

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

## AUTOMATIZOVANÉ OBCHODNÍ SYSTÉMY A JEJICH TESTOVÁNÍ

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN ZAPLETAL

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# AUTOMATIZOVANÉ OBCHODNÍ SYSTÉMY A JEJICH TESTOVÁNÍ

TESTING OF AUTOMATED TRADING SYSTEMS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN ZAPLETAL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PATRIK PETŘÍK

BRNO 2011

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav informačních systémů

Akademický rok 2010/2011

**Zadání bakalářské práce**

Řešitel: **Zapletal Martin**

Obor: Informační technologie

Téma: **Automatizované obchodní systémy a jejich testování  
Testing of Automated Trading Systems**

Kategorie: Softwarové inženýrství

Pokyny:

1. Seznamte se s problematikou testování automatizovaných obchodních systémů. Zaměřte se na metody zpětného a dopředného testování. Seznamte se s metodami porovnání výsledků.
2. Specifikujte požadavky na systém pro testování obchodních dat. Systém navrhnete.
3. Po konzultaci s vedoucím implementujte zvolenou část systému.
4. Po konzultaci s vedoucím testujte aplikaci při různých parametrech.
5. Zhodnoťte dosažené výsledky a diskutujte další možný vývoj.

Literatura:

- Frank J Fabozzi Cfa, Frank J. Fabozzi, CFA, Sergio M. Focardi, Petter N. Kolm, Quantitative Equity Investing: Techniques and Strategies, Wiley, 2010, ISBN 0470262478.
- George A. Fontanills, Trading Options For Dummies, For Dummies, 2008, ISBN 0470241764.
- John Wang, Grace Wang, AbleTrend: Identifying and Analyzing Market Trends for Trading Success, Wiley, 2010, ISBN 0470581204.

Při obhajobě semestrální části projektu je požadováno:

- Body 1-2.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Petřík Patrik, Ing.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2010

Datum odevzdání: 18. května 2011

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav informačních systémů  
612-66 Brno, Božetěchova 2

doc. Dr. Ing. Dušan Kolář  
vedoucí ústavu

## **Abstrakt**

Práce se zabývá testováním automatických systémů pro burzovní obchod. Čtenář je seznámen se základními pojmy z oblasti finančních trhů a burzovního obchodu. Následně jsou představeny nejen ukazatele výkonnosti obchodního systému, ale také ukazatele popisující výkonnost metody pro předpověď budoucího vývoje trhu, která je v obchodním systému použita. Stěžejní částí práce je vysvětlení významu vlivu volby vstupních dat na výkon obchodního systému. Prezentovány jsou také výsledky vybraných testů obchodních systémů, které byly provedeny s využitím aplikace vytvořené jako součást práce.

## **Abstract**

This thesis deals with testing of automated systems used for stock market trading. The reader acquires basic knowledge about financial markets and stock market. Subsequently, performance indicators for the whole trading system and for the prediction model used in the system are introduced. The explanation of influence of input data selection on trading system performance is the most crucial part of the paper. Selected trading system tests were done using the application, which was made as a part of this thesis. Results of these tests are also presented.

## **Klíčová slova**

Automatický obchodní systém, burza, testování, neuronová síť, vliv dat na výsledky, in-sample, out-of-sample, zpětné testování, dopředné testování.

## **Keywords**

Automated trading system, stock market, testing, neural network, influence of input data on results, in-sample, out-of-sample, backtesting, forward testing.

## **Citace**

Martin Zapletal: Automatizované obchodní systémy a jejich testování, bakalářská práce, Brno, FIT VUT v Brně, 2011



# Automatizované obchodní systémy a jejich testování

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Patrika Petříka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Martin Zapletal

18. května 2011

## Poděkování

Děkuji vedoucímu práce Ing. Patriku Petříkovi za odbornou pomoc při tvorbě práce a rady při konzultacích.

© Martin Zapletal, 2011.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1 Úvod</b>	<b>4</b>
<b>2 Obchodní systémy</b>	<b>5</b>
2.1 Finanční trh	5
2.1.1 Burza	5
2.1.2 Instrumenty finančního trhu	6
2.1.3 Rozdíly mezi jednotlivými trhy	6
2.1.4 Pákový efekt	6
2.1.5 Zprostředkování obchodu	7
2.1.6 OHLC Data	7
2.1.7 Likvidita	8
2.1.8 Volatilita	8
2.2 Obchodní systém	9
2.2.1 Analýza trhu	9
2.2.2 Poziční rozhodování	11
2.2.3 Money management, řízení rizika a psychologie	11
2.3 Dělení obchodních systémů dle doby držení pozice	12
2.4 Dělení obchodních systémů dle přístupu k rozhodování	12
2.4.1 Diskreční přístup	12
2.4.2 Automatické obchodní systémy	12
2.5 Použití neuronových sítí	13
2.5.1 Přehled	13
2.5.2 Příprava sítě	13
2.5.3 Vstupní data	13
2.5.4 Trénink sítě	14
2.5.5 Struktura sítě	15
2.5.6 Výkon neuronových sítí	15
<b>3 Testování automatických obchodních systémů</b>	<b>16</b>
3.1 Metody testování	17
3.1.1 Zpětné testování	17
3.1.2 Dopředné testování	17
3.2 Metody porovnání výsledků	17
3.2.1 In-sample	17
3.2.2 Out-of-sample	18
3.2.3 Vliv volby dat na výsledky	18
3.2.4 Curve-fitting	19
3.2.5 Porovnání modelů	19

3.3	Metody výběru dat . . . . .	20
3.3.1	Rozdělení dat . . . . .	20
3.3.2	Křížová validace . . . . .	20
3.3.3	Leave-one-out . . . . .	20
3.3.4	Bootstrap . . . . .	20
3.4	Ukazatele pro klasifikaci do tříd . . . . .	21
3.4.1	Matice záměn . . . . .	21
3.4.2	Celková správnost a celková chyba . . . . .	21
3.4.3	Správnost pro jednotlivé třídy . . . . .	22
3.4.4	Přesnost a úplnost . . . . .	22
3.4.5	Senzitivita a specificita . . . . .	22
3.5	Ukazatele pro numerické predikce . . . . .	22
3.5.1	Graf závislosti chyby předpovědi na čase . . . . .	23
3.5.2	Graf závislosti chyby předpovědi na předpovězené hodnotě . . . . .	23
3.5.3	Graf závislosti skutečné hodnoty na předpovězené hodnotě . . . . .	24
3.5.4	Graf závislosti chyby předpovědi na skutečné hodnotě . . . . .	24
3.5.5	Hodnocení ukazatelů a možné problémy . . . . .	25
3.6	Další ukazatele . . . . .	25
3.6.1	Křivka učení . . . . .	25
3.6.2	Křivka navýšení . . . . .	26
3.6.3	Křivka ROC . . . . .	26
3.7	Testování obchodního systému . . . . .	27
3.7.1	Vhodné ukazatele . . . . .	27
<b>4</b>	<b>Systém pro testování obchodních dat</b>	<b>28</b>
4.1	Neformální analýza a specifikace požadavků . . . . .	28
4.2	Návrh a implemetace . . . . .	29
4.2.1	Návrh aplikace . . . . .	29
4.2.2	Přijatá zjednodušení . . . . .	30
4.2.3	Načítání dat . . . . .	30
4.2.4	Formát CSV . . . . .	31
4.2.5	Předzpracování dat . . . . .	31
4.2.6	Implementace obchodní strategie . . . . .	31
4.2.7	Příprava testování . . . . .	32
4.2.8	Průběh testování . . . . .	32
4.2.9	Výsledky testování . . . . .	33
4.3	Použité technologie . . . . .	33
4.3.1	Technologie .NET . . . . .	33
4.3.2	Jazyk C# . . . . .	33
4.4	Použité knihovny . . . . .	33
4.4.1	Neuronové sítě . . . . .	33
<b>5</b>	<b>Experimenty s aplikací</b>	<b>34</b>
5.1	Implementované obchodní systémy . . . . .	34
5.1.1	Experimenty s numerickými predikcemi neuronové sítě . . . . .	34
5.1.2	Experimenty s klasifikací do tříd neuronové sítě . . . . .	34
5.1.3	Experimenty s klasifikací do tříd pomocí indikátorů . . . . .	35
5.2	Ukázky testování obchodního systému . . . . .	35

5.2.1	Ukázka ukazatelů pro klasifikaci . . . . .	35
5.2.2	Ukázka testů obchodního systému . . . . .	35
<b>6</b>	<b>Závěr</b>	<b>37</b>
	<b>Literatura</b>	<b>39</b>
	<b>Seznam příloh</b>	<b>40</b>
<b>A</b>	<b>Diagram tříd</b>	<b>41</b>
<b>B</b>	<b>Obsah DVD</b>	<b>43</b>
<b>C</b>	<b>Adresářová struktura</b>	<b>44</b>
<b>D</b>	<b>Tvorba vlastních testů a obchodních systémů</b>	<b>45</b>
D.1	Ukázka tvorby vlastního testu obchodního systému . . . . .	45
D.2	Ukázka tvorby vlastního obchodního systému . . . . .	46
D.2.1	Načtení dat . . . . .	46
D.2.2	Model pro předpověď budoucího vývoje trhu . . . . .	46
D.2.3	Rozhodovací strategie . . . . .	47
D.2.4	Výpočet chyby . . . . .	48

# Kapitola 1

## Úvod

Obchod na burze je jedním z nejvýznamnějších způsobů investice volných peněžních prostředků. Přestože v České republice není tato forma investování mezi veřejností příliš rozšířená, movitější subjekty, jako banky a jiné finanční instituce, ji ve velké míře využívají. Vždyť jen průměrný obrat na všech devizových trzích světa je téměř 4 biliony amerických dolarů denně. Prudký rozvoj informačních technologií umožnil výraznou změnu přístupu k burzovnímu obchodu. Převážná většina obchodů je v dnešní době prováděna počítačem za použití automatických obchodních systémů. Tím vzrostly nároky na tyto systémy a jejich testování, které je obvykle prováděno taktéž s pomocí počítače [4].

Tvorba investičních a obchodních strategií na burzovním trhu vyžaduje experty z oblasti matematiky, ekonomie, politologie a dalších oborů. V této práci si klademe za úkol vytvořit metodologii a také počítačový program, který dokáže co nejpřesněji odhadnout výkon a vlastnosti obchodního systému v reálném provozu pomocí jeho předcházejícího testování.

V kapitole 2 se zaměříme na ekonomické souvislosti okolo burz, burzovních obchodů a spekulací a také na jejich účel a přínos. Zmíněny budou i další důležité pojmy z oblasti finančních trhů související s tématem této práce. Těmto souvislostem se bude práce věnovat jen v rozsahu nutném pro potřeby zadání práce. Kapitola se bude dále podrobně zabývat pojmem obchodní systém se zdůrazněním důležitosti testování takového systému. V této kapitole se také čtenář seznámí s metodami analýzy dat, které se v současnosti v obchodních systémech používají. Smyslem těchto metod je získat nové informace o trhu a v nejlepším případě předpověď jeho budoucího vývoje.

V kapitole 3 se zaměříme na testování obchodního systému a na vliv výběru dat a parametrů systému na jeho výkon. Diskutovány budou nejen metody hodnocení obchodního systému, ale také metody hodnocení a porovnávání modelů použitých pro analýzu dat.

Kapitola 4 se bude zabývat návrhem systému pro testování obchodních dat, který dokáže otestovat zvolenou strategii pomocí metodiky prezentované v předcházejících kapitolách. Je zřejmé, že takový systém se neobejde bez použití výpočetní techniky, protože zde bude probíhat množství výpočtů nad velkými objemy dat. Následovat bude popis implementace prezentovaného systému pro testování obchodních dat a seznámení čtenáře s technologiemi a knihovnami použitými při tvorbě této aplikace.

Kapitola 5 nazvaná Experimenty s aplikací se poté zaměří na aplikaci poznatků z této práce v praxi s využitím právě navrženého systému na testování obchodních dat.

V závěru budou zhodnoceny výsledky a také budou diskutovány možnosti dalšího pokračování práce.

## Kapitola 2

# Obchodní systémy

### 2.1 Finanční trh

Znalost finančního trhu v kontextu této práce nám dává přehled o možnostech investování a přerozdělování volných finančních prostředků. Trh obecně můžeme definovat jako místo, kde se setkává nabídka a poptávka po různých produktech. Více o funkcích nabídky a poptávky lze najít v [16].

Finanční trh zabezpečuje pohyb peněz a kapitálu mezi ekonomickými subjekty. Hlavní funkcí trhu je přesun volných finančních prostředků od subjektů s jejich přebytkem k subjektům s jejich nedostatkem. Pořizovaný kapitál můžeme hodnotit dle rizika, likvidity a výnosnosti, přičemž tato kritéria jsou navzájem propojena. S rostoucí výnosností se obvykle zvyšuje také riziko.

Dle faktu, zda se jedná o prvotní prodej instrumentu finančního trhu, můžeme finanční trh dělit na primární a sekundární, či dle času vypořádání na promptní a termínované trhy. Nejčastějším rozdělením finančního trhu je dělení dle účelu, doby splatnosti, rizika, výnosů a obchodovaných instrumentů:

- Peněžní trh se zabývá krátkodobými úvěry a krátkodobými cennými papíry, tedy opcemi, šeky, směnkami, vládními dluhopisy apod.
- Kapitálový trh se zabývá cennými papíry, majetkovými cennými papíry a deriváty. Do těchto skupin řadíme například akcie, hypotéky, obligace nebo komodity.
- Úvěrový trh se zabývá půjčkami a úvěry.
- Devizový trh se zabývá obchodem s devizovými prostředky, jimiž jsou převážně měny jiných států.
- Trh s drahými kovy se zabývá obchodem s drahými kovy, tedy převážně se zlatem a stříbrem.

Podrobnější informace lze dohledat například v [23, 6, 5].

#### 2.1.1 Burza

Burza je instituce, která má na starosti organizaci trhu s investičními nástroji. Účastníkům obchodů burza poskytuje nejen potřebné informace, ale především záruku bezpečnosti investic a plnění závazků. Burzovní obchod je přesně vymezen platnými zákony v daném



státě a na jejich dodržování dohlíží kontrolní orgány burzy. K uzavírání obchodů na burze jsou zpravidla oprávněni pouze členové burzy.

*„Členem burzy může být jen obchodník s cennými papíry nebo zahraniční osoba s povolením k poskytování investičních služeb, která splňuje podmínky stanovené Burzovními pravidly. Mezi další povinnosti patří členství v Garančním fondu burzy (GFB), pouze tehdy je člen oprávněn nakupovat a prodávat cenné papíry“ [9].*

Burza určuje u každého obchodovaného instrumentu cenu dle pravidel tzv. oboustranné aukce. Tedy dle nabídky emitentů a poptávky investorů. Takto stanovená cena se nazývá kurz. Informace o aktuálním kurzu každého instrumentu je veřejně přístupná stejně jako další informace, například objem obchodů. Obchod na burze zpravidla probíhá v přesně určenou denní dobu.

Největšími světovými burzami jsou New York Stock Exchange, NASDAQ Stock Market a London Stock Exchange. V České republice je největší burzou Burza cenných papírů Praha [8].

Burzy můžeme dělit dle specializace na určitý druh zboží na komoditní burzy, peněžní burzy, mezi které můžeme zařadit také burzy s cennými papíry, dále na burzy s finančními deriváty, devizové burzy a burzy služeb. Další informace o burzách jsou například v [23, 10].

Můžeme také zmínit, že k burze existuje alternativa v podobě mimoburzovního trhu.

*„Organizátorem mimoburzovního trhu je akciová společnost, která na základě povolení Komise pro cenné papíry organizuje trh s investičními nástroji“ [3].*

Ten poskytuje nižší stupeň regulace, možnost obchodovat více subjektům, více obchodovaných produktů a často nižší poplatky při obchodování. Více například v [22].

### 2.1.2 Instrumenty finančního trhu

Mezi nejčastěji obchodované instrumenty patří úrokové míry, akcie a akciové indexy, komodity (obilí, kukuřice, ropa atd.), energie, kovy a cizí měny [19]. Každý z těchto instrumentů vyžaduje rozdílné analýzy a obchodování s nimi se řídí rozdílnými pravidly.

