

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

FOREXOVÝ AUTOMATICKÝ OBCHODNÍ SYSTÉM ZALOŽENÝ NA  
NEURONOVÝCH SÍTÍCH

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. PETR KAČER

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

# FOREXOVÝ AUTOMATICKÝ OBCHODNÍ SYSTÉM ZALOŽENÝ NA NEURONOVÝCH SÍTÍCH

FOREX AUTOMATED TRADING SYSTEM BASED ON NEURAL NETWORKS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PETR KAČER

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. VÁCLAV JIRSÍK, CSc.

BRNO 2015



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

# Diplomová práce

magisterský navazující studijní obor  
Kybernetika, automatizace a měření

**Student:** Bc. Petr Kačer

**ID:** 138951

**Ročník:** 2

**Akademický rok:** 2014/2015

## NÁZEV TÉMATU:

**Forexový automatický obchodní systém založený na neuronových sítích**

## POKYNY PRO VYPRACOVÁNÍ:

1. Provedte rešerši na téma využití neuronových sítí při obchodování na měnových trzích.
2. Navrhněte a realizujte obchodní systém s možností přidávat obchodní strategie jako moduly.
3. Navrhněte a realizujte obchodní strategii pro tento obchodní systém založenou na umělých neuronových sítích.
4. Strategii otestujte na historických datech a při obchodování na demo účtu u brokera.
5. Dosažené výsledky zhodnoťte.

## DOPORUČENÁ LITERATURA:

Dle doporučení vedoucího práce a dle vlastního výběru.

**Termín zadání:** 9.2.2015

**Termín odevzdání:** 18.5.2015

**Vedoucí práce:** doc. Ing. Václav Jirsík, CSc.

**Konzultanti diplomové práce:**

**doc. Ing. Václav Jirsík, CSc.**

*Předseda oborové rady*

## UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Hlavním cílem této práce je vytvoření forexového automatického obchodního systému s možností přidávat obchodní strategie jako moduly a realizace modulu obchodní strategie založené na neuronových sítích. Implementovaný obchodní systém se skládá z klientské části pro obchodní platformu MetaTrader 4 a ze serverové GUI aplikace. Moduly obchodních strategií jsou realizovány formou dynamických knihoven. Navržená obchodní strategie využívá vícevrstvé neuronové sítě pro predikci směru 45-ti minutového plovoucího průměru zavíracích hodnot ceny v časovém horizontu jedné hodiny. Neuronové sítě byly schopné najít souvislost mezi vstupy a výstupem a predikovat pokles či nárůst s úspěšností vyšší než 50%. Při živém obchodování na demo účtu se pro měnový pár EUR/USD strategie projevila jako zisková, pro měnový pár GBP/USD naopak jako ztrátová. Při testech strategie na historických datech za rok 2014 bylo dosaženo zisku v případě obchodování na měnovém páru EUR/USD ve směru dlouhodobého trendu. Při obchodování proti směru trendu na měnovém páru EUR/USD a ve směru, i proti směru trendu na měnovém páru GBP/USD byla strategie ztrátová.

## KLÍČOVÁ SLOVA

Vícevrstvá neuronová síť, forex, predikce, obchodování, strategie, automatický, systém, databáze, server, klient, meziprocesová komunikace, knihovna, C++, Qt5.

## ABSTRACT

Main goal of this thesis is to create forex automated trading system with possibility to add trading strategies as modules and implementation of trading strategy module based on neural networks. Created trading system is composed of client part for MetaTrader 4 trading platform and server GUI application. Trading strategy modules are implemented as dynamic libraries. Proposed trading strategy uses multilayer neural networks for prediction of direction of 45 minute moving average of close prices in one hour time horizon. Neural networks were able to find relationship between inputs and output and predict drop or growth with success rate higher than 50%. In live demo trading, strategy displayed itself as profitable for currency pair EUR/USD, but it was losing for currency pair GBP/USD. In tests with historical data from year 2014, strategy was profitable for currency pair EUR/USD in case of trading in direction of long-term trend. In case of trading against direction of trend for pair EUR/USD and in case of trading in direction and against direction of trend for pair GBP/USD, strategy was losing.

## KEYWORDS

Multilayer neural network, forex, prediction, trading, strategy, automated, system, database, server, client, interprocess communication, library, C++, Qt5.

KAČER, Petr. *Forexový automatický obchodní systém založený na neuronových sítích*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2015. 77 s. Vedoucí práce byl doc. Ing. Václav Jirsík, CSc.

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Forexový automatický obchodní systém založený na neuronových sítích“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora(-ky)

## PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce doc. Ing. Václavu Jirsíkovi CSc. za jeho cenné rady a připomínky. Dále děkuji své rodině za pomoc při kontrole pravopisu a v neposlední řadě děkuji také všem, kteří se rozhodli si tuto práci přečíst.

Brno .....

.....

podpis autora(-ky)

# OBSAH

Úvod	10
<b>1 Vícevrstvá neuronová síť</b>	<b>12</b>
1.1 Neuron . . . . .	12
1.2 Topologie . . . . .	13
1.3 Učení . . . . .	15
1.4 Neuronové sítě a predikce . . . . .	17
<b>2 Forex</b>	<b>18</b>
2.1 Historie forexu . . . . .	18
2.2 Základy forexu . . . . .	19
2.3 Obchodní strategie . . . . .	22
2.4 Neuronové sítě a forex . . . . .	25
<b>3 Obchodní systém AutoTrader</b>	<b>29</b>
3.1 Použité technologie . . . . .	29
3.2 Princip činnosti . . . . .	29
3.3 Komunikace mezi klientem a serverem . . . . .	30
3.4 Serverová část systému . . . . .	36
3.5 Klientská část systému . . . . .	39
3.6 Knihovny obchodních strategií . . . . .	43
3.7 Obchodní deník . . . . .	48
<b>4 Obchodní strategie založená na neuronových sítích</b>	<b>50</b>
4.1 Princip obchodní strategie . . . . .	50
4.2 Tvorba datových množin . . . . .	53
4.3 Učení a testování neuronových sítí . . . . .	58
4.4 Implementace obchodní strategie . . . . .	64
4.5 Testování strategie na historických datech . . . . .	66
4.6 Testování strategie na demo účtu . . . . .	68
<b>5 Závěr</b>	<b>70</b>
<b>Literatura</b>	<b>73</b>

# SEZNAM OBRÁZKŮ

1.1	Neuron . . . . .	12
1.2	Přenosové (aktivační) funkce neuronu [38] . . . . .	13
1.3	Značení neuronů, vah, vstupů a výstupů v neuronové síti . . . . .	14
2.1	Čárový graf . . . . .	19
2.2	Svícový graf . . . . .	20
2.3	Sloupcový graf . . . . .	20
2.4	Časy hlavních obchodních seancí . . . . .	21
2.5	Jednoduchý a exponenciální klouzavý průměr o periodě 14 . . . . .	23
2.6	Indikátor MACD o periodách 12, 26 a 9 . . . . .	24
2.7	Hybridní systém s neuronovou sítí a expertním systémem [16] . . . . .	27
2.8	Obchodní systém využívající samoorganizační mapy a SVR [2] . . . . .	27
3.1	Principiální schéma obchodního systému AutoTrader . . . . .	30
3.2	Průběh registrace klienta na serveru . . . . .	31
3.3	Komunikace mezi klientem a serverem . . . . .	32
3.4	AutoTrader server - výpis logu . . . . .	36
3.5	AutoTrader server - tabulka připojení . . . . .	36
3.6	AutoTrader server - seznam klientů . . . . .	37
3.7	AutoTrader server - správa strategií . . . . .	37
3.8	AutoTrader server - nastavení serveru . . . . .	38
3.9	AutoTrader server - nastavení databáze . . . . .	38
3.10	Obchodní platforma MetaTrader 4 . . . . .	39
3.11	Nastavení parametrů expert advisora AutoTrader . . . . .	42
3.12	Obchodní deník pro AutoTrader . . . . .	48
3.13	Entity relationship diagram databáze pro logování obchodů . . . . .	49
4.1	Schéma obchodní strategie založené na neuronových sítích . . . . .	50
4.2	Výpočet vstupů neuronové sítě . . . . .	51
4.3	Navržený a realizovaný rozhodovací mechanismus strategie . . . . .	52
4.4	Program DataConvertor . . . . .	54
4.5	Rozložení velikostí rozdílů průměrů (GBP/USD, 2014/08-10) . . . . .	55
4.6	Určení požadovaného výstupu neuronové sítě . . . . .	56
4.7	Průběh učení neuronové sítě (EUR/USD, 2014/06) . . . . .	58
4.8	Test klasifikace na validačních datech (EUR/USD, 2014/09) . . . . .	60
4.9	Stavový dialog obchodní strategie . . . . .	65
4.10	Tester strategií v platformě MetaTrader 4 . . . . .	66
4.11	Zisková obchodní pozice z obchodování na demo účtu . . . . .	69



## SEZNAM TABULEK

4.1	Meze tříd požadovaného výstupu (EUR/USD) . . . . .	57
4.2	Meze tříd požadovaného výstupu (GBP/USD) . . . . .	57
4.3	Úspěšnost predikce na tréninkových datech (EUR/USD) . . . . .	61
4.4	Úspěšnost predikce na tréninkových datech (GBP/USD) . . . . .	61
4.5	Úspěšnost predikce na testovacích datech (EUR/USD) . . . . .	62
4.6	Úspěšnost predikce na testovacích datech (GBP/USD) . . . . .	62
4.7	Úspěšnost predikce na validačních datech (EUR/USD) . . . . .	63
4.8	Úspěšnost predikce na validačních datech (GBP/USD) . . . . .	63
4.9	Výsledky obchodování na historických datech (EUR/USD) . . . . .	67
4.10	Výsledky obchodování na historických datech (GBP/USD) . . . . .	67
4.11	Úspěšnost predikce neuronových sítí pro test na demo účtu . . . . .	68
4.12	Výsledky obchodování na demo účtu (EUR/USD) . . . . .	68
4.13	Výsledky obchodování na demo účtu (GBP/USD) . . . . .	69

## SEZNAM VÝPISŮ

3.1	Požadavek na registraci . . . . .	32
3.2	Hlášení o stavu . . . . .	32
3.3	Požadavek na aktualizaci . . . . .	33
3.4	Požadavek na data pro aktualizaci . . . . .	33
3.5	Odpověď s daty pro aktualizaci . . . . .	34
3.6	Požadavek na obchodní rozhodnutí . . . . .	34
3.7	Odpověď s obchodním rozhodnutím . . . . .	35
3.8	Funkce klientské knihovny pro připojení a registraci na serveru . . . . .	40
3.9	Funkce klientské knihovny pro žádost o aktualizaci . . . . .	40
3.10	Funkce klientské knihovny pro tvorbu datové odpovědi . . . . .	40
3.11	Funkce klientské knihovny pro zpracování obchodního rozhodnutí . . . . .	41
3.12	Pomocné funkce klientské knihovny . . . . .	41
3.13	Základní funkce tvořící expert advisora . . . . .	42
3.14	Rozhraní knihovny obchodní strategie . . . . .	44
3.15	Hlavičkový soubor třídy strategie . . . . .	44
3.16	Konstruktor třídy a metoda <i>requestedData</i> . . . . .	45
3.17	Metoda <i>tradeData</i> . . . . .	45
3.18	Metoda <i>makeDecision</i> - 1. část . . . . .	46
3.19	Metoda <i>makeDecision</i> - 2. část . . . . .	46
3.20	Metoda <i>makeDecision</i> - 3. část . . . . .	47
3.21	Metoda <i>onClientDoubleClicked</i> . . . . .	47
3.22	Metoda <i>decisionStatus</i> . . . . .	47
4.1	Seznam neuronových sítí ve formátu xml . . . . .	64

# ÚVOD

Forex (někdy též FX) je často používaná zkratka pro foreign exchange, neboli měnový trh. Forex je co do objemu největším finančním trhem na světě, je to decentralizovaný trh tvořený propojením bank, investičních fondů, pojišťoven a brokerských společností pomocí sítě internet [37].

Obchodovat na forexu je výhodné hned z několika důvodů. Můžeme vydělat jak na poklesu, tak i na vzestupu ceny, obchoduje se 24 hodin denně, pákový efekt nám umožňuje začít obchodovat s malým počátečním kapitálem a díky vysoké likviditě máme možnost vždy prodat nebo koupit měnu za rozumnou cenu [37].

Na forexu se pohybuje nepřehledné množství různých automatických obchodních systémů. Rozšířeným použitím neuronových sítí při automatickém obchodování je predikce průběhu ceny či technických ukazatelů [25][31].

Vícevrstvé neuronové sítě jsou vhodným kandidátem pro predikci měnového trhu z toho důvodu, že tento trh je vysoce volatilním, komplexním a značně nelineárním prostředím obsahujícím šum [41].

Vícevrstvá neuronová síť je matematický model schopný naučit se nelineární závislosti mezi vstupy a výstupy ze zaznamenaných historických dat [17]. Díky tomu, že neuronovou síť vytváříme pouze na základě historických dat, není pro její vytvoření nutná znalost daného systému [40].

Vícevrstvou neuronovou síť můžeme použít například pro aproximaci nelineární funkce, klasifikaci dat, predikci, simulaci systémů či filtraci signálu [13][15]. Další výhodou vícevrstvé neuronové sítě je fakt, že dovede pracovat s velkými a komplexními systémy s větším množstvím vzájemně závislých parametrů [40].

Prvním cílem této práce je vytvoření forexového automatického obchodního systému s možností přidávat obchodní strategie jako moduly a s funkcí automatického logování uskutečněných obchodů do databáze. Druhým cílem práce je potom realizace modulu obchodní strategie založené na neuronových sítích.

První kapitola je věnována teorii vícevrstvých neuronových sítí. Nejprve je zde popsán samotný neuron a jeho přenosové funkce, dále potom topologie vícevrstvé neuronové sítě a nakonec algoritmus učení backpropagation. Závěr kapitoly je věnován rešerši na téma využití (nejen) vícevrstvých neuronových sítí k predikci.

Druhá kapitola má za úkol podat základní informace o forexu. V první části kapitoly se nachází stručné shrnutí historie forexu, druhá část kapitoly vysvětluje základní typy grafů používaných pro zobrazování průběhu ceny a popisuje nejobchodovanější měnové páry a časy obchodních seancí. Třetí část kapitoly se věnuje analýze trhu a obchodním strategiím a poslední, čtvrtá část, je výsledkem rešerše na téma využití neuronových sítí při obchodování na forexu.

Třetí kapitola se zabývá návrhem a realizací forexového automatického obchodního systému AutoTrader. Kapitola nejprve popisuje technologie použité pro realizaci tohoto systému a princip jeho činnosti. Vzhledem k tomu, že se jedná o systém typu klient-server, nechybí zde ani popis zpráv posílaných mezi klientem a serverem během jejich komunikace.

Kapitola dále popisuje grafické uživatelské rozhraní serverové aplikace, klientskou knihovnu obsahující funkce pro komunikaci se serverem a implementaci klientské části systému v obchodní platformě MetaTrader 4 s využitím této knihovny.

Dále následuje část kapitoly, která na příkladu se strategií založené na překřížení dvou klouzavých aritmetických průměrů o rozdílné periodě demonstruje způsob, jakým lze vytvořit modul obchodní strategie.

Poslední část třetí kapitoly je věnována obchodnímu deníku, tj. programu sloužícímu k vyhledávání a zobrazování uskutečněných obchodů zapsaných v databázi a popisu účelu tabulek této databáze.

Poslední, čtvrtá kapitola má za cíl návrh, realizaci a otestování obchodní strategie založené na vícevrstvých neuronových sítích. Kapitola nejprve vysvětluje princip činnosti této strategie a způsob tvorby datových množin pro neuronové sítě.

Další část kapitoly řeší učení neuronových sítí a jejich testování na testovacích a validačních datech a poslední dvě části kapitoly jsou věnovány samotnému testování obchodní strategie na historických datech a při zkušebním obchodování na demo účtu u vybraného brokera.

Závěr práce shrnuje dosažené výsledky a postupy použité při realizaci obchodního systému i obchodní strategie. První část závěru je věnována obchodnímu systému AutoTrader, druhá část potom samotné obchodní strategii.

# 1 VÍCEVRSTVÁ NEURONOVÁ SÍŤ<sup>1</sup>

Tato kapitola vysvětluje základy teorie vícevrstevných neuronových sítí. První část kapitoly se zabývá popisem neuronu, jakožto prvku, ze kterého je složena celá neuronová síť. Vícevrstvá neuronová síť včetně několika doporučení k volbě topologie je popsána v druhé části kapitoly. Třetí část kapitoly je věnována učení neuronové sítě algoritmem backpropagation a poslední, čtvrtá část, je výsledkem rešerše a obsahuje příklady použití (nejen) vícevrstevných neuronových sítí k predikci.

## 1.1 Neuron

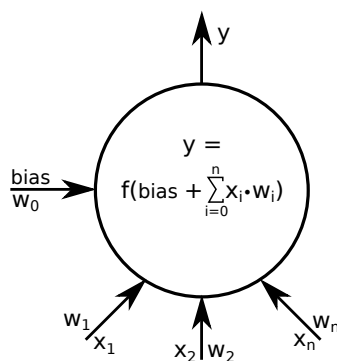
Neuron (Obr. 1.1) je základním stavebním kamenem každé neuronové sítě. Každý neuron má vstupy  $x_1$  až  $x_n$  opatřené vahami  $w_1$  až  $w_n$ . Váhy vstupů mohou být i záporné, čímž je vyjádřen jejich inhibiční charakter [38].

Vážená suma vstupních hodnot představuje vnitřní potenciál neuronu, který lze vypočítat podle vztahu

$$y_{in} = bias + \sum_{i=0}^n x_i \cdot w_i, \quad (1.1)$$

kde  $x_1$  až  $x_n$  je vektor vstupů neuronu,  $w_1$  až  $w_n$  je vektor vah těchto vstupů a bias je konstantní vstup neuronu [38].

Pokud k neuronu přidáme vstup o stálé hodnotě 1, můžeme bias včlenit do vztahu jako hodnotu váhy tohoto vstupu. Konečný výstup neuronu potom dostaneme dosazením vnitřního potenciálu neuronu do jeho přenosové funkce [38].



Obr. 1.1: Neuron

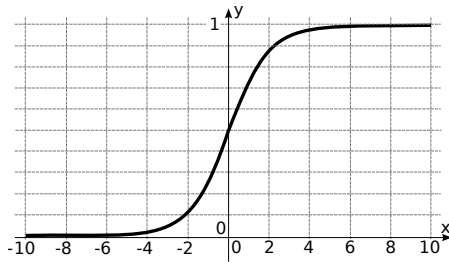
Nejdůležitějšími přenosovými (aktivačními) funkcemi v neuronových sítích typu backpropagation jsou unipolární sigmoida a hyperbolický tangens (Obr. 1.2). I když se většinou předpokládá použití stejné přenosové funkce pro celou neuronovou síť

<sup>1</sup>Text převzat z [13] vyjma kapitoly 1.4.

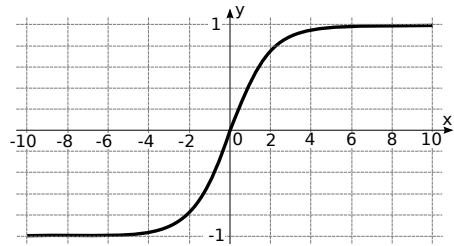
nebo alespoň její jednotlivé vrstvy, není to pravidlem a každý neuron teoreticky může mít jinou přenosovou funkci [6].

Uvedené přenosové funkce jsou užitečné u neuronových sítí typu backpropagation zejména proto, že je lze snadno zderivovat a dosazení do již zderivované funkce snižuje výpočetní náročnost algoritmu během učení neuronové sítě. Strmost průběhu funkce je dána hodnotou parametru  $s$  [6].

Unipolární sigmoida se používá v sítích, u kterých požadujeme binární výstup, nebo výstup v intervalu 0 až 1. Hyperbolický tangens se používá u sítí s požadovaným výstupem v rozsahu -1 až 1 [6].



$$f(x) = \frac{1}{1 + e^{-sx}} \quad (1.2)$$



$$f(x) = \frac{1 - e^{-sx}}{1 + e^{-sx}} \quad (1.3)$$

Obr. 1.2: Přenosové (aktivační) funkce neuronu [38]

## 1.2 Topologie

Vícevrstvá neuronová síť (Obr. 1.3) je síť s jednou nebo více skrytými vrstvami neuronů umístěnými mezi vstupní a výstupní vrstvou. Neuron vstupní vrstvy posílá svůj vstup ke všem neuronům vnitřní vrstvy. Výstupy vnitřní vrstvy jsou přivedeny na vstupy každého neuronu vyšší vrstvy a jsou vynásobeny příslušnými vahami [6].

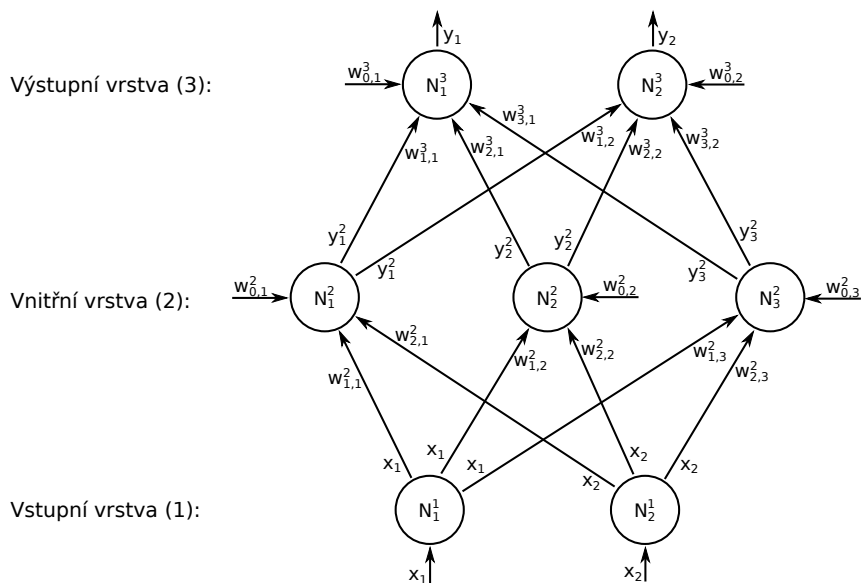
Výstup  $k$ -tého neuronu nacházejícího se v  $n$ -té skryté nebo výstupní vrstvě vícevrstvé neuronové sítě lze vypočítat podle vzorce

$$y_k^n = f \left( w_{0,k}^n + \sum_{i=1}^m y_i^{n-1} \cdot w_{i,k}^n \right), \quad (1.4)$$

kde  $f(x)$  je přenosová funkce neuronu,  $w_{0,k}^n$  je bias neuronu a  $m$  je počet vah neuronu, kterému taktéž odpovídá počet výstupů nižší vrstvy sítě [38].

Vícevrstvá neuronová síť má potenciál řešit nelineárnější problémy než jednovrstvá, učení takové sítě je ale mnohem složitější. V některých případech je učení vícevrstvé sítě úspěšnější, protože je možné, že se jedná o problém, který nedokáže jednovrstvá síť správně vyřešit [6].

K určení počtu vrstev vícevrstvé neuronové sítě a počtu neuronů v těchto vrstvách neexistují žádná pevně daná pravidla, topologii neuronové sítě je nutno určit



Obr. 1.3: Značení neuronů, vah, vstupů a výstupů v neuronové síti

experimentálně. Přesto však existuje několik pravidel a doporučení, která mohou usnadnit volbu její topologie [21].

Pro řešení daného problému obvykle stačí neuronová síť s jednou vnitřní vrstvou. Topologie se dvěma vnitřními vrstvami může být potřebná, když se neuronová síť má naučit funkci mající nespojitosti. Pro použití více jak dvou vnitřních vrstev neexistuje žádný teoretický důvod. Je zde však možnost, že problém bude efektivněji řešitelný pomocí neuronové sítě s více vrstvami o menším počtu neuronů, než pomocí sítě s menším počtem vrstev s neprakticky velkým počtem neuronů [21].

Po zvolení počtu vrstev je třeba zvolit počet neuronů v těchto vrstvách. Volba správného počtu neuronů je velmi důležitá. Při použití malého počtu neuronů nemá neuronová síť kapacitu k naučení daného problému. Naopak při použití příliš velkého počtu neuronů dochází k výraznému prodloužení doby učení a také může nastat problém zvaný přeučení neuronové sítě [21].

K přeučení neuronové sítě dochází, má-li tato síť příliš mnoho prostředků ke zpracování informací. Je to stav, kdy se neuronová síť příliš přesně naučí množinu tréninkových dat (chyba sítě se blíží k 0) a to včetně jejích náhodných chyb nebo šumu a ztrácí schopnost generalizace. Přeučená neuronová síť dosahuje výborných výsledků s tréninkovými daty. Při použití v reálné aplikaci nebo při práci s testovací množinou dat jsou však výsledky velmi špatné [21].

Určit počet neuronů můžeme například tak, že začneme s počtem neuronů, který je příliš malý. Pokud tento počet nelze určit, začneme se dvěma neurony. Dále si určíme způsob výpočtu chyby sítě, což je kritérium posuzující, jak dobře je neuronová síť naučená. Potom postupně zvyšujeme počet neuronů a znovu trénujeme a testujeme síť až do doby, než chyba sítě klesne pod přijatelnou mez, nebo už nedochází

k žádnému zlepšení [21]. Počet neuronů ve vnitřních vrstvách neuronové sítě můžeme určit například také s využitím genetických [4] nebo i jiných algoritmů.

### 1.3 Učení

Učící algoritmus backpropagation je nejpoužívanějším algoritmem v oblasti učení neuronových sítí (přibližně 80% všech aplikací). Algoritmus lze rozdělit do tří hlavních částí, kterými jsou dopředné šíření vstupního signálu, zpětné šíření chyby a adaptace váhových hodnot vstupů neuronů. V praxi se tyto části cyklicky opakují, dokud není dosaženo dostatečně malé chyby sítě, mezního počtu iterací nebo jiného kritéria pro zastavení procesu učení [38].

Celkovou chybu neuronové sítě je možné vypočítat, pokud známe skutečné a požadované hodnoty jejich výstupů. Pro hodnotu této chyby potom platí

$$E = \sum_{i=0}^q \frac{1}{2} \sum_{k=0}^n (y_{i,k} - t_{i,k})^2, \quad (1.5)$$

kde  $q$  je počet vzorů tréninkové množiny dat,  $n$  je počet výstupů neuronové sítě,  $y_{i,k}$  je reálný výstup neuronové sítě a  $t_{i,k}$  je požadovaná hodnota pro daný výstup neuronové sítě uvedená v tréninkovém vzoru [38].

Prvním krokem k adaptaci vah je dopředné šíření signálu přivedeného na vstup neuronové sítě. *"Během dopředného šíření signálu obdrží každý neuron ve vstupní vrstvě vstupní signál a zprostředkuje jeho přenos ke všem neuronům vyšší (vnitřní) vrstvy. Každý neuron ve vnitřní vrstvě opět vypočítá svůj výstup a pošle ho na vstup další vrstvy. Pokud je další vrstva výstupní, tak je její výstup zároveň výstupem neuronové sítě po předložení vstupního vzoru"* ([38], strana 36).

