIMT
4	1	10	11647	DWCH
5	0	15	11906	MSFT
6	0.5	13	11929	MSFT
7	NaN	20	11929	KOPN
8	0	19	11929	KOPN
9	0	16	12503	AMKR
10	NaN	16	12503	PDFS
11	NaN	7	12503	INFA
12	NaN	1	12503	CIMT
13	NaN	18	12503	RCMT
14	0	14	12653	RCMT
15	NaN	15	12653	RCMT
16	NaN	12	12653	ASUR
17	NaN	7	12653	BCOR
18	NaN	11	12653	BCOR
19	0	7	13267	RCMT
20	NaN	18	13267	AFOP
21	0.8	12	12789	HHS

Table 10: output for test 3

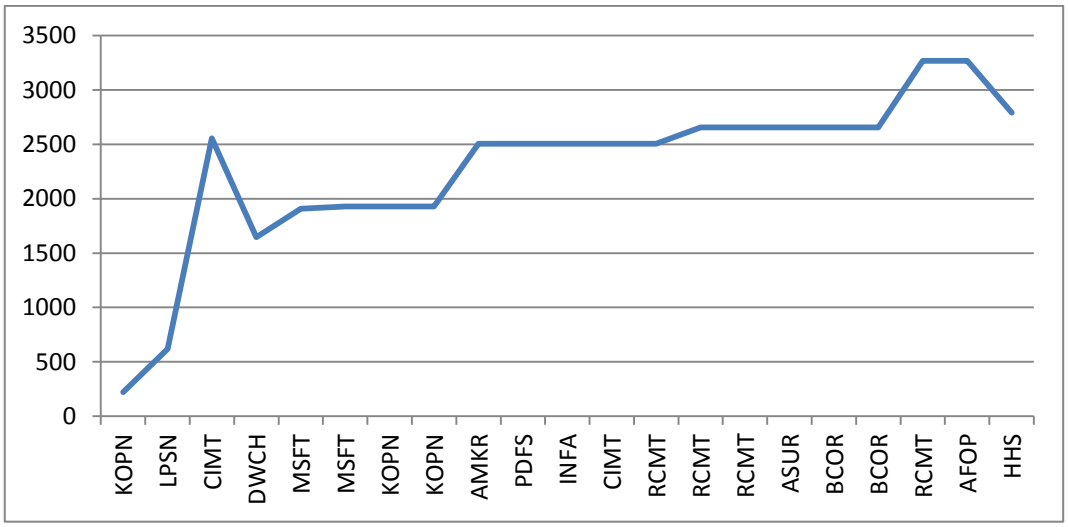

Figure 61: Profit for test 3

- Starting account balance: $10000
- Final account balance: $12789
- Average of wrong buy signal prediction for trading days: 32.7%

5.4.4.1 Test 3 conclusion

The last test performed very well. The trading process has generated $2789 in profit which means 28.9% in return on investments. The ratio of wrong buy signal predictions equal to 32.7% has a very good buy prediction rate. In comparison to previous tests, the account balance never went below ten thousand dollars, which makes the test case less “risky” for investors.

6 Conclusion

The aim of the thesis was to build automated trading systems which would proceed trades over historical data. The whole process of the system design requires knowledge from several parts of natural science, physics, economics or programing. These knowledge have increased and tailored throughout the whole process of model design. It resulted in many, more or less successful, test cases.

For automated trading systems to be successful, even a small help can result in success on the market. The final version of the system provides the trader with around 30% probability for wrong direction moves, which in other words means that around 70% buy orders are executed when the price is lower than next day's and results in a profit. Note that it does not guarantee a profit, but it gives one a very good probability to make one.

Presented automated system is successful in trading on historical data set, which has been proved within three test cases. It simulates real trading accuracy, therefore every inner happenings follow the same behavior as one may find in real trades. It is recommended, prior of using the system in real trade, to validate the system on actual data. The situation on the market might require some changes in order to generate profit in current circumstances.