### 2.1.3 Rozdíly mezi jednotlivými trhy

Jedním z prvních důležitých rozhodnutí obchodníka je, na jakých trzích obchodovat. Každý trh má rozdílné vlastnosti a dokonce každý instrument v rámci jednoho trhu má svá specifika.

Při výběru trhu je potřeba se zaměřit na vlastnosti jako likvidita, volatilita, míra ovlivnění fundamentálními zprávami, dostupnost, náklady, poplatky nebo doba, ve kterou se na daném trhu obchoduje. Další informace o instrumentech finančního trhu jsou například v [6, 7].

### 2.1.4 Pákový efekt

Pomocí pákového efektu může obchodník dosáhnout násobení zisků a ztrát. Jedná se o vypůjčení finančních prostředků k zvýšení potenciální návratnosti investice. Spekulant může po složení zálohy disponovat množstvím instrumentu větším, než by za tuto zálohu sám nakoupil. Například po složení zálohy asi 20 dolarů lze spekulovat na vývoj ceny jedné

trojské unce zlata, která má cenu okolo 430 dolarů. Při změně kurzu na 431 dolarů by tedy spekulant vydělal jeden dolar.

Díky tomuto fenoménu může spekulant vydělat nebo prodělat na změně kurzu několika-násobně více než při klasickém investování. Výše zmiňované zálohy závisí na obchodovaném instrumentu a na ekonomické situaci. Zajímavou možností je také možnost spekulace na pokles kurzu, tedy prodej instrumentu bez toho, aby spekulat nějaký vlastnil. Těchto možností lze využít při obchodování s finančními deriváty, mezi které patří například komoditní futures nebo opce.

### 2.1.5 Zprostředkování obchodu

Obchodníkovi je obchod na burze zprostředkován brokerskou (makléřskou) společností. Za zprostředkování jsou účtovány poplatky, které jsou stanoveny dle obchodovaného instrumentu a jsou důležitým faktorem při obchodování. Přesnou aktuální výši poplatků lze nalézt v ceníku služeb jednotlivých brokerských společností.

Brokerskou společnost obchodník informuje o požadovaném výkonu pomocí obchodních příkazů. Tyto příkazy bývají přesně definovány a jejich popis lze dohledat v popisu služeb jednotlivých brokerů. Zmíňme příkazy používané v aplikaci, která je součástí práce.

- Příkaz Market říká brokerovi, že má nakoupit nebo prodat instrument okamžitě za momentální cenu. Cena zde není omezena, a tak je příkaz proveden hned, jak je to možné a to s aktuálním kurzem. Většina společností nabízí také alternativy tohoto příkazu, ve kterých může obchodník stanovit cenové rozpětí, v jakém je ochoten provedení příkazu akceptovat.
- Každým obchodníkem používaný je také příkaz Stop-loss, který je základní formou ochrany proti riziku a přílišné ztrátě. Příkaz typu Stop-loss je nastaven na určitou hodnotu kurzu a pokud se kurz sledovaného instrumentu vyrovná této hodnotě, příkaz Stop-loss se mění na příkaz typu Market a je proveden. V praxi to znamená, že po nakoupení instrumentu příkazem Market pokládá obchodník příkaz Stop-loss na nižší úroveň kurzu, než byla při provedení příkazu Market. Na jakou úroveň obchodník příkaz stanoví, závisí na jeho uvážení. Musí zde počítat s mnohými faktory v čele s volatilitou trhu.

Dále budou v práci použity příkazy, které nejsou přímo obchodními příkazy, ale jsou složeny z více příkazů. Jejich názvy nejsou jednotné, a proto si je zvolíme pro následné odkazování na ně v dalších částech práce.

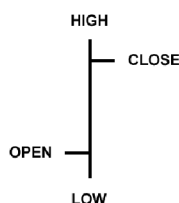
- Pohybující se Stop-loss se používá pro kontrolu zisku posouváním Stop-lossu při očekávaném vývoji kurzu.
- Last-trading-day-order je proveden při dosažení určitého času. Příkaz je používán například při obchodu s komoditami, kdy se obchodník chce vyhnout převzetí fyzické podoby komodity.
- Příkaz Profit-target bývá nastaven na požadovanou úroveň kurzu, a při jejím dosažení obchodník vystupuje z pozice s dostatečným ziskem.

### 2.1.6 OHLC Data

Data poskytovaná burzami a brokerskými společnostmi bývají často ve formátu OHLC. Tato zkratka vyjadřuje v daném intervalu následující hodnoty.

- Otevírací kurz (open).
- Nejvyšší úroveň kurzu (high).
- Nejnižší kurz (low).
- Uzavírací kurz (close).

Tato data bývají zveřejňována v různých intervalech. Ke každému intervalu se poté váže tato čtveřice hodnot, doprovázená přesným časem a datem, a obvykle také objemem obchodů provedených v tomto intervalu. Obvyklé grafické znázornění zmíněných čtyř hodnot v jednom intervalu můžeme vidět na obrázku 2.1. Zmíňme ještě skutečnost, že kdykoliv bude v této práci uvedena v daném intervalu pouze jedna hodnota, budeme uvažovat uzavírací hodnotu kurzu.



Obrázek 2.1: Obvyklé znázornění OHLC dat jednoho intervalu v grafu

Historická data pro jednotlivé instrumenty finančního trhu s intervalem jednoho dne jsou obvykle volně dostupná na internetu. Přesnější data s kratšími intervaly či okamžitá aktuální data lze obvykle získat za poplatek.

### 2.1.7 Likvidita

V oblasti finančních trhů a burzovního obchodu je likvidita úzce spjata s objemem obchodů (volume). V podstatě to znamená, že čím likvidnější je trh a tedy čím více obchodů v něm probíhá, tím rychleji dojde k vyřízení podaného příkazu. V likvidních trzích dochází k vyřazení příkazu téměř okamžitě a s kurzem stejným nebo blízkým kurzu aktuálnímu při podání příkazu. Naopak v nelikvidním prostředí může nějakou dobu trvat, než bude obchodníkem prodáváný instrument někým zakoupen a naopak. V horizontu této doby může dojít ke změně kurzu. Tento rozdíl mezi hodnotou kurzu při podání příkazu a jeho vyřízením se nazývá slippage.

Hodnota objemu obchodů bývá považována za důležitý ukazatel a je obecně uznávanou teorií, že změny kurzu instrumentu bývají doprovázeny změnou objemu obchodů. Tohoto faktu využívají analýzy zmíněné v podkapitole 2.2.1.

### 2.1.8 Volatilita

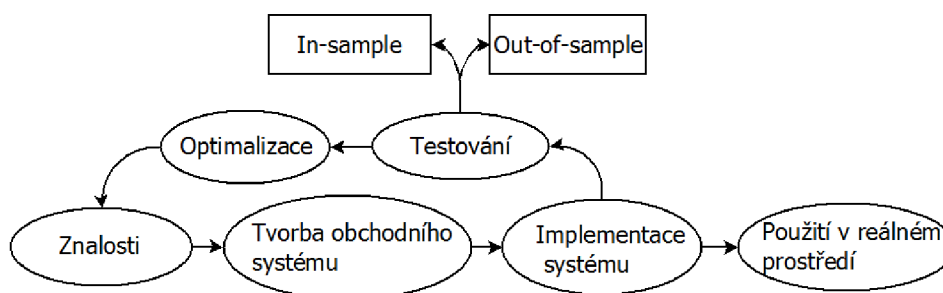
Pojem volatilita vyjadřuje míru změny kurzu aktiva nebo jeho výnosové míry. Volatilita souvisí s rizikovostí investice. Vysoká volatilita poskytuje možnost vyšších výnosů, ale za cenu vyššího rizika. V praxi volatilita znamená, že dochází k častým a nezanedbatelným změnám kurzu, které jsou často nezávislé na dlouhodobém trendu vývoje kurzu. Vysoce volatilní jsou například akcie firem ve státech s rozvíjející se ekonomikou.

## 2.2 Obchodní systém

Obchodní systémem představuje množinu pravidel, kterými obchodník řídí svoje akce na trhu. Systém zahrnuje nejen signály ke vstupu do trhu, signály pro výstup z trhu, chování na trhu, ale také volbu instrumentů a diverzifikaci portfolia.

Důležité je zmínit, že s rozvojem technologií se burzovní obchod stává stále více závislým na počítačích a výpočetní technice, a také obchodní systémy využívají stále ve větší míře vysoký výpočetní výkon soudobých počítačů.

Tvorba obchodního systému je vysoce subjektivní a pro zajištění jeho efektivity musí být neustále hledány nové způsoby rozpoznání příležitostí a využití dostupných informací. Typický je iterativní vývoj systému znázorněný na schématu 2.2.



Obrázek 2.2: Fáze vývoje obchodního systému

Obchodní systém musí být navržen nejen na základě ekonomických souvislostí, matematických znalostí a modelů trhu nebo zkušeností analytiků, ale také důkladně otestován na historických datech, aby došlo alespoň k částečnému potvrzení ekonomických a matematických teorií použitých při jeho tvorbě.

Obchodní systém se skládá z několika částí, a protože pojem obchodního systému nemůžeme naprosto exaktně definovat, nelze jej ani přesně rozdělit na jednotlivé části. Každý obchodní systém se může zakládat na jiných pravidlech, přesto mají některé části společné. Jmenovitě jsou to části pro analýzu trhu, poziční rozhodování a řízení rizika. Nejobtížněji určitelnou částí systému je analýza trhu, ostatní části systému jsou snadno určitelné na základě pravděpodobnosti a statistiky.

### 2.2.1 Analýza trhu

Úkolem analýzy trhu je předpověď budoucího vývoje trhu a generování signálů pro vstup a výstup z trhu. V kapitole 3 budou prezentovány způsoby, jak tyto metody testovat a optimalizovat jejich výkon. Každý obchodní systém je v tomto ohledu naprosto rozdílný. Někteří obchodníci analyzují trh pouhým pohledem na graf vývoje kurzu, zatímco jiní využívají moderní statistické metody zkoumající vztahy mezi mnoha instrumenty finančního trhu. Tato část bude často dále v textu nazývána model pro předpověď budoucího vývoje kurzu nebo zkráceně model.

Než se začaly používat počítače, obchodníci využívali k obchodování na finančních trzích převážně intuici. S rostoucí důležitostí těchto způsobů investice volných prostředků začali lidé hledat způsoby a metody vedoucí ke zvýšení zisků a minimalizaci rizika. Existuje množství přístupů, teorií a metod, které se snaží modelovat a předpovídat trh. Jejich často spekulativní přesnost lze s jistou mírou nadsázky parafrázovat následujícím výrokem Kena

Oslena, zakladatele společnosti Digital Equipment Corporation, z roku 1977, i když přímo nesouvisí s finančními trhy.

*„Neeexistuje důvod, proč by měli někteří jedinci mít doma počítač“ [2].*

Přestože jsou metody pro předpověď vývoje trhu často nepřesné a existují teorie, které jejich význam naprosto vyvracejí (například teorie efektivních trhů, o níž lze nalézt informace v [21]), jsou metody pro analýzu finančních trhů v literatuře prezentovány jako často používané. Více například v [15, 14, 25].

Analytické metody pro předpověď budoucího vývoje trhu lze rozdělit na technickou, fundamentální a psychologickou analýzu. Ke každému z těchto přístupů existuje velké množství metod, které jsou předmětem zkoumání při vývoji obchodního systému. Obvyklé jsou i metody kombinující více přístupů.

### 2.2.1.1 Technická analýza

Technická analýza předpokládá, že se ve vývoji kurzu objevují rozeznatelné a opakující se obrazce a lze tedy z předchozího vývoje kurzu odhadnout vývoj budoucí. Dle [14] je tento typ analýzy používán až devadesáti procenty významných obchodníků s akcemi. Tento typ analýzy může být automatizován a jeho použití v automatických obchodních systémech je běžné.

S technickou analýzou je spojeno množství indikátorů. Obvykle se jedná o matematickou funkci zvýrazňující některou vlastnost časové řady, tedy například vyhlazení průběhu a odstranění šumu. Tyto indikátory jsou obvykle používány jako součást komplexnějšího přístupu. Indikátory lze dělit například na klouzavé průměry, oscilátory, objemové indikátory, sentiment indikátory a indikátory šíře trhu. Více o indikátorech lze dohledat například v [18, 26, 20].

$$SMA_t = \frac{C_t + C_{t-1} + C_{t-2} + C_{t-3} + C_{t-4}}{5} \quad (2.1)$$

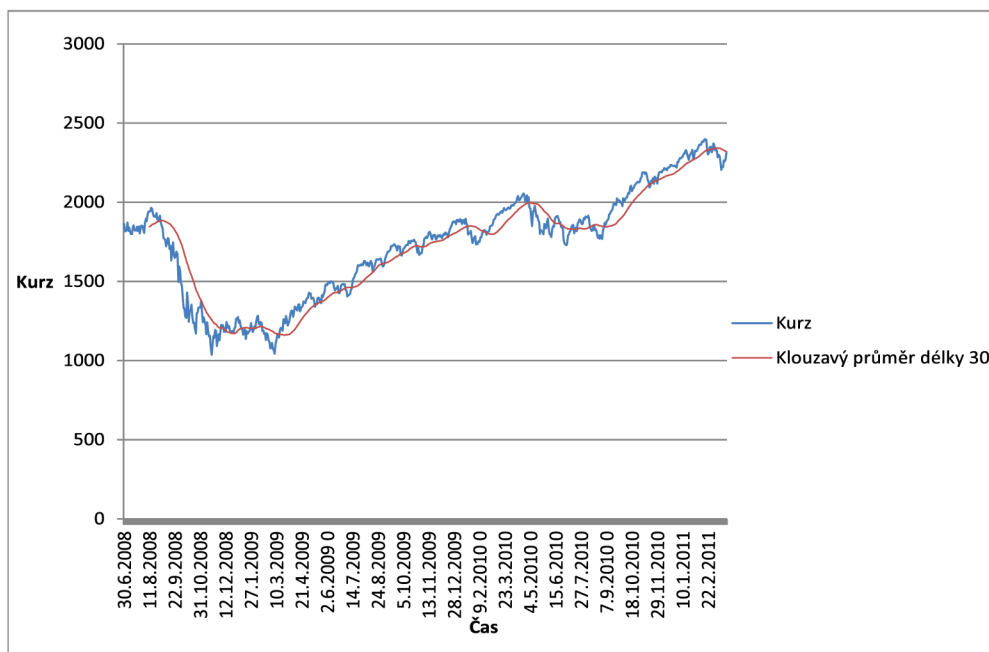
Mezi nejpoužívanější patří klouzavý průměr (moving average). Výpočet hodnoty jednoduchého klouzavého průměru délky 5 (průměr je vypočítán z pěti intervalů) v čase  $t$  popisuje vzorec 2.1. Hodnota  $C$  vyjadřuje kurz použitý k výpočtu klouzavého průměru v čase  $t$ .

*„Když se uzavírací cena dostane nad úroveň klouzavého průměru, je generován signál k nákupu“ [14].*

Na grafu 2.3 je vidět klouzavý průměr délky 30 aplikovaný na vývoj kurzu indexu Nasdaq.

### 2.2.1.2 Fundamentální analýza

Fundamentální analýza je hloubkovou analýzou instrumentu, jeho ekonomického pozadí, ziskovosti, substitutů, vztahů s jinými instrumenty, nabídky a poptávky a dalších kurzotvorných faktorů. Bohužel vliv všech faktorů ovlivňujících kurz akcie je obtížně formalizovatelný a převoditelný do automatizované podoby a interpretace těchto znalostí může být subjektivní. Přesto existují automatické systémy, které například extrahují klíčová slova z článků v ekonomických periodikách.



Obrázek 2.3: Klouzavý průměr

### 2.2.1.3 Psychologická analýza

Psychologická analýza se snaží odhadnout chování lidí dle aktuální situace na trhu. Zabývá se psychologií davu a zkoumá příčiny lidského chování vedoucí k nákupu nebo prodeji na daném trhu.