Poté jsou pro každý tréninkový vzor porovnány vypočtené hodnoty výstupů s požadovanými a na základě rozdílu těchto hodnot je vypočten faktor  $\delta_k$  ( $k = 1 \dots m$ ), který reprezentuje část chyby neuronové sítě šířené z daného neuronu ke všem neuronům nižší vrstvy. Úprava váhových hodnot vstupů neuronů závisí na chybovém faktoru  $\delta_k$  a hodnotách výstupů neuronů nižší vrstvy [38].

Chybový faktor  $\delta_k$  pro  $k$ -tý neuron výstupní vrstvy  $n$  neuronové sítě můžeme vypočítat podle vztahu

$$\delta_k^n = (t_k - y_k) \cdot f' \left( w_{0,k}^n + \sum_{i=1}^m y_i^{n-1} \cdot w_{i,k}^n \right), \quad (1.6)$$

kde  $t_k$  je požadovaný výstup neuronové sítě,  $y_k$  je skutečný výstup neuronové sítě,  $f'(x)$  je derivace přenosové funkce neuronu a  $m$  je počet výstupů nižší vrstvy [38].



Chybový faktor  $\delta_k$  pro k-tý neuron n-té vnitřní vrstvy neuronové sítě lze vypočítat podle vztahu

$$\delta_k^n = \sum_{i=1}^q (\delta_i^{n+1} \cdot w_{k,i}^{n+1}) \cdot f' \left( w_{0,k}^n + \sum_{i=1}^m (y_i^{n-1} \cdot w_{i,k}^n) \right), \quad (1.7)$$

kde  $q$  je počet neuronů vyšší vrstvy,  $\delta_i^{n+1}$  je chybový faktor i-tého neuronu ve vyšší vrstvě,  $w_{k,i}^{n+1}$  je váha spoje mezi k-tým neuronem n-té vrstvy a i-tým neuronem vyšší vrstvy,  $f'(x)$  je derivace přenosové funkce neuronu a  $m$  je počet výstupů nižší vrstvy, tj. počet vstupů a vah k-tého neuronu n-té vrstvy [38].

Posledním krokem prováděným v rámci jedné iterace algoritmu je samotná adaptace vah neuronů. Pro hodnotu změny biasu k-tého neuronu v n-té vrstvě platí

$$w_{0,k}^n = \alpha \cdot \delta_k^n, \quad (1.8)$$

kde  $\alpha$  je koeficient učení, který ovlivňuje rychlost učení neuronové sítě a  $\delta_k^n$  je vypočtený chybový faktor neuronu [38].

Hodnotu změny váhy spoje mezi k-tým neuronem n-té vrstvy a i-tým neuronem vrstvy předchozí vypočítáme podle vzorce

$$w_{i,k}^n = \alpha \cdot \delta_k^n \cdot y_i^{n-1}, \quad (1.9)$$

kde  $\alpha$  je koeficient učení neuronové sítě,  $\delta_k^n$  je vypočtený chybový faktor neuronu a  $y_i^{n-1}$  je výstup i-tého neuronu nižší vrstvy [38].

*"Cílem adaptace je minimalizace chyby sítě ve váhovém prostoru. Vzhledem k tomu, že chyba sítě přímo závisí na komplikované nelineární složené funkci vícevrstvé sítě, představuje tento cíl netriviální optimalizační problém. Pro jeho řešení se v základním modelu používá nejjednodušší varianta gradientní metody, která vyžaduje diferencovatelnost chybové funkce"* ([38], strana 37).

Přenosová funkce neuronu v neuronových sítích používajících adaptační algoritmus backpropagation musí splňovat následující požadavky: musí být spojitá, diferencovatelná a monotónně neklesající [38].

K naučení neuronové sítě je kromě patřičného algoritmu učení nutné mít také množinu tréninkových dat. Každý vzor tréninkové množiny popisuje požadovaný výstup neuronové sítě při daném vstupu. Za tréninkovou množinu  $T$  můžeme považovat množinu prvků definovaných dvojicemi vektorů vstupu a výstupu

$$T = \{\{S_1, T_1\}\{S_2, T_2\} \dots \{S_q, T_q\}\}, \quad (1.10)$$

$$S_i = [s_1 \ s_2 \ s_3 \ \dots \ s_m], \quad T_i = [t_1 \ t_2 \ t_3 \ \dots \ t_n], \quad (1.11)$$

kde  $T$  je množina tréninkových dat,  $S_i$  je vektor vstupů neuronové sítě,  $T_i$  je vektor požadovaných výstupů neuronové sítě,  $q$  je počet vzorů tréninkové množiny,  $m$  je počet vstupů neuronové sítě a  $n$  je počet výstupů neuronové sítě [38].

## 1.4 Neuronové sítě a predikce

Vícevrstvé neuronové sítě, ale i neuronové sítě jiného typu můžeme v praxi použít k predikci nejrůznějších dějů, například spotřeby energie bioklimatických budov [22], pravděpodobnosti poruchy vodovodních systémů [15], sesuvů půdy [17], sekundární struktury proteinů [24], výkonu solárních termálních elektráren [40], kvality vody v řekách [3], rychlosti větru v horských regionech Indie [30], či k predikci rychlosti dopravy s využitím dat z mikrovlnných senzorů [20].

V článku [22] je použita neuronová síť typu NARX s algoritmem učení Levenberg-Marquardt ke krátkodobé predikci požadavků na energii u bioklimatické budovy v časovém horizontu o délce jedné hodiny (60 kroků), kde byla dosažena přijatelná přesnost predikce pro reálná data se střední chybou 11,48 %.

Pokud potřebujeme předpovídat sesuvy půdy, můžeme vzít v úvahu článek [17], který navrhuje predikci založenou na kombinaci výstupů několika vícevrstevných neuronových sítí. V první fázi práce bylo vytvořeno několik neuronových sítí diverzifikovaných rozdílnými faktory prostředí a/nebo rozdílnými algoritmy učení. Predikce s využitím několika neuronových sítí byla realizována pomocí algoritmů ELM a LSSVM. Dosažené výsledky naznačují, že použití více neuronových sítí může vést ke zlepšení kvality predikce.

Vícevrstvou neuronovou sítí lze podle [40] použít k predikci výkonu solární termální elektrárny. Pro experiment byly použity algoritmy učení Levenberg-Marquardt a scaled conjugate gradient. Získané neuronové sítě umožnily predikovat teploty rozvrstvení přehřívacího vodního tanku s přesností  $\pm 3\%$  a solární frakci termálního energetického systému s přesností  $\pm 10\%$ .

Pro predikci rychlosti dopravy s využitím dat ze vzdálených mikrovlnných senzorů může být podle [20] použita neuronová síť typu LSTM (Long Short-Term Memory). Neuronová síť typu LSTM překonává z hlediska přesnosti a stability predikce ostatní algoritmy (Elman NN, TDNN, NARX NN, SVM, ARIMA, Kalman filter).

Dalším možným využitím vícevrstevných neuronových sítí spolu s genetickými algoritmy a analýzou hlavních komponent je podle [3] predikce kvality vody v řekách. Analýza hlavních komponent (PCA) byla využita pro redukci dimenze vstupních dat, čímž byla zvýšena efektivita učení a genetické algoritmy k následnému nastavení parametrů neuronové sítě. Tímto postupem bylo dosaženo přesnosti predikce o hodnotě přibližně 91%.

Článek [30] popisuje použití vícevrstevné neuronové sítě s algoritmem učení Levenberg-Marquardt pro předvídání rychlosti větru v 11-ti lokacích Himalájí za účelem nalezení možností výroby větrné energie. Pomocí neuronové sítě bylo dosaženo vysoké přesnosti predikce, hodnoty korelačních koeficientů mezi skutečnými a predikovanými průběhy se pohybují okolo 0,99.

## 2 FOREX

Tato kapitola se skládá ze čtyř částí. První část kapitoly stručně shrnuje historii forexu, druhá část obsahuje základní informace o typech grafů ceny, obchodních hodinách a měnových párech, třetí část je věnována obchodním strategiím a analýze trhu a poslední, čtvrtá část kapitoly, je výsledkem rešerše na téma využití neuronových sítí při obchodování na forexu.

### 2.1 Historie forexu

Začátky obchodování s měnou můžeme datovat do roku 1875, kdy byl vytvořen tzv. zlatý standard, podle kterého vlády jednotlivých zemí zaručovaly konverzní poměr své měny za zlato a naopak. Před jeho vytvořením byly jako hlavní platidlo používány především drahé kovy, zejména zlato a stříbro. Díky tomu, že po vytvoření zlatého standardu byla jednotka měny různých zemí kryta rozdílným množstvím zlata, došlo ke vzniku kurzů pro výměnu jednotlivých měn, díky čemuž mohlo následně dojít také k prvním obchodům s měnou [10].

Zlatý standard přestal být ve velké míře respektován během začátku první světové války. Kvůli politickému tlaku mezi Německem a ostatními evropskými zeměmi a jejich potřebě financovat vojenské projekty bylo vytištěno větší množství peněz, než bylo možné pokrýt zásobami zlata. I přes skutečnost, že zlatý standard přestal být dodržován, zůstává zlato stále velmi důležitou formou peněžní hodnoty [10].

Jako náhrada za zlatý standard byl v roce 1944 vytvořen tzv. Brettonwoodský systém. V tomto systému nahradil americký dolar zlato, jakožto primární rezervní měna. Americký dolar se stal jedinou měnou, která byla podložena zlatem. Kolem roku 1970 však začalo docházet k problémům, protože americké zlaté rezervy byly na tak nízké úrovni, že nebyly schopné pokrýt všechny dolary, které měly zahraniční centrální banky v rezervě. To v roce 1971 společně s odepřením možnosti výměny dolaru za zlato znamenalo konec Brettonwoodského systému [10].

Po pádu Brettonwoodského systému se od roku 1976 začal používat systém plovoucích měnových kurzů, což znamenalo úplné opuštění zlatého standardu [10]. V systému s plovoucími měnovými kurzy je cena měny dána nabídkou a poptávkou, může ale také být regulována prodejem či nákupem svou centrální bankou [8].

V současnosti je forex největším finančním trhem na světě s objemem transakcí převyšujícím hodnotu 4 biliony ( $4 \cdot 10^{12}$ ) dolarů, na kterém se prodávají a nakupují měny. Je otevřen 24 hodin denně, 5 dní v týdnu a díky velkému množství zúčastněných obchodníků je možné téměř okamžitě měnu prodat nebo nakoupit. K tomu abychom mohli začít obchodovat nám stačí jen internetové připojení, počítač a otevřený obchodní účet u námi vybraného brokera [10].

## 2.2 Základy forexu

Před tím, než můžeme začít obchodovat, si musíme vybrat brokera. Volbu brokera provádíme podle našich individuálních požadavků, ať již jde o typ brokera, velikost nabízené finanční páky<sup>1</sup>, minimální vklad pro otevření účtu, nabízené spready<sup>2</sup>, ohlasy a hodnocení od jeho klientů nebo nabízenou obchodní platformu.

Obchodní platforma je software, který umožňuje obchodníkům otevírat, modifikovat či uzavírat obchodní pozice. Součástí obchodní platformy často bývají také nástroje pro analýzu trhu nebo pro vývoj a testování automatických obchodních strategií a vlastních technických indikátorů [35]. Velmi důležitým prostředkem potřebným pro analýzu trhu je bezesporu grafické zobrazení kurzovních dat.

V praxi se nejčastěji můžeme setkat se třemi typy grafů zobrazujících cenu měnového páru v závislosti na čase. Jedná se o grafy čárové, svícové a sloupcové [37].



Obr. 2.1: Čárový graf

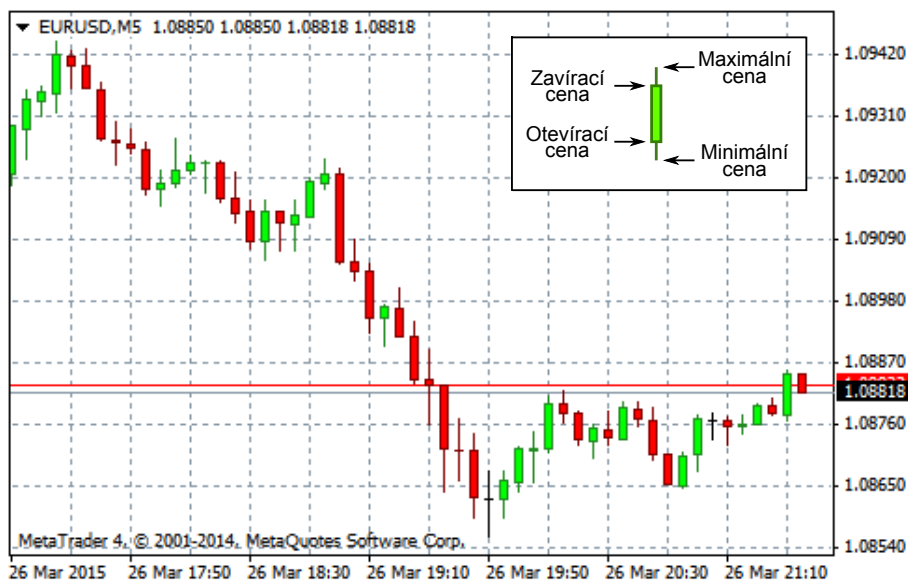
Čárový graf (Obr. 2.1) nejčastěji zobrazuje hodnotu zavírací ceny v daném časovém období. Výhodou čárového grafu je přehledné zobrazení pohybu ceny, nevýhodou je, že nezobrazuje cenovou rozkolísanost trhu [37][19].

Dalším z nejčastěji používaných typů cenového grafu je svícový graf (Obr. 2.2). Samotná svíce má dvě části, tělo a knoty. Tělo svíce zobrazuje interval mezi otevírací a zavírací cenou za daný časový úsek, zatímco spodní a horní knot zobrazují minimální a maximální cenu. Barva svíce udává, zdali se jedná o pohyb rostoucí

<sup>1</sup>Finanční páka nám při malém vlastním kapitálu umožňuje obchodovat s využitím výrazně většího půjčeného kapitálu za účelem zvýšení potenciálních zisků a ztrát [10].

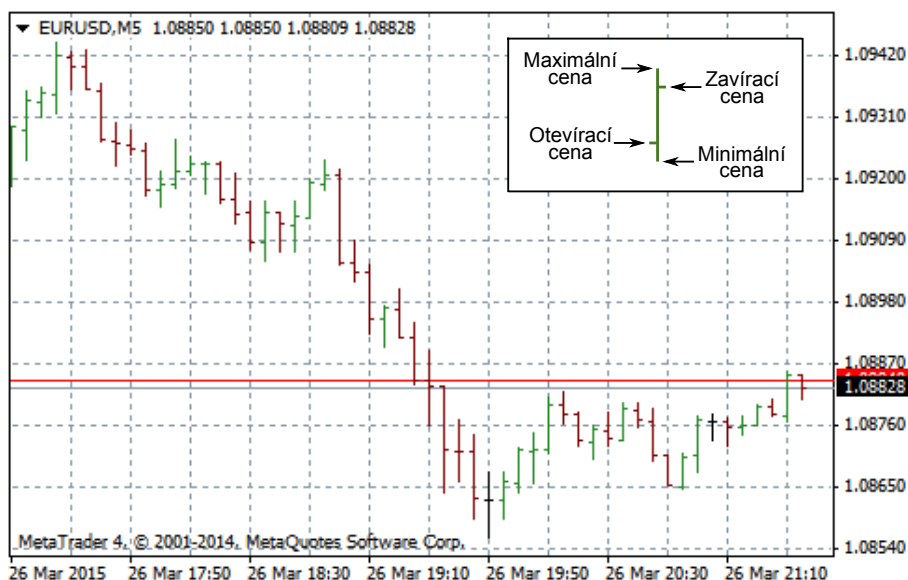
<sup>2</sup>Spread je rozdíl mezi nákupní a prodejní cenou měny [10].

nebo klesající. V případě obrázku 2.2 je svíce vybarvena zelenou barvou pokud se jedná o pohyb rostoucí a červenou barvou pokud jde o pohyb klesající [37][10].



Obr. 2.2: Svícový graf

Sloupcový graf (Obr. 2.3) zobrazuje stejné informace jako graf svícový. Jednotlivé sloupce, podobně jako svíce svícového grafu, zobrazují hodnoty otevírací, zavírací, maximální a minimální ceny za daný časový úsek [37].

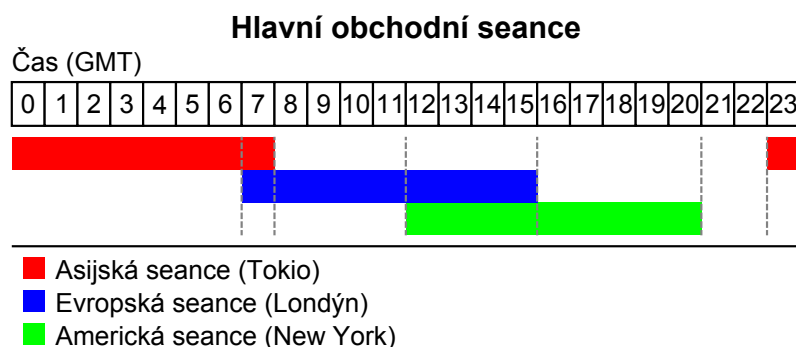


Obr. 2.3: Sloupcový graf

Ať již jde o bod v čárovém grafu, svíci ve svícovém grafu nebo sloupec v grafu sloupcovém, pokaždé se jedná o zobrazení ceny za určitou časovou periodu. Obvykle používané periody jsou M1 (1 minuta), M5 (5 minut), M15 (15 minut), M30 (30 minut), H1 (1 hodina), H4 (4 hodiny), D1 (1 den), W1 (1 týden), MN (1 měsíc). Například pro graf o periodě M15 tedy platí, že jedna jeho svíce zobrazuje informace o průběhu ceny za 15-ti minutové období [37].

I přes to, že forex je trh otevřený 24 hodin denně, je pro efektivní obchodování důležité vědět, v jakých hodinách dochází u kterého měnového páru k největším změnám kurzu, tj. kdy je nejvíce volatilní. To ve velké míře závisí na tom, která obchodní seance právě probíhá [18].

Na obrázku 2.4 jsou znázorněny časy, ve kterých probíhají tři nejvýznamnější obchodní seance. Jedná se o asijskou seanci, probíhající v čase od 23:00 do 8:00 GMT, evropskou seanci probíhající v čase od 7:00 do 16:00 GMT a americkou seanci probíhající v čase od 12:00 do 21:00 GMT [7].



Obr. 2.4: Časy hlavních obchodních seancí

Nejvýznamnější obchodní centra působící během asijské seance jsou Tokio, Hongkong a Singapur, během americké seance potom New York a nakonec, během evropské seance Londýn [18].

Kromě forexových obchodních seancí je vhodné se seznámit také s nejvíce obchodovanými měnovými páry. Jedná se o páry EUR/USD (euro / americký dolar), USD/JPY (americký dolar / japonský yen), GBP/USD (britská libra / americký dolar) a USD/CHF (americký dolar / švýcarský frank) [10].

Nejobchodovanějším měnovým párem je bezesporu EUR/USD, protože Spojené státy a Evropská unie jsou dvě největší ekonomické mocnosti na světě a protože americký dolar a euro jsou dvě nejrozšířenější měny. Výhodou tohoto měnového páru jsou nízké spready a vysoká konstantní likvidita. Kombinace likvidity a volatility dělá EUR/USD ideálním měnovým párem pro začínající obchodníky [10].

Druhým nejčastěji obchodovaným měnovým párem je USD/JPY. Podobně jako EUR/USD nabízí nízké spready, vynikající likviditu a je vhodný jak pro začínající, tak i pro pokročilé obchodníky. Díky faktu, že japonský yen je ve velké míře obchodován během asijské obchodní seance, je měnový pár USD/JPY vhodným měnovým párem pro obchodníky, kteří chtějí obchodovat v nočních hodinách [10].

Dalším z nejobchodovanějších měnových párů je GBP/USD. Tento měnový pár nabízí, podobně jako první dva nejobchodovanější měnové páry, vysokou likviditu a nízké spready [10].

Posledním ze čtyř nejobchodovanějších měnových párů je USD/CHF. I přes to, že tento měnový pár nabízí nižší likviditu než GBP/USD, je stále snadno obchodovatelný. Velké pohyby v ceně švýcarského franku jsou často způsobeny mezinárodní politickou a ekonomickou nestabilitou, kdy investoři začnou nakupovat frank kvůli jeho vysoké stabilitě [10].

## 2.3 Obchodní strategie

Obchodní strategie je soubor pravidel a podmínek pro vstup do obchodní pozice a pro její modifikaci či uzavření. Obchodní strategie také zahrnuje management rizika a zisku a může být zpětně testována na historických datech [36].

Forexová obchodní strategie je v drtivé většině případů založena na výsledku specifické analýzy trhu. Analýza trhu může být fundamentální nebo technická.

Fundamentální analýza trhu je analýza ekonomických, finančních a jiných kvalitativních i kvantitativních údajů. Fundamentální analýza zpracovává všechny informace, které mohou souviset s vývojem ceny, včetně makroekonomických faktorů (celkový stav ekonomiky a průmyslu) a faktorů týkajících se jednotlivých společností, jako je například jejich finanční situace s cílem zjistit pravděpodobný směr, jakým se bude trh v budoucí době ubírat [11].

Technická analýza je opakem k analýze fundamentální. Technická analýza zpracovává historická data (data která byla vytvořena předchozí aktivitou trhu) s cílem najít určité vzory a za účelem předpovědět budoucí tržní pohyby jak v krátkodobém, tak i v dlouhodobém měřítku [10].

Technickou analýzu lze použít nejen na forexu, ale například i při obchodování s akcemi, komoditami, futures atd. Pokaždé je však podle [10] založena na třech základních předpokladech:

- Průběh kurzu měnového páru obsahuje všechny dostupné informace včetně fundamentálních faktorů.
- Cenové pohyby mají sklon následovat trend.
- Historie pohybu ceny má tendenci se opakovat.

Nedílnou součástí technické analýzy jsou technické indikátory. Technický indikátor je takový indikátor, jehož hodnota je vypočtena z kurzovních dat a který neanalyzuje žádné fundamentální faktory. Technické indikátory jsou nejčastěji používány při krátkodobém obchodování, jsou vypočítány z kurzovních dat za určitou časovou periodu, díky čemuž dochází k jejich určitému zpoždění, o kterém obecně platí, že roste se vzrůstající periodou, za kterou je indikátor vypočten [37][34].

Indikátory můžeme rozdělit do dvou hlavních skupin. První skupinou jsou trendové indikátory. Trendové indikátory nám umožňují generovat obchodní signály do směru trendu, v situaci bez trendu jsou však vygenerované obchodní signály obvykle ztrátové. Druhou skupinou indikátorů jsou oscilátory. Oscilátory je na rozdíl od trendových ukazatelů vhodné používat v situaci bez trendu [37].

Typickým zástupcem kategorie trendových indikátorů jsou jednoduché a exponenciální klouzavé průměry. Jednoduchý klouzavý průměr (SMA) je aritmetický průměr vypočtený z daného počtu zavíracích cen svíci, který je s každou nově dokončenou svíci aktualizován.

Exponenciální průměr (EMA) se liší od jednoduchého tím, že jednotlivým hodnotám, ze kterých je počítán, přiřazuje váhy, přičemž největší váhu má nejnovější hodnota a nejnižší váhu má hodnota nejstarší. Příklad jednoduchého a exponenciálního plovoucího průměru o periodě 14 je zobrazen na obrázku 2.5 [37].



Obr. 2.5: Jednoduchý a exponenciální klouzavý průměr o periodě 14

Často používaným indikátorem založeným na plovoucích průměrech je oscilátor MACD (Moving Average Convergence Divergence). Hodnota indikátoru MACD se vypočítá jako rozdíl dvou exponenciálních plovoucích průměrů, nejčastěji o periodách 12 a 26. Takto vypočítané hodnoty se zobrazují ve formě histogramu.



Součástí indikátoru je také signální křivka (jednoduchý plovoucí průměr vypočtený z histogramu nejčastěji o periodě 9), která se používá pro generování obchodních signálů. Ukázkou indikátoru MACD nalezneme na obrázku 2.6 [37].



Obr. 2.6: Indikátor MACD o periodách 12, 26 a 9

Součástí každé obchodní strategie by kromě vybrané metodiky analýzy trhu a vstupních a výstupních podmínek pro obchodní pozice mělo být také řízení risku na obchod. V praxi to znamená, že každou otevřenou obchodní pozici musíme pojistit stop-lossem a že nesmíme v jednom obchodu riskovat příliš velkou část svého celkového dostupného kapitálu [26].

Stop-loss je příkaz obvykle posílaný na trh společně se vstupním příkazem, který nám umožňuje předem určit maximální ztrátu, kterou jsme ochotni při daném obchodu podstoupit. Díky stop-lossu tedy můžeme řídit risk daného obchodu [26].

S řízením risku souvisí také pojem risk-reward-ratio, neboli poměr zisku a ztráty. Poměr slouží pro porovnání očekávaného zisku se ztrátou, kterou je obchodník pro dosažení tohoto zisku ochotný podstoupit. Při volbě poměru zisku a ztráty pro obchodování na forexu je třeba počítat s úspěšností dané obchodní strategie. Obchodník se strategií s úspěšností obchodů 50% si například nemůže dovolit poměr zisku a ztráty menší než 1:1 [26].

Obchodní strategii můžeme provozovat manuálně, to znamená vykonávat analýzu trhu a jednotlivé akce podle pravidel, ze kterých je strategie sestavena nebo ji můžeme naprogramovat. Naprogramováním obchodní strategie získáváme automatický obchodní systém, který zcela autonomně nebo s občasnými zásahy obchodníka vykonává obchodní činnost.

## 2.4 Neuronové sítě a forex

Vícevrstvé neuronové sítě se při obchodování na forexu používají zejména pro účely predikce budoucího vývoje kurzu, obvykle jako součást obchodních systémů, kde podle výsledku jejich predikce rozhoduje obchodní strategie o nákupu či prodeji měny. Samoorganizační mapy nalézají uplatnění při předzpracování dat, kdy nám umožňují vytvoření shluků statisticky podobných dat.

K predikci kurzu je možné s úspěchem použít vícevrstvou neuronovou síť s algoritmem učení backpropagation. V nejjednodušším případě můžeme na její vstupy přivést přímo několik posledních vzorků průběhu kurzu, nevýhodou této metody je skutečnost, že dochází k naučení neuronové sítě na šum a to zejména při zvyšování počtu vstupů neuronové sítě [41].

Další možností je použití technických indikátorů, především plovoucích průměrů, jako vstupů pro neuronovou síť. V případě článku [41] se jedná o plovoucí průměry o periodách 5, 10, 20, 60 a 120, kde perioda 5 znamená, že jde o plovoucí aritmetický průměr za posledních 5 obchodních dní. Klouzavé průměry nám pomohou odfiltrovat značnou část šumu, čímž můžeme získat více užitečných informací [41].