Working on automated trading system has taken me into a new environment, the trading environment. Studying the market made me gain knowledge about trading in technology industry, trading methodologies and tools. It also helped me learn how to use neural networks in real life problems. Last but not least, I became such a huge fan of neural networks and their power that I am sure I will use them in my future projects and researches.

List of Figures

List of Tables

7 Bibliography

1. BHATTACHARYA, U. a H. DAOUK, 2002. The World Price of Insider Trading. [online]. Bloomington:2002 [cit. 2013-02-10]. Dostupné z: https://faculty.fuqua.duke.edu/~charvey/Teaching/BA453_2005/BD_The_world.pdf

2. CHICAGO BOARD OPTIONS EXCHANGE, 2012. VOLATILITY INDEXES at CBOE. [online]. 1. 1. 2012 [cit. 2013-04-21]. Dostupné z: http://www.cboe.com/micro/VIX/pdf/VolatilityIndexQRG2012-01-30.pdf

3. DEMUTH, a BEALE, 2000. *Neural Network Toolbox For Use with MATLAB* [Document].

4. DIACONESCU, E., 2008. *The use of NARX Neural Networks to predict Chaotic Time Series*. University of Pitesti. ISSN 1991-8755.

5. DOSTÁL, P., 2011. *Advanced decision making in business and public services*. Brno: CERM. ISBN 978-80-7204-747-5.

6. DOSTÁL, P., 2012. *Pokročilé metody rozhodování v podnikatelství a veřejné správě*. Brno: CERM. ISBN 978-80-7204-798-7.

7. ELLISON , B. a M. BOYD , 2007. Journal of Computer-Mediated Communication. *Social Network Sites: Definition, History, and Scholarship* [online] [cit. 2012-11-12]. Dostupné z: http://jcmc.indiana.edu/vol13/issue1/boyd.ellison.html

8. HAJEK, M., 2005. *NEURAL NETWORKS* [online] [cit. 2013-01-04]. Dostupné z: http://www.cs.ukzn.ac.za/notes/NeuralNetworks2005.pdf

9. HENNESSEY, P., 2010. *Using the Hurst Exponent in Forex Trading* [online] [cit. 2013-04-18]. Dostupné z: http://www.trade-forex-harmonic-patterns.com/hurst-exponent.html

10. HOLLMEN , , 1996. *Artificial neural networks* [online]. 8. 3. 1996 [cit. 2013-02-12]. Dostupné z: http://users.ics.aalto.fi/jhollmen/dippa/node8.html

11. HONG, H., J. D. KUBIK a C. STEIN, 2006. Harvard Institute of Economic Research. In: *http://time.dufe.edu.cn* [online].2006 [cit. 2012-11-11]. Dostupné z: http://time.dufe.edu.cn/spti/article/harvard/2006.pdf

12. INDIANA UNIVERSITY SOUTH BEND, 2008. [online].2008 [cit. 25-04-2013]. Dostupné z: http://www.cs.iusb.edu/~danav/teach/c463/neuron_real.gif

13. investopedia, 2013. *Bond* [online] [cit. 2013-02-20]. Dostupné z: http://www.investopedia.com/terms/b/bond.asp#axzz2LMwFT4lR

14. investopedia, 2013. *Fence (Options)* [online] [cit. 2013-02-20]. Dostupné z: http://www.investopedia.com/terms/f/fence-options.asp#axzz2LMwFT4lR

15. investopedia, 2013. *Forex Tutorial: Introduction to Currency Trading* [online] [cit. 2013-02-19]. Dostupné z: http://www.investopedia.com/university/forexmarket/#axzz2LMwFT4lR

16. investopedia, 2013. *Futures Fundamentals: Introduction* [online] [cit. 2013-02-19]. Dostupné z: http://www.investopedia.com/university/futures/#axzz2LMwFT4lR

17. investopedia, 2013. *Random Walk Theory* [online] [cit. 2013-04-18]. Dostupné z: http://www.investopedia.com/terms/r/randomwalktheory.asp

18. investopedia, 2013. *Stock* [online] [cit. 2013-02-20]. Dostupné z: http://www.investopedia.com/terms/s/stock.asp#axzz2LMwFT4lR