### 2.2.2 Poziční rozhodování

Na základě výsledků provedených analýz se obchodník rozhoduje o dalších akcích. Tato etapa je opět velice subjektivní. Takové rozhodnutí může znamenat vstup do pozice (obvykle nákup instrumentu) nebo naopak vystoupení z pozice (obvykle prodej). Tuto část obchodního systému budeme nazývat pro potřeby práce rozhodovací strategií nebo pouze strategií.

### 2.2.3 Money management, řízení rizika a psychologie

Ani nejlepší metody nedokáží dlouhodobě předpovídat budoucí vývoj s pravděpodobností o moc vyšší než 50 %, jsou proto důležité i další části obchodního systému.

*„úspěšné obchodování je 60 % psychologie, 30 % money-management a pouze 10 % vstupní strategie“ [25].*

Psychologií je zde myšlena náchylnost lidí na psychologický tlak, který vzniká když se obchodní systém potýká se sérií ztrát. Komplexní obchodní systém by měl počítat i s touto variantou a také očekávat, že obchod bude úspěšný jen s určitou pravděpodobností (často menší než 50 %).

Důležitou charakteristikou je money-management. Tento pojem vyjadřuje nakládání s financemi v obchodním systému. I systém s méně než 50 % ziskových obchodů může dlouhodobě generovat zisk. Obchodníci v tomto kontextu zavádějí pojem „Poměr riziku



a zisku“ (Risk reward ratio, RRR), který vyjadřuje poměr průměrného zisku ze ziskového obchodu a průměrné ztráty ze ztrátového obchodu. Velikost ztráty lze totiž korigovat například pomocí Stop-lossu a výši zisku například vhodným vystoupením z trhu, případným přikoupením instrumentu nebo posouváním Stop-lossu ve směru vývoje trhu. Obchodník samozřejmě hledá trh, kde bude tento poměr co nejvyšší, díky čemuž může následně přistoupit na nižší procento úspěšnosti obchodů.

## 2.3 Dělení obchodních systémů dle doby držení pozice

Existuje několik přístupů k obchodování, které se liší dobou trvání obchodů. Každý z těchto přístupů vyžaduje odlišný obchodní systém. Obchodníky můžeme rozdělit na intradenní, kteří nenechávají probíhající obchody otevřené přes noc, iterdenní (poziční), jejichž obchody trvají v řádu dní a dlouhodobé investory. Relativně novým přístupem je vysokofrekvenční obchodování. Při tomto přístupu dochází k velkým objemům obchodů, ale obchodník je v pozici jen zlomek vteřiny. Změna kurzu o setiny dolaru dokáže při velkém objemu obchodu vytvořit velké zisky, ale i velké ztráty. Tyto obchodní systémy jsou obvykle automatické a zásadní je vliv technického vybavení, výpočetní techniky a připojení k síti. Dle [24, 4] je více než 70 % z objemu obchodů na finančních trzích ve Spojených státech amerických realizováno právě tímto přístupem.

## 2.4 Dělení obchodních systémů dle přístupu k rozhodování

### 2.4.1 Diskreční přístup

Přístup je založený na zkušenostech obchodníka, jeho analýzách, intuici a momentálním rozhodnutí. Velkým mínusem je předešlá náročnost na psychiku obchodníka, protože pod tíhou ztráty peněz může docházet k dalším chybám vedoucím k vyšším ztrátám nebo k promeškání obchodu, který by vedl k zisku. Jako odstrašující případ jmenujme obchodníka Nicholase Leesona, který se snažil zvrátit finanční ztrátu, kterou vytvořil při nákupu indexu Nikkei a jeho následném propadu po zemětřesení 17. ledna 1995 dalšími riskantními obchody. Ztráta, kterou Leeson způsobil v několika dnech, byla asi 1,4 miliardy dolarů. Více o tomto incidentu lze dohledat v [1].

### 2.4.2 Automatické obchodní systémy

Do této kategorie jsou zahrnuty obchodní systémy s přesně určenými pravidly vstupu do obchodní pozice, pohybu v trhu i výstupu z trhu, ať už jsou obchody prováděny obchodníkem nebo přímo počítačem. Postupuje se přesně podle předem určeného algoritmu, ve kterém jsou ošetřeny všechny situace, které mohou nastat a nedochází tak k chybným rozhodnutím. Obtížným úkolem je tvorba takového systému, který by i přes ztráty dlouhodobě generoval zisk.

Velkou výhodou je také právě fakt, že pravidla jsou přesně určena, a je proto možné je algoritmizovat a zapsat formou počítačového programu. Podíl obchodních příkazů podaných automaticky přímo počítačem byl dle [4] v roce 2008 odhadován v některých trzích až na 80 %.

Právě testováním algoritmizovatelných obchodních systémů se tato práce zabývá.

## 2.5 Použití neuronových sítí

Pro předpověď vývoje burzovního trhu je v praxi používáno mnoho metod známých z informačních technologií. Jejich použití je možné ve všech typech analýz, prezentovaných v podkapitole 2.2.1. Počínaje jednoduchými programy ke tvorbě grafů a konče expertními systémy, rozhodovacími stromy, metodami známými z oblasti dolování dat nebo fuzzy logikou. Dle [21] v této oblasti analýz dosáhly v posledních letech vysoké popularity neuronové sítě. Jejich schopnost generalizace a modelování nelineárních vztahů ve vstupních datech z nich dělá ideální nástroj pro popis nelineárních systémů jako jsou burzovní trhy. Jsou používány pro doplnění jiných metod nebo samostatně. Jejich případnou alternativou s podobnými vlastnostmi jsou metody známé ze statistiky, například metody ARMA, ARIMA nebo ARIMA-GARCH [13].

V literatuře je prezentováno velké množství různých neuronových sítí použitých k analýze trhu, z nichž mnohé používají jako vstup také odvozená data získaná technickou nebo fundamentální analýzou. [21, 15].

### 2.5.1 Přehled

Neuronová síť je výpočetní model inspirovaný strukturou a funkcionalitou biologických neuronových sítí. Skládá se ze vzájemně propojených neuronů. Existuje množství typů sítí, z nichž nejčastěji zmiňovanou pro aplikaci v analýze trhu je vícevrstvá perceptronová síť, ve které jsou neurony uspořádány do vrstev, přičemž mezi neurony jedné vrstvy nejsou vazby, ale každý neuron jedné vrstvy je propojen se všemi neurony vrstvy sousední. Mezi vstupní a výstupní vrstvou existuje libovolné množství skrytých vrstev. Obvykle jsou neuronové sítě používány k hledání komplexních vztahů v datech a hledání vzorů v datech.

### 2.5.2 Příprava sítě

Neuronová síť musí být před použitím trénována. Tento proces zahrnuje nastavování vah vazeb mezi neurony dle učícího algoritmu. Nejdůležitější pro tento proces je vybrat vhodná data a učící algoritmus.

### 2.5.3 Vstupní data

Pravděpodobně nejdůležitější při tvorbě neuronové sítě je vyřešit otázku, jaká budou vstupní data. Určení, které veličiny budou použity, je obtížným problémem a vyžaduje zkušeného ekonomického analytika.

Vstupy mohou být surová a nezpracovaná data, jako například objem obchodů, kurz, změna kurzu za předchozí časový interval nebo i odvozená data, jako indikátory technické analýzy nebo určitým způsobem zpracovaná data fundamentální analýzy.

Jako příklad vysoce komplexní neuronové sítě můžeme uvést síť, známou jako JSE-system, která byla použita k modelování Johannesburgské burzy [14]. Tato síť měla celkem 69 vstupních veličin a snažila se vyhodnotit kompletní přehled o dění na burze. Z toho 3 vstupní veličiny byly výstupy fundamentální analýzy (výnos, kurzy, objem obchodů) a 17 vstupů byly výstupy technické analýzy (různé klouzavé průměry atd.). Dalšími vstupy byly vlastní ukazatele, mezinárodní ukazatele (Index Dow Jones atd.), ceny zlata a kurzy vybraných měnových párů, úrokové míry a některé ekonomické statistiky (export, import atd.). Zajímavé je, že analýzy této sítě ukázaly, že 20 z těchto vstupů nemělo žádný vliv na výkon sítě. Typy analýzy, kterými můžeme podobného zjednodušení dosáhnout budou

prezentovány dále v této práci. Jednodušší sítě používají jako vstupní data například jen předchozí hodnoty kurzu instrumentu a objem obchodů. Volnost volby vstupních dat je obrovská a záleží pouze na znalostech analytika, jaká data použije.

#### 2.5.4 Trénink sítě

Při učení síť upravuje váhy spojení mezi neurony po porovnání svého výstupu s ideálním tak, aby došlo k minimalizaci chyby. Tento přístup se nazývá učení s učitelem. Pro výpočet chyby se používá střední kvadratická chyba, jejíž výpočet je založen na druhé mocnině rozdílu mezi skutečným a očekávaným výstupem sítě [11].

$$Err(w) = \frac{1}{2} \sum_{v \in \text{vystupy}} (y_{i,v} - \hat{y}_{i,v})^2 \quad (2.2)$$

V literatuře [21] bývá pro finanční sféru nejčastěji zmiňováno použití vícevrstevných perceptronových sítí trénovaných algoritmem zpětného šíření chyby (backpropagation). Protože výstupní vrstva jako jediná může porovnávat s ideální očekávanou hodnotou, chyba se šíří sítí od výstupní ke vstupní vrstvě a dochází k úpravě vah spojení. Po ukončení trénování jsou tyto váhy zafixovány. Svoji roli zde hraje také aktivační (přenosová) funkce. Pro potřeby práce jen zmíníme, že existuje množství používaných přenosových funkcí, například sigmoidální funkce nebo funkce hyperbolické tangenty. Často je také potřeba normalizovat vstupy neuronové sítě například do rozmezí mezi 0 a 1. Normalizaci do potřebného rozsahu provedeme následovně.

$$\text{normalizovanaHodnota} = (\text{vstup} - \text{min}) / (\text{max} - \text{min}) \quad (2.3)$$

Hodnota vstup vyjadřuje číslo, které budeme normalizovat a min a max jsou nejnížší respektive nejvyšší číslo z těch, které budeme normalizovat. Analogický je i případný převod zpět na původní hodnoty.

$$\text{puvodniHodnota} = \text{vstup} \times (\text{max} - \text{min}) + \text{min} \quad (2.4)$$

Trénování je ukončeno, až když považujeme chybu sítě za dostatečně malou nebo proběhl dostatečný počet iterací (epoch). Ukončení trénování ve správnou chvíli je nejdůležitějším rozhodnutím v trénovacím procesu. Při přílišném tréninku může dojít k přetrénování sítě, kdy je síť naučena na příklady, na kterých se učila, ale ztrácí schopnost generalizace a její výkon na nových příkladech není dobrý. Právě schopnost generalizace je nejvíce ceněnou vlastností neuronových sítí. Přetrénování se lze vyhnout vhodným testováním, které bude předvedeno v kapitole 3.

Dále musíme určit jaká data a jaké množství dat síti prezentovat. Větší množství vzorků nemusí být v tomto případě vždy výhodnější. Některé v praxi použité sítě byly naučené i na data za posledních 50 let [21]. Jelikož je burzovní obchod velice dynamicky se vyvíjející odvětví, některé principy a zákonitosti, které platily dříve, již dnes platit nemusí. Síť by se tak naučila vzory a chování, které již není důležité nebo je úplně bezcenné. Množství dat má také vliv na rychlost trénování sítě. Trénink je výpočetně velice náročný proces, jehož doba trvání závisí na úrovni natrénování sítě, množství vzorků dat a jejich velikost nebo množství neuronů a skrytých vrstev. Podle [21] jsou známy případy sítí, jejichž trénink trval i několik dní.

### 2.5.5 Struktura sítě

Nejčastěji jsou podle [21] v oblasti burzovního obchodu využívány právě neuronové sítě fungující na principu zpětného šíření chyby.

Setkáváme se i s dalšími přístupy jako například v [21] zmiňované neuronové sítě spolupracující s fuzzy logikou, neuronové sítě navržené s pomocí genetických algoritmů, modulární sítě složené z více neuronových sítí, kde každá je trénována na jiných datech, rekurentní sítě se schopností pamatovat si předchozí vstupy, samoorganizující se systémy nebo neuronové sítě spolupracující s expertním systémem. Volba vhodného přístupu a parametrů sítě je opět obtížným úkolem a vyžaduje mnohé analýzy a testování.

### 2.5.6 Výkon neuronových sítí

Dle [21], neuronové sítě vycházejí velice úspěšně ze srovnání s moderními statistickými metodami a metodami regresní analýzy a jejich procento úspěšnosti předpovědi je vyšší. V některých testech vykazují neuronové sítě úspěšnost předpovědi vývoje trhu vyšší než 90 %. Příkladem takové sítě je výše zmiňovaný JSE-system. Bohužel není nikde uvedeno, za jakých podmínek a na jakých datech byly provedeny testy vedoucí k těmto výsledkům. Více o problematice neuronových sítí lze nalézt například [11, 12].

## Kapitola 3

# Testování automatických obchodních systémů

Tato práce se nezabývá tvorbou obchodního systému, ale jeho testováním. Pro úplné ověření systému jsou potřeba znalosti z několika disciplín, a to hlavně z matematiky a ekonomie. Testováním systému není myšleno jen otestovat navržený systém na historických datech burzy a zhodnotit jeho funkčnost. Diskutován bude také vliv parametrů systému a testovací množiny dat na odhad jeho výkonu na datech jiných.

Přestože přesné určení výkonu systému v reálném prostředí není možné, pomocí vhodného testování na různých testovacích množinách a pomocí úprav parametrů systému lze docílit zvýšení pravděpodobnosti úspěšnosti systému. Ani důkladně otestovaný obchodní systém tedy nemusí kvůli náhodným vlivům a změnám v trhu prosperovat v reálném tržním prostředí.

Obchodní systém se většinou skládá z více částí, ale většinou je velice důležitý odhad budoucího vývoje trhu. Na tuto část systému se zaměříme nejvíce. Ostatní části jako například již zmiňovaný money-management jsou relativně přímočaré a k jejich ověření stačí ukazatele prezentované v podkapitole 3.7.

Část obchodního systému zabývající se odhadem vývoje trhu může být, jak už bylo zmiňováno, založena na různých přístupech. Například na neuronových sítích, rozhodovacích stromech, matematických metodách popisu časových řad a dalších. Rozdílné jsou také vstupy a výstupy těchto přístupů v případě každého obchodníka či analytika. Mnohé z následně prezentovaných metod v podkapitolách Ukazatele pro klasifikaci do tříd a Ukazatele pro numerické predikce jsou používány v oboru dolování dat, ale často mohou být použity i pro modely nespádající přímo do této kategorie. Tyto informace jsou čerpány převážně z [11, 12]. Dále je nutné zmínit, že toto jsou jen některé z velkého množství ukazatelů, z nichž množství může být specifických pro jeden určitý model.

Je potřeba si také všimnout, že testy modelu pro předpovídání vývoje či rozhodování o nákupu nebo prodeji nemají stejný význam jako testy celého systému. Zatímco neprovedený možný ziskový obchod se na výsledcích testů celého systému nepromítne, na testech modelu pro předpovídání uvidíme, že odhad byl proveden nesprávně a můžeme se pokusit o kalibraci tohoto modelu. Každý z těchto přístupů má své přínosy a omezení, a proto se zaměříme na oba.



## 3.1 Metody testování

Předem musíme uvést, že model není možné úplně verifikovat a testovat jen z pohledu určitého intervalu historického vývoje. Důležitá je také znalost modelované řady, v našem případě finančního trhu. Expert sestavující model může použít proměnné, které podle ekonomických teorií mají na pohyb kurzu vliv. V datech, na kterých je model sestaven nebo testován, tato proměnná vliv na pohyb kurzu nevykazuje a vypuštění proměnné z modelu nemá na jeho chování téměř žádný vliv. Přesto tato proměnná vypuštěna být nemůže, protože prokazatelně existuje její vliv na trh. Správný model tedy musí být založen i na ekonomických teoriích, ne jen na přesnosti aproximace historických dat.

### 3.1.1 Zpětné testování

Pojem zpětné testování (backtesting) se používá k vyjádření aplikování pravidel obchodního systému na historická data. Obchodní systém tak může být otestován, aniž by docházelo k riziku ztráty peněz. Zpětné testování bývá často využito k účelu nastavení parametrů systému, což ale, jak uvidíme dále v této kapitole, nemusí být vhodný přístup.