Neuronová síť využívající plovoucí průměry jako vstupní veličiny dosahuje dle [41] lepších výsledků než síť používající přímo hodnoty cen. Je však třeba brát na vědomí, že predikční schopnosti takto vytvořených neuronových sítí se budou postupem času zhoršovat a bude třeba je přeučit s využitím aktualizovaných historických dat [41].

Dalším přístupem, který podle [1] nabízí slibné výsledky je využití predikce pomocí fuzzy časových řad a Kohonenovy samoorganizační mapy. Princip predikce časové řady pomocí fuzzy logiky je následující: nejprve rozdělíme univerzum na  $n$  intervalů, definujeme fuzzy množiny na tomto univerzu a provedeme fuzzyfikaci časových řad, dále nalezneme logické vztahy existující ve fuzzyfikovaných časových řadách a sestavíme soubor pravidel, provedeme samotnou predikci podle nalezených pravidel a nakonec defuzzyfikujeme predikovaný výstup [1].

Kohonenova samoorganizační mapa je v tomto případě použita k rozdělení univerza do  $n$  rozdílně velkých intervalů z důvodu vyšší výpočetní rychlosti než mají obvykle používané genetické algoritmy [1].

Neuronové sítě predikující budoucí tržní pohyby se používají také jako součást obchodních systémů, jak je ukázáno v článku [4]. Za účelem usnadnění procesu generalizace jsou kurzovní data v tomto případě normalizována s využitím logaritmické diference, kterou lze vypočítat podle vztahu

$$r = \ln \left( \frac{P_t}{P_{t-1}} \right), \quad (2.1)$$

kde  $P_t$  je cena v čase  $t$  a  $P_{t-1}$  cena v čase  $t - 1$  [4].

Pro vyhlazení dat získaných transformací je nutné použít některý z technických indikátorů, velmi rozšířené jsou například plovoucí průměry. Jejich nevýhodou je pomalá reakce na změnu směru kurzu, z toho důvodu je v [4] použita upravená verze plovoucího průměru nazvaná Incremental Window Moving Average (IWMA).

Pro účely predikce byla vybrána vícevrstvá neuronová síť s algoritmem učení Levenberg-Marquardt mající 20 neuronů ve vstupní vrstvě a 1 neuron ve výstupní vrstvě. Jako aktivační funkce neuronů této sítě byla zvolena funkce hyperbolický tangens. Určení optimálního počtu neuronů ve dvou skrytých vrstvách této neuronové sítě proběhlo s využitím genetických algoritmů s fitness funkcí založenou na vyhodnocení predikce naučené neuronové sítě o dané topologii [4].

Výsledkem optimalizace je neuronová síť s 9-ti neurony v první a s 8-mi neurony ve druhé vnitřní vrstvě [4]. Pomocí této neuronové sítě využité k predikci bylo při testech dosaženo roční zhodnocení vkladu ve výši cca 27% [4].

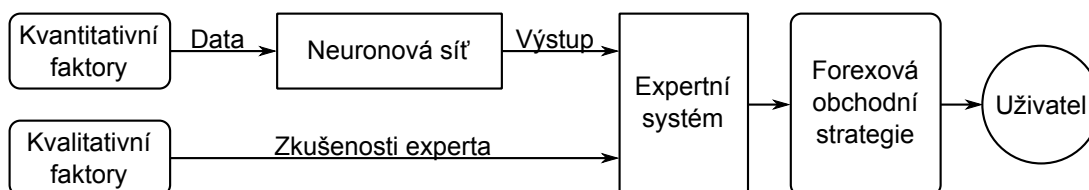
Vícevrstvou neuronovou síť můžeme dle [14] použít také pro účely filtrace obchodních signálů získaných jiným způsobem, než pomocí neuronové sítě. V případě článku [14] jsou obchodní signály generovány pomocí strategie založené na překřížení dvou plovoucích exponenciálních průměrů o periodách 5 a 10 a pro jejich filtraci použita byla vícevrstvá neuronová síť s algoritmem učení Levenberg-Marquardt.

Tato neuronová síť má 29 vstupů, 40 neuronů v první skryté vnitřní vrstvě, 20 neuronů ve druhé vnitřní vrstvě a jeden výstup. Na vstupy sítě je přivedeno 28 technických indikátorů, z nichž 13 jsou plovoucí průměry o periodách 3, 5, 8, 13, 21, 34, 50, 75, 100, 150, 200, 250 a 300, na dalších 8 vstupů jsou přivedeny hodnoty ukazatelů relative strength index (RSI) o periodách 3, 5, 8, 13, 21, 21, 34 a 55 a nakonec, posledních 7 ukazatelů jsou standardní odchylky o periodách 8, 13, 21, 34, 55, 89 a 144. Poslední vstup sítě udává, zdali došlo k sestupnému nebo k vzestupnému překřížení signálních exponenciálních průměrů [14].

Pokud dojde k překřížení signálních průměrů, hodnoty indikátorů jsou vypočteny a přivedeny na vstup neuronové sítě, která posléze určí, zdali by potenciální obchod skončil ztrátou nebo ziskem. Dosažené výsledky prokazují, že obchodní signály filtrované neuronovou sítí jsou podstatně výnosnější než signály náhodně vybrané, a to zejména v případech, kdy původní signály před filtrací jsou méně kvalitní [14].

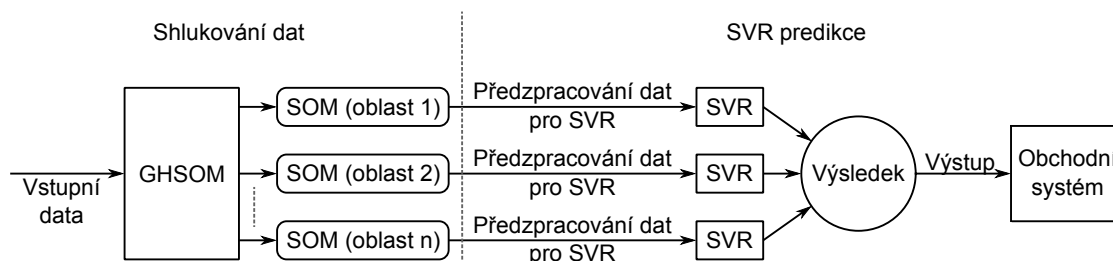
Pokud zkombinujeme vícevrstvou neuronovou síť a expertní systém, získáme hybridní systém, který můžeme použít jako podpůrný nástroj při rozhodování v oblasti obchodování na forexu [16].

Hybridní systém zobrazený na obrázku 2.7 nám umožňuje zpracovávat jak kvantitativní, tak i kvalitativní informace. Neuronová síť slouží ke zpracování kvantitativních informací a její výstup je přiveden na vstup expertního systému. Na druhý vstup expertního systému je přivedena kvalitativní informace získaná například od experta v oblasti analýzy trhu [16].



Obr. 2.7: Hybridní systém s neuronovou sítí a expertním systémem [16]

Pro vytvoření obchodního systému můžeme podle [2] použít rostoucí hierarchickou samoorganizační mapu (Growing Hierarchical Self Organizing Map) v kombinaci s predikční metodou strojového učení support vector regression (SVR). Blokové schéma systému je zobrazeno na obrázku 2.8.



Obr. 2.8: Obchodní systém využívající samoorganizační mapy a SVR [2]

Rostoucí hierarchická samoorganizační mapa použitá v první části systému má na starosti rozdělení dat do statisticky podobných shluků a byla vybrána z důvodu, že není nutné předem specifikovat její strukturu. Ve druhé části dochází pro vybraný region k predikci pomocí metody support vector regression. Výsledek získaný predikcí je dále předán na vstup obchodní strategie, která na jeho základě provádí obchodní rozhodnutí o koupi či prodeji [2].

Pro vytvoření modelu predikujícího tržní pohyby je dle [42] možné využít také principu nazývaného metalearning. Model se v takovém případě skládá z daného počtu dílčích modelů a jednoho metamodelu, který pracuje s jejich výstupy.

Pro realizaci dílčích modelů i metamodelu byla použita vícevrstvá neuronová síť. Výhodami tohoto přístupu jsou zlepšení přesnosti a stability predikce, zamezení ztráty potenciálně cenné informace výběrem pouze jedné nejlépe predikující neuronové sítě či široké možnosti paralelizace úlohy [42].

Pro vytvoření modelu jsou nejprve dostupná data rozdělena na tréninková, testovací a validační. Tréninková data jsou dále rozdělena na tréninkové podmnožiny pro jednotlivé dílčí modely. V dalším kroku potom probíhá učení dílčích modelů,

které jsou navzájem diverzifikovány rozdílným počátečním nastavením vah neuronů a rozdílnými tréninkovými daty [42].

V další fázi tvorby modelu probíhá selekce dílčích modelů pomocí metody analýzy hlavních komponent (PCA). Výstupy vybraných dílčích modelů můžeme dále použít pro vytvoření meta-tréninkové datové množiny, která je posléze použita pro naučení neuronové sítě sloužící jako metamodel [42].

## 3 OBCHODNÍ SYSTÉM AUTOTRADER

Tato kapitola popisuje princip činnosti serverové a klientské části navrženého a realizovaného obchodního systému AutoTrader a na příkladu demonstruje postup tvorby obchodních strategií pro tento systém. Konec kapitoly je věnován programu obchodního deníku a databázi pro logování uskutečněných obchodů.

### 3.1 Použité technologie<sup>1</sup>

Obchodní systém je napsán v jazyce C++14 s použitím Qt5 frameworku a vývojového prostředí QtCreator. Qt5 je multiplatformní framework pro tvorbu aplikací s grafickým uživatelským rozhraním. Qt5 framework je napsán v jazyce C++, můžeme ho ale použít i v jiných jazycích, jako jsou například Java, Python, Ruby a další. Mezi výhody tohoto frameworku patří přenositelnost mezi většinou používaných platform (Windows, Linux, Mac, Android, iOS a další), velmi dobře zpracovaná dokumentace, vysoká produktivita práce, skvělé nástroje pro vývoj (QtCreator), otevřenost zdrojových kódů nebo i jeho oblíbenost u programátorů.

Qt5 framework či jeho předchozí verze Qt4 byly použity při vývoji mnoha známých projektů, jako jsou například multimediální přehrávač VLC, linuxové desktopové prostředí KDE 4, komunikační program Skype, 3D kreslicí program Autodesk Maya, Google Earth, program na kreslení plošných spojů Eagle, virtualizační program VirtualBox a mnoho dalších.

### 3.2 Princip činnosti

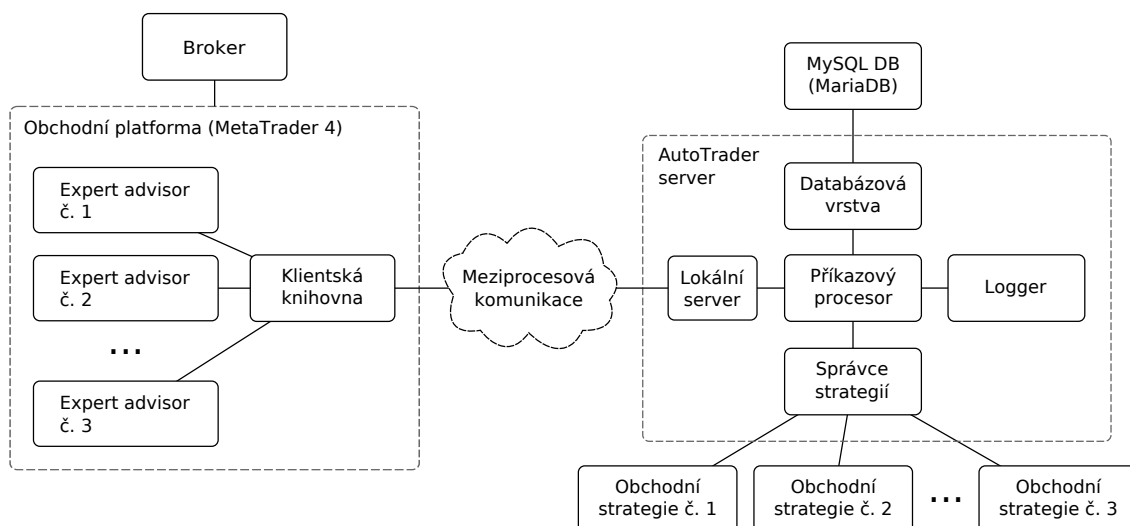
Pro realizaci obchodního systému byl zvolen koncept klient-server. Výhodou tohoto konceptu je možnost vytvoření jednotného rozhraní pro tvorbu obchodních strategií v moderním programovacím jazyce C++14 nezávislého na obchodní platformě, možnost automatického logování uskutečněných obchodů nezávisle na obchodní strategii, snadná portovatelnost klientské části systému na další obchodní platformy, jednodušší údržba systému a jeho snadné použití.

Obchodní systém AutoTrader (Obr. 3.1) můžeme rozdělit na tři hlavní části. První částí systému je klientská knihovna. Jedná se o dynamicky linkovanou knihovnu obsahující funkce pro připojení k serveru a pro přijímání a odesílání zpráv na tento server. Knihovna je načtena obchodní platformou a její funkce jsou volány jednotlivými expert advisory<sup>2</sup> běžícími na této platformě.

---

<sup>1</sup>Převzato a upraveno z [13].

<sup>2</sup>Expert advisor je automatická obchodní strategie implementovaná v obchodní platformě MetaTrader 4, která provádí analýzu trhu, na jejímž základě následně obchoduje.



Obr. 3.1: Principiální schéma obchodního systému AutoTrader

Druhou částí systému je samotný AutoTrader server, jedná se o aplikaci s grafickým uživatelským rozhraním, která přijímá příchozí připojení, registruje nově připojené klienty a provádí obchodní činnost na základě dané obchodní strategie.

Server dále obsahuje správu obchodních strategií a systém pro logování obchodních událostí (otevření, změna a uzavření pozice) do SQL databáze. Server je podle nastavení možné spustit a minimalizovat do system tray, v nastavení je také možné povolit zapisování zobrazovaných informací o příchozích připojeních, registracích klientů a dalších událostí do souboru.

Třetí, neméně důležitou částí obchodního systému jsou samotné moduly obchodních strategií. Moduly obchodních strategií jsou realizovány formou dynamicky linkovaných knihoven implementujících příslušné rozhraní. Tyto knihovny jsou za běhu načítány správcem obchodních strategií. Při registraci klienta dojde k vytvoření instance obchodní strategie obsažené v knihovně a jejímu uložení v instanci klienta, který ji následně využívá při své další činnosti.

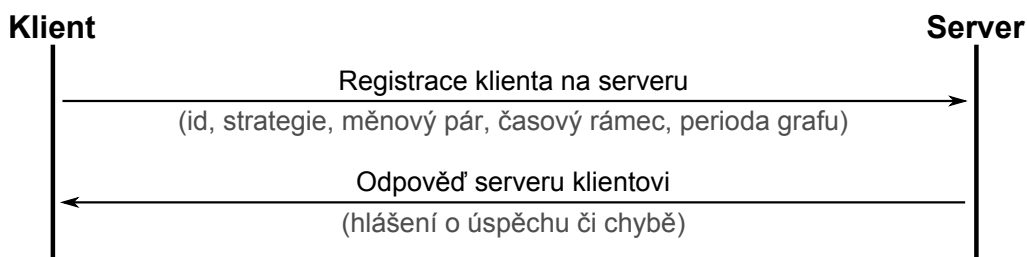
### 3.3 Komunikace mezi klientem a serverem

Z výkonnostních i bezpečnostních důvodů běží obchodní platforma i AutoTrader server na stejném počítači. V případě komunikace mezi klientem (expert advisorem) a AutoTrader serverem tedy budeme hovořit o meziprocesové komunikaci.

Meziprocesová komunikace je řešena pomocí předávání zpráv ve formátu xml. Výhodou xml zpráv je transparentnost komunikace a možnost jejího logování za účelem ladění. Odpadá také synchronizace, která by byla nutná v případě použití

sdílené paměti. Pro realizaci částí klientské knihovny a serveru zabývajících se mezi-procesovou komunikací byly využity třídy `QLocalSocket` a `QLocalServer` z modulu `QNetwork`, který je součástí Qt5 frameworku.

Třída `QLocalServer` umožňuje přijímat příchozí připojení lokálních socketů [28]. Nově navázaná připojení jsou v programu ukládána a po dobu co jsou aktivní od nich mohou být přijímány zprávy. Třída `QLocalSocket` poskytuje funkčnost pro navázání připojení a pro posílání a přijímání dat (zpráv) [29].



Obr. 3.2: Průběh registrace klienta na serveru

Komunikace začíná připojením klienta k serveru. V případě, že se mu úspěšně podaří vytvořit připojení, odešle klient požadavek na registraci, server zprávu přijme, zjistí zdali je možné klienta zaregistrovat a odešle mu hlášení o úspěchu či chybě (Obr. 3.2). Klienta je možné registrovat, pokud jeho id ještě není zaregistrováno a pokud je požadovaná obchodní strategie dostupná.

Průběh komunikace zaregistrovaného klienta se serverem je zobrazen na obrázku 3.3. Klient nejprve odešle požadavek na aktualizaci při změně kurzu a server na to reaguje odpovědí, ve které klienta žádá o data potřebná k aktualizaci.

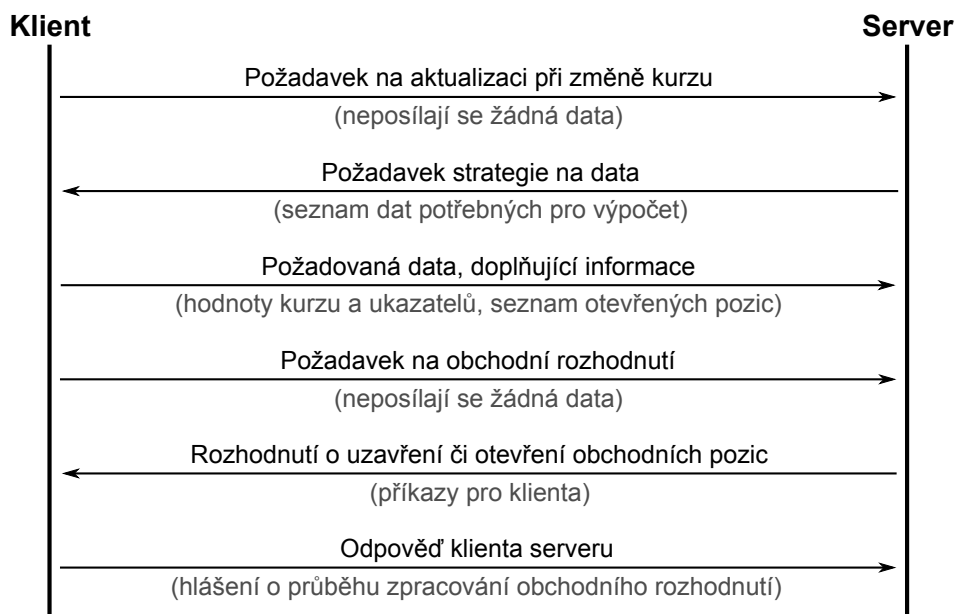
Klient tuto zprávu přijme a odešle serveru požadovaná data, server klientovi na základě zaregistrované strategie po příslušném požadavku odpoví zprávou obsahující obchodní rozhodnutí. Klient zpracovává přijaté obchodní rozhodnutí a informuje server o výsledku jeho zpracování, čímž je dokončena komunikace iniciovaná změnou kurzu.

Každá zpráva obsahuje id klienta, které je uloženo během registrace a podle kterého je potom možné přiřadit příchozí zprávu s instancí zaregistrovaného klienta na serveru nebo rozpoznat, že klient není registrován. Každá zpráva dále obsahuje element akce, jejíž typ určuje účel a obsah zprávy. Typy akcí jsou následující:

- **StatusReport** - hlášení o stavu po vykonání dané akce.
- **RegistrationRequest** - žádost klienta o registraci na serveru.
- **UpdateRequest** - požadavek na seznam dat potřebných pro aktualizaci.
- **UpdateDataRequest** - seznam dat potřebných pro aktualizaci.
- **UpdateDataResponse** - odpověď s daty pro aktualizaci.



- **DecisionRequest** - požadavek na aktualizované obchodní rozhodnutí.
- **DecisionResponse** - odpověď s aktualizovaným obchodním rozhodnutím.
- **DecisionStatus** - zpráva o výsledku vykonání obchodního rozhodnutí.



Obr. 3.3: Komunikace mezi klientem a serverem

Požadavek na registraci (akce typu `RegistrationRequest`), jehož příklad je uveden ve výpisu 3.1 obsahuje informace o požadované strategii, měnovém páru, periodě grafu a počtu desetinných míst ceny daného měnového páru.

```

1 | <?xml version="1.0"?>
2 | <message>
3 |   <client id="1234"/>
4 |   <action type="RegistrationRequest">
5 |     <strategy>MovingAverages</strategy>
6 |     <currency>EURUSD</currency>
7 |     <timeframe>M5</timeframe>
8 |     <digits>5</digits>
9 |   </action>
10| </message>

```

Výpis 3.1: Požadavek na registraci

Na požadavek o registraci server reaguje hlášením o stavu (akce typu `StatusReport`), jehož příklad je uveden ve výpisu 3.2. Element `status` nabývá hodnoty `OK` v případě, že k žádné chybě nedošlo, nebo `ERROR` v případě chyby. Řetězec popisující chybu můžeme najít v elementu `error`. Element `action` nám říká, ke kterému typu akce přijaté hlášení o stavu náleží.

```

1 <?xml version="1.0"?>
2 <message>
3   <client id="1234"/>
4   <action type="StatusReport">
5     <status>OK</status>
6     <error></error>
7     <action>RegistrationResponse</action>
8   </action>
9 </message>

```

Výpis 3.2: Hlášení o stavu

Požadavek na aktualizaci (akce typu `UpdateRequest`), nenese žádná data, pouze říká serveru o seznam požadovaných dat nutných k aktualizaci, tj. k výpočtům a učinění obchodního rozhodnutí. Příklad požadavku na aktualizaci je uveden ve výpisu 3.3.

```

1 <?xml version="1.0"?>
2 <message>
3   <client id="1234"/>
4   <action type="UpdateRequest"/>
5 </message>

```

Výpis 3.3: Požadavek na aktualizaci

Požadavek na data (akce typu `UpdateDataRequest`, výpis 3.4) obsahuje seznam elementů s názvem *data*. Každý z těchto elementů má atribut *name*, který určuje druh požadovaných dat a element *count*, který určuje množství dat od současnosti směrem k minulosti. Hodnota 5 elementu *count* tedy znamená, že server obdrží data za aktuální nedokončenou svíci a za poslední 4 dokončené svíce.

```

1 <?xml version="1.0"?>
2 <message>
3   <client id="1234"/>
4   <action type="UpdateDataRequest">
5     <data name="PRICE" count="10" />
6     <data name="MACD" count="3" />
7     <data name="EMA10" count="1" />
8   </action>
9 </message>

```

Výpis 3.4: Požadavek na data pro aktualizaci

Odpověď s daty pro aktualizaci (akce typu `UpdateDataResponse`, Výpis 3.5) obsahuje data vyžádaná serverem v požadavku na data. Data mohou být typu *ORDER*, *PRICE*, *VALUE* a *VECTOR*. Data typu *ORDER* obsahují informace o všech momentálně otevřených obchodních pozicích, data typu *PRICE* obsahují požadované svíce a data typu *VALUE* nebo *VECTOR* obsahují požadovaný počet hodnot zvoleného technického indikátoru, pokud je v klientské části systému podporován.

```

1  <?xml version="1.0"?>
2  <message>
3    <client id="1234"/>
4    <action type="UpdateDataResponse">
5      <data type="ORDER">
6        <order>
7          <ticket>123456</ticket>
8          <type>SELL</type>
9          <openPrice>1.45467</openPrice>
10         <lots>0.02</lots>
11         <slippage>3</slippage>
12         <stopLoss>1550</stopLoss>
13         <takeProfit>110</takeProfit>
14         <profit>5.20</profit>
15         <error></error>
16       </order>
17     </data>
18     <data type="PRICE">
19       < candle datetime="07.05.2012_00:05">
20         <open>1.4904</open>
21         <high>1.4911</high>
22         <low>1.4904</low>
23         <close>1.4908</close>
24         <volume>39</volume>
25       </ candle>
26     </data>
27     <data type="VALUE" name="EMA10">
28       <value datetime="07.05.2012_00:05">0.000172</value>
29     </data>
30   </action>
31 </message>

```

Výpis 3.5: Odpověď s daty pro aktualizaci

Požadavek na obchodní rozhodnutí (akce typu `DecisionRequest`), nenesení žádná data, pouze říká serveru o obchodní rozhodnutí založená na aktualizovaných datech. Příklad požadavku na obchodní rozhodnutí je uveden ve výpisu 3.6.

```

1  <?xml version="1.0"?>
2  <message>
3    <client id="1234"/>
4    <action type="DecisionRequest"/>
5  </message>

```

Výpis 3.6: Požadavek na obchodní rozhodnutí

Obchodním rozhodnutím (akce typu `DecisionResponse`) je libovolná kombinace následujících čtyř příkazů, tj. hodnot, kterých může nabývat element *type*:

- **BUY** (nákup) - otevření nové obchodní pozice na stranu long.
- **SELL** (prodej) - otevření nové obchodní pozice na stranu short.
- **MODIFY** - modifikace existující obchodní pozice.
- **CLOSE** - uzavření části nebo celé otevřené obchodní pozice.

Údaje důležité pro otevření nové pozice, tj. pro nákup nebo prodej jsou hodnoty take-profit a stop-loss v pipech<sup>3</sup>, velikost pozice v lotech<sup>4</sup> a hodnota skluzu<sup>5</sup>. Příklad příkazu pro prodej je uveden ve výpisu 3.7. Při modifikaci obchodní pozice jsou pro nás důležité údaje stop-loss a take-profit v pipech a ticket, který identifikuje otevřenou pozici. Pokud chceme část nebo celou otevřenou obchodní pozici uzavřít, musíme v lotech uvést, jak velkou část pozice s daným ticketem a při jakém maximálním skluzu si přejeme uzavřít.

```

1  <?xml version="1.0"?>
2  <message>
3    <client id="1"/>
4    <action type="DecisionResponse">
5      <order command="SEND">
6        <time></time>
7        <openTime></openTime>
8        <magic>-1</magic>
9        <ticket>-1</ticket>
10       <type>SELL</type>
11       <openPrice>0</openPrice>
12       <closePrice>0</closePrice>
13       <lots>1</lots>
14       <slippage>3</slippage>
15       <stopLoss>0</stopLoss>
16       <takeProfit>0</takeProfit>
17       <stopLossPt>250</stopLossPt>
18       <takeProfitPt>400</takeProfitPt>
19       <profit>0</profit>
20       <error></error>
21     </order>
22   </action>
23 </message>

```

Výpis 3.7: Odpověď s obchodním rozhodnutím

Poslední zprávou je zpráva obsahující hlášení o úspěchu či chybách při zpracování obchodního rozhodnutí (akce typu DecisionStatus). Tato zpráva obsahuje seznam přijatých obchodních příkazů (Výpis 3.7), přičemž v případě chyby je vyplněn element *error*, který obsahuje chybovou hlášku.