19. investopedia.com, 2013. *Fundamental Analysis* [online] [cit. 2013-10-02]. Dostupné z: http://www.investopedia.com/terms/f/fundamentalanalysis.asp#axzz2KXP6bx2s

20. KARAA, , BOYACIOGLU a BAYK, 2011. In: *Predicting direction of stock price index movement using artificial neural* [online].2011 [cit. 2013-04-15]. Dostupné z: http://www.cecm.usp.br/~felipeh/fama_french_ann_fcst.pdf

21. MALÍKOVÁ, K., 2013. Fundamentální analýza. In: *Business partners* [online].2008-2013 [cit. 2013-02-02]. Dostupné z: http:// www.businesspartners.cz/fundamentalni-analyza.html

22. MASHBURN, B., 2010. How to Download Historical Stock Data into Matlab. *luminouslogic* [online] [cit. 2013-02-10]. Dostupné z: http://luminouslogic.com

23. MATHWORKS, 2012. *Neural Network Toolbox, help*. The MathWorks, Inc. 1994-2012.

24. MURPHY, J. J., 1999. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and …*. New York: New York Institute of Finance. ISBN 978-0735200661.

25. NASDAQ, 2013. What Is NASDAQ? [online] [cit. 2013-04-21]. Dostupné z: http://www.nasdaqomx.com/aboutus/whatisnasdaq/

26. NIKOLAEV, N., 2012. [online].2012 [cit. 2013-04-25]. Dostupné z: http:// homepages.gold.ac.uk/nikolaev/perceptr.gif

27. NTNU, 2010. Department of Computer and Information Science. In: *Unsupervised Learning in Neural Networks* [online]. 7. 3. 2010 [cit. 2013-02-12]. Dostupné z: http://www.idi.ntnu.no/emner/it3708/lectures/notes/learn-unsup.pdf

28. ORACLE, 2013. oracle.com. *The java tutorials* [online] [cit. 2013-02-10]. Dostupné z: http://docs.oracle.com/javase/tutorial/essential/regex/intro.html

29. PEARSON, K., 1905. The Problem of the Random Walk. Nature.

30. SIMPSON, S. D., 2012. A Primer On Investing In The Tech Industry. *investopedia* [online]. 8. 8. 2012 [cit. 2013-04-21]. Dostupné z: http:// www.investopedia.com/articles/stocks/10/primer-on-the-tech-industry.asp

31. SPITZER, , 1976. Principles of Random Walk. Springer-Verlag. ISBN 9780387901503.

32. STOCKCHARTS, 2013. stockcharts. *Rate of Change (ROC)* [online] [cit. 2013-02-21]. Dostupné z: http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:rate_of_change

33. TEXAS A&M AGRILIFE RESEARCH & EXTENSION CENTER AT DALLAS, 2001. Types of Charts. [online].2001 [cit. 2013-04-25]. Dostupné z: http://dallas.tamu.edu/econ/Technical%20Analysis/types_of_charts.html

34. VARIAN, H. a H. CHOI, 2009. In: *Predicting the Present with Google Trends* [online]. 10. 4. 2009 [cit. 2013-02-11]. Dostupné z: http://static.googleusercontent.com/external_content/untrusted_dlcp/www.google.com/cs//googleblogs/pdfs/google_predicting_the_present.pdf

35. VRBA, P., 2011. *Využití prostředků umělé inteligence na finančních trzích.* [78 s.]. Brno.

36. YAHOO, 2012. Yahoo! help. *Historical prices and adjusted close* [online]. 9. 8. 2012 [cit. 2013-02-13]. Dostupné z: http://help.yahoo.com/kb/index?locale=en_US&page=content&y=PROD_FIN&id=SLN2311&impressions=true

37. YOSSI , M., 2012. The official google search blog. *Inside search* [online]. 27. 9. 2012 [cit. 2013-02-11]. Dostupné z: http://insidesearch.blogspot.co.il/2012/09/insights-into-what-world-is-searching.html

38. YU, a K. SUBHASH, 2012. *A Survey of Prediction Using Social Media*. Oklahoma: Department of Computer Science, Oklahoma State University.