### 3.1.2 Dopředné testování

Dopředné testování (forward testing) se používá ve významu testování obchodního systému na nově se objevujících datech. Jde tedy o simulaci obchodování za reálných podmínek na aktuálních datech, ale bez reálné finanční investice. Všechny provedené obchody jsou zaznamenány stejně, jako by tomu bylo při reálném použití testovaného obchodního systému na burze, ale žádný obchod není ve skutečnosti proveden. Bohužel důkladné otestování systému tímto způsobem trvá dlouhou dobu, protože nová data se objevují v daných časových intervalech, a čekání na dostatečné množství dat je velice zdlouhavé.

## 3.2 Metody porovnání výsledků

V následujících podkapitolách bude často používán pojem sestavení modelu. Sestavením modelu v tomto kontextu myslíme nastavování, ladění a optimalizaci parametrů systému. Některé modely musí být nejdříve natrénovány nebo musí proběhnout jiné výpočty (výpočet metody aproximace časové řady atd.). Tyto výpočty mohou být provedeny na jiných datech, než na kterých je model následně optimalizován. Volba těchto dat je obtížným problémem a pro účely práce budeme uvažovat provedení těchto výpočtů na stejných datech, jako je provedena ona optimalizace. V této podkapitole se zaměříme na volbu množin dat k sestavení modelu a jeho testování.

### 3.2.1 In-sample

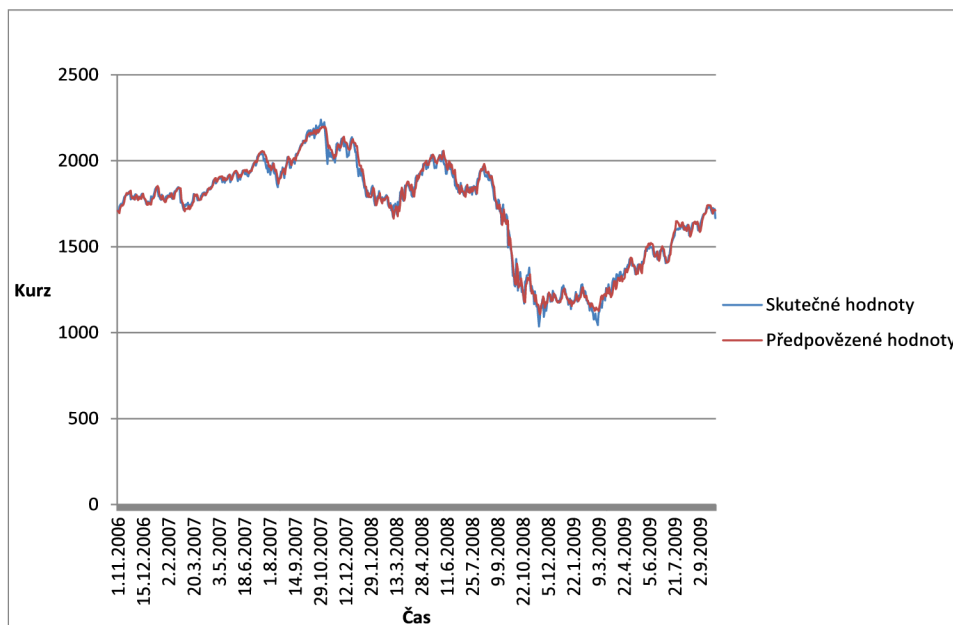
Pokud je model sestaven na vzorku dat a na těch samých datech je posléze testována jeho předpověď schopnost, nazývá se tato předpověď in-sample. Potom tato data, na kterých byl model sestaven i otestován, nazýváme in-sample data a procesu testování na těchto datech analogicky testování in-sample. Upozorníme jen na to, že pokud je provedeno například trénování neuronové sítě a její následné testování a optimalizace parametrů na jiné množině dat, než na které byla trénována, bude tato testovací množina taktéž in-sample.



### 3.2.2 Out-of-sample

Název out-of-sample se používá vždy, pokud je model sestaven na jedné množině dat a poté testován na jiné. Tato množina použitá k testování se nazývá out-of-sample a analogicky proces testování na těchto datech se nazývá testování out-of-sample.

Neznamená to ale, že by analytik musel čekat na nová data, aby mohl systém podrobit out-of-sample testům pro odhad správnosti modelu a jeho případnou kalibraci. Efektu out-of-sample množiny se dá dosáhnout i vhodným rozdělením dat, která máme k dispozici. Množinu dat, na které bude model sestaven, budeme nazývat trénovací a množinu, na které bude model testován budeme nazývat testovací.



Obrázek 3.1: Předpovědi in-sample

### 3.2.3 Vliv volby dat na výsledky

Out-of-sample testy mají výrazně lepší vypovídající schopnost o kvalitě modelu než testy in-sample. Je ale potřeba si dát pozor na strukturu obou množin, protože se může stát, že každá z nich bude obsahovat výrazně rozdílné vzorky dat. Out-of-sample předpovědi zpravidla také nejsou tak přesné, jak by mohlo být očekáváno při testování modelu in-sample.

Na grafech 3.1 a 3.2 je vidět rozdíl v přesnosti předpovědi neuronové sítě. Na obou grafech jsou předpovědi téže sítě a na stejných datech (index Nasdaq od listopadu 2006 do září 2009). Ale zatímco v prvním byl model sestaven na stejných datech, v druhém na jiných. U out-of-sample testů je především v některých úsecích grafu patrná značná odchylka od skutečných hodnot. Zajímavá je také informace, že zatímco obchodní systém obchodující na základě těchto předpovědí v případě in-sample testů vykazoval zisk, v případě out-of-sample naopak nezanedbatelnou ztrátu. Je tedy vidět, že přestože i out-of-sample předpovědi ve většině případů kopírují skutečný trend, odchylky jsou poměrně velké. Více o významu a analýze odchylek předpovědi bude zmíněno později v této kapitole. Můžeme ale už nyní vidět význam out-of-sample testování a také testování jak vlastností modelu

určeného k předpovídání budoucího vývoje tak i obchodního systému jako celku.



Obrázek 3.2: Předpovědi out-of-sample

### 3.2.4 Curve-fitting

Při sestavování modelu hrozí výrazné riziko tzv. curve-fitting (někdy také over-fitting, pře-učení, přílišná optimalizace). Tento jev nastává při přílišné optimalizaci a přizpůsobení modelu jedné množině dat. Takto optimalizovaný model vykazuje výborné výsledky na této množině (tedy in-sample). Je ale přizpůsoben datům a šumu v této množině, takže ztrácí schopnost generalizace. Tento model se pak ukazuje velice nevhodný k použití na jiné množině dat.

Proto je tedy důležité zakládat model na matematických a ekonomických teoriích a ne jen na určitém vzorku dat, aby byla zaručena obecnost modelu. Tomuto chování se lze vyhnout vhodným testováním na out-of-sample datech.

### 3.2.5 Porovnání modelů

Při porovnávání modelů pro předpovídání je důležité porovnávat je ve shodných podmínkách. Naštěstí v literatuře (například [13, 12]) se obvykle setkáváme s porovnáváním modelů na testovací množině. Je ale poměrně snadné „vylepšit“ metodu, například přizpůsobením předpovědicích schopností modelu právě této specifické testovací množině dat nebo vyzkoušením několika modelů s různými parametry a použití k předpovědi toho modelu, co vykazuje na testovací množině nejlepších výsledků, přestože na jiné by nemusel.

Ideálním přístupem je tedy sestavit model dle teoretických podkladových znalostí a testovat jej na jedné množině a při případném objevení chyby v teoretických předpokladech nebo pouze snaze o optimalizaci a vylepšení výsledků model upravit a testovat opět na stejné množině. Po dokončení sestavování jej nakonec otestovat na jiné množině dat. Tímto získáme plnohodnotné out-of-sample testování a výsledky budou více vypovídající o skutečných vlastnostech a výkonu modelu.

### 3.3 Metody výběru dat

Pro testování se nabízí celá řada variant podle toho, jaká data použijeme pro sestavení a jaká pro testování.

Nejjednodušším způsobem je testování in-sample a následné ověření na out-of-sample datech. Otázkou je, jak taková data získat. Jedním problémem může být, že dat je k dispozici málo. Jiným problémem je, že by se vlastnosti dat použitých pro testování podobaly vlastnostem dat trénovacích. Oba problémy umožní řešit různé způsoby výběru trénovacích a testovacích dat.

Některé z následně prezentovaných metod nelze použít na testování automatického obchodního systému jako celku, ale pouze na algoritmus předpovídající budoucí vývoj. Metody totiž předpokládají rozdělení množin na jednotlivé příklady, ale na otestování celého systému potřebujeme těchto příkladů výrazně větší množství než pouze jeden. Přesto mohou sloužit k validaci některých rozhodovacích přístupů, například neuronových sítí nebo rozhodovacích stromů.

#### 3.3.1 Rozdělení dat

První možností je rozdělení dat na dvě části. Jedna bude použita k sestavení modelu a druhá k jeho otestování. Tato metoda může být nevypovídající, pokud mají data v obou částech výrazně rozdílné charakteristiky. Bohužel literatura se nezmiňuje o žádné metodě určení poměru velikostí těchto dvou množin. Dle [11] je doporučený poměr mezi testovacími a trénovacími daty 75 % dat pro sestavení a 25 % pro testování. V [13] je k problematice doporučeno dle praktických zkušeností rozdělení 90 % dat pro sestavení modelu a 10 % pro testování s dodatkem, že toto rozdělení není založeno na žádných teoretických podkladech.

#### 3.3.2 Křížová validace

V této metodě dochází k rozdělení dat například na 10 částí. Z toho se poté devět desetin použije na sestavení modelu a zbývající desetina k testování. Tento postup opakujeme, aby každá z deseti částí byla testovací a z výsledků uděláme průměr. Tomuto způsobu říkáme desetinásobná křížová validace.

#### 3.3.3 Leave-one-out

Tato metoda je obdobná jako křížová validace, tentokrát jsou ale data rozdělena na tolik částí, kolik existuje v datech příkladů. Jedná se tedy o n-násobnou křížovou validaci. Tato metoda dává odhad, jak by se model sestavený na všech známých datech choval při klasifikaci nových příkladů.

#### 3.3.4 Bootstrap

Tato metoda je rozdílná od předchozích v tom, že se příklady užití pro sestavení modelu mohou opakovat. Dle [11] bude vybráno:

$$\lim_{n \rightarrow \infty} \left(1 - \frac{1}{n}\right)^n = e^{-1} = 0,368, \quad (3.1)$$

tedy 36,8 % příkladů pro testování a 63,2 % příkladů pro sestavení modelu.

### 3.4 Ukazatele pro klasifikaci do tříd

Následující ukazatele taktéž nejsou určeny pro testování obchodního systému jako celku, ale pouze pro ověření jeho části určené pro předpovězení budoucího vývoje. Tato podkapitola je zaměřena na ukazatele, které slouží k testování metod, jež klasifikují do tříd. Pokud tedy například rozhodovací metoda klasifikuje do tříd „nákup/prodej“, „nákup/prodej/nedělat nic“ nebo do tříd vyjadřujících budoucí odhad chování kurzu jako „bude hodně klesat/málo klesat/nedělat nic/málo stoupat/hodně stoupat“ nebo jakkoliv jinak, patří do této skupiny.

#### 3.4.1 Matice záměn

Nejobvyklejším ukazatelem je tzv. matice záměn (confusion matrix). V této matici je zachyceno, kolikrát se model při testování dopustil chyby a kolikrát naopak predikoval hodnotu správně.

Předvedme si tento princip na klasifikaci do tříd nákup (+) a prodej (-). Pro tento účel si musíme zavést několik pojmů.

- TP (true positive) je počet příkladů, které systém správně zařadil do třídy +.
- FP (false positive) je počet příkladů, které systém chybně zařadil do třídy + a patří do třídy -.
- TN (true negative) je počet příkladů, které systém správně zařadil do třídy -.
- FN (false negative) je počet příkladů, které systém nesprávně zařadil do třídy - a patří do +.

Správné zařazení	Klasifikace systémem	
	+	-
+	TP	FN
-	FP	TN

Tabulka 3.1: Matice záměn pro dvě třídy + a -  
Zdroj: [11]

Matice záměn ukazuje pouze počet správně a nesprávně zařazených příkladů. Někdy je ale potřeba odlišit různé chyby. Neprovedený ziskový obchod je určitě závažnějším problémem než neprovedený ztrátový. Tento problém lze zahrnout v tzv. matici cen. Zde je správnému rozhodnutí přiřazena cena 0 a různě závažným chybám je přidělena různá cena.

#### 3.4.2 Celková správnost a celková chyba

Celková správnost se vypočítá jako relativní počet správných rozhodnutí modelu.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.2)$$

Celková chyba naopak jako relativní počet rozhodnutí špatných.

$$Err = \frac{FP + FN}{TP + TN + FP + FN} \quad (3.3)$$

Pokud chceme podobně jako u matice cen započítávat cenu jednotlivých chyb, můžeme pro výpočet celkové chyby použít následující vzorec.

$$Err = FP \times c(P, n) + FN \times c(N, p) \quad (3.4)$$

Kde  $c(P, n)$  je cena chyby zařazení příkladu do třídy +, zatímco správně měl být zařazen do třídy -. Obdobně  $c(N, p)$  je cena, pokud byl příklad zařazen do třídy -, ale měl být zařazen do třídy +.

### 3.4.3 Správnost pro jednotlivé třídy

Je poměr správně klasifikovaných příkladů třídy k součtu správně klasifikovaných a špatně klasifikovaných do této třídy. Výsledkem je číslo ukazující, jaká část z příkladů, které model přiřadil dané třídě, k této třídě skutečně patřilo.

$$Acc_+ = \frac{TP}{TP + FP} \quad (3.5)$$

$$Acc_- = \frac{TN}{TN + FN} \quad (3.6)$$

### 3.4.4 Přesnost a úplnost

Přesnost je stejným ukazatelem jako správnost pro některou třídu. Úplnost říká, jakou část z příkladů dané třídy model této třídě skutečně přiřadil.

$$Presnost = \frac{TP}{TP + FP} \quad (3.7)$$

$$Uplnost = \frac{TP}{TP + FN} \quad (3.8)$$

### 3.4.5 Senzitivita a specificita

Dalšími běžně využívanými ukazateli jsou senzitivita a specificita. Senzitivita je totéž co úplnost.

$$Senzitivita = \frac{TP}{TP + FN} \quad (3.9)$$

$$Specificita = \frac{TN}{TN + FP} \quad (3.10)$$

## 3.5 Ukazatele pro numerické predikce

Podobně jako předchozí podkapitola, která se věnovala klasifikacím, se tato podkapitola nezabývá testováním obchodního systému jako celku, ale je zaměřena pouze na model použitý pro předpověď budoucího vývoje. Narozdíl od předchozí podkapitoly je tato podkapitola zaměřena na numerické předpovědi. Do této kategorie patří například model, který jako předpověď vrací reálné číslo v intervalu (-1,1) vyjadřující pravděpodobnost směru pohybu

trhu (-1 nejvíce klesá, 1 nejvíce stoupá) nebo model, který předpovídá přímo následující předpovězenou hodnotu kurzu a samozřejmě mnohé další. Do této kategorie můžeme zařadit například některé neuronové sítě nebo modely pro předpověď časové řady. Testování je založeno na faktu, že máme k dispozici ideální výstup, tedy skutečnou očekávanou hodnotu, kterou můžeme porovnávat s tou, kterou předpověděl model. Zavedeme si pojem chyba předpovědi (residual value, zbytková hodnota, zbytková chyba), který vyjadřuje rozdíl mezi skutečnou a předpovězenou hodnotou.

$$r = a - p \tag{3.11}$$

Hodnota  $r$  je chyba předpovědi,  $a$  je skutečná hodnota a  $p$  je předpovězená hodnota [12]. Tyto chyby by měly tvořit nejlépe náhodnou řadu a jejich průměr by měl být blízký nule.