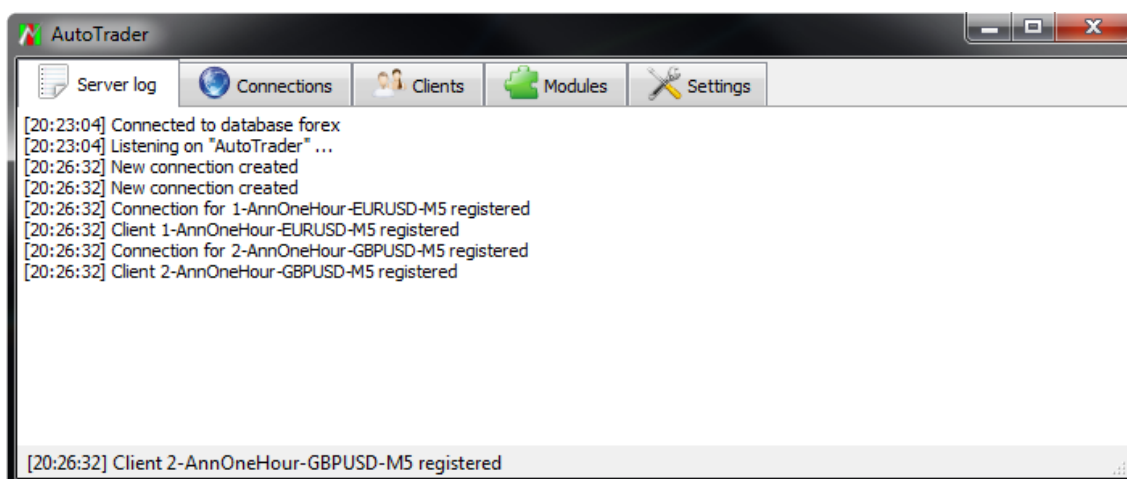
<sup>3</sup>Pip je jednotka o velikosti nejmenší možné změny kurzu měnového páru [27].

<sup>4</sup>Lot (standardní) je ekvivalent 100 000 jednotek nakupované nebo prodávané měny [33].

<sup>5</sup>Skluz je rozdíl mezi skutečnou a očekávanou cenou při které je obchod proveden [32].

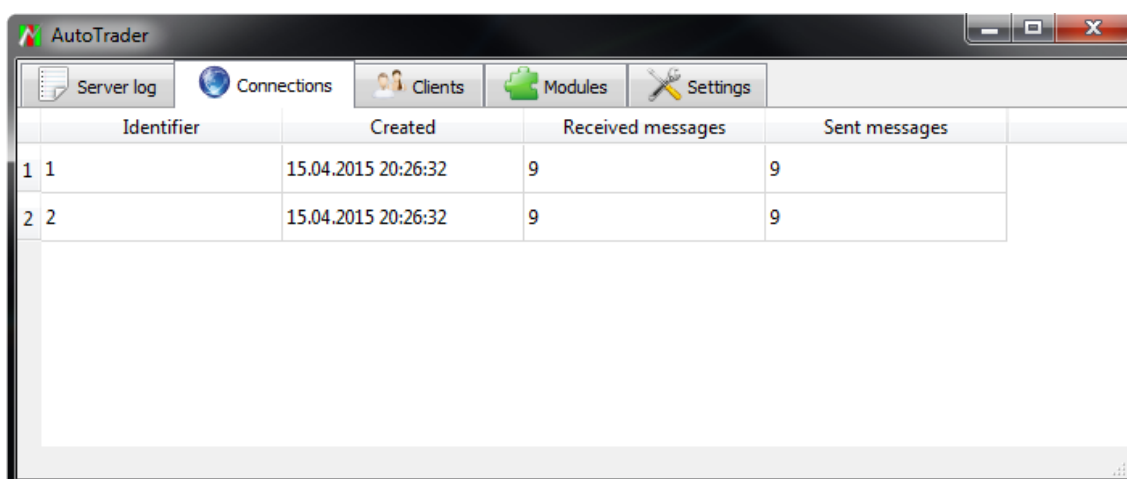
### 3.4 Serverová část systému

Serverovou část systému tvoří program AutoTrader, jehož účel a hlavní funkce jsou popsány v kapitole 3.2. Grafické rozhraní programu je uspořádáno do záložek. Pod první záložkou (Obr. 3.4) nalezneme výpis logu od posledního spuštění programu. Ve výpisu logu můžeme najít například informace o tom, zdali se serveru podařilo připojit k databázi pro logování obchodů nebo o aktivitě klientů.



Obr. 3.4: AutoTrader server - výpis logu

Pod druhou záložkou grafického rozhraní programu AutoTrader (Obr. 3.5) se nachází seznam aktuálně existujících připojení jak registrovaných, tak i dosud nezaregistrovaných klientů.



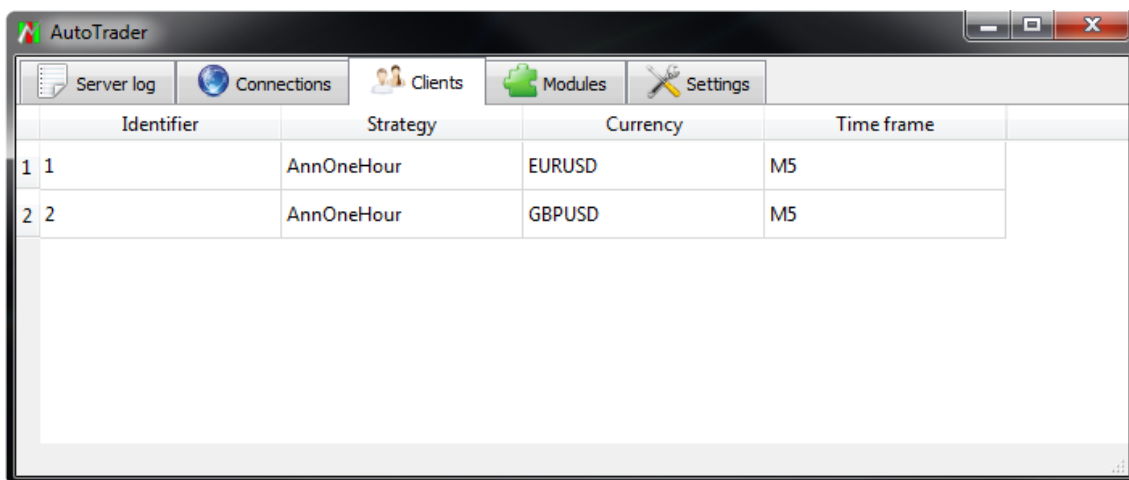
The screenshot shows the 'Connections' tab in the AutoTrader application. It displays a table with the following data:

	Identifier	Created	Received messages	Sent messages
1	1	15.04.2015 20:26:32	9	9
2	2	15.04.2015 20:26:32	9	9

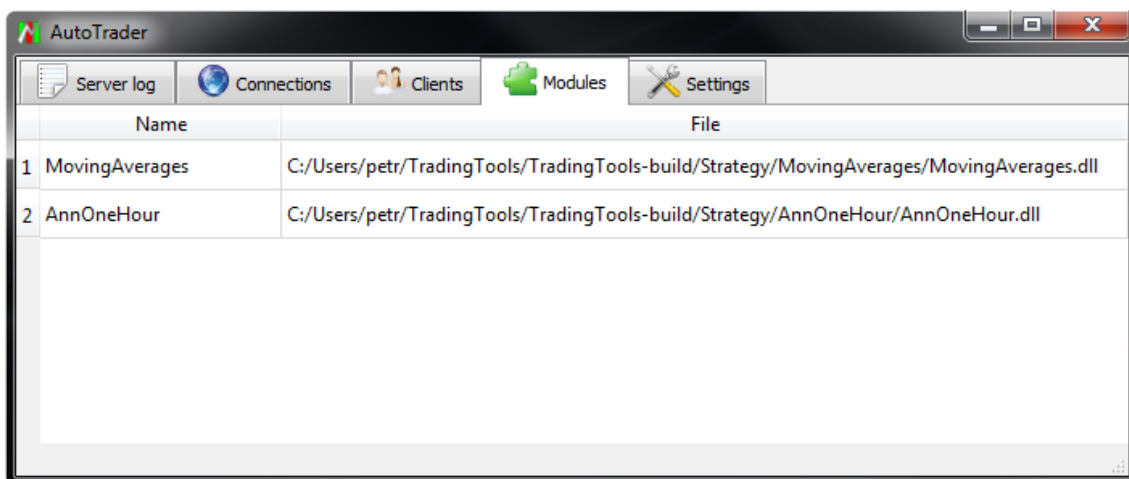
Obr. 3.5: AutoTrader server - tabulka připojení

Tabulka zobrazuje id klienta (-1 pro nezaregistrované připojení), datum a čas, kdy bylo spojení navázáno a počet přijatých a odeslaných zpráv pro dané připojení. Seznam připojení také nabízí možnost sledování probíhající komunikace, která je zobrazena v dialogovém okně po dvojkliku na vybrané připojení.

Třetí záložka (Obr. 3.6) ukrývá seznam zaregistrovaných klientů. Řádek této tabulky zobrazuje id, název strategie, měnový pár a časový rámec zaregistrovaného klienta. Dvojklikem na vybraného klienta můžeme navíc zobrazit okno s dodatečnými informacemi a volbami pro jeho strategii, pouze však pokud je tato funkce ve strategii podporována.



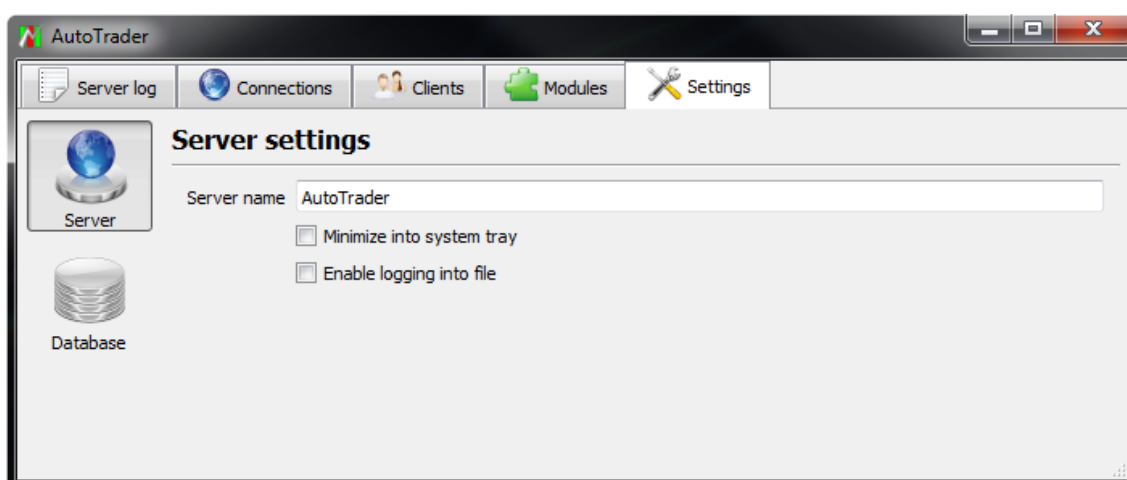
Obr. 3.6: AutoTrader server - seznam klientů



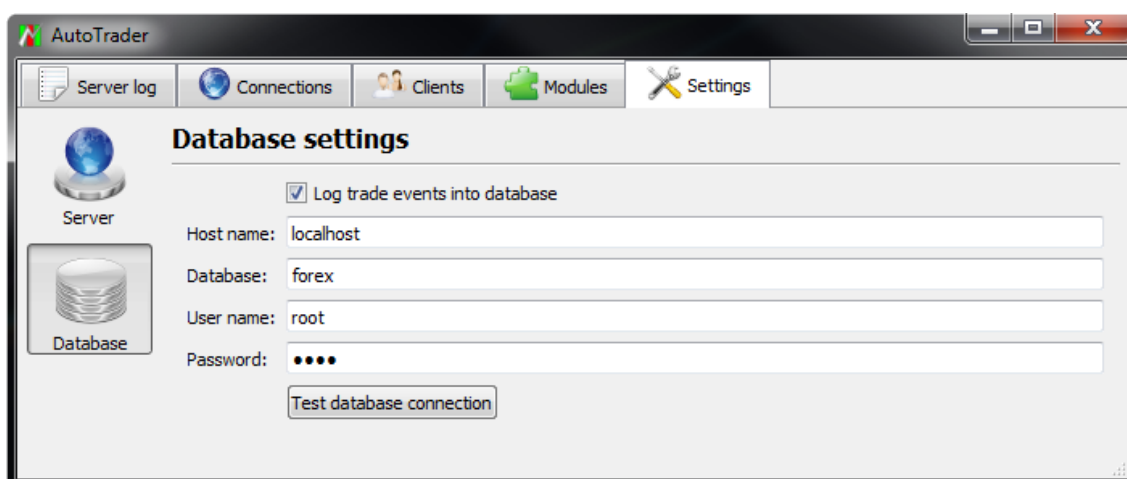
Obr. 3.7: AutoTrader server - správa strategií

Čtvrtá záložka (Obr. 3.7) obsahuje tabulku dostupných obchodních strategií. Řádek tabulky zobrazuje jméno strategie a cestu k souboru knihovny této strategie. Strategie je možné přidávat a odebírat pomocí kontextového menu vyvolaného kliknutím pravého tlačítka myši.

Pod poslední, pátou záložkou, se nachází nastavení programu, které obsahuje 2 podkategorie. První podkategorie (Obr. 3.8) je věnována nastavení serveru a umožňuje zvolit jméno serveru, zdali bude program křížkem ukončen nebo minimalizován do system tray a zdali se log vypisovaný v první záložce bude zapisovat také do souboru. Druhá podkategorie nastavení (Obr. 3.9) slouží k nastavení přístupu do databáze pro logování uskutečněných obchodů.



Obr. 3.8: AutoTrader server - nastavení serveru

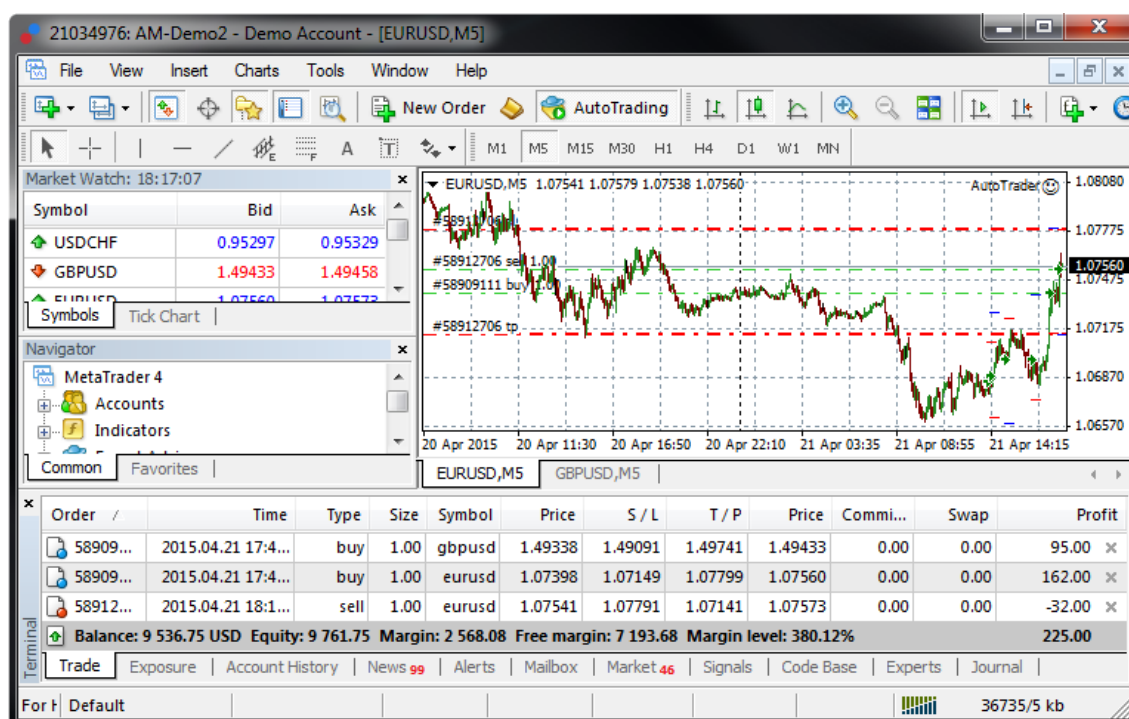


Obr. 3.9: AutoTrader server - nastavení databáze

## 3.5 Klientská část systému

Klientská část systému se skládá ze dvou hlavních částí, první částí je klientská knihovna obsahující funkce potřebné pro komunikaci s AutoTrader serverem a druhou částí je expert advisor naprogramovaný přímo v obchodní platformě MetaTrader 4, který s klientskou knihovnou pracuje.

Platforma MetaTrader 4 (Obr. 3.10) je obchodní platforma poskytovaná brokerem vybraným pro testování obchodních strategií na demo účtu v rámci této diplomové práce (více o výběru brokera v kapitole 4.6). MetaTrader 4 nabízí svým uživatelům nejen nástroje pro manuální obchodování a velké množství technických indikátorů, ale i možnost naprogramování vlastních technických indikátorů, skriptů a automatických obchodních strategií v jazyce Mql4.



Obr. 3.10: Obchodní platforma MetaTrader 4

Funkce obsažené v klientské knihovně můžeme rozdělit do pěti skupin. První skupinou (Výpis 3.8) jsou funkce pro připojení k serveru a registraci klienta, druhá skupina (Výpis 3.9) obsahuje funkce pro odeslání požadavku na aktualizaci a přijetí odpovědi obsahující seznam požadovaných dat, třetí skupina funkcí (Výpis 3.10) má na starosti vytvoření a odeslání požadovaných dat a čtvrtá skupina funkcí (Výpis 3.11) obstarává příjem a zpracování obchodních rozhodnutí. Poslední, pátá skupina (Výpis 3.12) obsahuje pomocné funkce.

Pokud chceme, aby expert advisor pracoval s klientskou knihovnou, musíme v této knihovně nejprve vytvořit jeho instanci. Toho docílíme voláním funkce *initia-*



*lize*. Knihovna může obsahovat instance libovolného počtu expert advisorů, přičemž instance každého z nich je přístupná pomocí unikátního identifikátoru. Knihovna také počítá s tím, že každý expert advisor poběží ve svém vlastním vlákne.

Pro připojení expert advisora k serveru daného jména voláme funkci *connect*. Pokud se připojení podaří vytvořit, můžeme expert advisora na serveru registrovat voláním funkce *registerClient*.

```
1 void LIB_EXPORT initialize(int id);
2 void LIB_EXPORT deinitialize(int id);
3 void LIB_EXPORT connect(int id, const char* server);
4
5 void LIB_EXPORT registerClient(
6     int id, const char* strategy, const char* currency,
7     const char* timeframe, int digits);
```

Výpis 3.8: Funkce klientské knihovny pro připojení a registraci na serveru

Po úspěšné registraci klienta můžeme začít obchodovat. Prvním krokem je zjistit, jaká data potřebuje server pro učinění obchodního rozhodnutí. To zjistíme pomocí funkce *requestUpdate*, která o tuto informaci požádá server a uloží si jeho odpověď.

Počet položek odpovědi zjistíme voláním funkce *requestedNameCount*, o jaký typ dat se jedná nám řekne funkce *requestedDataName* a o požadovaném počtu vzorků nám podá informaci funkce *requestedDataCount*.

```
1 void LIB_EXPORT requestUpdate(int id);
2 int LIB_EXPORT requestedNameCount(int id);
3 void LIB_EXPORT requestedDataName(int id, int index, char* symbol);
4 int LIB_EXPORT requestedDataCount(int id, int index);
```

Výpis 3.9: Funkce klientské knihovny pro žádost o aktualizaci

Nyní již máme všechny potřebné informace pro vytvoření zprávy obsahující serverem požadovaná data. Data svíce grafu přidáme do odpovědi pomocí funkce *appendCandle*, hodnotu technického indikátoru můžeme do datové zprávy přidat pomocí funkce *appendValue* a vektor hodnot, například v situaci, kdy má technický indikátor větší počet výstupů, připojíme do zprávy voláním funkce *appendVector*.

Každá zpráva s daty také musí obsahovat seznam aktuálně otevřených obchodních pozic, ten vytvoříme pomocí funkce *appendOrder*. Nakonec zprávu odešleme na server voláním funkce *dataResponse*.

```
1 void LIB_EXPORT appendOrder(
2     int id, const char* event, int magic, int ticket,
3     const char* type, double openPrice, double closePrice,
4     double lots, double stopLoss, double takeProfit,
5     double profit, const char* time, const char* openTime);
6
7 void LIB_EXPORT appendCandle(
```

```

8 |     int id, const char* dateTime, double open, double high,
9 |     double low, double close, long volume);
10 |
11 | void LIB_EXPORT appendValue(
12 |     int id, const char* name, const char* dateTime, double value);
13 |
14 | void LIB_EXPORT appendVector(
15 |     int id, const char* name, const char* dateTime,
16 |     double* array, int size);
17 |
18 | void LIB_EXPORT dataResponse(int id);

```

Výpis 3.10: Funkce klientské knihovny pro tvorbu datové odpovědi

Po odeslání zprávy s aktuálními daty můžeme server požádat o nové obchodní rozhodnutí. To provedeme voláním funkce *requestDecision*. Počet příkazů, obsažených v získaném obchodním rozhodnutí můžeme zjistit díky funkci *orderCount*. Parametry potřebné pro vykonání daného příkazu dostaneme voláním příslušných funkcí uvedených ve výpisu 3.11.

Při vykonávání příkazů může docházet k chybám (například při nedostatku peněz na účtu). Z toho důvodu je třeba odeslat na server zprávu o výsledku zpracování přijatých příkazů. Tato zpráva obsahuje přijaté obchodní příkazy, ke kterým můžeme pomocí funkce *setOrderError* přiřadit řetězec s chybovou hláškou. Zprávu na server odešleme voláním funkce *decisionStatus*.

```

1 | void LIB_EXPORT requestDecision(int id);
2 | int LIB_EXPORT orderCount(int id);
3 | void LIB_EXPORT orderCommand(int id, int index, char* command);
4 | int LIB_EXPORT orderTicket(int id, int index);
5 | void LIB_EXPORT orderType(int id, int index, char* type);
6 | double LIB_EXPORT orderLots(int id, int index);
7 | int LIB_EXPORT orderSlippage(int id, int index);
8 | int LIB_EXPORT orderStopLoss(int id, int index);
9 | int LIB_EXPORT orderTakeProfit(int id, int index);
10 | void LIB_EXPORT setOrderError(int id, int index, const char* error);
11 | void LIB_EXPORT decisionStatus(int id);

```

Výpis 3.11: Funkce klientské knihovny pro zpracování obchodního rozhodnutí

Pomocné funkce *hasError* a *lastError* (Výpis 3.12) slouží ke kontrole, zdali při volání ostatních funkcí klientské knihovny nedošlo k chybě. Pokud funkce *hasError* vrací hodnotu true, můžeme pomocí funkce *lastError* získat popis poslední chyby, která pro daného experta nastala. Zbývající pomocnou funkcí je funkce *magicNumber*, kterou můžeme použít pro získání unikátního čísla napříč všemi experty.

```

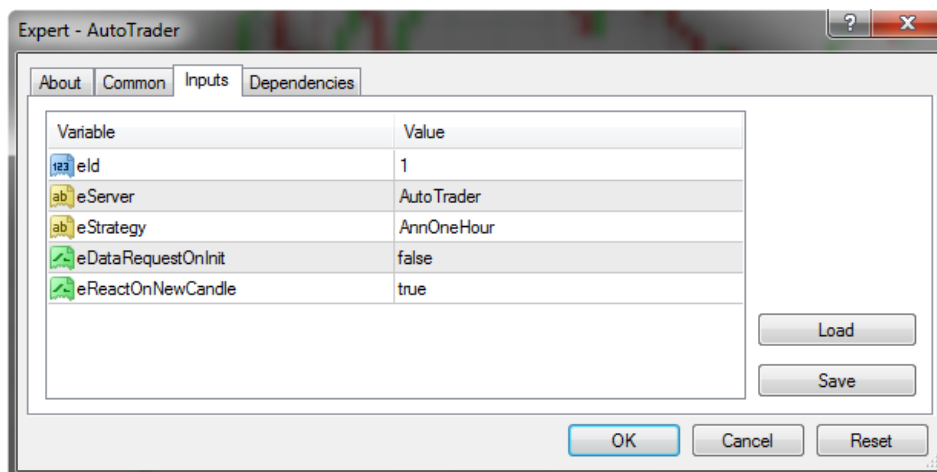
1 | int LIB_EXPORT magicNumber();
2 | void LIB_EXPORT lastError(int id, char* error);

```

```
3 || bool LIB_EXPORT hasError(int id);
```

Výpis 3.12: Pomocné funkce klientské knihovny

Druhou částí klientské části systému je již zmíněný expert advisor pojmenovaný AutoTrader shodně se serverovou aplikací. Ten s využitím klientské knihovny propojuje AutoTrader server s obchodní platformou MetaTrader 4. Experta můžeme spustit například jeho přetažením ze seznamu expertů do okna grafu. Během spuštění experta je třeba v dialogovém okně (Obr. 3.11) nastavit jeho parametry.



Obr. 3.11: Nastavení parametrů expert advisora AutoTrader

Prvním parametrem je identifikátor experta, pod kterým bude vytvořena jeho instance v klientské knihovně a pod kterým bude registrován na serveru. Dalšími parametry jsou název serveru ke kterému se má expert připojit a název strategie, kterou bude expert vyžadovat při registraci.

Parametr *eRequestDataOnInit* udává, zdali si má expert vyžádat seznam požadovaných dat od serveru pouze při svém spuštění (hodnota true), což je výhodné v případě, že strategie vyžaduje stále stejná data nebo při každé aktualizaci (hodnota false). Posledním parametrem můžeme nastavit, že expert bude reagovat na změnu kurzu pouze při vzniku nové svíce.

Při vývoji experta AutoTrader byly implementovány 3 funkce pro obsluhu událostí uvedené ve výpisu 3.13. První funkcí je funkce *OnInit*, která obsluhuje událost inicializace experta a je volána ihned po jeho spuštění. Druhou funkcí, je funkce *OnDeinit* obsluhující událost deinicializace experta. Třetí funkce (*OnTick*) obsluhuje událost změny hodnoty kurzu, tj. přijetí nového ticku na měnovém páru pro který je expert advisor spuštěn [5].

```
1 || int OnInit()
2 || { ... }
3 ||
```

```

4 | void OnDeinit(const int reason)
5 | { ... }
6 |
7 | void OnTick()
8 | { ... }

```

Výpis 3.13: Základní funkce tvořící expert advisora

Činnost těchto tří funkcí je v případě experta `AutoTrader` následující. Funkce `OnInit` nejprve inicializuje data pro svého experta v klientské knihovně a pokusí se připojit a registrovat na serveru.