8 Appendices

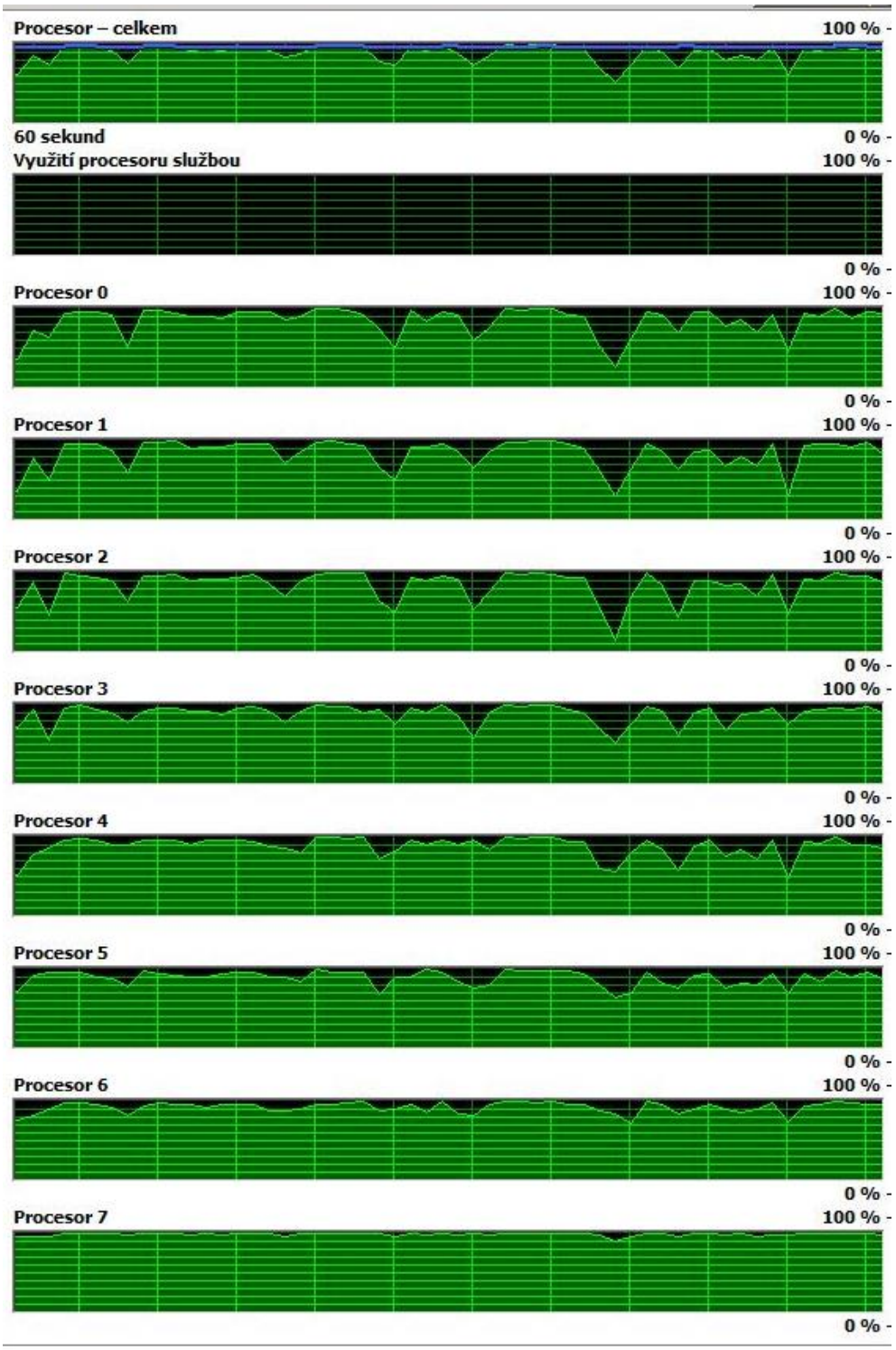

Figure 62: The usage of eight matlabPC procesors during testing phase.