### 3.5.1 Graf závislosti chyby předpovědi na čase

Toto je nejjednodušší graf, který ale nemá velkou vypovídací hodnotu o vlastnostech modelu. Slouží jen jako jednoduché zobrazení vypovídající o velikosti chyb. Na osu  $x$  vyneseme čas a při každé předpovědi vyneseme na osu  $y$  k ní příslušnou chybu předpovědi. Alternativou, u které můžeme podobně rychle vidět velikost chyby je, že osa  $x$  opět reprezentuje čas, ale na ose  $y$  vyneseme dvě sady hodnot. Jednou z nich jsou skutečné hodnoty a druhou předpovězené. Tento typ grafu je např. na obrázcích 3.1 a 3.2.

### 3.5.2 Graf závislosti chyby předpovědi na předpovězené hodnotě

Na ose  $x$  vyneseme seřazené předpovězené hodnoty a na ose  $y$  k nim příslušné chyby předpovědi. Tento graf slouží jako základní ověření, zda model nedělá systematickou chybu.

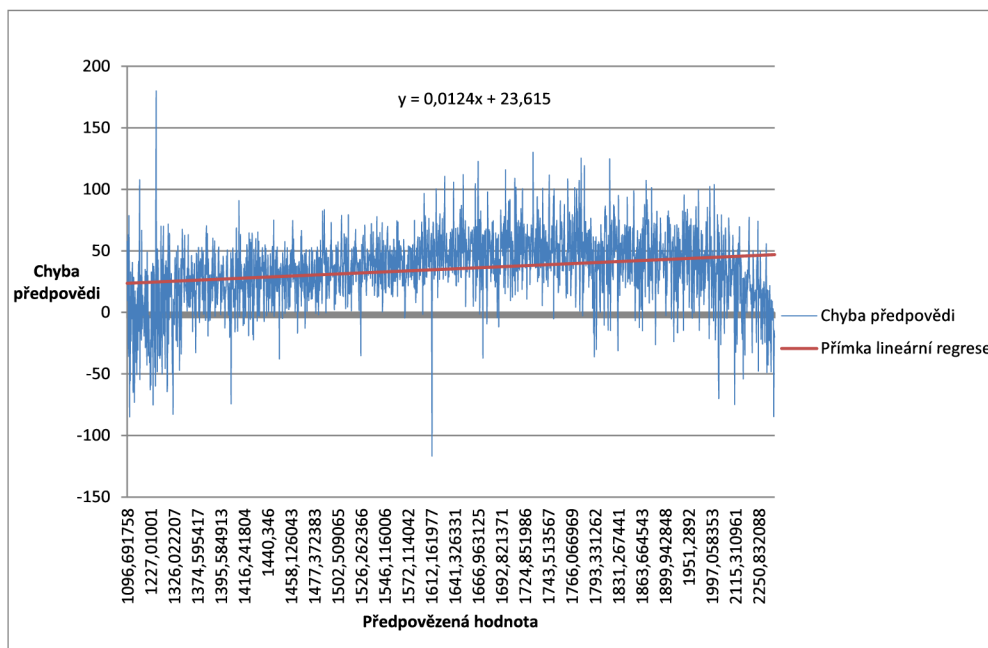
Průměr druhých mocnin rozdílů očekávaných a skutečných hodnot by se měl pohybovat co nejbližší nule. Pokud se nule neblíží při testování na testovací množině dat, je potřeba zjistit její hodnotu na trénovací množině. Pokud se ani v tomto případě neblíží tato hodnota číslu nula, může to signalizovat problém jako například špatný model, nedostatek dat, nerovnoměrná data a jiné.

Dalším ukazatelem je proložení bodů grafu lineární regresí. Tato přímka by měla být rovnoběžná s osou  $x$  grafu a měla by procházet nulou. Pokud tato přímka nesplňuje tyto vlastnosti nebo jsou amplitudy (velikosti) chyb výrazné, pravděpodobně to signalizuje problém. Na grafu 3.3 je znázorněna právě závislost chyby předpovědi na předpovězené hodnotě. Rovnice v grafu je rovnicí přímky lineární regrese, která byla grafem proložena. V tomto případě vidíme nejen velikost chyb, ale především zaujatost modelu, který systematicky předpovídá hodnoty nižší, než jsou skutečné.

Rozložení chyb předpovědi by mělo být blízko normálnímu rozložení, což je další z významných ukazatelů. Odchylky od normálního rozložení mohou symbolizovat šum v datech nebo rozdíl trénovací a testovací množiny, pokud jsou ale tyto odchylky výrazné, je pravděpodobné, že s modelem není něco v pořádku. Pokud chyby předpovědi na trénovacích datech tvoří přibližně normální rozložení, ale v testovacích ne, problémem je pravděpodobně rozdílnost mezi těmito daty. Pokud se rozložení neblíží normálnímu ani v jedné z množin, chyba je s největší pravděpodobností v modelu.

Ideálně by graf závislosti chyby na predikované hodnotě měl být rovnoměrný, chyby co možná nejmenší a přímka lineární regrese by měla procházet nulou a měla by být rovnoběžná s osou  $x$ .





Obrázek 3.3: Graf závislosti chyby předpovědi na předpovězené hodnotě

### 3.5.3 Graf závislosti skutečné hodnoty na předpovězené hodnotě

V tomto případě reprezentuje osa x seřazené předpovězené hodnoty a osa y k nim příslušné skutečné. Tento graf patří také k nejzákladnějším nástrojům. Jedním z ukazatelů, na které se zde můžeme zaměřit, je porovnání rozsahů skutečných a předpovězených hodnot. Další věcí, která musí být provedena, je porovnání rozložení skutečných a předpovězených hodnot. Pokud se tato rozložení či rozsahy liší, je potřeba předpovědi analyzovat a zjistit, zda se jedná o chybu v modelu nebo například nedostatečně reprezentativní trénovací data.

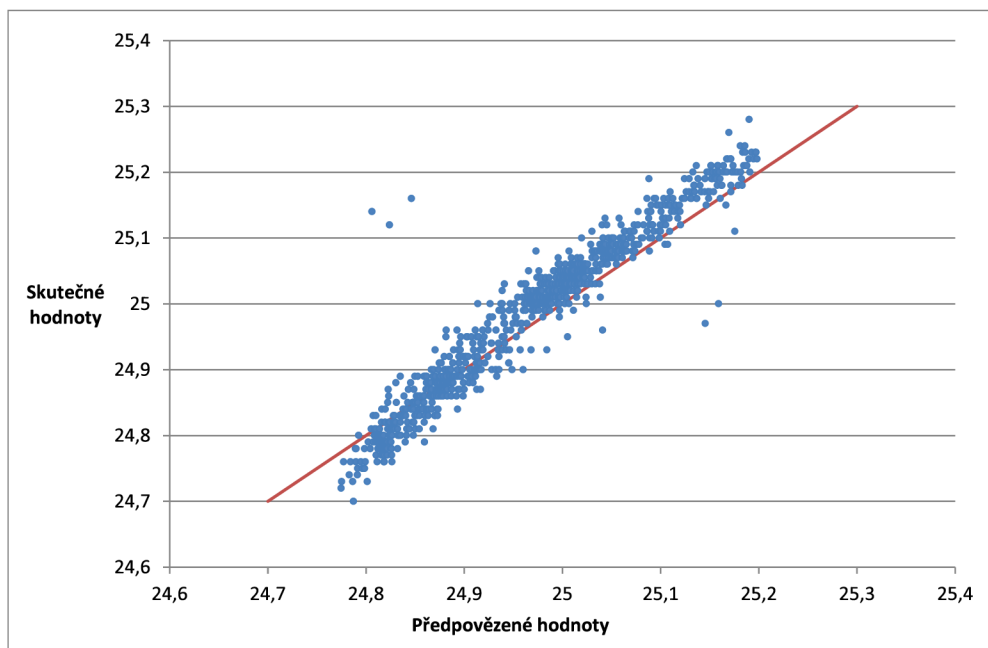
Zajímavé výsledky může mít také proložení regresní křivky tímto grafem. Vztah mezi skutečnými a předpovězenými hodnotami by měl být lineární a tato křivka by se tedy měla blížit přímce. Pokud je tato křivka výrazně odchylená od diagonály vedoucí z bodu  $[0,0]$ , je možné, že není model dostatečně specifikován a úroveň generalizace je příliš vysoká.

Proložení dat přímkou (lineární regresi) bychom měli ideálně dostat přímku procházející bodem  $[0,0]$  a stoupající pod úhlem 45 stupňů. V ideálním případě, kdy by všechny předpovězené hodnoty odpovídaly skutečným, by všechny body grafu ležely právě na této přímce. Také pokud rozsah hodnot je například od 0 do 1, přímka lineární regrese by měla procházet body  $[0,0]$  a  $[1,1]$  nebo se jim alespoň blížit. Na grafu 3.4 můžeme pozorovat závislost skutečné hodnoty na předpovězené. Každý bod zde vyjadřuje jednu předpověď a k ní příslušnou skutečnou hodnotu. Přímka v grafu je diagonála z bodu  $[0,0]$  stoupající pod úhlem 45 stupňů.

### 3.5.4 Graf závislosti chyby předpovědi na skutečné hodnotě

Tento graf je dalším grafem, který by měl být rutinně zkoumán. Tentokrát seřadíme skutečné hodnoty a vyneseme je na osu x a k nim příslušné chyby předpovědi vyneseme na osu y.

Body opět proložíme lineární regresi a zkoumáme výsledek. I zde si musíme dát pozor, zda je nerovnoběžnost přímky značící lineární regresi s osou x výsledkem systematické



Obrázek 3.4: Graf závislosti skutečné hodnoty na předpovězené

chyby nebo jen vlastností modelu. Podobně jako u předchozích, i z tohoto grafu lze pouhým pohledem odhalit případné systematické chyby. Předpovědi modelu však mohou být vylepšeny i v případě, že nevykazují systematickou chybu, a to například zmenšením chyb.

### 3.5.5 Hodnocení ukazatelů a možné problémy

Výše zmíněné metody by za správného testování out-of-sample měly ukázat vlastnosti modelu určeného na předpovídání vývoje trhu. Tyto vlastnosti by se měly podobat vlastnostem v reálném provozu. Takové ověření může obchodníkovi na burze ušetřit velké množství finančních prostředků.

Že je s modelem něco v nepořádku můžeme poznat často také jednoduše tak, že jeho výsledky předpovídání nejsou o mnoho lepší než náhodné tipování, chyby nesou dodatečnou informaci, která může být použita k vylepšení předpovědi, obchodní systém používající tento model je dlouhodobě ztrátový a další. Tyto problémy často značí nedostatky v použitých datech nebo právě v modelu.

## 3.6 Další ukazatele

### 3.6.1 Křivka učení

Křivka je zaměřena na předpoklad, že čím více příkladů v trénovací množině, tím bude model přesnější. Dává tedy do souvislosti velikost trénovací množiny dat a správnost rozhodnutí modelu. Využití této křivky je významné především u modelů, které vyžadují provedení výpočtů předcházející samotné schopnosti předpovídat.

### 3.6.2 Křivka navýšení

Křivka navýšení je ukazatelem, který můžeme použít jen u modelů, které doprovází klasifikaci do třídy vahou. Výstupem modelu je tedy reálné číslo, které následně dle jeho hodnoty zařadíme do některé ze tříd. Křivka navýšení vyjadřuje vztah mezi TP a  $(TP+TN+FP+FN)$ .

Křivku vytvoříme tak, že sestavíme tabulku předpovězených hodnot a k nim příslušných tříd a tuto tabulku seřadíme od největší váhy po nejmenší.

Pořadí	Skutečná třída	Předpovězená váha
1	+	0,97
2	+	0,92
3	-	0,84
4	+	0,82
5	-	0,76
...	...	...

Tabulka 3.2: Tabulka pro konstrukci křivky navýšení.

Zdroj: [11]

Poté v tabulce odshora (od nejvyšší váhy) pro libovolnou část tabulky vytvoříme tzv. dílčí matici záměn. Její hodnoty značíme  $TP'$ ,  $TN'$ ,  $FP'$  a  $FN'$ . Poté vyneseme do grafu hodnoty  $TP'/TP$  a  $(TP' + TN' + FP' + FN') / (TP + TN + FP + FN)$ .

Křivka, kterou tímto způsobem získáme prochází body  $[0,0]$  (neprovedeme-li jediný obchod, nebude tedy žádný ziskový) a  $[1,1]$  (provedeme-li největší možné množství obchodů, zachytíme všechny ziskové). Křivka by se měla nacházet co nejvýše nad diagonálou, tedy přímkou vedoucí z bodu  $[0,0]$  do  $[1,1]$ .

V případě, že máme k dispozici informace o průměrné ztrátě při neúspěšném obchodu a zisku při úspěšném, můžeme křivku navýšení převést na křivku návratnosti investic, označovanou obvykle jako ROI. Z této křivky lze vyčíst změny zisku v závislosti na počtu provedených obchodů.

### 3.6.3 Křivka ROC

Tato křivka reprezentuje souvislost mezi  $TP[\%]$  a  $FP[\%]$ .

$$TP[\%] = \frac{TP}{TP + FN} \quad (3.12)$$

$$FP[\%] = \frac{FP}{FP + TN} \quad (3.13)$$

U této křivky předpokládáme model, který predikci doprovází vahou podobně jako u křivky navýšení. Křivka je vytvořena tak, že je měněn právě tento práh, kdy bude výstup vyhodnocen jako třída +. Bodu  $[0,0]$  (žádná predikce třídy +) odpovídá práh 1, bodu  $[1,1]$  (všechny příklady přiřazeny do třídy +) naopak odpovídá práh 0. Změnou tohoto prahu můžeme testovat, při jaké hodnotě bude model nejpřesnější nebo také změny poměru mezi počty příkladů obou tříd. Bod  $[0,1]$  zde vyjadřuje bezchybné predikce a křivka, která v grafu vznikne, by se ideálně měla pohybovat právě co nejblíže tohoto bodu [11].

## 3.7 Testování obchodního systému

Tato kapitola se zaměří na testování obchodního systému jako celku. Výsledky testů obchodního systému mohou být ovlivněny například participovaným trhem (trhy), jeho volatilitou a cenami, výší poplatků a komisí spojenými s burzovním obchodem, které mohou být u každé brokerské společnosti rozdílné, velikostí konta obchodníka, kdy každý vstupuje do trhu s rozdílnou částkou a za přijatelnou ztrátu považuje rozdílnou sumu a dalšími faktory. Výsledky testování jsou proto hodnoceny subjektivně jednotlivými obchodníky.

### 3.7.1 Vhodné ukazatele

Použité ukazatele by tak měly být dostatečně informativní a také objektivní. Zahrnout sem můžeme ukazatele:

- Nejlepší provedený obchod v průběhu testování.
- Nejhorší provedený obchod.
- Nejvyšší stav účtu.
- Nejnižší stav účtu.
- Počet obchodů otevřených nákupem.
- Počet obchodů otevřených prodejem (možné jen u vybraných instrumentů finančního trhu).
- Počet obchodů uzavřených strategií.
- Počet obchodů ukončených Stop-lossem, případně jinak.
- Celkový počet provedených obchodů.
- Počet ziskových obchodů.
- Počet ztrátových obchodů.
- Poměr úspěšnosti vstupu do trhu (poměr počtu ziskových a ztrátových obchodů).
- Celkový zisk.
- Celková ztráta.
- Nejdelší obchod.
- Nejkratší obchod.
- Nejdelší sérii ziskových obchodů a její peněžní vyjádření.
- Nejvyšší počet po sobě jdoucích ztrátových obchodů a součet těchto ztrát.

Vhodné jsou i statistiky ke každému jednotlivému obchodu, tedy čas vstupu do trhu, čas výstupu z trhu, typ obchodu, počáteční a koncová cena. Dalšími vhodnými ukazateli jsou Risk reward ratio nebo zaznamenávání výše účtu v čase.

## Kapitola 4

# System pro testování obchodních dat

Každý burzovní obchodník, ať už drobný investor nebo banka, potřebuje způsob, jak nějakým způsobem otestovat svůj obchodní systém. Použití nevyhovujícího systému může přinést obchodníkovi nemalé ztráty, a proto je potřeba před jeho uvedením do provozu nějakým způsobem zjistit, jak si systém povede. Bohužel trh je nevyzpytatelný, nelze jej popsat exaktně a závisí na mnoha náhodných jevech. Funkcí aplikace, která je součástí této práce a je nazvána System pro testování obchodních dat, je tedy co nejpřesnější odhad výkonu obchodního systému v budoucím reálném provozu.