V případě, že je nastaven parametr experta `eRequestDataOnInit` si inicializační funkce po registraci experta vyžádá od serveru seznam požadovaných dat, který po celou dobu běhu experta zůstane neměnný. V případě neúspěchu vrací funkce chybový kód `INIT_FAILED`, který vyvolá událost deinicializace, kde dojde k odstranění dat experta z klientské knihovny a přerušení spojení experta se serverem.

Funkce `OnTick` po svém zavolání nejprve zkontroluje, zdali je nastaven parametr `eReactOnNewCandle`, pokud ano a poslední tick není tickem začínajícím novou svíci, funkce končí svou činnost a expert čeká na další tick.

V případě, že není nastaven parametr `eReactOnNewCandle` nebo došlo ke vzniku nové svíce, pokračuje expert kontrolou parametru `eRequestDataOnInit`. Pokud tento parametr není nastaven, expert si vyžádá od serveru aktuální seznam požadovaných dat, pokud parametr nastaven je, bude expert dále pracovat se seznamem požadovaných dat získaných při své inicializaci.

Dalším krokem je příprava datové odpovědi a její odeslání, poté může v případě úspěchu expert požádat server o aktualizované obchodní rozhodnutí, které následně vykoná a odešle serveru zprávu o výsledku.

V případě libovolné chyby v průběhu vykonávání funkce `OnTick` dojde k jejímu ukončení, k zastavení experta a k jeho odstranění z grafu, ke kterému byl přiřazen. Aktuálně otevřené obchodní pozice můžeme v takové situaci kontrolovat manuálně nebo počkat až skončí stop-lossem nebo take-profitem.

## 3.6 Knihovny obchodních strategií

Knihovna obchodní strategie je dynamicky linkovaná knihovna s rozhraním v jazyce C. Princip tvorby knihovny obchodní strategie si ukážeme na příkladu se strategií založenou na překřížení dvou plovoucích aritmetických průměrů.

Rozhraní knihovny pro naši strategii zobrazené ve výpisu 3.14 obsahuje 3 funkce. První funkce slouží k získání jména strategie, podle kterého je identifikována a její instance přiřazena klientovi při registraci. Druhá funkce vrací řetězec popisující obchodní strategii, který spolu se jménem strategie můžeme najít v seznamu načtených

modulů v AutoTrader serveru. Poslední funkce rozhraní vrací ukazatel na nově vytvořenou instanci samotné strategie.

```
1 #include "../TradeCore/abstract_strategy.hh"
2 #include "moving_averages.hh"
3
4 #define DllExport __declspec(dllexport)
5
6 extern "C"{
7
8 DllExport const char* name() {
9     return "MovingAverages";
10 }
11
12 DllExport const char* description() {
13     return "Strategy based on crossing of two moving averages.";
14 }
15
16 DllExport AbstractStrategy* getInstance() {
17     return new MovingAverages();
18 }
19
20 }
```

Výpis 3.14: Rozhraní knihovny obchodní strategie

Třída implementující konkrétní obchodní strategii dědí třídu *AbstractStrategy* a musí reimplementovat abstraktní metody společné pro všechny obchodní strategie. Hlavičkový soubor pro třídu strategie založené na překřížení dvou klouzavých průměrů je uveden ve výpisu 3.15.

```
1 #pragma once
2
3 #include "control_dialog.hh"
4 #include "../TradeCore/abstract_strategy.hh"
5 #include <memory>
6
7 class MovingAverages : public AbstractStrategy
8 {
9 public:
10     MovingAverages();
11     DataRequest requestedData() override;
12     bool tradeData(const TradeData& data) override;
13     OrderVector makeDecision() override;
14     void onClientDoubleClicked() override;
15     void decisionStatus(const OrderVector& status) override;
16
17 private:
```

```

18 |     DataRequest dataRequest_;
19 |     TradeData tradeData_;
20 |     OrderVector decisionStatus_;
21 |     std::unique_ptr<ControlDialog> dialog_;
22 |     bool dataValid_;
23 | };

```

Výpis 3.15: Hlavičkový soubor třídy strategie

První reimplementovanou metodou je metoda *requestedData* (výpis 3.16), která říká, jaká data momentálně potřebuje strategie pro svou činnost. Vzhledem k tomu, že naše strategie svůj požadavek na data nemění, je vytvořen již v konstruktoru.

```

1 | MovingAverages::MovingAverages()
2 |     : dataValid_(false)
3 | {
4 |     dataRequest_.append("PRICE");
5 |     dataRequest_.append("EMA10");
6 |     dataRequest_.append("EMA20");
7 | }
8 |
9 | DataRequest MovingAverages::requestedData()
10 | {
11 |     return dataRequest_;
12 | }

```

Výpis 3.16: Konstruktor třídy a metoda *requestedData*

Data přijatá od klienta obdrží strategie pomocí metody *tradeData*. V této metodě také můžeme provádět různé výpočty či předat část těchto dat dialogovému oknu, které je zobrazí. Funkce vrací *true*, pokud jsou přijatá data validní, nebo *false* pokud část dat chybí. Příklad funkce, která přijatá data uloží, předá dialogovému oknu k zobrazení a validuje, je pro naši strategii uveden ve výpisu 3.17.

```

1 | bool MovingAverages::tradeData(const TradeData& data)
2 | {
3 |     tradeData_ = data;
4 |
5 |     if (dialog_ && dialog_->isVisible())
6 |         dialog_->setData(tradeData_);
7 |
8 |     dataValid_ = tradeData_.validate(dataRequest_);
9 |     return dataValid_;
10 | }

```

Výpis 3.17: Metoda *tradeData*

Třetí reimplementovanou metodou je metoda *makeDecision*, která na základě přijatých dat provádí rozhodovací činnost a vrací seznam obchodních příkazů pro

odeslání do klientské části systému. Implementace této metody v naší strategii ověří, zdali jsou poslední přijatá data validní (Výpis 3.18), pokud validní nejsou, metoda končí aniž by učinila jakékoliv rozhodnutí, pokud data validní jsou, pokračuje metoda správou momentálně otevřených obchodních pozic.

```
1 | OrderVector MovingAverages::makeDecision()
2 | {
3 |     OrderVector decision;
4 |
5 |     if (!dataValid_)
6 |         return decision;
```

Výpis 3.18: Metoda *makeDecision* - 1. část

Strategie je navržena tak, že pracuje maximálně s jednou otevřenou obchodní pozicí. Pokud existuje otevřená pozice, strategie nové pozice neotevřívá a namísto toho sleduje překročení střední úrovně mezi otevírací cenou a hladinou take-profit. Při překročení této úrovně dochází k posunu úrovní stop-loss a take-profit o 150 pipů směrem k zisku a v případě prvního překročení je navíc uzavřena polovina obchodní pozice. Zdrojový kód nalezneme ve výpisu 3.19.

```
1 |     if (tradeData_.orderCount() > 0) {
2 |         int ticket = tradeData_.order(0).ticket;
3 |         double price = tradeData_.candle(0).close();
4 |
5 |         double orderOpen = tradeData_.order(0).openPrice;
6 |         double takeProfit = tradeData_.order(0).takeProfit;
7 |         double profitPoint = (orderOpen + takeProfit) / 2;
8 |
9 |         if ((tradeData_.order(0).type == "BUY" && price > profitPoint) ||
10 |            (tradeData_.order(0).type == "SELL" && price < profitPoint))
11 |         {
12 |             decision.push_back(Order::modify(ticket, 150, 150));
13 |
14 |             if (tradeData_.order(0).lots > 0.06)
15 |                 decision.push_back(Order::close(ticket, 0.05, 3));
16 |         }
17 |
18 |         return decision;
19 |     }
```

Výpis 3.19: Metoda *makeDecision* - 2. část

Poslední část metody *makeDecision* (Výpis 3.20) má na starosti otevření nové obchodní pozice. Nová pozice může být otevřena pouze v případě, že žádná otevřená obchodní pozice pro daného experta momentálně neexistuje. Nová pozice je

otevřena nákupem v případě překřížení plovoucího průměru o delší periodě plovoucím průměrem o kratší periodě zdola, v případě překřížení shora je obchodní pozice otevřena prodejem.

```
1 |     double EMA10_0 = tradeData_.value("EMA10", 0).second;  
2 |     double EMA10_1 = tradeData_.value("EMA10", 1).second;  
3 |     double EMA20_0 = tradeData_.value("EMA20", 0).second;  
4 |     double EMA20_1 = tradeData_.value("EMA20", 1).second;  
5 |  
6 |     if (EMA10_1 > EMA20_1 && EMA10_0 < EMA20_0)  
7 |         decision.push_back(Order::send("SELL", 0.1, 3, 300, 300));  
8 |     else if (EMA10_1 < EMA20_1 && EMA10_0 > EMA20_0)  
9 |         decision.push_back(Order::send("BUY", 0.1, 3, 300, 300));  
10 |  
11 |     return decision;  
12 | }
```

Výpis 3.20: Metoda *makeDecision* - 3. část

Další metodou ze třídy *AbstractStrategy*, kterou je třeba reimplementovat je metoda *onClientDoubleClicked* (Výpis 3.21). Tato metoda je volána po dvojkliku na řádek klienta v seznamu klientů serverové aplikace *AutoTrader*. V případě naší strategie dojde k vytvoření a zobrazení dialogového okna informujícího o momentálně otevřené obchodní pozici.

```
1 | void MovingAverages::onClientDoubleClicked()  
2 | {  
3 |     if (!dialog_)  
4 |         dialog_.reset(new ControlDialog());  
5 |  
6 |     dialog_ -> setData(tradeData_);  
7 |     dialog_ -> show();  
8 | }
```

Výpis 3.21: Metoda *onClientDoubleClicked*

Poslední reimplementovanou metodou je metoda *decisionStatus* (Výpis 3.22). Ta je volána po odeslání obchodního rozhodnutí a získání výsledku jeho zpracování od experta. V našem případě je získaná odpověď pouze uložena pro zpracování či zobrazení, které můžeme v budoucnu implementovat.

```
1 | void MovingAverages::decisionStatus(const OrderVector& status)  
2 | {  
3 |     decisionStatus_ = status;  
4 | }
```

Výpis 3.22: Metoda *decisionStatus*

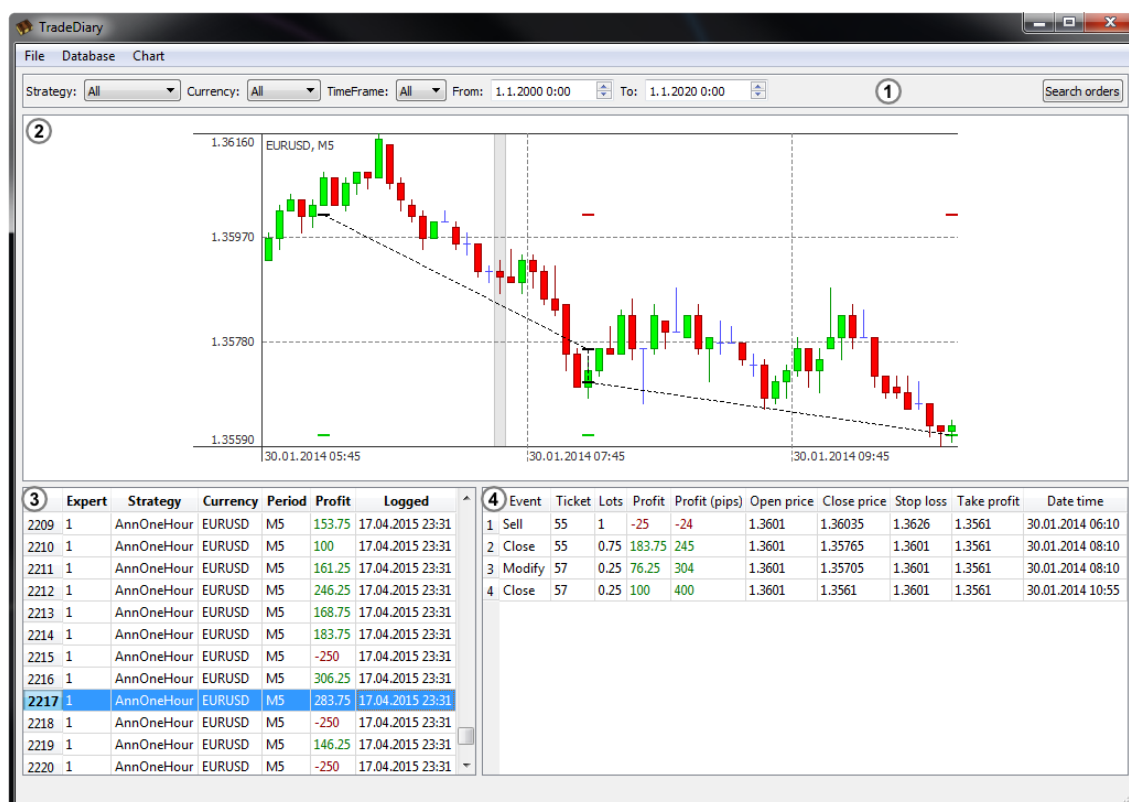


## 3.7 Obchodní deník

Logování uskutečněných obchodů nám dává cennou možnost jejich zpětného prohlížení, které je klíčové pro další ladění a optimalizaci obchodních strategií. Obchodní deník (Obr. 3.12) je program, jehož hlavním účelem je zobrazování informací o již uskutečněných obchodech zalogovaných v databázi a o událostech, které v rámci těchto obchodů proběhly.

Program nabízí filtr pro výběr zalogovaných obchodů a umožňuje jejich grafické i tabulkové zobrazení. Díky filtru pro výběr obchodů (1) si můžeme zvolit, jaké obchody chceme zobrazit. Filtrovat obchody můžeme podle názvu obchodní strategie, měnového páru a časového rámce, v jakém byl obchod uskutečněn. Můžeme také zvolit časový interval, ze kterého chceme obchody zobrazit.

Seznam nalezených obchodů je zobrazen v tabulce (3), která obsahuje informace o zisku nebo ztrátě z obchodu, o tom kdy a kterým klientem byl zalogován a ke které strategii, měnovému páru a časovému rámci přísluší.



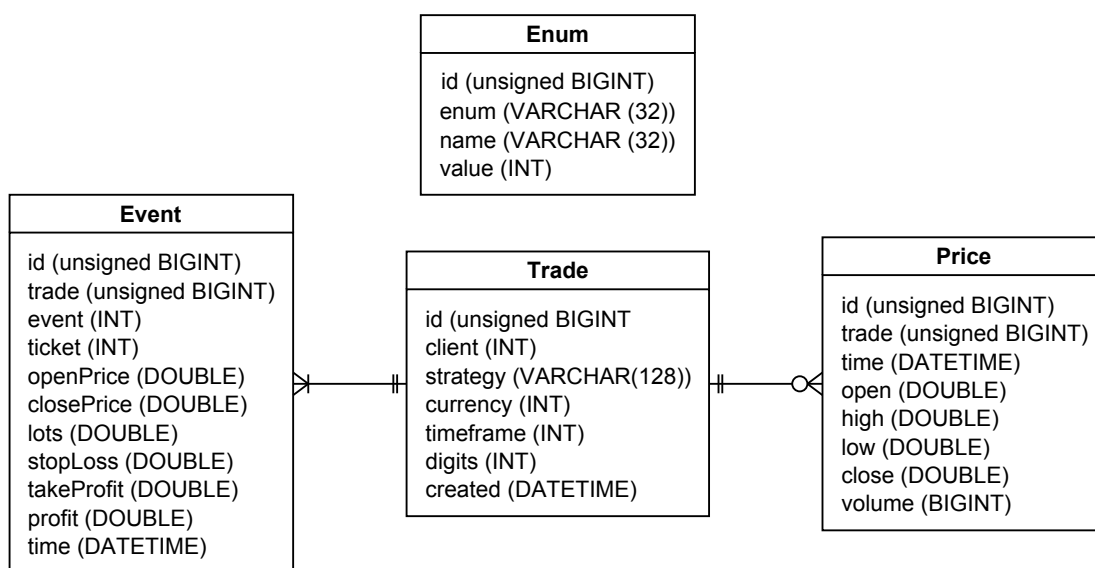
Obr. 3.12: Obchodní deník pro AutoTrader

Po kliknutí na libovolný obchod dojde k jeho zobrazení v grafu (2) a k výpisu jednotlivých událostí, které během něho proběhly. Řádek tabulky (4) odpovídá jedné této události a zobrazuje její typ, profit v jednotkách měny a v pipech příslušný

k datu a času kdy tato událost proběhla a důležité cenové úrovně, kterými jsou otevírací cena, zavírací cena, stop-loss a take-profit.

AutoTrader server zapisuje uskutečněné obchody do databáze v případě, že na-konfigurujeme připojení k databázi v jeho nastavení.

Databáze pro logování obchodů (Obr. 3.13) se skládá ze čtyř tabulek. Tabulka *Trade* obsahuje záznamy obchodů, tabulka *Event* obsahuje události, které v rámci těchto obchodů proběhly, v tabulce *Price* jsou uloženy kurzovní data z doby, kdy obchody probíhaly a tabulka *Enum* slouží pro ukládání hodnot enumerací.



Obr. 3.13: Entity relationship diagram databáze pro logování obchodů

Záznam obchodu uložený v tabulce *Trade* obsahuje identifikátor klienta v AutoTrader serveru, název strategie, hodnoty enumerací pro měnový pár a časovou periodu, počet desetinných míst, na které je udávána hodnota ceny a datum, kdy byl záznam vytvořen.

Záznam obchodní události uložený v tabulce *Event* obsahuje identifikátor obchodu, ke kterému náleží, hodnotu enumerace určující o jaký typ události se jedná, ticket obchodu, hodnoty cen, take-profitu, stop-lossu a profitu v čase události, velikost pozice v lotech a čas, kdy událost nastala.

Záznam ceny uložený v tabulce *Price* odpovídá jedné svíci grafu, obsahuje otevírací, maximální, minimální a zavírací cenu, objem obchodů, čas a identifikátor obchodu ke kterému je svíce zaznamenána.

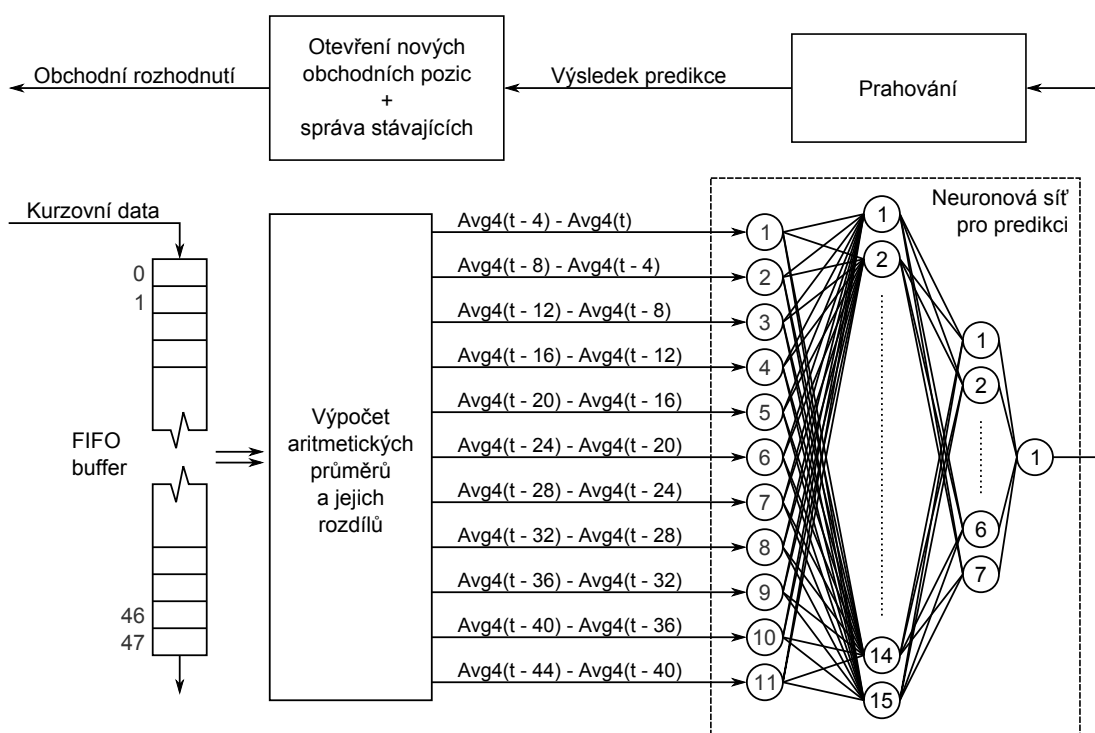
Záznam enumerace uložený v tabulce *Enum* se skládá z názvu enumerace ke které přísluší, názvu hodnoty enumerace a samotné hodnoty enumerace.

## 4 OBCHODNÍ STRATEGIE ZALOŽENÁ NA NEURONOVÝCH SÍTÍCH

Tato kapitola se zabývá tvorbou a testováním obchodní strategie založené na vícevrstvých neuronových sítích. Kapitola popisuje princip fungování této strategie, postup tvorby datových množin, učení a testování neuronových sítí a testování obchodní strategie na historických datech a na demo účtu u vybraného brokera.

### 4.1 Princip obchodní strategie

Obchodní strategii, jejíž schema je zobrazeno na obrázku 4.1, si můžeme představit jako systém skládající se z pěti hlavních částí. První částí je fronta (buffer typu FIFO) o maximální délce 48 prvků, druhou částí je blok pro výpočet aritmetických průměrů a jejich diferencí, třetí částí systému je samotná neuronová síť, ve čtvrté části systému dochází ke zpracování výstupu neuronové sítě prahováním a poslední, pátou částí, je mechanismus vykonávající obchodní rozhodnutí.



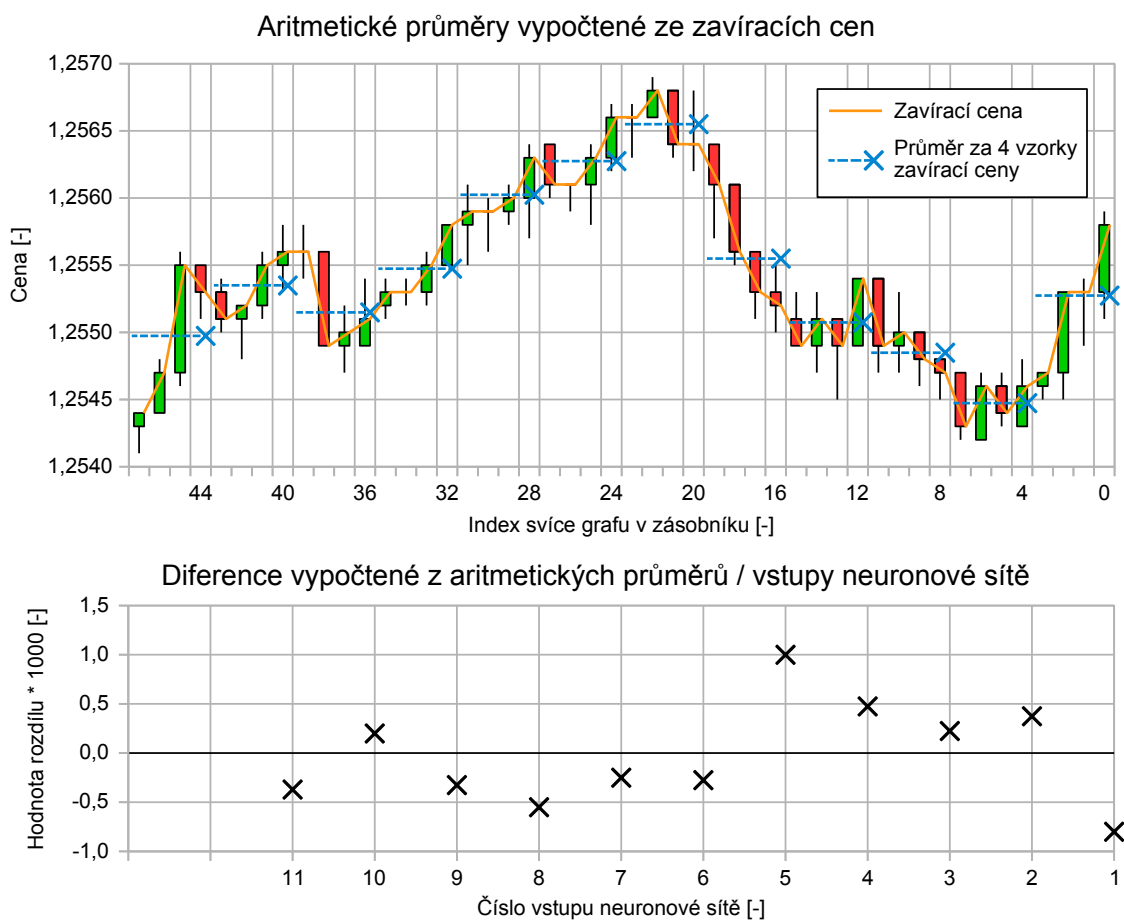
Obr. 4.1: Schéma obchodní strategie založené na neuronových sítích

Princip fungování celé strategie je následující. Celý proces začíná tím, že strategie přijme poslední dokončenou svíci grafu, kterou vloží do fronty. Pokud již fronta

dosáhla požadované velikosti, je před vložení nového prvku odstraněn poslední prvek tak, aby byl zachován požadovaný počet prvků.

Po aktualizaci fronty je ze závíracích cen svíček, které jsou v ní uloženy, spočteno 12 aritmetických průměrů, vždy pro 4 v čase po sobě jdoucí hodnoty cen. Pokud fronta ještě nedosáhla velikosti 48 prvků, nic se neděje a strategie čeká než je na požadovaný počet prvků doplněna.

V praxi by čekání na naplnění fronty trvalo 48 period grafu (například 4 hodiny při periodě grafu M5), proto strategie po svém spuštění v prvním požadavku na data požádá o všechna data potřebná k jejímu naplnění, to znamená o 48 posledních dokončených svíček.



Obr. 4.2: Výpočet vstupů neuronové sítě

Jednotlivé průměry jsou počítány právě ze čtyř hodnot ze dvou důvodů. Prvním důvodem je podmínka dělitelnosti délky fronty počtem hodnot pro výpočet průměru, což nám dává možnost počítat průměry ze dvou, tří, čtyř, šesti, osmi, dvanácti, šestnácti nebo čtyřiceti hodnot. Druhým důvodem je, že od těchto průměrů požadujeme nejen dostatečně detailní popis tvaru signálu, ale zároveň i dobrou schopnost filtrace šumu.