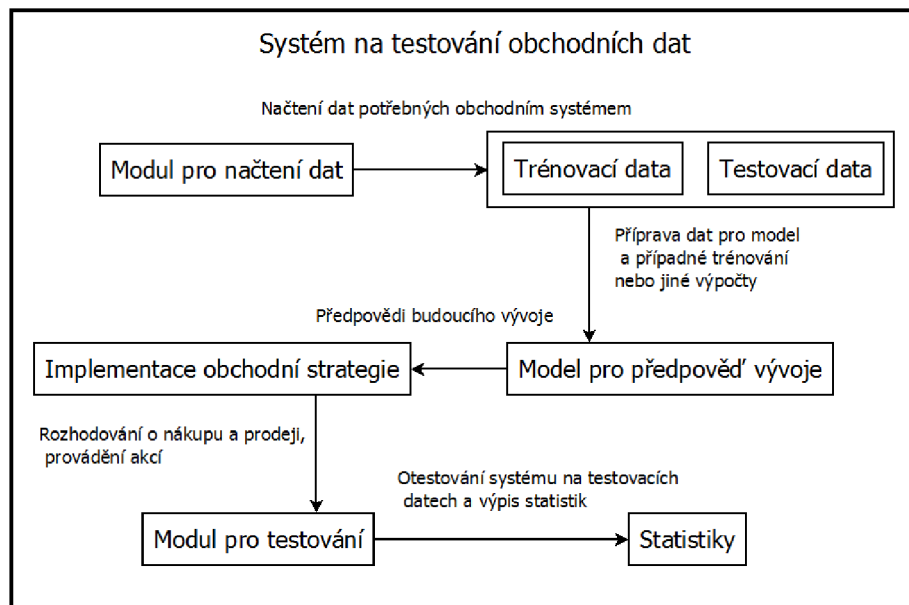
Další důležitou vlastností je rychlost testování. Obchodní systém je často založen na nedávných znalostech vývoje trhu a je potřeba uvést jej na trh pokud možno co nejdříve, aby zajistil určitou konkurenční výhodu.

### 4.1 Neformální analýza a specifikace požadavků

Na trhu existuje množství softwarových nástrojů pro otestování obchodního systému. Bohužel obvykle dokáží pracovat jen se sadou jednoduchých indikátorů. Některé z nich poskytují možnost vývoje komplexnějších obchodních systémů díky rozsáhlým knihovnám nebo vlastním programovacím jazykům [20]. Navíc svádí obchodníka k testování a ladění systému in-sample s očekáváním, že si systém povede stejně dobře i v reálném prostředí. Pro otestování obecného obchodního systému musí mít obchodník k dispozici programátora, který dokáže navržená pravidla obchodního systému algoritmizovat a také metodiku pro komplexnější in-sample i out-of-sample testování na historických, případně i aktuálních datech.

Buhožel každý obchodní systém je unikátní a proto nelze systém na testování vytvořit obecně a každý obchodní systém musí být samostatně implementován. System pro testování obchodních dat by měl umožnit volbu množiny testovacích dat pro testování in-sample i out-of-sample spojené s optimalizací parametrů obchodního systému pro zlepšení výkonu. Důležité je také množství statistik, které bude aplikace ukládat.

Vyhodnocení výsledků testů a způsob ladění už je ponechán na obchodníkovi. System ale při správném dodržení postupu testování zmenšuje riziko curve-fitting. I po důkladném otestování systému jsou ale stále nejdůležitější znalosti obchodníka při tvorbě a ladění systému.



Obrázek 4.1: Moduly systému pro testování obchodních dat a vztahy mezi nimi

## 4.2 Návrh a implemetace

V této podkapitole se zaměříme na návrh Systému pro testování obchodních dat. Při popisu návrhu budou pro lepší přiblížení aplikace zmíněny některé důležité třídy a stěžejní detaily implementace.

### 4.2.1 Návrh aplikace

Návrh aplikace byl proveden s důrazem na obecnost. Předpokládáme možnost dodatečné implementace libovolných obchodních systémů využívajících různorodé přístupy a různorodá data. Obchodní systém je reprezentován třídou `TradingSystem`. Tento systém obsahuje třídu `PredictionModel` pro model předpovědi budoucího vývoje a třídu `TradingStrategy` pro strategii rozhodující o nákupech, prodeji a dalších náležitostech na základě výsledků předpovědi. Dále obsahuje třídy pro načtení testovací a trénovací množiny dat, což jsou třídy dědicí od třídy `InputData`, jejíž jediná metoda `loadData()` zajistí načtení potřebných dat. Vzhledem k navržené obecnosti systému je možné vytvořit další třídy, dědicí od zmíněných tříd, které díky principu dědičnosti zachovají potřebné vlastnosti a rozhraní komunikace mezi jednotlivými částmi systému. Je tedy například možné jen vyměnit třídu s rozhodovací strategií při zachování zbytku systému nebo třeba navrhnout naprosto odlišný obchodní systém, pracující s odlišnými daty, který ale díky zachování komunikačního rozhraní může být aplikací otestován. Na schématu 4.1 jsou znázorněny části systému na testování obchodních dat a vztahy mezi nimi.

Aplikace dokáže pracovat s komoditními futures, indikátorem technické analýzy, akciemi a měnovým párem systému Forex. Indikátor technické analýzy reprezentuje třída `StockMarketIndikator`. Oproti tomu `Futures`, `Forex` a `Stock` jsou třídy dědicí od třídy `StockMarketProduct`. Tato třída je specifická tím, že se s ní dají provádět obchody. Zákonitosti obchodu s komoditními futures, forexem a akciemi jsou rozdílné a výpočty zisků, ztrát či poplatků jsou provedeny na základě rozdílného principu. `StockMarketProduct`



i `StockMarketIndicator` jsou třídy dědicí od `StockMarketEntity`. Všechny entity, se kterými systém pracuje jsou ukládány do seznamu, aby bylo možno implementovat obchodní systém využívající jejich libovolné množství. Každá z těchto entit poté obsahuje testovací a trénovací data k ní příslušná. Tato data jsou načtena právě metodami `loadData()`, které jsou volány při inicializaci systému.

Test obchodního systému je reprezentován třídou `TradingSystemTest`.

```
/*
 * Inicializace obchodniho systemu, na kterem budou provedeny testy
 */

System.Initialize
(
    new RPROPClassifyingNeuralNetwork(), //predpovidaci model
    new RPROPClassifyingNNTradingStrategy(), //strategie
    new LivechartsMSInSampleData(), //trida pro nacteni trenovacich dat
    new LivechartsMSOutOfSampleData(), //trida pro nacteni testovacich dat
    new RPROPClassifyingNNResidualError(), //trida pro vypocet chyb
    1000, //pocatecni suma na uctu
    files //soubory pro testovani a trenovani
);
```

#### 4.2.2 Přijatá zjednodušení

Testování obchodního systému probíhá v podmínkách simulujících reálné prostředí trhu. Některé faktory jsou ale příliš náhodné na to, aby byly napodobeny. Jedním z nich je slippage. Toto zpoždění by bylo možné odhadovat podle objemu obchodů, ale závisí na více faktorech. Problémem je, že máme dostupné hodnoty kurzu jen na konci daných časových intervalů. Nelze tedy přesně určit hodnotu kurzu v době opožděně provedeného příkazu. Proto budeme tuto prodlevu zanedbávat a všechny příkazy budou provedeny okamžitě a za aktuální cenu.

#### 4.2.3 Načítání dat

Systém dokáže načíst data do dvou množin. První z množin je použita pouze pro účel výpočtů, které jsou potřebné před samotným testováním. Například trénink neuronové sítě. Pro každou z těchto množin zadá uživatel soubor nebo soubory, ze kterých mají být data načítána a také rozsah dat, který má být načten. Tyto údaje jsou v implementaci představovány procentuelně vzhledem k velikosti souboru.

```
/*
 * Vlozeni s testovacimi a trenovacimi daty
 */

InputDataFileList files = new InputDataFileList();

//vlozeni dat od zacatku souboru do 20 % delky souboru mezi trenovaci data
files.addTrainFile(new InputDataFile("livechartscoukMS.txt", 0,20));
```

```
//vlozeni dat od 20% souboru do 40 % souboru mezi testovací data
files.addTestFile(new InputDataFile("livechartscoukMS.txt", 20,40));
```

#### 4.2.4 Formát CSV

System dokáže importovat data, což obvykle znamená hodnoty kurzu v časových rozestupech například pěti minut, hodiny nebo dne. Obvykle jsou poskytována data ve formátu OHLC doplněná o údaj o čase, kdy byly hodnoty naměřeny a o objem obchodů. Soubor ve formátu CSV se skládá z řádků, na kterých jsou jednotlivé položky odděleny čárkou. Bohužel často různé zdroje poskytují tyto údaje v rozdílném pořadí a datum a čas bývají také často formátovány rozdílně (DD-MM-RRRR, MM:DD:RR:Hours:Minutes atd.), proto je potřeba vytvořit pro každý zdroj dat speciální parser. Tento parser je třída dědicí od třídy `FileLoader` a je používán v metodě `loadData()`, která je volána při inicializaci obchodního systému a zajistí korektní načtení všech dat potřebných obchodním systémem.

Obchodní systémy také často pracují i s dalšími informacemi jako jsou hodnoty kurzů dalších instrumentů nebo hodnoty vybraných indikátorů technické analýzy. Tato data mohou být načtena stejným způsobem.

Nutno opět dodat, že jelikož každý obchodní systém vyžaduje naprosto rozdílná data, musí být způsob načtení dat ze souborů implementován individuálně pro každý systém ve třídě dědicí od třídy `InputData`.

#### 4.2.5 Předzpracování dat

Prvním krokem před samotným testováním obchodního systému je předzpracování dat. V tomto kroku dochází k napravení chyb v datech nebo k vypočítání některých nových hodnot. Může se stát, že v datech, která máme k dispozici jsou chybějící hodnoty nebo hodnoty, které se výrazně liší od ostatních. V oblasti kurzů na burzovních trzích může chybějící hodnota nebo hodnoty značit neobchodní den a hodnota vyčnívající z řady může být skokovým pohybem, tzv. gap. Pokud jsou ale data opravdu chybná, musíme chybu odstranit. Většinou bývá doplněna průměrná hodnota nebo se uvažují všechny hodnoty dané relativními četnostmi [11]. V oblasti kurzů na burzovním trhu ale většinou bývá účelné doplnit hodnotu navazující na okolní vývoj trhu, aby chybějící nebo příliš vyčnívající hodnota neovlivňovala výpočty různých ukazatelů.

V kroku předzpracování dat také můžeme zjistit některé atributy, které nejsou přímo v dodaných datech, jako například průměrná hodnota, nejvyšší a nejnižší hodnota, počet položek atd. Ty mohou být použity při výpočtech nebo pro následné statistiky.

#### 4.2.6 Implementace obchodní strategie

System na testování obchodních dat musí samozřejmě obsahovat implementaci dané rozhodovací strategie. Jak již bylo dříve zmíněno, je tato strategie implementována na míru a nelze ji vytvořit obecně. Práce je zaměřena na testování obchodních systémů, a proto jsou implementovány jen jednoduché obchodní systémy, které jsou vhodné právě pro vyzkoušení znalostí o jejich testování.

Pro implementaci libovolné strategie je připraveno množství nástrojů. V metodě pro rozhodovací obchodní strategii jsou k dispozici výsledky předpovědi modelu pro předpovídání, aktuální stav účtu a aktuální seznam trhů a otevřených transakcí. Dále jsou

k dispozici metody pro nákup a prodej. Takových metod je implementováno hned několik, například metoda třídy `StockMarketProduct` `openNewTransactionBuy()` provede nákup požadovaného množství daného instrumentu finančního trhu. Dále jsou k dispozici metody `closeExistingTransaction()` a `closeAllTransactions()`, kde první jmenovaná uzavře již probíhající transakci, zatímco druhá uzavře všechny transakce, které má obchodník aktuálně otevřené. Dále systém poskytuje obchodníkovi možnost zadat příkazy, které budou provedeny až při dosažení určité podmínky. K tomuto účelu slouží metoda `placeOrder()`, která po určení typu příkazu a jeho parametrů, tento příkaz vytvoří. V aplikaci jsou implementovány všechny příkazy, které byly zmíněny v podkapitole 2.1.5. Příkaz se váže k dané transakci a po jejím ukončení je příkaz také zrušen.

#### 4.2.7 Příprava testování

V metodě `runTest()` třídy reprezentující test je vytvořen obchodní systém, kterému jsou jako parametry předány výše zmíněné součásti. Poté proběhne načtení dat. V případě, že model pro předpověď vyžaduje trénink nebo jiné výpočty předcházející jeho schopnosti předpovídání, jsou tyto výpočty provedeny. Všechny tyto činnosti včetně např. přípravy trénovacích dat a ideálních výstupů zabezpečuje metoda třídy `PredictionModel`, nazvaná `train()`. Tato metoda nemusí být v každém obchodním systému implementována.

#### 4.2.8 Průběh testování

Testování systému probíhá obdobně jako skutečné obchodování. Nejprve jsou seřazeny vzorky dat testovací množiny dle času a do proměnné reprezentující aktuální čas je načten nejnižší čas a probíhá následující cyklus.

```
while(aktualniCas <= koncovyCas)
{
    executeOrders();
    predict();
    decide();
    aktualniCas = najdiCasDalsihoZaznamu();
}
```

Metoda `executeOrders()` projde všechny položené obchodní příkazy a pokud je splněna podmínka jejich provedení, jsou provedeny.

Metoda `predict()` je metodou třídy `PredictionModel`. Provede výpočet modelu potřebný k předpovědi budoucího vývoje. Nejprve si připraví potřebná data pomocí metody `getDataForPrediction()`, která vrátí všechna potřebná data pro predikci v aktuálním čase. Metoda `predict()` je v rodičovské třídě `PredictionModel` definována jako virtuální, proto pokud ji daný model neimplementuje (například není potřeba provést žádný výpočet, protože rozhodnutí je učiněno na základě již vypočtených hodnot) je použita prázdná implementace.

Metoda třídy `TradingStrategy` nazvaná `decide()` obsahuje implementaci rozhodovací strategie obchodníka. Provádějí se zde nákupy, prodeje a rozhodování o nich.

Pokud funkci některé z metod nelze provést, například z důvodu chybějícího záznamu, je vyvolána výjimka a je posunut čas na další záznam.

O každém rozhodnutí a každém provedeném příkazu a obchodu jsou ukládány statistiky, které jsou po skončení testování přístupné.

### 4.2.9 Výsledky testování

Výsledky testování jsou vypsány do souboru po provedení samotného testování. Vypsány jsou statistiky výkonu obchodního systému, které byly popsány v podkapitole 3.7 a další informace nutné pro určení a výpočet všech ukazatelů prezentovaných v kapitole 3. Především jsou to odchylky předpovězených hodnot od skutečných nebo správně klasifikovaná a předpovězená třída.

## 4.3 Použité technologie

Rozhodujícím kritériem pro výběr implementačního prostředí byla vhodnost pro tvořenou aplikaci. Zvolený jazyk C# obsahuje vestavěné knihovny pro všechny požadované činnosti. Významným faktorem byla také dostupnost externích knihoven, v tomto případě především knihoven pro práci s neuronovými sítěmi.

Aplikace je přenositelná na operační systémy, kde je nainstalované prostředí .NET nebo prostředí Mono. Použitá verze prostředí .NET je 4.0 a to z důvodu vyžadování této verze externí knihovnou pro práci s neuronovými sítěmi.

### 4.3.1 Technologie .NET

Framework .NET je softwarový framework pro operační systémy Windows. Podporuje několik jazyků (C#, Visual Basic, Visual C++ a další). Programy vytvořené pro .NET framework jsou spouštěny v softwarovém prostředí, které se nazývá CLR (Common Language Runtime). Toto CLR je virtuálním strojem, který poskytuje určitou abstrakci od hardwaru. Tím by měla být zajištěna přenositelnost na jakékoliv zařízení, které má podporu .NET nebo Mono frameworku.

Virtuální stroj také poskytuje pokročilou ochranu například před útoky typu buffer overflow nebo pokročilou správu paměti obsahující garbage collector, což je funkce, která automaticky uvolňuje dříve alokovanou, ale nyní již nepoužívanou paměť.

Mezi nejčastěji zmiňované nevýhody .NET patří vyšší potřeba zdrojů než u jazyků, které přistupují k hardware více přímo (například jazyk C nebo assembler) a tedy pomalejší běh aplikací.

### 4.3.2 Jazyk C#

Jazyk C# byl vyvinut společností Microsoft jako součást prostředí .NET. Jazyk C# je moderním, víceúčelovým, objektově orientovaným jazykem. Jazyk využívá principy jako silná typová kontrola, sledování hranic polí, detekce použití neinicizovaných proměnných, garbage collector a další.

## 4.4 Použité knihovny

### 4.4.1 Neuronové síť

Pro práci s neuronovými sítěmi byla použita knihovna Encog. Tato knihovna je zaměřena na dolování dat, strojové učení a primárně neuronové síť. Encog obsahuje nástroje k tvorbě různorodých neuronových sítí a také třídy pro normalizaci dat nebo zpracování dat pro tyto síť. Knihovna také nabízí několik algoritmů učení, v čele s algoritmem resilient propagation, který je optimalizován pro běh ve více vláknech. Více o knihovně Encog lze najít v [17].

## Kapitola 5

# Experimenty s aplikací

V této kapitole bude představen způsob, jakým lze provést test se zvoleným obchodním systémem a prezentovány výsledky vybraných testů. Testy jsou zvoleny pro demonstraci možností aplikace.

### 5.1 Implementované obchodní systémy

Součástí této práce je také implementovaný Systém pro testování obchodních dat popsán a navržený v předchozí kapitole. Kromě možnosti implementovat různorodé obchodní systémy nabízí také systémy již implementované. Grafy 3.1, 3.2, 3.3, 3.4 byly vytvořeny za pomoci tohoto Systému na testování obchodních dat a obchodních systémů, které budou představeny v následujících podkapitolách 5.1.1, 5.1.2. Tyto obchodní systémy byly zvoleny s ohledem na podobu s reálně využívanými systémy.

#### 5.1.1 Experimenty s numerickými predikcemi neuronové sítě

První implementovaný systém byl vytvořen pro experimenty s ukazateli pro numerické predikce, které byly prezentovány v podkapitole 3.5. Tento systém používá pro předpovědi neuronovou síť, která přijímá na vstupu  $N$  uzavíracích hodnot intervalu kurzu, které předcházely aktuálnímu času a hodnotu objemu obchodů v aktuálním čase. Na výstupu sítě je reálné číslo  $C$ , které reprezentuje předpokládaný vzestup nebo pokles trhu. Toto číslo je v intervalu  $\langle 0,1 \rangle$ , kde 0 znamená velké klesání, 1 velké stoupání a 0,5 pohyb „do strany“, tedy žádnou nebo malou změnu kurzu v následujících  $M$  intervalech. Dalšími výstupy sítě je  $O$  čísel, což jsou předpovědi kurzu v  $O$  následujících časových intervalech. Síť je vícevrstvou perceptronovou sítí, má jednu skrytou vrstvu a je trénována algoritmem Resilient propagation, což je varianta algoritmu zpětného šíření chyby.

Nákup je proveden, když je v předpovězených  $M$  číslech zjištěn vzestup kurzu o  $P$  procent a zároveň je číslo  $C$  větší než hodnota  $H$ . Analogicky pro prodej. Obchodní systém obchoduje jen na jediném trhu.

#### 5.1.2 Experimenty s klasifikací do tříd neuronové sítě

Další implementovaný obchodní systém byl navržen pro experimentování s ukazateli pro klasifikaci do tříd z podkapitoly 3.4. Taktéž využívá pro předpovídání o budoucím vývoji neuronovou síť. Vstupy této sítě jsou totožné se sítí představenou v podkapitole 5.1.1. V tomto případě je ale ideálním výstupem při učení pouze jedno číslo, které má hodnotu



1, pokud je v následujících  $M$  hodnotách zaznamenán vzestup, hodnotu 0, pokud je zaznamenán pokles a 0,5 při malé změně či neměnicím se kurzu.

Po ukončení učícího procesu a při samotném testování obchodního systému je síti vždy na vstup předloženo  $N$  předcházejících close hodnot a jedna hodnota objemu obchodů a síť odhadne výstup, který je poté zařazen do jedné ze tříd 0, 0,5 nebo 1. Tyto hodnoty můžeme interpretovat jako prodat, nedělat nic a nakoupit. Obchodní systém obchoduje pouze na jediném trhu.

### 5.1.3 Experimenty s klasifikací do tříd pomocí indikátorů

Tento obchodní systém využívá pro rozhodování jednoduché klouzavé průměry délky 9 a 19. Pokud se tyto dva klouzavé průměry protnou, provede se akce. Pokud byla hodnota klouzavého průměru délky 19 vyšší než hodnota klouzavého průměru délky 9 a po protnutí je situace opačná, dojde k nákupu instrumentu. Pokud dojde k protnutí opačnému, dojde k prodeji instrumentu.

## 5.2 Ukázky testování obchodního systému

### 5.2.1 Ukázka ukazatelů pro klasifikaci

Cílem této podkapitoly je prezentovat funkci Systému na testování obchodních dat v případě, že model pro předpověď vývoje trhu klasifikuje do tříd. Je použit systém popsáný v podkapitole 5.1.3. Když obchodní systém rozhodne o nákupu, zařadíme tento příklad do třídy nákup. Pokud naopak systém rozhodne o prodeji, zařadíme příklad do třídy prodej. Tato rozhodnutí poté porovnáme se skutečným pohybem trhu.

Správné zařazení	Klasifikace systémem	
	nákup	prodej
nákup	17	18
prodej	12	10

Tabulka 5.1: Matice záměn

Hodnoty v matici záměn, která je zobrazena v tabulce 5.1 byly získány provedením testů na indexu Nasdaq za pět let (od 15.4.2006 do 15.4.2011) s intervalem jednoho dne. Vidíme, že správných odhadů bylo 27 a nesprávných 30.

Zlepšení výsledků by pravděpodobně mohlo být dosaženo po ladění obchodního systému například optimalizací délky klouzavých průměrů nebo přidáním dalších indikátorů.

### 5.2.2 Ukázka testů obchodního systému

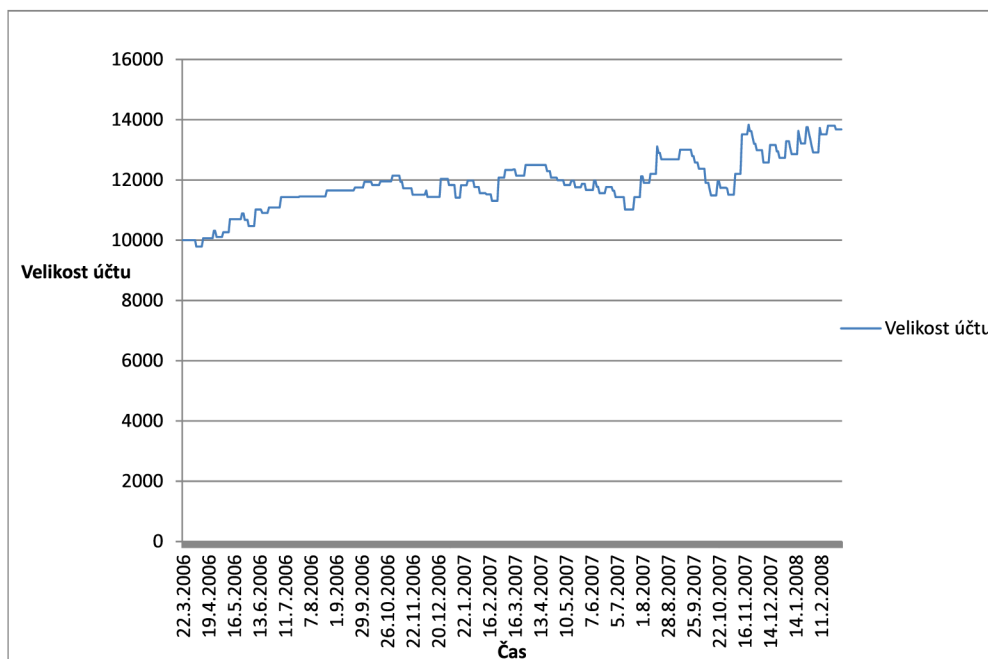
V tomto experimentu se zaměříme na testování obchodního systému jako celku. Využit bude obchodní systém 5.1.1. Experiment byl uskutečněn obchodováním s indexem Nasdaq s hodnotami v intervalu jednoho dne po dobu pěti let. Obchodování probíhá na principu obchodu s futures kontrakty, kde poplatek za nákup 1 kusu instrumentu je stanoven na 5 amerických dolarů a záloha za toto množství je 162,5 dolarů. Test proběhl tak, že neuronová síť byla natrénována na datech za dobu osmi let a poté byl obchodní systém otestován na jiných pěti letech dat.



Zisk z nejlepšího obch.	2 646,5	Počet obch. otevř. nákupem	1
Ztráta z nejhoršího obch.	810	Počet obch. otevř. prodejem	197
Nejvyšší stav účtu	20 695,6	Celkový počet obch.	298
Nejnižší stav účtu	9 789,4	Nejdelší obch.	14 400
Poč. obch. zavř. Last-trading-day	136	Nejkratší obch.	1 440
Poč. obch. zavř. Moving-stop-loss	158	Celkový zisk	45 592,4
Poč. obch. zavřených strategií	3	Celková ztráta	41 749,9
Počet ziskových obch.	101	Nejdelší série zisk. obch.	6
Počet ztrátových obch.	197	Nejdelší série ztrát. obch.	13

Tabulka 5.2: Výsledky obchodního systému in-sample

Pro optimalizaci výkonu obchodního systému lze v tomto případě měnit velké množství parametrů. Například množství nakupovaných jednotek, umístování Stop-lossu, kdy a o kolik případně Stop-loss posouvat, rozsah a výběr trénovacích dat neuronové sítě, počet vrstev sítě, míra naučení sítě, rozhodnutí při jakém výstupu předpovědi sítě nakoupit a kdy prodat, počet přecházejících hodnot uvažovaných při předpovídání sítě (označené jako  $N$  v podkapitole 5.1.1) a další. Pomocí Systému na testování obchodních dat lze sledovat vliv změny těchto parametrů na výsledky testů. Po optimalizaci by bylo vhodné provedení out-of-sample testů.



Obrázek 5.1: Vývoj stavu účtu při testování obchodního systému

V tabulce 5.2 jsou zobrazeny výsledky obchodního systému za pět let testování in-sample, tedy na množině, na které byly laděny jeho parametry. V těchto výsledcích nejsou promítnuty poplatky. Hodnoty jsou v amerických dolarech a čas v minutách. Na obrázku 5.1 lze vidět vývoj konta obchodníka při používání tohoto systému za první dva roky běhu testování. V grafu není ukázáno blokování zálohy.

## Kapitola 6

### Závěr

Práce byla zaměřena na testování obchodních systémů. K tomuto účelu byly vysvětleny důležité pojmy a prezentována metodika a množství ukazatelů. Jednou ze stěžejních částí práce bylo vysvětlení vlivu volby dat na výsledky testování. Na základě analýzy požadavků kladených na testování obchodních systémů byla vytvořena aplikace, která poskytuje nástroje právě pro tento účel. V aplikaci byly také vytvořeny jednoduché obchodní systémy, na nichž byly experimentálně ověřeny znalosti získané v teoretické části práce. Výsledky těchto experimentů byly také v práci rozebrány.

Na poznatky získané v této práci lze navázat dalším studiem a rozšířením repertoáru ukazatelů vhodných pro určení výkonu obchodního systému před jeho případným použitím v reálných podmínkách obchodování na finančních trzích. Velice rozsáhlou a náročnou oblastí možného dalšího zkoumání je tvorba komplexního obchodního systému, který by dokázal po důkladném otestování prezentovanou metodikou a po případném nasazení do provozu vykazovat dlouhodobě dobré výsledky a generovat zisk.

Další možná zdokonalení by se mohla týkat Systému na testování obchodních dat, tedy aplikace, která byla vytvořena v rámci této práce. Pravděpodobně nejvýznamnějším rozšířením by bylo vhodné grafické uživatelské rozhraní, které by dokázalo usnadnit volbu dat a modulů aplikace bez nutnosti zásahů do kódu. Velkým plusem by bylo také automatické znázornění vybraných sledovaných ukazatelů ve formě grafů po dokončení běhu testů.

# Literatura

- [1] *Wikipedia.org : the free encyclopedia [online]*. St. Petersburg : Wikimedia Foundation, 2003-03-03 [cit. 2011-05-01], Nick Leeson. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Nick\\_Leeson](http://en.wikipedia.org/wiki/Nick_Leeson)>.
- [2] *Wikipedia.org : the free encyclopedia [online]*. St. Petersburg : Wikimedia Foundation, 2004-03-20 [cit. 2011-05-01], Ken Olsen. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Ken\\_Olsen](http://en.wikipedia.org/wiki/Ken_Olsen)>.
- [3] *Nakladatelství Sagit [online]*. 2004-05-01 [cit. 2011-05-01], Organizátor mimoburzovního trhu. Dostupné z WWW: <[http://www.sagit.cz/pages/lexikonheslatxt.asp?cd=152&typ=r&levelid=ob\\_176a.htm](http://www.sagit.cz/pages/lexikonheslatxt.asp?cd=152&typ=r&levelid=ob_176a.htm)>.
- [4] *Wikipedia : the free encyclopedia [online]*. St. Petersburg : Wikimedia Foundation, 2005-08-18 [cit. 2011-05-01], Algorithmic trading. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Algorithmic\\_trading](http://en.wikipedia.org/wiki/Algorithmic_trading)>.
- [5] *Regulace finančního trhu (ekonomické a právní aspekty)*. Brno : Masarykova univerzita, Právnická fakulta, 2006, diplomová práce.
- [6] *Wikipedia : the free encyclopedia [online]*. St. Petersburg : Wikimedia Foundation, 2007-01-06 [cit. 2011-05-01], Finanční trh. Dostupné z WWW: <[http://cs.wikipedia.org/wiki/Finanční\\_trh](http://cs.wikipedia.org/wiki/Finanční_trh)>.
- [7] *Wikipedia : the free encyclopedia [online]*. St. Petersburg : Wikimedia Foundation, 2007-02-16 [cit. 2011-05-01], Forex. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Forex>>.
- [8] *Reuters [online]*. 2007-06-23 [cit. 2011-05-01], World's largest stock exchanges. Dostupné z WWW: <<http://www.reuters.com/article/2007/06/23/uk-borsa-takeover-factbox-idUSL2224181520070623>>.
- [9] *Burza cenných papírů Praha [online]*. 2011-01-01 [cit. 2011-05-01], Členové burzy. Dostupné z WWW: <<http://www.bcpp.cz/dokument.aspx?k=Clenove-Burzy>>.
- [10] Beneš, V.; Musílek, P.: *Burzy a burzovní obchody*. 2. aktualiz. a rozšíř. vyd. Praha : Informatorium, 1992, 252 s. ISBN 80-85427-25-7.
- [11] Berka, P.: *Dobývání znalostí z databází*. 1. vyd. Praha : Academia, 2003, 366 s. ISBN 80-200-1062-9.
- [12] Chakrabarti, S.: *Data mining : know it all*. Amsterdam : Elsevier, 2009, 460 s. ISBN 978-012-3746-290.