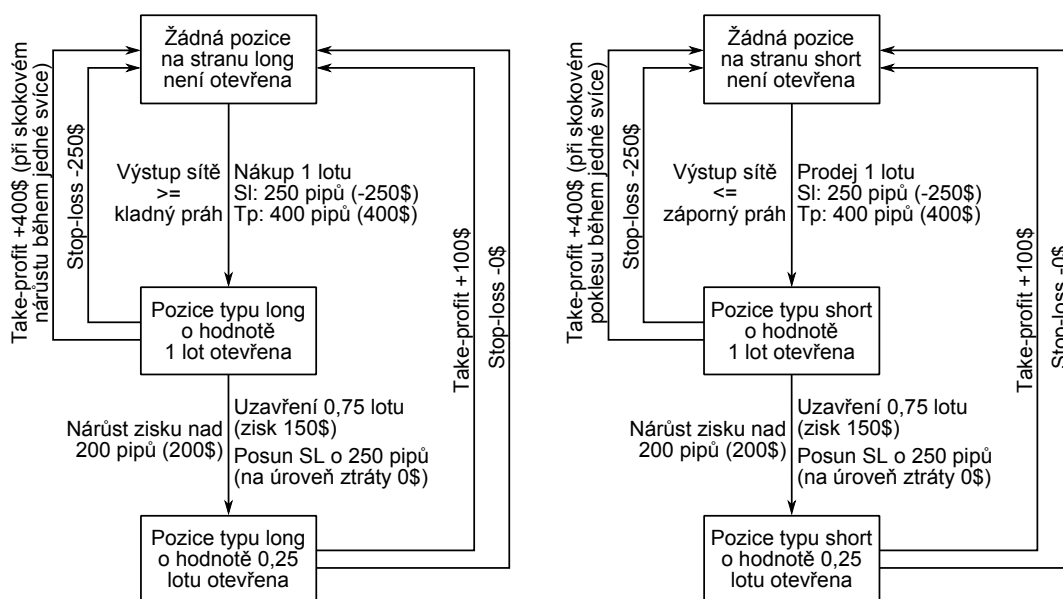
Z těchto 12-ti průměrů je dále vypočteno 11 diferencí, které jsou dále normalizovány konstantou úměrnou velikosti jednoho pipu tak, aby difference o velikosti 100 pipů měla hodnotu 1. Vypočtené normalizované difference jsou poté přivedeny na vstup neuronové sítě. Data ve formě svícového grafu, zavírací ceny svíci, vypočtené průměry a jejich normalizované difference jsou znázorněny na obrázku 4.2.

Neuronová síť je trénována tak, aby na základě normalizovaných diferencí aritmetických průměrů za poslední 4 hodiny predikovala, zdali bude hodnota 45-ti minutového průměru za jednu hodinu výrazně vyšší než současná hodnota výstupu 1 nebo výrazně nižší než současná hodnota výstupu -1.

Protože hodnota výstupu v případě predikce růstu či poklesu nemusí být přesně 1 nebo -1, je výsledek predikce určen prahováním. Pokud si hodnotu prahu označíme jako  $th$  a výstup sítě jako  $y$ , může být výsledek predikce dvojitý:

1. **Pokles** pro  $y \leq -th$
2. **Nárůst** pro  $y \geq th$

Se získaným výsledkem predikce pracuje obchodní rozhodovací mechanismus (Obr. 4.3), což je mechanismus, který na základě výsledku predikce a informací o otevřených obchodních pozicích provádí rozhodnutí o otevření nových obchodních pozic či o modifikaci a uzavření stávajících.



Obr. 4.3: Navržený a realizovaný rozhodovací mechanismus strategie

Mechanismus se po spuštění za běžné situace nachází ve stavu, kdy není otevřena žádná obchodní pozice. Nová obchodní pozice na stranu long je v takové situaci otevřena pokud neuronová síť predikuje nárůst ceny, nová pozice na stranu short je

otevřena naopak v případě, že neuronová síť predikuje pokles ceny. V obou případech navíc platí, že nové pozice mohou být otevřeny pouze během Evropské obchodní seance a pouze během úterý, středy a čtvrtka.

Evropská obchodní seance byla zvolena z důvodu vysoké volatility měnových párů EUR/USD a GBP/USD a také z důvodu, že v našem časovém pásmu probíhá v denních pracovních hodinách. Svíce za první 4 hodiny Evropské seance využijeme pro naplnění fronty, zbylý čas seance potom k obchodování.

Pondělky jsou vynechány, protože mohou být obtížně obchodovatelné z důvodu, že velké množství obchodníků čeká na vyhlášení ekonomických zpráv předtím, než otevřou své obchodní pozice. To také mimo jiné znamená nižší volatilitu trhu [39][9].

Pátky byly vynechány i přesto, že ve své první polovině stále nabízejí dobré obchodní příležitosti, protože během dne může dojít ke změně týdenního trendu v době, kdy obchodníci před víkendem uzavírají své pozice aby předešli riziku při jejich držení přes víkend. Ve druhé polovině dne navíc výrazně klesá celkový objem obchodů, což způsobuje značné zvýšení spreadů [39][9].

Velikost nově otevřené pozice je vždy 1 lot. Hodnota stop-lossu je nastavena na 250 pipů (ztráta 250\$) a hodnota take-profitu na 400 pipů (zisk 400\$). Zisk 400\$ (400 pipů s pozicí o velikosti celého lotu) je však vybrán pouze ve výjimečných případech, pokud dojde ke strmému vzrůstu (long pozice) nebo ke strmému poklesu (short pozice) ceny během jedné svíce.

V běžném případě, kdy nenastane skokové zvýšení ceny během jedné svíce, dojde po dosažení poloviny úrovně take-profit k uzavření části pozice o velikosti 0,75 lotu (zisk 150\$) a k posunu stop-lossu o 250 pipů na nulovou úroveň ztráty. Zbylá část pozice o velikosti 0,25 lotu tedy může skončit stop-lossem (ztráta 0\$) nebo ziskem ve výši 400 pipů (150\$). Celkový zisk tedy může být 300\$ a celková ztráta 250\$, což znamená poměr zisku a ztráty 6:5.

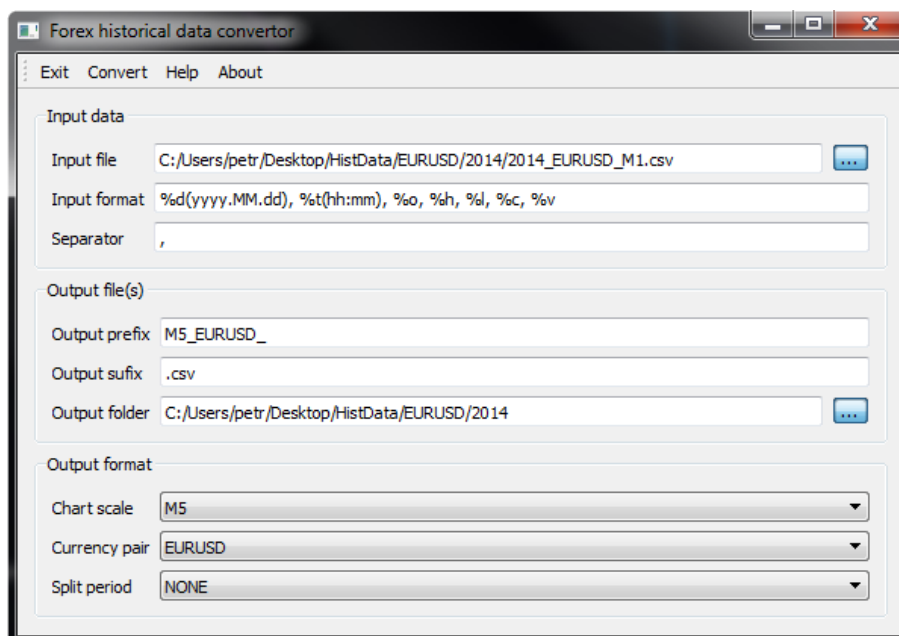
## 4.2 Tvorba datových množin

Pro vytvoření datových množin bylo nejprve nutné získat historická data. Na internetu je možné najít různé poskytovatele historických dat, pro účely této práce byla historická data stažena z webových stránek <http://www.histdata.com/>.

Webové stránky nabízí možnost stažení historických dat 66-ti měnových párů jako minutová nebo ticková data pro obchodní platformy MetaTrader 4 / MetaTrader 5, NinjaTrader, MetaStock a nebo jako textový soubor pro import v libovolném jiném programu [12].

Protože pro vytvoření tréninkových a testovacích datových množin potřebujeme data o periodě M5, byl za účelem konverze dat o periodě M1 na data o periodě

M5 vytvořen program DataConverter. Program DataConverter (Obr. 4.4) slouží ke konverzi dat získaných z libovolného dostupného zdroje do formátu dále použitého k vytvoření datových množin.



Obr. 4.4: Program DataConverter

Pro konverzi dat je nejprve nutné zvolit vstupní soubor. V našem případě se jedná o minutová data za jeden rok. Pro vstupní soubor je třeba nastavit formát jednoho jeho řádku a oddělovač jednotlivých hodnot.

Dále je třeba nastavit výstupní složku, do které budou uloženy výsledné soubory z konverze. Můžeme také zadat prefix a sufix pro tyto soubory. Výchozí prefix je generován na základě vybraného měnového páru a měřítka grafu. Prostřední část jmen výstupních souborů je tvořena datem a časem první svíčky ukládané do souboru.

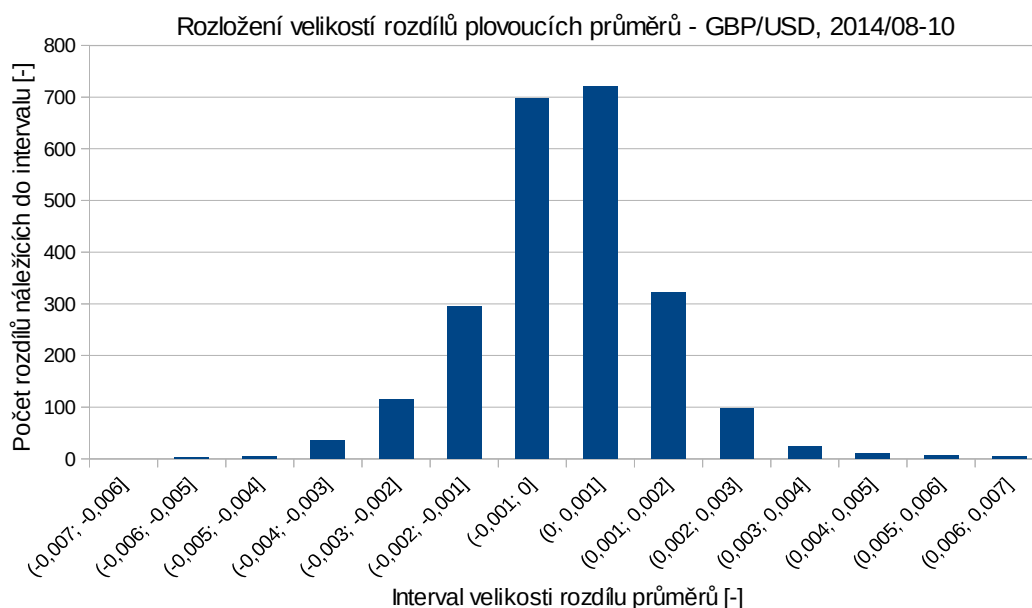
Posledním nastavením je měřítka grafu. Pokud vstupní soubor obsahuje minutová data a výsledný graf má být v měřítka pětiminutovém, konvertor přepočítá minutový graf na pětiminutový. Také je možno nastavit období, po jakém konvertor rozdělí vstupní soubor na několik výstupních souborů.

Po provedení všech potřebných nastavení již stačí stisknout tlačítko *Convert* na nástrojové liště a konvertor provede požadovanou konverzi nebo upozorní uživatele na chybu v nastavení.

Obchodní strategii budeme testovat při obchodování na měnových párech EUR/USD a GBP/USD za rok 2014. Tréninkové a testovací datové množiny pro neuronovou síť na následující měsíc vygenerujeme z historických dat za poslední 3 měsíce, to znamená, že datové množiny pro neuronovou síť obchodující například v květnu budeme generovat s využitím historických dat za únor, březen a duben.

Kromě tréninkových a testovacích datových množin si vytvoříme také datové množiny validační. Ty budou sloužit k ověření reálných predikčních schopností naučených neuronových sítí a budou vygenerovány z historických dat za měsíc, který bude daná neuronová síť obchodovat.

Datové množiny vytvoříme následujícím postupem. Pro vstupní data o  $n$  svíčkách pocházející z evropské obchodní seance vypočteme pro každou  $i$ -tou svíčku v rozsahu  $0 \leq i < n$  hodnoty 45-ti minutových plovoucích průměrů pro  $i$  a pro  $i + 12$  a velikosti jejich rozdílů roztrídíme do 2000 tříd v rozsahu od  $-0,02$  do  $0,02$ . Rozložení velikostí rozdílů pro měnový pár GBP/USD za měsíce srpen, září a říjen roku 2014 rozdělených do tříd o velikosti  $0,001$  je zobrazeno v grafu na obrázku 4.5.



Obr. 4.5: Rozložení velikostí rozdílů průměrů (GBP/USD, 2014/08-10)

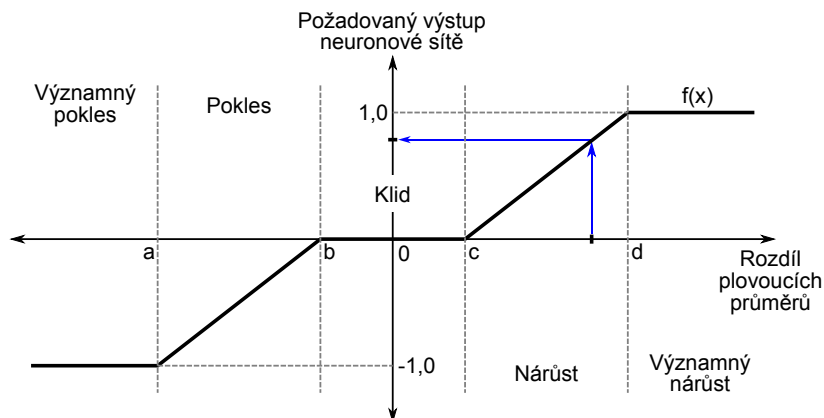
Za významný pokles považujeme všechny velikosti rozdílů ve všech třídách směrem od  $-\infty$  až po třídu, u které překročil součet počtu prvků hodnotu 175. Za významný nárůst naopak všechny velikosti rozdílů ve všech třídách směrem od  $+\infty$  až po třídu, u které překročil součet počtu prvků hodnotu 175. Za klidový stav jsou považovány prvky ve všech třídách na obě strany od hranice nulového rozdílu dokud součet počtu prvků nepřesáhne 175.

Dostaneme tedy 5 tříd, které nazveme *Významný pokles*, *Pokles*, *Klid*, *Nárůst* a *Významný nárůst*. V každé ze tříd *Významný pokles*, *Klid* a *Významný nárůst* bude přibližně 175 prvků přičemž zbylé prvky budou patřit do tříd *Pokles* a *Nárůst*. Meze těchto tříd si uložíme a následně je použijeme při tvorbě datových množin.

Při výpočtu jednotlivých tréninkových vzorů postupujeme tak, že pro každou



$i$ -tou svíci v rozsahu  $0 \leq i < n$ , kde  $n$  je celkový počet svíci z evropské obchodní seance vypočteme vstupní vektor s využitím aktuální svíce a 47 předchozích svíci postupem popsáním v kapitole 4.1.



Obr. 4.6: Určení požadovaného výstupu neuronové sítě

V rámci výpočtu požadovaného výstupu sítě opět spočteme rozdíl plovoucích průměrů pro  $i$  a pro  $i + 12$ . Hodnota požadovaného výstupu je -1 pokud rozdíl spadá do třídy *Významný pokles*, 0 pokud rozdíl spadá do třídy *Klid* a 1 pokud spadá do třídy *Významný nárůst*. Požadovaný výstup pro zbylé hodnoty rozdílů je vypočten lineární interpolací viz obrázek 4.6.

Vzorů s výstupem vypočteným interpolací (třídy *Pokles* a *Nárůst*) je největší množství, proto jsou pro urychlení procesu učení vybírány do výsledné datové množiny jen s takovou pravděpodobností, aby jich bylo v ideálním případě stejné množství jako v ostatních třídách. Pravděpodobnosti výběru vzoru pro třídy *Pokles* a *Nárůst* vypočteme podle vztahů

$$P_{pokles} = \frac{\text{Součet počtu prvků ve třídách Významný pokles, Klid a Významný nárůst}}{\text{Počet prvků ve třídě Pokles} \cdot 3}, \quad (4.1)$$

$$P_{nárůst} = \frac{\text{Součet počtu prvků ve třídách Významný pokles, Klid a Významný nárůst}}{\text{Počet prvků ve třídě Nárůst} \cdot 3}. \quad (4.2)$$

Po vytvoření všech vzorů dojde dále k jejich rozdělení do tréninkové a testovací datové množiny. Každý vzor má 15% pravděpodobnost, že bude zařazen do testovací množiny dat, v opačném případě bude zařazen do tréninkové množiny dat.

Dále potřebujeme vytvořit validační datovou množinu. Vzory pro tuto množinu vytvoříme z dat za měsíc, který bude neuronová síť obchodovat stejným způsobem, jako vzory pro množiny tréninkové a testovací s tím rozdílem, že nejsou vynechány žádné vzory patřící do tříd *Pokles* a *Nárůst*.

Tabulky 4.1 a 4.2 obsahují meze jednotlivých tříd požadovaného výstupu neuronové sítě a počty prvků v nich obsažené. Řádek tabulky udává měsíc, který bude daná neuronová síť obchodovat, to znamená, že meze a počty v řádku tabulky pro měsíc květen byly vygenerovány z dat za únor, březen a duben. Uvedené meze jsou použity také při generování validačních datových množin.

	Význ. nárůst			Nárůst	Klid			Pokles	Význ. pokles		
	Od	Do	Počet	Počet	Od	Do	Počet	Počet	Od	Do	Počet
Leden	$-\infty$	-0,00156	175	749	-0,0001	0,0001	211	1027	0,0018	$+\infty$	175
Únor	$-\infty$	-0,00176	175	805	-0,00012	0,00012	202	856	0,0017	$+\infty$	179
Březen	$-\infty$	-0,00176	175	833	-0,00012	0,00012	209	824	0,0016	$+\infty$	179
Duben	$-\infty$	-0,00176	177	861	-0,0001	0,0001	178	827	0,00156	$+\infty$	177
Květen	$-\infty$	-0,0014	175	903	-0.00008	0.00008	187	839	0,00142	$+\infty$	176
Červen	$-\infty$	-0,00136	175	1047	-0.00008	0.00008	216	723	0,00124	$+\infty$	179
Červenec	$-\infty$	-0,00134	180	1044	-0.00008	0.00008	225	712	0,00122	$+\infty$	179
Srpen	$-\infty$	-0,00134	177	1148	-0.00006	0.00006	201	699	0,001	$+\infty$	175
Září	$-\infty$	-0,00132	176	997	-0.00006	0.00006	208	782	0,00096	$+\infty$	177
Říjen	$-\infty$	-0,00156	176	1094	-0.00006	0.00006	198	750	0,00104	$+\infty$	182
Listopad	$-\infty$	-0,00186	175	908	-0.00008	0.00008	192	888	0,00176	$+\infty$	176
Prosinec	$-\infty$	-0,00176	175	870	-0,00012	0,00012	198	919	0,00194	$+\infty$	177

Tab. 4.1: Meze tříd požadovaného výstupu (EUR/USD)

	Význ. nárůst			Nárůst	Klid			Pokles	Význ. pokles		
	Od	Do	Počet	Počet	Od	Do	Počet	Počet	Od	Do	Počet
Leden	$-\infty$	-0,00188	177	848	-0,00012	0,00012	178	954	0,00204	$+\infty$	180
Únor	$-\infty$	-0,002	179	770	-0,00014	0,00014	195	893	0,00204	$+\infty$	180
Březen	$-\infty$	-0,002	178	756	-0,00016	0,00016	189	922	0,00208	$+\infty$	175
Duben	$-\infty$	-0,00186	176	715	-0,00014	0,00014	175	978	0,00218	$+\infty$	176
Květen	$-\infty$	-0,0015	177	731	-0,00012	0,00012	194	1002	0,00198	$+\infty$	176
Červen	$-\infty$	-0,00136	178	819	-0,0001	0,0001	218	946	0,00144	$+\infty$	179
Červenec	$-\infty$	-0,00122	180	879	-0.00008	0.00008	191	914	0,00128	$+\infty$	176
Srpen	$-\infty$	-0,0013	178	955	-0.00008	0.00008	225	867	0,00124	$+\infty$	175
Září	$-\infty$	-0,00122	177	945	-0.00008	0.00008	215	825	0,00118	$+\infty$	178
Říjen	$-\infty$	-0,00156	177	977	-0.00008	0.00008	184	882	0,00132	$+\infty$	180
Listopad	$-\infty$	-0,00184	175	890	-0,00012	0,00012	177	921	0,00174	$+\infty$	176
Prosinec	$-\infty$	-0,00222	178	838	-0,00016	0,00016	186	960	0,00212	$+\infty$	177

Tab. 4.2: Meze tříd požadovaného výstupu (GBP/USD)

## 4.3 Učení a testování neuronových sítí

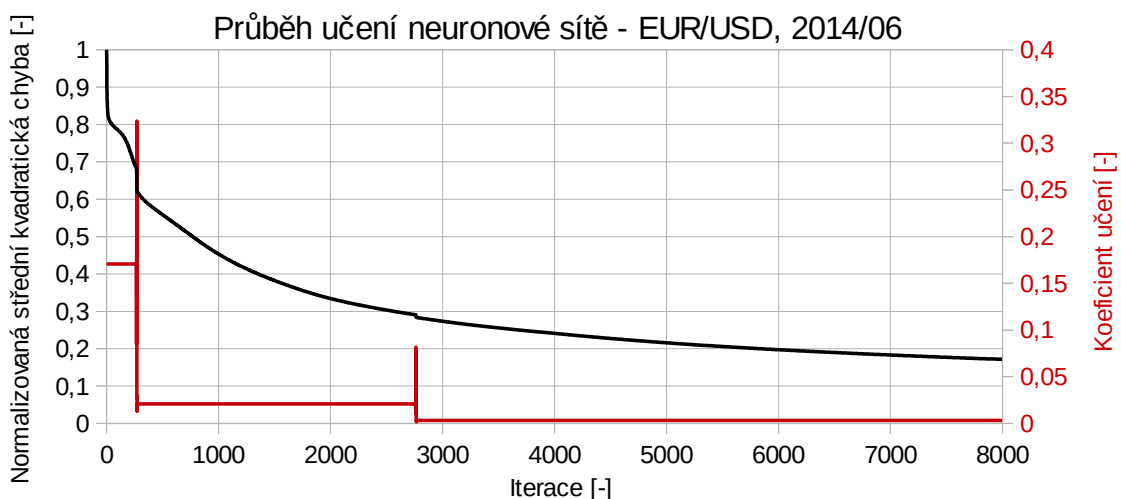
Implementace neuronové sítě použitá pro účely našeho projektu pochází z knihovny OpenNN, kterou je možné najít na webu <http://sourceforge.net/projects/opennn/>. OpenNN je knihovna napsaná v jazyce C++, jejíž hlavní výhodou je vysoký výkon a nízké nároky na paměť. Knihovna také podporuje paralelizaci výpočtů s využitím API OpenMP [23].

Knihovna nabízí možnost učení neuronové sítě pomocí algoritmů gradient descent, Levenberg-Marquardt, conjugate gradient a quasi Newton metody, dále je také možné nastavit parametry neuronové sítě pomocí algoritmu random search nebo pomocí evolučního algoritmu [23].

Pro naučení našich neuronových sítí byl zvolen algoritmus gradient descent (back-propagation) s proměnným koeficientem učení a počátečním náhodným nastavením vah v rozsahu od -0,3 do 0,3. Jako kritérium naučenosti sítě slouží normalizovaná střední kvadratická chyba, kterou můžeme vypočítat podle vzorce

$$NMSE = \frac{\sum_{i=0}^q \sum_{k=0}^n (y_{i,k} - t_{i,k})^2}{\sum_{i=0}^q \sum_{k=0}^n (t_{i,k} - \bar{t}_k)^2}, \quad (4.3)$$

kde  $q$  je počet vzorů v množině dat,  $n$  je počet výstupů neuronové sítě,  $y_{i,k}$  je  $k$ -tý skutečný výstup neuronové,  $t_{i,k}$  je  $k$ -tý požadovaný výstup neuronové sítě a  $\bar{t}_k$  je střední hodnota  $k$ -tého požadovaného výstupu neuronové sítě vypočtená za všechny tréninkové vzory v množině dat [23]. Ukázka průběhu koeficientu učení a chyby sítě pro červen 2014 a pro měnový pár EUR/USD je zobrazena v grafu na obrázku 4.7.



Obr. 4.7: Průběh učení neuronové sítě (EUR/USD, 2014/06)

Topologie všech trénovaných neuronových sítí (Obr. 4.1) je shodná, 11 neuronů ve vstupní vrstvě, 15 neuronů v první skryté vrstvě, 7 neuronů ve druhé skryté vrstvě a jeden neuron ve výstupní vrstvě.

Výsledná topologie neuronových sítí byla zjištěna experimentálně, pomocí opakovaného trénování a přidávání neuronů až dokud se nepodařilo neuronovou síť naučit na dostatečně malou chybu o hodnotě kolem 0,1 za omezený počet iterací učicího algoritmu. Protože výstup sítě požadujeme v rozsahu od -1 do 1, byl všem vrstvám jako přenosová funkce nastaven hyperbolický tangens.

Pro obchodování v daném měsíci bylo vytvořeno vždy 5 neuronových sítí diverzifikovaných rozdílnou tréninkovou množinou (náhodný výběr prvků ze tříd *Pokles* a *Nárůst*) a rozdílným náhodným počátečním nastavením vah. Pro výběr nejlepší sítě z této pětky byly použity výsledky predikce na testovacích datech, protože při reálné aplikaci nejsou validační data k dispozici.

Výběr neuronové sítě byl proveden na základě úspěšnosti predikce a počtu případů predikce významného nárůstu a poklesu, ke kterým došlo při daném prahu. Pro posouzení úspěšnosti predikce neuronové sítě v rámci této práce byla navržena a vypočtena dvě rozdílná kritéria.

První kritérium posuzuje počet správných a špatných predikcí tříd *Významný nárůst* a *Významný pokles*. Vysoká úspěšnost predikce podle tohoto kritéria je důležitá, protože v případě že neuronová síť predikuje významný nárůst namísto významného poklesu či naopak, může dojít k otevření obchodní pozice na opačnou stranu, což znamená vyšší pravděpodobnost ztráty.

Kritérium počítáme zvlášť pro třídy *Významný pokles* a *Významný nárůst*. Pro třídu *Významný pokles* provedeme výpočet podle vztahu

$$Krit_{1P} = \frac{p}{p+n} \cdot 100 [\%], \quad (4.4)$$

kde  $p$  je počet správných predikcí třídy *Významný pokles* a  $n$  je počet případů kdy je namísto třídy *Významný nárůst* chybně predikována třída *Významný pokles*.

Pro třídu *Významný nárůst* provedeme výpočet obdobně podle vztahu

$$Krit_{1N} = \frac{n}{n+p} \cdot 100 [\%], \quad (4.5)$$

kde  $n$  je počet správných predikcí třídy *Významný nárůst* a  $p$  je počet případů kdy je namísto třídy *Významný pokles* chybně predikována třída *Významný nárůst*.

Druhým kritériem je kritérium, které hodnotí úspěšnost predikce směru kurzu. Pro predikci třídy *Významný pokles* můžeme jeho hodnotu vypočítat podle vztahu

$$Krit_{2P} = \frac{q}{p} \cdot 100 [\%], \quad (4.6)$$

kde  $p$  je počet všech predikcí třídy *Významný pokles* a  $q$  je počet úspěšných predikcí. Za úspěšnou predikci považujeme v tomto případě každou predikci, která znamená pokles, tj. výstup neuronové sítě menší než nula.

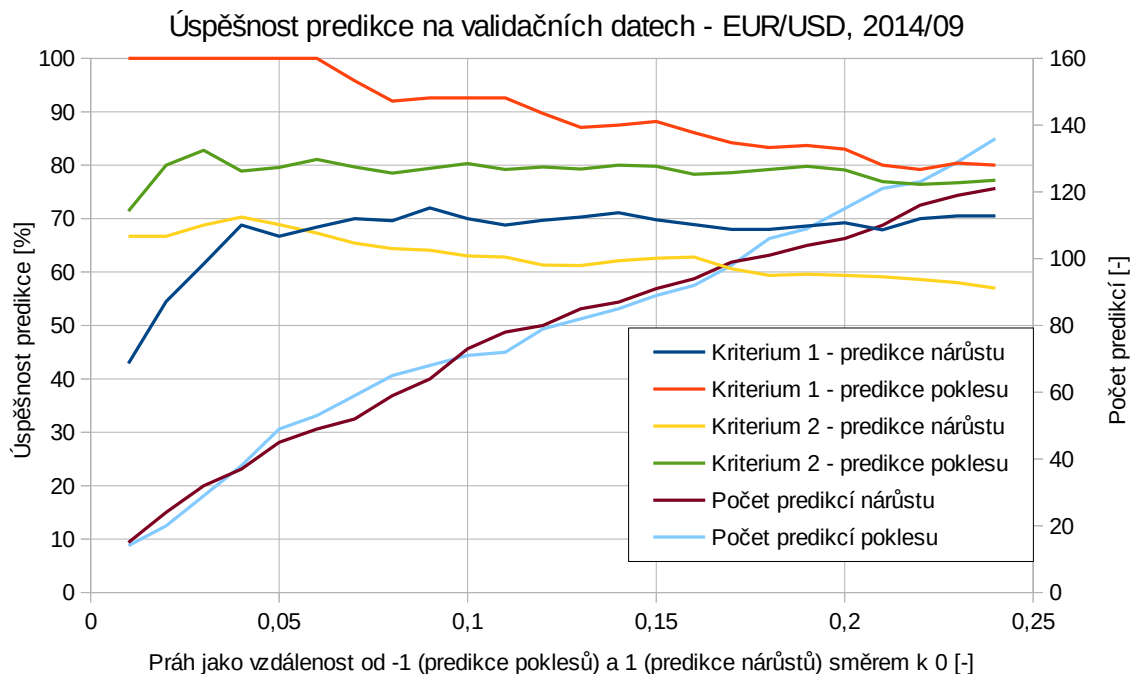
Pro predikci třídy *Významný nárůst* můžeme hodnotu druhého kritéria vypočítat obdobným způsobem podle vztahu

$$Krit_{2N} = \frac{q}{n} \cdot 100 [\%], \quad (4.7)$$

kde  $n$  je počet všech predikcí třídy *Významný nárůst* a  $q$  je počet úspěšných predikcí. Za úspěšnou predikci považujeme každou predikci, která znamená nárůst, tj. výstup neuronové sítě větší nebo roven nule.

Úspěšnost predikce podle obou dvou navržených kritérií pro 24 vybraných neuronových sítí určených k obchodování v roce 2014 na měnových párech EUR/USD a GBP/USD je zaznamenána v tabulkách 4.3 a 4.4 pro tréninkovou množinu dat, v tabulkách 4.5 a 4.6 pro testovací množinu dat a v tabulkách 4.7 a 4.8 pro validační množinu dat.

Závislost celkového počtu predikcí významného nárůstu a poklesu a úspěšnosti predikce podle obou kritérií na hodnotě prahu získaná z testu na validačních datech pro neuronovou síť určenou k obchodování na měnovém páru EUR/USD v září 2014 je zobrazena na obrázku 4.8.



Obr. 4.8: Test klasifikace na validačních datech (EUR/USD, 2014/09)

Úspěšnost predikce					
		Kritérium 1		Kritérium 2	
	Práh	Nárůst	Pokles	Nárůst	Pokles
Leden	±0,87	95,2% (59/62)	93,2% (41/44)	91,1% (72/79)	82,8% (48/58)
Únor	±0,96	98,6% (69/70)	95,9% (70/73)	92,6% (87/94)	84,2% (96/114)
Březen	±0,96	97,3% (36/37)	92,2% (47/51)	91,9% (57/62)	78,7% (74/94)
Duben	±0,97	94,5% (52/55)	97,4% (37/38)	87,2% (68/78)	78,5% (51/65)
Květen	±0,93	97,1% (67/69)	100,0% (60/60)	87,7% (93/106)	76,6% (82/107)
Červen	±0,92	96,8% (60/62)	98,0% (48/49)	87,2% (75/86)	80,0% (56/70)
Červenec	±0,96	98,3% (59/60)	93,6% (44/47)	93,7% (74/79)	84,1% (53/63)
Srpen	±0,95	100,0% (35/35)	95,3% (41/43)	95,9% (47/49)	93,9% (46/49)
Září	±0,94	100,0% (30/30)	95,7 % (22/23)	83,3% (35/42)	87,5% (28/32)
Říjen	±0,97	100,0% (22/22)	100,0% (55/55)	91,7% (33/36)	93,0% (66/71)
Listopad	±0,95	100,0% (56/56)	95,2% (60/63)	91,0% (71/78)	90,0% (72/80)
Prosinec	±0,94	91,8% (45/49)	98,4% (62/63)	83,0% (73/88)	81,7% (76/93)
<b>Celkem</b>		97,2% (590/607)	96,4% (587/609)	89,5% (785/877)	83,5% (748/896)

Tab. 4.3: Úspěšnost predikce na tréninkových datech (EUR/USD)

Úspěšnost predikce					
		Kritérium 1		Kritérium 2	
	Práh	Nárůst	Pokles	Nárůst	Pokles
Leden	±0,95	100,0% (76/76)	98,7% (74/75)	91,4% (117/128)	76,2% (93/122)
Únor	±0,96	92,5% (98/106)	98,8% (81/82)	83,1% (123/148)	78,3% (101/129)
Březen	±0,96	94,7% (71/75)	96,7% (59/61)	84,0% (105/125)	85,1% (86/101)
Duben	±0,95	93,2% (69/74)	98,4% (60/61)	88,7% (102/115)	85,9 % (85/99)
Květen	±0,97	94,3% (50/53)	94,4% (51/54)	86,0% (80/93)	79,6% (74/93)
Červen	±0,96	97,7% (43/44)	98,1% (52/53)	87,0% (67/77)	76,8% (63/82)
Červenec	±0,94	88,3% (68/77)	95,3% (61/64)	84,1% (95/113)	78,0% (78/100)
Srpen	±0,90	98,1% (52/53)	98,6% (69/70)	89,2% (66/74)	80,0% (88/110)
Září	±0,95	95,7% (44/46)	100,0% (45/45)	93,8% (60/64)	80,6% (50/62)
Říjen	±0,94	98,2% (55/56)	100,0% (43/43)	88,6% (70/79)	85,9% (55/64)
Listopad	±0,95	89,4% (59/66)	95,1% (39/41)	83,0% (83/100)	81,8% (54/66)
Prosinec	±0,94	98,4% (60/61)	98,4% (62/63)	89,7% (87/97)	78,7% (85/108)
<b>Celkem</b>		94,7% (745/787)	97,8% (696/712)	87,0% (1055/1213)	80,3% (912/1136)

Tab. 4.4: Úspěšnost predikce na tréninkových datech (GBP/USD)

		Úspěšnost predikce			
		Kritérium 1		Kritérium 2	
	Práh	Nárůst	Pokles	Nárůst	Pokles
Leden	±0, 87	100,0% (8/8)	100,0% (8/8)	90,9% (10/11)	83,3% (10/12)
Únor	±0, 96	100,0% (15/15)	100,0% (5/5)	95,2% (20/21)	73,3% (11/15)
Březen	±0, 96	100,0% (6/6)	100,0% (8/8)	100,0% (9/9)	91,7% (11/12)
Duben	±0, 97	100,0% (13/13)	100,0% (11/11)	100,0% (14/14)	81,2% (13/16)
Květen	±0, 93	100,0% (12/12)	100,0% (15/15)	94,1% (16/17)	85,7% (18/21)
Červen	±0, 92	100,0% (8/8)	100,0% (9/9)	100,0% (14/14)	100,0% (9/9)
Červenec	±0, 96	100,0% (7/7)	100,0% (6/6)	100,0% (12/12)	100,0% (8/8)
Srpen	±0, 95	100,0% (6/6)	100,0% (9/9)	100,0% (9/9)	100,0% (10/10)
Září	±0, 94	100,0% (4/4)	100,0% (8/8)	100,0% (8/8)	100,0% (8/8)
Říjen	±0, 97	100,0% (6/6)	100,0% (13/13)	90,9% (10/11)	100,0% (15/15)
Listopad	±0, 95	100,0% (7/7)	100,0% (13/13)	100,0% (7/7)	100,0% (15/15)
Prosinec	±0, 94	100,0% (9/9)	100,0% (12/12)	86,7% (13/15)	88,9% (16/18)
<b>Celkem</b>		100,0% (101/101)	100,0% (117/117)	95,9% (142/148)	90,6% (144/159)

Tab. 4.5: Úspěšnost predikce na testovacích datech (EUR/USD)

		Úspěšnost predikce			
		Kritérium 1		Kritérium 2	
	Práh	Nárůst	Pokles	Nárůst	Pokles
Leden	±0, 95	100,0% (12/12)	100,0% (11/11)	94,4% (17/18)	88,9% (16/18)
Únor	±0, 96	100,0% (13/13)	100,0% (10/10)	92,6% (25/27)	88,2% (15/17)
Březen	±0, 96	100,0% (15/15)	100,0% (12/12)	90,5% (19/21)	100,0% (15/15)
Duben	±0, 95	100,0% (17/17)	100,0% (8/8)	95,2% (20/21)	100,0% (13/13)
Květen	±0, 97	100,0% (10/10)	100,0% (7/7)	90,9% (10/11)	90,0% (9/10)
Červen	±0, 96	100,0% (6/6)	100,0% (10/10)	88,9% (8/9)	92,3% (12/13)
Červenec	±0, 94	90,0% (9/10)	100,0% (8/8)	92,9% (13/14)	71,4% (10/14)
Srpen	±0, 90	100,0% (6/6)	100,0% (9/9)	81,8% (9/11)	90,9% (10/11)
Září	±0, 95	100,0% (9/9)	100,0% (3/3)	100,0% (16/16)	85,7% (6/7)
Říjen	±0, 94	100,0% (8/8)	100,0% (7/7)	100,0% (13/13)	88,9% (8/9)
Listopad	±0, 95	87,5% (7/8)	100,0% (9/9)	75,0% (9/12)	100,0% (12/12)
Prosinec	±0, 94	100,0% (12/12)	100,0% (15/15)	100,0% (14/14)	89,5% (17/19)
<b>Celkem</b>		98,4% (124/126)	100,0% (109/109)	92,5% (173/187)	90,5% (143/158)

Tab. 4.6: Úspěšnost predikce na testovacích datech (GBP/USD)

Úspěšnost predikce					
		Kritérium 1		Kritérium 2	
	Práh	Nárůst	Pokles	Nárůst	Pokles
Leden	±0, 87	66,7% (14/21)	90,0% (27/30)	58,8% (50/85)	66,2% (53/80)
Únor	±0, 96	83,3% (5/6)	50,0% (13/26)	65,3% (32/49)	44,2% (42/95)
Březen	±0, 96	90,0% (9/10)	75,0% (6/8)	67,8% (40/59)	51,9% (41/79)
Duben	±0, 97	50,0% (2/4)	100,0% (9/9)	49,1% (28/57)	80,4% (41/51)
Květen	±0, 93	55,6% (10/18)	100,0% (17/17)	46,2% (30/65)	57,7% (60/104)
Červen	±0, 92	81,8% (9/11)	46,2% (12/26)	56,3% (40/71)	52,1% (25/48)
Červenec	±0, 96	100,0% (2/2)	100,0% (11/11)	76,9% (20/26)	71,9% (23/32)
Srpen	±0, 95	81,8% (9/11)	90,0% (9/10)	53,1% (26/49)	76,9% (20/26)
Září	±0, 94	68,4% (13/19)	100,0% (23/23)	67,3% (33/49)	81,1% (43/53)
Říjen	±0, 97	72,0% (36/50)	77,8% (35/45)	63,0% (63/100)	66,1% (76/115)
Listopad	±0, 95	100,0% (15/15)	71,4% (5/7)	66,7% (40/60)	60,5% (23/38)
Prosinec	±0, 94	66,7% (12/18)	67,7% (21/31)	52,9% (36/68)	58,5% (62/106)
<b>Celkem</b>		73,5% (136/185)	77,4% (188/243)	59,3% (438/738)	61,5% (509/827)

Tab. 4.7: Úspěšnost predikce na validačních datech (EUR/USD)

Úspěšnost predikce					
		Kritérium 1		Kritérium 2	
	Práh	Nárůst	Pokles	Nárůst	Pokles
Leden	±0, 95	85,2% (23/27)	71,8% (28/39)	72,6% (90/124)	52,9% (82/155)
Únor	±0, 96	92,0% (23/25)	62,5% (10/16)	78,4% (76/97)	54,7% (52/95)
Březen	±0, 96	54,5% (6/11)	20,0% (1/5)	68,3% (71/104)	40,8% (31/76)
Duben	±0, 95	20,0% (1/5)	50,0% (1/2)	70,7% (65/92)	49,3% (36/73)
Květen	±0, 97	100,0% (1/1)	- (0/0)	69,4% (43/62)	64,4% (29/45)
Červen	±0, 96	52,6% (10/19)	26,7% (4/15)	70,1% (54/77)	64,7% (44/68)
Červenec	±0, 94	28,6% (4/14)	77,3% (17/22)	51,5% (35/68)	54,2% (65/120)
Srpen	±0, 90	0,0% (0/1)	68,8% (11/16)	51,1% (24/47)	64,3% (45/70)
Září	±0, 95	83,3% (30/36)	58,1% (25/43)	80,4% (74/92)	53,8% (49/91)
Říjen	±0, 94	71,0% (49/69)	65,4% (17/26)	63,5% (108/170)	52,3% (46/88)
Listopad	±0, 95	48,3% (14/29)	68,2% (15/22)	60,0% (72/120)	54,7% (41/75)
Prosinec	±0, 94	67,9% (19/28)	71,4% (25/35)	68,4% (80/117)	59,5% (75/126)
<b>Celkem</b>		67,9% (180/265)	63,9% (154/241)	67,7% (792/1170)	55,0% (595/1082)

Tab. 4.8: Úspěšnost predikce na validačních datech (GBP/USD)



## 4.4 Implementace obchodní strategie

Strategie využívající neuronové sítě popsaná v kapitole 4.1 byla nazvána AnnOneHour a implementována způsobem popsáním v kapitole 3.6.

Instance obchodní strategie si po svém vytvoření nejprve načte seznam dostupných neuronových sítí ze souboru (Výpis 4.1). Seznam neuronových sítí (soubor *network.xml*) musí být umístěn v domovském adresáři uživatele, pod kterým je spuštěn AutoTrader server, ve složce *.AnnOneHour*.

Pro každou neuronovou síť je v seznamu uveden soubor, ve kterém je tato síť uložena, data od kdy do kdy bude obchodovat, měnový pár a perioda grafu pro kterou je určena, práh pro predikci a zdali budou při obchodování s danou sítí otevírány nové pozice prodejem a/nebo nákupem.

```
1 <?xml version="1.0"?>
2 <networks>
3   <network>
4     <file>D04_EURUSD_1.net</file>
5     <from>2015.04.01</from>
6     <to>2015.04.30</to>
7     <currency>EURUSD</currency>
8     <period>M5</period>
9     <threshold>0.06</threshold>
10    <buy>true</buy>
11    <sell>true</sell>
12  </network>
13  <network>
14    ...
15  </network>
16  ...
17 </networks>
```

Výpis 4.1: Seznam neuronových sítí ve formátu xml

Po načtení seznamu neuronových sítí je strategie připravena a čeká, až bude požádána klientem o seznam dat, která potřebuje pro svou činnost (funkce *requestedData*). Požadavek strategie na data se liší v závislosti na tom, zdali její fronta obsahuje dostatečný počet prvků nebo ne. Pokud ano, strategie si vyžádá pouze poslední 2 svíce grafu, aktuální nedokončenou a poslední dokončenou. V případě, že je fronta prázdná si strategie vyžádá více svíci tak, aby mohlo dojít k jejímu doplnění.

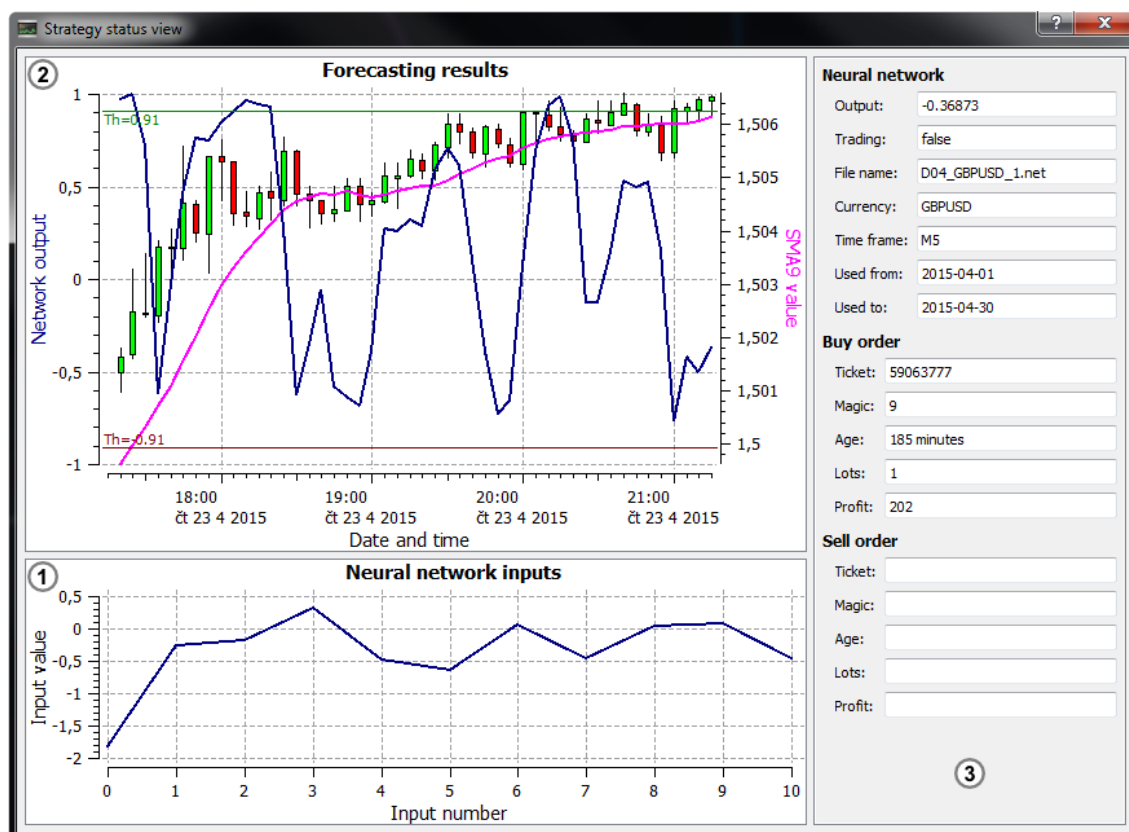
Po přijetí vyžádaných dat (ve funkci *tradeData*) dojde nejprve k jejich validaci a k vložení přijatých dokončených svíci do fronty. Pokud je fronta naplněna, je vypočten vstupní vektor pro neuronovou síť, proběhne kontrola, zdali je aktuálně

načtená neuronová síť určena pro obchodování v čase poslední svíce vložené do fronty a nakonec je vypočten výstup neuronové sítě.

Metoda *makeDecision* dále provádí obchodní rozhodnutí způsobem popsáním v kapitole 4.1, pouze však pokud je splněno, že poslední přijatá data jsou validní, že momentální čas je časem zvoleným k obchodování a že fronta obsahuje požadovaný počet svíci. Pokud momentálně neprobíhá evropská obchodní seance, strategie neotevřává nové obchodní pozice a pouze upravuje stávající.

Pokud aktuálně načtená neuronová síť není určena pro obchodování v daném čase, je v seznamu dostupných sítí vyhledána a načtena nová neuronová síť. Pokud neuronová síť pro dané období v seznamu neexistuje, je proměnná obsahující výstup neuronové sítě nastavena na nulovou hodnotu. Po provedení všech výpočtů je nakonec aktualizován stavový dialog strategie.

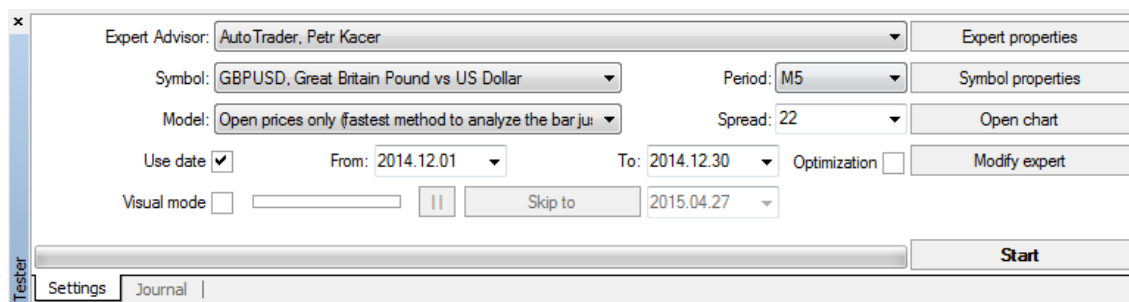
Stavový dialog (Obr. 4.9) zobrazuje v grafu (1) vstupy neuronové sítě vypočtené na základě nejaktuálnějších přijatých dat, v grafu (2) je zobrazen průběh ceny ve formě svíci, průběh 45-ti minutového klouzavého průměru a průběh výstupu neuronové sítě v závislosti na čase. V pravé části okna se nachází panel (3) zobrazující dodatečné informace o načtené neuronové síti a otevřených obchodních pozicích.



Obr. 4.9: Stavový dialog obchodní strategie

## 4.5 Testování strategie na historických datech

Testování obchodní strategie na historických za rok 2014 bylo realizováno s využitím testeru strategií dostupného v platformě MetaTrader 4 (Obr. 4.10). Testy byly provedeny na měnových párech EUR/USD a GBP/USD.



Obr. 4.10: Tester strategií v platformě MetaTrader 4

Při testování byla použita metoda simulace *Open prices only* vhodná pro naši strategii z důvodu, že tato strategie rozhoduje pouze při vzniku nové svíce. Hodnota spreadu pro test byla stanovena jako roční průměr jeho hodnot v čase zvoleném pro obchodování. Po provedení příslušných výpočtů byly zjištěny hodnoty spreadu 0,00012 pro měnový pár EUR/USD a 0,00022 pro měnový pár GBP/USD.

Počáteční kapitál na testovacím účtu byl před každým testem nastaven na hodnotu 10000\$, což při možné ztrátě 250\$ na obchod znamená, že v jednom obchodu riskujeme 2,5% z našeho celkového vstupního kapitálu.

Hodnota počátečního kapitálu však není při testech relevantní, protože hodnoty zisku a ztráty jsou dané velikostí otevřených obchodních pozic a nastavením úrovní stop-loss a take-profit, nikoliv kapitálem dostupným na obchodním účtu.

Celkem bylo provedeno 48 jednotlivých testů, vždy pro každý měnový pár, každý měsíc v roce a zvláště pro prodeje a nákupy. Výsledky testů obchodní strategie pro měnový pár EUR/USD jsou zobrazeny v tabulce 4.9 a pro měnový pár GBP/USD v tabulce 4.10.

Řádek tabulky zobrazuje počet prodejů a nákupů při obchodování za daný měsíc, jejich úspěšnost vyjádřenou v procentech a dosažený profit v dolarech. Poslední sloupec tabulky (*Trend*) udává měsíční trend vypočtený jako rozdíl zavírací a otevírací ceny měsíční svíce v pipech.

Pokud je otevřená obchodní pozice strategií uzavřena jen částečně, zůstává její zbývající část otevřena, avšak pod jiným ticketem. Toto chování obchodní platformy způsobuje, že při testování strategie je na zbylou část částečně uzavřené obchodní pozice z hlediska určení celkového počtu obchodních pozic a výpočtu jejich úspěšnosti pohlíženo jako na novou obchodní pozici, což má za následek, že při jejím

uzavření na úrovni stop-lossu, jehož úroveň je nastavena na hodnotu nulového nebo kladného zisku je do výsledné statistiky započtena jako úspěšná.

	Počet [-]		Úspěšnost [%]		Profit [\$]		Trend [pip]
	Nákupy	Prodeje	Nákupy	Prodeje	Nákupy	Prodeje	
Leden	29	29	55,17	65,52	-256,50	173,85	-2620
Únor	24	21	62,50	66,67	615,20	1005,65	3176
Březen	18	24	61,11	58,33	-97,25	-148,10	-25
Duben	18	19	44,44	73,68	-1396,85	676,65	943
Květen	18	21	33,33	76,19	-2214,30	1092,60	-2343
Červen	14	16	28,57	43,75	-1958,30	-636,70	600
Červenec	9	11	33,33	72,73	-864,40	935,00	-3027
Srpen	19	12	47,37	41,67	-563,75	-302,75	-2489
Září	19	23	68,42	82,61	317,50	1542,75	-4996
Říjen	42	38	73,81	65,79	900,90	-580,05	-1060
Listopad	24	16	54,17	62,50	-955,50	-316,70	-696
Prosinec	18	26	38,89	65,38	-1299,85	337,25	-3501
<b>Celkem</b>	<b>252</b>	<b>256</b>	<b>53,97</b>	<b>65,63</b>	<b>-7773,10</b>	<b>3779,45</b>	<b>-16038</b>

Tab. 4.9: Výsledky obchodování na historických datech (EUR/USD)

	Počet [-]		Úspěšnost [%]		Profit [\$]		Trend [pip]
	Nákupy	Prodeje	Nákupy	Prodeje	Nákupy	Prodeje	
Leden	33	38	60,61	55,26	-345,20	-1682,10	-1285
Únor	31	26	80,65	53,85	1979,60	-1748,30	3150
Březen	24	28	62,50	46,43	405,65	-1935,00	-660
Duben	20	24	65,00	54,17	54,40	-738,15	2081
Květen	14	19	35,71	68,42	-1280,65	70,70	-1096
Červen	21	20	57,14	45,00	-129,25	-746,02	3422
Červenec	22	26	50,00	57,69	-583,70	-877,75	-2212
Srpen	14	21	42,86	80,95	-1182,05	1189,53	-2822
Září	21	29	52,38	65,52	-275,75	-162,40	-3774
Říjen	42	34	73,81	58,82	1213,75	-1254,90	-2195
Listopad	35	30	57,14	80,00	-634,55	1148,35	-3270
Prosinec	39	37	69,23	51,35	622,30	-2288,45	-608
<b>Celkem</b>	<b>316</b>	<b>332</b>	<b>62,03</b>	<b>59,34</b>	<b>-155,45</b>	<b>-9024,49</b>	<b>-9269</b>

Tab. 4.10: Výsledky obchodování na historických datech (GBP/USD)

## 4.6 Testování strategie na demo účtu

Pro účely testování strategie *AnnOneHour* byl vybrán broker Admiral Markets <<http://www.admiralmarkets.cz>>. Broker byl vybrán na základě vlastností poskytovaných služeb s přihlédnutím k pozitivnímu hodnocení od jeho zákazníků.

Na zvoleném typu obchodního účtu s názvem *Admiral.Markets* nabízí broker minimální hodnotu počátečního vkladu 2000 Kč, finanční páku v rozsahu od 1:25 do 1:500, nízké spready, hedging, možnost otevřít obchodní pozici o velikosti od 0,01 do 100 lotů, obchodní platformu MetaTrader 4 a bezplatný demo účet.

Testování strategie při živém obchodování na demo účtu bylo provedeno ve dvou posledních týdnech dubna 2015. Úspěšnosti predikce vybraných neuronových sítí jsou uvedeny v tabulce 4.11. Výsledky testu jsou zaznamenány v tabulce 4.12 pro měnový pár EUR/USD a v tabulce 4.13 pro měnový pár GBP/USD.

		Kritérium 1		Kritérium 2	
		Nárůst	Pokles	Nárůst	Pokles
	<b>Práh</b>	<b>Úspěšnost predikce na tréninkových datech</b>			
EUR/USD	±0,94	97,4% (75/77)	92,8% (77/83)	86,3% (107/124)	80,0% (96/120)
GBP/USD	±0,91	93,3% (70/75)	100,0% (73/73)	85,1% (103/121)	88,5% (100/113)
	<b>Práh</b>	<b>Úspěšnost predikce na testovacích datech</b>			
EUR/USD	±0,94	100,0% (12/12)	100,0% (12/12)	95,7% (22/23)	83,3% (15/18)
GBP/USD	±0,91	100,0% (8/8)	100,0% (9/9)	87,5% (14/16)	70,6% (12/17)

Tab. 4.11: Úspěšnost predikce neuronových sítí pro test na demo účtu

	Počet [-]		Úspěšnost [%]		Profit [\$]	
	Nákupy	Prodeje	Nákupy	Prodeje	Nákupy	Prodeje
Út 21.04.2014	6	3	83,3	0,0	772,25	-774,00
St 22.04.2014	1	4	0,0	100,0	-252,00	516,25
Čt 23.04.2014	5	5	100,0	40,0	971,25	61,50
Út 28.04.2014	2	2	100,0	0,0	295,00	-501,00
St 29.04.2014	11	4	81,8	25,0	1196,50	-296,75
Čt 30.04.2014	3	6	0,0	16,7	-761,00	-853,50
<b>Celkem</b>	28	24	75,00	33,33	2222	-1847,5

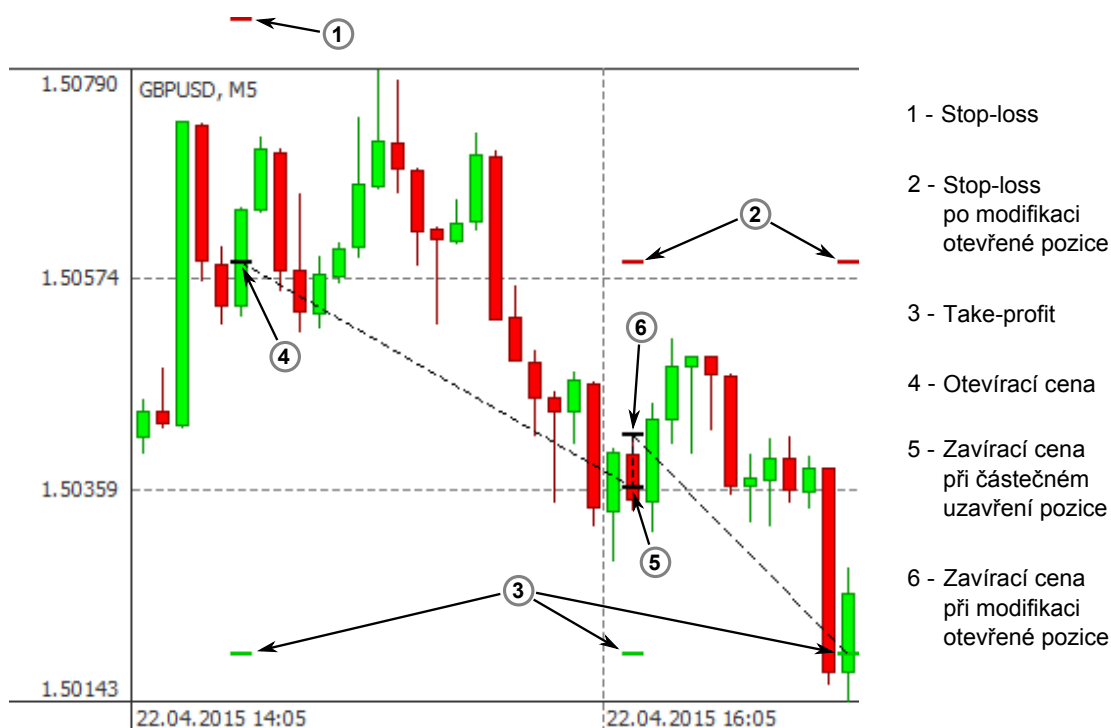
Tab. 4.12: Výsledky obchodování na demo účtu (EUR/USD)

Uvedená úspěšnost obchodů se od úspěšnosti uvedené v kapitole 4.5 liší tím, že pozice uzavřená při protnutí úrovně stop-lossu cenou je vždy považována za neúspěšnou, a to včetně případů, kdy je uzavřena se ziskem.

	Počet [-]		Úspěšnost [%]		Profit [\$]	
	Nákupy	Prodeje	Nákupy	Prodeje	Nákupy	Prodeje
Út 21.04.2014	3	2	33,3	0,0	-99,00	-504,00
St 22.04.2014	3	3	33,3	66,7	-30,50	22,25
Čt 23.04.2014	6	1	66,7	0,0	568,75	-253,00
Út 28.04.2014	4	3	100,0	33,3	543,75	-125,00
St 29.04.2014	7	2	71,4	0,0	430,25	-495,00
Čt 30.04.2014	8	3	25,0	66,7	-1231,00	42,50
<b>Celkem</b>	<b>31</b>	<b>14</b>	<b>54,84</b>	<b>35,71</b>	<b>182,25</b>	<b>-1312,25</b>

Tab. 4.13: Výsledky obchodování na demo účtu (GBP/USD)

Obrázek 4.11 zobrazuje průběh vybraného ziskového obchodu z testování strategie při obchodování na demo účtu. Jednotlivé značky zobrazují otevírací cenu pozice a úrovně stop-loss a take-profit po otevření pozice a po její modifikaci při protnutí ceny s polovinou úrovně take-profit.



Obr. 4.11: Zisková obchodní pozice z obchodování na demo účtu

## 5 ZÁVĚR

Prvním cílem této práce, jehož řešení je popsáno v kapitole 3, byl vývoj forexového obchodního systému, který by umožnil pohodlnou tvorbu automatických obchodních strategií v jazyce C++14 nezávisle na obchodní platformě poskytnuté brokerem.

Implementovaný obchodní systém se skládá z klientské části pro obchodní platformu MetaTrader 4 a ze serverové části. V případě potřeby podpory dalších obchodních platform tedy stačí pouze implementovat klientskou část systému pro danou platformu a serverovou část systému již není třeba modifikovat.

Z bezpečnostních a výkonnostních důvodů a z důvodu, že obchodní platforma a serverová část obchodního systému poběží na stejném počítači, byla jako způsob komunikace mezi klientskou a serverovou částí systému zvolena meziprocesová komunikace založená na výměně xml zpráv pomocí lokálních socketů. Výhodou tohoto přístupu je možnost snadného logování komunikace a zbavení se nutnosti synchronizace sdílené paměti.

Klientská část obchodního systému implementovaná v platformě MetaTrader 4 využívá funkce z klientské DLL knihovny, která byla vytvořena pro usnadnění implementace klientské části systému a která obsahuje funkce pro procházení a vytváření zpráv a komunikaci se serverem.

Serverová část obchodního systému (program AutoTrader) je samostatnou GUI aplikací, která po svém spuštění přijímá příchozí připojení, registruje a spravuje připojené klienty, na základě daných obchodních strategií provádí rozhodovací činnost a zapisuje provedené obchody do databáze.

Moduly obchodních strategií pro AutoTrader server jsou realizovány formou dynamických knihoven, které je možné v programu přidávat a odebírat pomocí správce modulů. Knihovny strategií jsou načítány za běhu programu, díky čemuž je možné pohodlně vytvářet nové obchodní strategie bez nutnosti neustále rekompilovat serverovou aplikaci.

Pro vývoj knihoven obchodních strategií bylo navrženo a využito jednotné rozhraní, které serveru umožňuje s těmito knihovnami pracovat. Třídy strategií obsažené v těchto knihovnách reimplementují abstraktního předka, díky kterému je možné pracovat se všemi strategiemi stejným způsobem. Tento přístup také umožnil implementaci na strategii nezávislého logování uskutečněných obchodů do databáze.

Za účelem logování uskutečněných obchodů byla navržena a realizována databáze o čtyřech tabulkách, do které jsou zapisovány údaje o jednotlivých obchodních pozicích a událostech ke kterým v rámci nich došlo a záznamy průběhu ceny z doby, kdy byly dané obchodní pozice otevřeny.

Z důvodu lepšího pochopení chování obchodních strategií a jejich budoucí optimalizace je důležité mít možnost zpětného procházení obchodů, které byly těmito

strategiemi provedeny. Za tímto účelem byl vytvořen program TradeDiary (obchodní deník), který slouží k zobrazování uskutečněných obchodů zaznamenaných v databázi. Tento program umožňuje filtrovat zaznamenané obchody podle času, strategie, měnového páru a časového rámce a zobrazit je v tabulkách a v cenovém grafu.

Druhým cílem této práce, který je řešen v kapitole 4, bylo vytvoření knihovny obchodní strategie pro AutoTrader založené na neuronových sítích a její otestování na historických datech a při obchodování na demo účtu u vybraného brokera.

Obchodní strategie navržená a realizovaná v rámci této práce je určena k obchodování během Evropské obchodní seance a používá vícevrstvé neuronové sítě k predikci nárůstu nebo poklesu hodnoty 45-ti minutového plovoucího průměru vypočteného ze zavíracích cen svíci o periodě M5 v časovém horizontu jedné hodiny na základě historických dat za poslední 4 hodiny.

Testování strategie na historických datech bylo provedeno pro měnové páry EUR/USD a GBP/USD za rok 2014. Pro otestování strategie bylo nejprve třeba vytvořit datové množiny a naučit a otestovat samotné neuronové sítě.

Tréninkové a testovací datové množiny pro neuronovou síť na daný měsíc byly vytvořeny rozdělením vzorů vygenerovaných z historických dat za předchozí 3 měsíce (85% tréninkové, 15% testovací). Vzory s požadovaným výstupem náležitým do tříd s nejvyšším počtem prvků byly vybrány náhodně tak, aby byl zachován přibližně stejný počet prvků ve všech třídách.

Validační data byla vytvořena z historických dat za měsíc, ve kterém bude neuronová síť obchodovat a umožnila její otestování z hlediska skutečných predikčních schopností. Do validační datové množiny byly zahrnuty všechny dostupné vzory, nedocházelo zde k žádnému náhodnému výběru ani dodatečnému rozdělování.

Pro účely testování strategie na historických datech bylo vytvořeno celkem 120 neuronových sítí, vždy 5 pro daný měsíc a měnový pár. Jednotlivé neuronové sítě jsou diverzifikovány rozdílnými tréninkovými datovými množinami (náhodný výběr prvků) a rozdílným náhodným počátečním nastavením vah.

Topologie všech neuronových sítí je shodná (11 vstupů, 15 neuronů v první vnitřní vrstvě, 7 neuronů ve druhé vnitřní vrstvě a jeden neuron ve výstupní vrstvě). Vzhledem k tomu, že výstup sítě požadujeme v rozsahu od -1 do 1, byl jako přenosová funkce vybrán hyperbolický tangens. Váhy všech neuronů v síti byly náhodně inicializovány v rozsahu od -0,3 do 0,3 a následné učení probíhalo až do dosažení 8000 iterací algoritmu učení, kterým byl algoritmus backpropagation s proměnným koeficientem učení.

Ze získaných pěti neuronových sítí pro daný měsíc a měnový pár byla vždy vybrána jedna, která byla použita k následnému testu strategie při obchodování. Výběr neuronové sítě byl proveden na základě úspěšnosti její predikce zjištěné na testovacích datech. Úspěšnost predikce zjištěná na validačních datech nebyla pro



výběr sítí použita z důvodu, že při praktické aplikaci nejsou data pro vytvoření validační datové množiny dostupná.

Skutečná úspěšnost predikce neuronových sítí vybraných na základě testovacích dat ověřená na validačních datech je uvedena v tabulkách 4.7 a 4.8. Získané hodnoty úspěšnosti predikce jsou v drtivé většině případů vyšší než 50% a prokazují, že neuronové sítě dokázaly najít jistý vztah mezi vstupy a výstupem.

Výsledky testu strategie získané simulací na historických datech z roku 2014 jsou uvedeny v tabulkách 4.9 a 4.10. Při obchodování ve směru dlouhodobého trendu na měnovém páru EUR/USD byla strategie zisková, v případě obchodování proti směru trendu na měnovém páru EUR/USD a v případě obchodování ve směru, i proti směru trendu na měnovém páru GBP/USD naopak ztrátová.

Neuronové sítě pro test strategie při živém obchodování na demo účtu byly opět vybrány z pětky kandidátů podle úspěšnosti jejich predikce na testovacích datech (Tabulka 4.11). Výsledky obchodování jsou uvedeny v tabulkách 4.12 a 4.13. Při obchodování na měnovém páru EUR/USD byla strategie zisková, obchodování na měnovém páru GBP/USD skončilo se ztrátou.

Ztrátové obchodování může být způsobeno nejistotou výběru nejlepší dostupné neuronové sítě, otevíráním obchodních pozic i proti směru dlouhodobého trendu, obchodováním v nevhodnou dobu a v případě měnového páru GBP/USD také vyšší hodnotou spreadu.

Do budoucna je plánováno zvýšení efektivity obchodní strategie s cílem dosažení celkového ročního zhodnocení kapitálu cca 10% a vylepšení obchodního systému za účelem dalšího usnadnění vývoje nových obchodních strategií. Plánováno je také vytvoření několika nových pomocných aplikací, jakými jsou tester strategií pracující s tickovými daty a obchodní deník pro tablety a mobilní telefony.

## LITERATURA

- [1] BAHREPOUR, Majid, Mohammad-R. AKBARZADEH-T., Mahdi YAGHOBI a Mohammad-B. NAGHIBI-S. An adaptive ordered fuzzy time series with application to FOREX. *Expert Systems with Applications*. 2011, vol. 38, issue 1, s. 475-485. DOI: 10.1016/j.eswa.2010.06.087. Dostupné z: <<http://linkinghub.elsevier.com/retrieve/pii/S0957417410005919>>
- [2] DE BRITO, Rodrigo F. B. a Adriano L. I. OLIVEIRA. A foreign exchange market trading system by combining GHSOM and SVR. *The 2012 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2012, s. 1-7. DOI: 10.1109/IJCNN.2012.6252496. Dostupné z: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6252496>>
- [3] DING, Y.R., Y.J. CAI, P.D. SUN a B. CHEN. The Use of Combined Neural Networks and Genetic Algorithms for Prediction of River Water Quality. *Journal of Applied Research and Technology* [online]. 2014, vol. 12, issue 3, s. 493-499 [cit. 2015-04-13]. DOI: 10.1016/S1665-6423(14)71629-3. Dostupné z: <<http://linkinghub.elsevier.com/retrieve/pii/S1665642314716293>>
- [4] EVANS, Cain, Konstantinos PAPPAS a Fatos XHAFA. Utilizing artificial neural networks and genetic algorithms to build an algo-trading model for intra-day foreign exchange speculation. *Mathematical and Computer Modelling*. 2013, vol. 58, 5-6, s. 1249-1266. DOI: 10.1016/j.mcm.2013.02.002. Dostupné z: <<http://linkinghub.elsevier.com/retrieve/pii/S0895717713000290>>
- [5] Event Handling Functions: MQL4 Documentation. MQL5 LTD. *MQL4: automated forex trading, strategy tester and custom indicators with MetaTrader* [online]. 2015 [cit. 2015-04-15]. Dostupné z: <<http://docs.mql4.com/basis/function/events>>
- [6] FAUSETT, Laurene. *Fundamentals of Neural Networks: Architectures, Algorithms And Applications*. Prentice-Hall, 1994. ISBN 978-0-13-334186-7.
- [7] FOREXMARKETHOURS.COM. *Forex market hours: London, New York, Tokyo, Sydney sessions* [online]. 2006 [cit. 2015-04-20]. Dostupné z: <<http://www.forexmarkethours.com/>>
- [8] Floating Exchange Rate Definition. INVESTOPEDIA, LLC. *Investopedia* [online]. 2004 [cit. 2015-03-26]. Dostupné z: <<http://www.investopedia.com/terms/f/floatingexchangerate.asp>>

- [9] Forex Market Hours: The Best Time to Trade - Hours, Days, Months. CASHBACKFOREX. *CashBackForex: Free Forex Trading and Training School Online* [online]. 2013 [cit. 2015-04-06]. Dostupné z: <<https://www.cashbackforex.com/en-us/school/tabid/426/ID/436818/best-hours-days-and-months-to-trade>>
- [10] *Forex Walkthrough*. INVESTOPEDIA, LLC. *Investopedia* [online]. 2010 [cit. 2015-03-26]. Dostupné z: <<http://www.investopedia.com/walkthrough/forex/>>
- [11] Fundamental Analysis Definition. INVESTOPEDIA, LLC. *Investopedia* [online]. 2003 [cit. 2015-04-01]. Dostupné z: <<http://www.investopedia.com/terms/f/fundamentalanalysis.asp>>
- [12] HISTDATA.COM. *Free Forex Historical Data* [online]. 2015 [cit. 2015-04-07]. Dostupné z: <<http://www.histdata.com/>>
- [13] KAČER, Petr. *Vícevrstvá neuronová síť*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2013. 52 s. Vedoucí práce byl doc. Ing. Václav Jirsík, CSc.
- [14] KAYAL, Abdulah. A Neural Networks filtering mechanism for foreign exchange trading signals. *2010 IEEE International Conference on Intelligent Computing and Intelligent Systems*. IEEE, 2010, č. 3, s. 159-167. DOI: 10.1109/ICICISYS.2010.5658495. Dostupné z: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5658495>>
- [15] KUTYŁOWSKA, Małgorzata. Neural network approach for failure rate prediction. *Engineering Failure Analysis* [online]. 2015, vol. 47, s. 41-48 [cit. 2015-04-13]. DOI: 10.1016/j.engfailanal.2014.10.007. Dostupné z: <<http://linkinghub.elsevier.com/retrieve/pii/S1350630714003070>>
- [16] LEAN YU, KIN KEUNG LAI a SHOUYANG WANG. Designing a hybrid AI system as a forex trading decision support tool. *17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)*. IEEE, 2005, 5 pp.-93. DOI: 10.1109/ICTAI.2005.56. Dostupné z: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1562920>>
- [17] LIAN, Cheng, Zhigang ZENG, Wei YAO a Huiming TANG. Multiple neural networks switched prediction for landslide displacement. *Engineering Geology* [online]. 2015, vol. 186, s. 91-99 [cit. 2015-04-13]. DOI:

- 10.1016/j.enggeo.2014.11.014. Dostupné z: <<http://linkinghub.elsevier.com/retrieve/pii/S0013795214003202>>
- [18] LIEN, Kathy. *Forex: Ziskové intradenní a swingové obchodní strategie*. Druhé, rozšířené vydání. Praha: FXstreet.cz, 2013, 263 s. ISBN 978-809-0441-828.
- [19] Line Chart Definition. INVESTOPEDIA, LLC. *Investopedia* [online]. 2006 [cit. 2015-03-26]. Dostupné z: <<http://www.investopedia.com/terms/1/linechart.asp>>
- [20] MA, Xiaolei, Zhimin TAO, Yinhai WANG, Haiyang YU a Yunpeng WANG. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies* [online]. 2015, vol. 54, s. 187-197 [cit. 2015-04-13]. DOI: 10.1016/j.trc.2015.03.014. Dostupné z: <<http://linkinghub.elsevier.com/retrieve/pii/S0968090X15000935>>
- [21] MASTERS, Timothy. *Practical Neural Network Recipes in C++*. San Francisco: Morgan Kaufmann, 1993. ISBN 978-0-12-479040-7.
- [22] MENA, R., F. RODRÍGUEZ, M. CASTILLA a M.R. ARAHAL. A prediction model based on neural networks for the energy consumption of a bioclimatic building. *Energy and Buildings* [online]. 2014, vol. 82, s. 142-155 [cit. 2015-04-13]. DOI: 10.1016/j.enbuild.2014.06.052. Dostupné z: <<http://linkinghub.elsevier.com/retrieve/pii/S0378778814005349>>
- [23] OpenNN documentation. INTELNICS. *Analytics, Data Mining, Big Data* [online]. 2015 [cit. 2015-04-08]. Dostupné z: <<http://www.intelnics.com/opennn/doc-opennn>>
- [24] PATEL, Maulika S. a Himanshu S. MAZUMDAR. Knowledge base and neural network approach for protein secondary structure prediction. *Journal of Theoretical Biology* [online]. 2014, vol. 361, s. 182-189 [cit. 2015-04-13]. DOI: 10.1016/j.jtbi.2014.08.005. Dostupné z: <<http://linkinghub.elsevier.com/retrieve/pii/S0022519314004536>>
- [25] PEKÁREK, J. *Analýza a predikce vývoje devizových trhů pomocí chaotických atraktorů a neuronových sítí*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2014. 99 s. Vedoucí diplomové práce Ing. Jan Budík, Ph.D.
- [26] PODHAJSKÝ, Petr a Tomáš NESNÍDAL. *Kompletní průvodce úspěšného finančníka*. Praha: Centrum finančního vzdělávání, 2010, 338 s. Finančník. ISBN 978-809-0387-454.

- [27] Pip Definition. INVESTOPEDIA, LLC. *Investopedia* [online]. 2003 [cit. 2015-04-29]. Dostupné z: <<http://www.investopedia.com/terms/p/pip.asp>>
- [28] QLocalServer Class. THE QT COMPANY LTD. *Qt Documentation* [online]. 2015 [cit. 2015-03-03]. Dostupné z: <<http://doc.qt.io/qt-5/qlocalserver.html>>
- [29] QLocalSocket Class. THE QT COMPANY LTD. *Qt Documentation* [online]. 2015 [cit. 2015-03-03]. Dostupné z: <<http://doc.qt.io/qt-5/qlocalsocket.html>>
- [30] RAMASAMY, P., S.S. CHANDEL, Amit Kumar YADAV a B. CHEN. Wind speed prediction in the mountainous region of India using an artificial neural network model. *Renewable Energy* [online]. 2015, vol. 80, issue 3, s. 338-347 [cit. 2015-04-13]. DOI: 10.1016/j.renene.2015.02.034. Dostupné z: <<http://linkinghub.elsevier.com/retrieve/pii/S0960148115001342>>
- [31] STANÍČEK, L. *Predikce burzovního trhu pomocí neuronových sítí*. Zlín: Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2012. 78 s. Vedoucí diplomové práce Ing. Petr Chalupa, Ph.D.
- [32] Slippage Definition. INVESTOPEDIA, LLC. *Investopedia* [online]. 2003 [cit. 2015-04-29]. Dostupné z: <<http://www.investopedia.com/terms/s/slippage.asp>>
- [33] Standard Lot Definition. INVESTOPEDIA, LLC. *Investopedia* [online]. 2010 [cit. 2015-04-29]. Dostupné z: <<http://www.investopedia.com/terms/s/standard-lot.asp>>
- [34] Technical Indicator Definition. INVESTOPEDIA, LLC. *Investopedia* [online]. 2006 [cit. 2015-04-01]. Dostupné z: <<http://www.investopedia.com/terms/t/technicalindicator.asp>>
- [35] Trading Platform Definition. INVESTOPEDIA, LLC. *Investopedia* [online]. 2010 [cit. 2015-03-27]. Dostupné z: <<http://www.investopedia.com/terms/t/trading-platform.asp>>
- [36] Trading Strategy Definition. INVESTOPEDIA, LLC. *Investopedia* [online]. 2010 [cit. 2015-04-01]. Dostupné z: <<http://www.investopedia.com/terms/t/trading-strategy.asp>>
- [37] VOBOŘIL, Tomáš. COLOSSEUM, a.s. *Forex eBook* [online]. Praha, 2011 [cit. 2014-12-14]. Dostupné z: <<http://www.colosseum.cz/>>.

- [38] VOLNÁ, Eva. *Neuronové sítě 1*. Ostrava, 2002. Učební texty. Ostravská Univerzita, Přírodovědecká fakulta.
- [39] What is a "good" time to trade forex?. THE FOREX CHAMPIONSHIP. *The Forex Championship Blog* [online]. 2001 [cit. 2015-04-06]. Dostupné z: <<http://theforexchampionship.com/blog/trading-school/good-time-trade-forex/>>
- [40] YAĀCI, Wahiba a Evgueniy ENTCHEV. Performance prediction of a solar thermal energy system using artificial neural networks. *Applied Thermal Engineering* [online]. 2014, vol. 73, issue 1, s. 1348-1359 [cit. 2015-04-13]. DOI: 10.1016/j.applthermaleng.2014.07.040. Dostupné z: <<http://linkinghub.elsevier.com/retrieve/pii/S1359431114006164>>
- [41] YAO, Jingtao a Chew Lim TAN. A case study on using neural networks to perform technical forecasting of forex. *Neurocomputing*. 2000, vol. 34, 1-4, s. 79-98. DOI: 10.1016/S0925-2312(00)00300-3. Dostupné z: <<http://linkinghub.elsevier.com/retrieve/pii/S0925231200003003>>
- [42] YU, Lean, Shouyang WANG a Kin Keung LAI. A neural-network-based nonlinear metamodeling approach to financial time series forecasting. *Applied Soft Computing*. 2009, vol. 9, issue 2, s. 563-574. DOI: 10.1016/j.asoc.2008.08.001. Dostupné z: <<http://linkinghub.elsevier.com/retrieve/pii/S156849460800118X>>