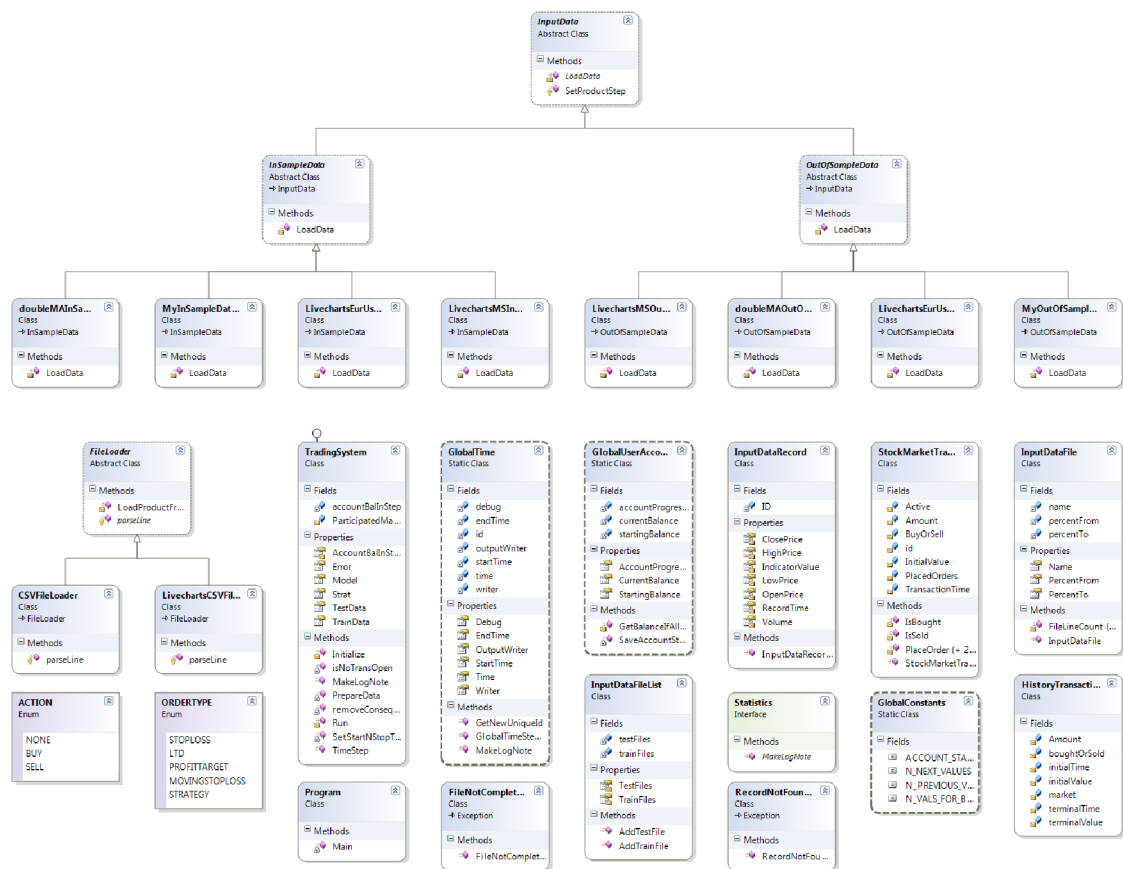
- [13] Chatfield, C.: *Time-Series Forecasting*. 1. vyd. New York : Chapman and Hall, 2001, 267 s. ISBN 15-848-8063-5.
- [14] Eyden, R. J. V.: *The Application of Neural Networks in the Forecasting of Share Prices*. Haymarket : Finance and Technology Publishing, 2006, 285 s. ISBN 0-9651332-0-6.
- [15] Fanta, J.: *Technologie umělé inteligence na kapitálových trzích*. 1. vyd. Praha : Karolinum, 1999, 89 s. ISBN 80-718-4866-2.
- [16] Giovannini, E.: *Ekonomická statistika srozumitelně : z pohledu OECD*. 1. vyd. Praha : Wolters Kluwer Česká republika, 2010, 206 s. ISBN 978-807-3575-366.
- [17] Heaton, J.: *HeatonResearch.com [online]*. 2011-01-01 [cit. 2011-05-01], About the Encog Project. Dostupné z WWW: <<http://www.heatonresearch.com/encog>>.
- [18] Ježková, M.: *Indikátory technické analýzy akcií*. Brno : Masarykova univerzita, Přírodovědecká fakulta, 2010, bakalářská práce. Vedoucí práce Gabriela Oškrdalová.
- [19] Kolmaš, R.: *Investclub.cz [online]*. 2010-01-01 [cit. 2011-05-01], Alternativní investice do managed futures. Dostupné z WWW: <<http://www.investclub.cz/cz/clanky/18-alternativni-investice-managed-futures>>.
- [20] Lanc, M.: *System pro testování obchodní strategie : System for Testing of Business Strategy*. Brno : Vysoké učení technické, Fakulta informačních technologií, 2008, diplomová práce. Vedoucí práce Vladimír Bartík.
- [21] Lawrence, R.: *Using Neural Networks to Forecast Stock Market Prices [online]*. Winnipeg : University of Manitoba, Department of Computer Science, 1997 [cit. 2011-05-01], 22 s. Dostupné z WWW: <<https://people.ok.ubc.ca/rlawrenc/research/Papers/nn.pdf>>.
- [22] Podhajský, P.: *Financnik.cz [online]*. 2008-09-11 [cit. 2011-05-01], V čem se liší burzovní a mimoburzovní brokeri. Dostupné z WWW: <[http://www.financnik.cz/komodity/fin\\_home/burzovni-a-mimoburzovni-brokeri.html](http://www.financnik.cz/komodity/fin_home/burzovni-a-mimoburzovni-brokeri.html)>.
- [23] Rejnuš, O.: *Cenné papíry a burzy*. 1. vyd. Brno : Akademické nakladatelství CERM, 2009, 400 s. ISBN 978-80-214-3805-7.
- [24] Sussman, A.; Tabb, L.; Iati, R.: *TabbGroup.com [online]*. 2009-02-09 [cit. 2011-05-01], US Equity High Frequency Trading: Strategies, Sizing and Market Structure. Dostupné z WWW: <<http://www.tabbgroup.com/PublicationDetail.aspx?PublicationID=505&MenuID=13&ParentMenuID=2&PageID=8>>.
- [25] Tharp, K. V.: *Trade your way to financial freedom*. 1998, 343 s. ISBN 978-0070647626.
- [26] Tupý, J.: *Invsetujeme.cz [online]*. 2008-03-19 [cit. 2011-05-01], Indikátory v technické analýze. Dostupné z WWW: <<http://www.investujeme.cz/clanky/indikatory-v-technicke-analize/>>.

# Seznam příloh

<b>A</b>	<b>Diagram tříd</b>	<b>41</b>
<b>B</b>	<b>Obsah DVD</b>	<b>43</b>
<b>C</b>	<b>Adresářová struktura</b>	<b>44</b>
<b>D</b>	<b>Tvorba vlastních testů a obchodních systémů</b>	<b>45</b>
D.1	Ukázka tvorby vlastního testu obchodního systému . . . . .	45
D.2	Ukázka tvorby vlastního obchodního systému . . . . .	46
D.2.1	Načtení dat . . . . .	46
D.2.2	Model pro předpověď budoucího vývoje trhu . . . . .	46
D.2.3	Rozhodovací strategie . . . . .	47
D.2.4	Výpočet chyby . . . . .	48

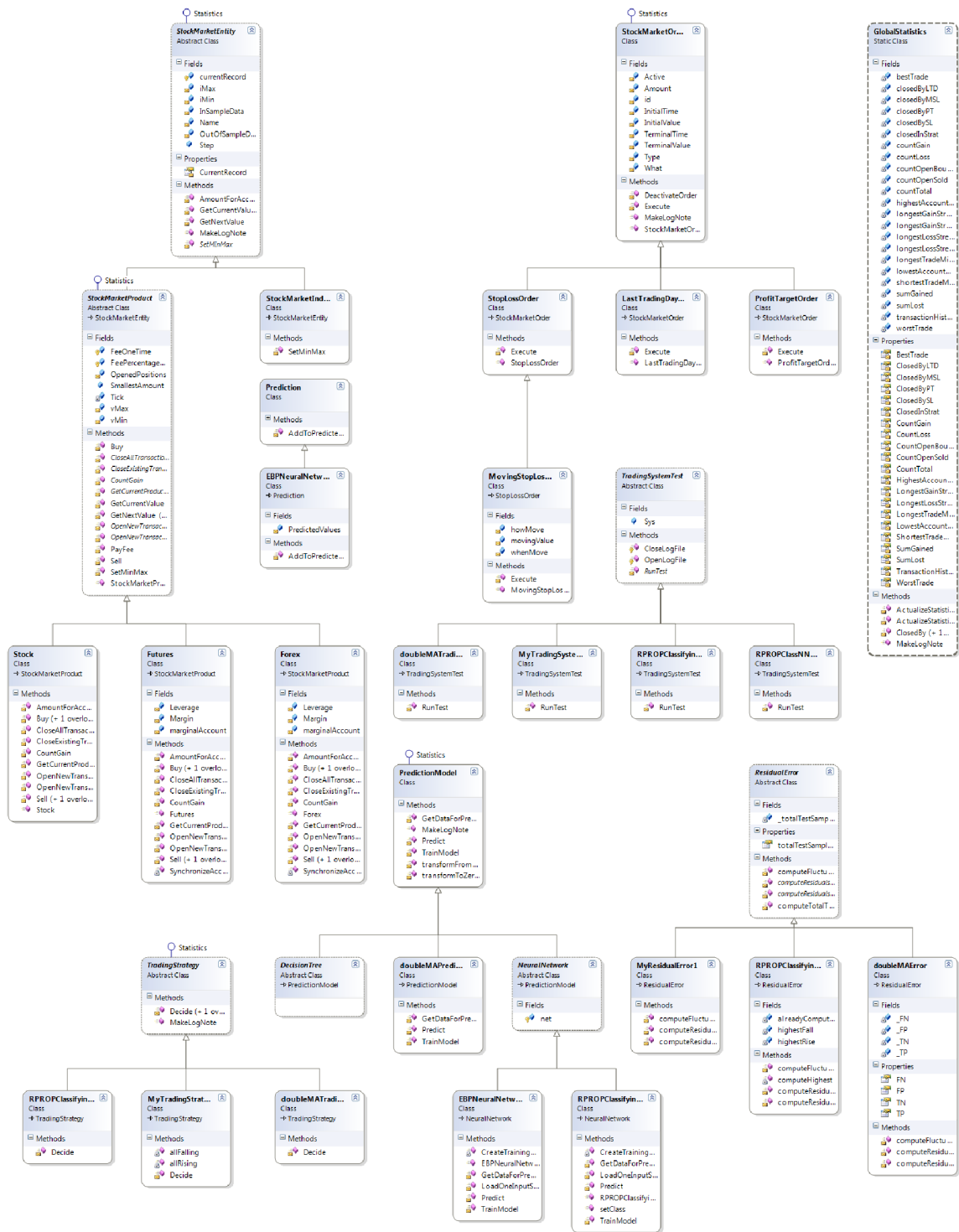
# Příloha A

## Diagram tříd



Obrázek A.1: Diagram tříd Systému pro testování obchodních dat, první část





Obrázek A.2: Diagram tříd Systému pro testování obchodních dat, druhá část

# Příloha B

## Obsah DVD

DVD dodané s prací obsahuje následující součásti.

- Zdrojové kódy aplikace Systém pro testování obchodních dat i s celým projektem v Microsoft Visual Studio 2010.
- Zdrojové kódy technické zprávy a součásti potřebné pro vysázení práce systémem LaTeX.
- Technickou zprávu ve formátu pdf.
- Obrázky, které jsou součástí technické zprávy.
- Soubory s burzovními daty, které byly potřebné při experimentech s aplikací.
- XML dokumentaci Systému pro testování obchodních dat v souboru bpDoc.XML.
- Diagram tříd aplikace.
- Výsledky vybraných testů v souborech formátu xls.

## Příloha C

# Adresářová struktura

Zdrojové kódy aplikace jsou organizovány v následující adresářové struktuře:

- `./Encog/`. Obsahuje knihovnu Encog pro práci s neuronovými sítěmi.
- `./InputData/`. Obsahuje základní třídy pro načtení dat.
- `./InputData/InSample/`. Třídy pro načtení trénovacích dat pro neuronové sítě.
- `./InputData/OutOfSample/`. Třídy potřebné pro načtení dat pro testování.
- `./InputHandler/`. Třídy pro načítání dat.
- `./InputHandler/TextFile/`. Třídy pro načítání dat ze souboru.
- `./Market/`. Obsahuje třídy související s trhem, tedy entity a příkazy.
- `./Market/Entities/`. Entity finančního trhu, tedy akcie, forex atd.
- `./Market/Orders/`. Příkazy podávané na burze.
- `./Model/`. Modely pro předpověď vývoje trhu.
- `./Prediction/`. Třídy pro předávání hodnot predikovaných modelem strategií.
- `./Statistics/`. Třídy ukládající statistiky.
- `./Strategy/`. Třídy pro strategie rozhodující se na základě předpovědi modelu.
- `./Test/`. Jednotlivé testy.
- Nezařazené jsou globální třídy, třídy s konstantami a jiné nezařaditelné třídy.

Pro volbu testu, který má být proveden, je ve třídě `Program` potřeba tento test odkommentovat. Změny parametrů implementovaných systémů mohou být provedeny přímo ve zdrojovém kódu nebo v konfigurační třídě s konstantami `GlobalConstants`. Více informací je v XML komentářích přímo ve zdrojových textech nebo v dokumentaci, která je z těchto komentářů generována a dostupná v souboru `bpDoc.XML`. Tato dokumentace obsahuje popis tříd a metod aplikace.

## Příloha D

# Tvorba vlastních testů a obchodních systémů

### D.1 Ukázka tvorby vlastního testu obchodního systému

Pro vytvoření vlastního testu obchodního systému je potřeba vytvořit třídu, která bude dědit od třídy `TradingSystemTest`. Tato nová třída poté musí přepsat metodu `runTest()`. V této metodě je následně vytvořen obchodní systém, vybrány jeho části a soubory pro testování a následně lze spustit test.

```
TradingSystem CompleteSystem = new TradingSystem();

//vybereme soubory, ze kterych se budou nacistat data
InputDataFileList files = new InputDataFileList();
files.addTrainFile(new InputDataFile("nasdaqNew.txt", 0, 10));
files.addTestFile(new InputDataFile("nasdaqNew.txt", 0, (decimal)19.75));

//vybereme komponenty do systemu
//napr error backpropagation nn, strategii cislo 1 atp
CompleteSystem.Initialize(new doubleMAPredictionModel(),
                          new doubleMATradingStrategy(),
                          new doubleMAInSampleData(),
                          new doubleMAOutOfSampleData(),
                          new doubleMAError(),
                          1000,
                          files
                          );

Sys = CompleteSystem;

//spustime beh simulace/testu
Sys.run();
```

Rozsah dat načtených ze souboru je volen parametry metody `addTrainFile()` resp. `addTestFile()`. Tato čísla vyjadřují procentuální část souboru, která bude načtena.

## D.2 Ukázka tvorby vlastního obchodního systému

Obchodní systém se skládá z několika částí. V této podkapitole bude předvedena jejich tvorba. Přesný popis metod a tříd lze dohledat v dokumentaci, která se nachází v souboru bpDoc.XML. Po vytvoření součástí obchodního systému lze systém sestavit a testovat. Samozřejmě lze také vytvořit pouze vybrané části obchodního systému a jako ostatní části využít již implementované třídy. Aplikace byla navržena s důrazem na obecnost a je tedy možné vytvořit libovolný obchodní systém.

### D.2.1 Načtení dat

Pro načtení zvolených dat je nutné vytvořit třídy dědicí od `InSampleData` a `OutOfSampleData`. Druhá jmenovaná třída zabezpečuje k načtení dat pro testování. Úkolem třídy dědicí od `InSampleData` je pouze načtení dat pro provedení výpočtů předcházejících testování. Obě tyto třídy musí přepsat metodu `loadData()`, která má k dispozici seznam souborů a rozsah dat, který z nich má být načten. Načtení dat je provedeno do seznamu entit, což je seznam objektů typu `StockMarketEntity`. Tento seznam je součástí třídy `TradingSystem`. Entitou v tomto kontextu myslíme instrument finančního trhu nebo indikátor technické analýzy. Další entity lze samozřejmě implementovat děděním právě od třídy `StockMarketEntity`. Následující příklad ukazuje vytvoření nové entity, představující komoditní futures. Tato entita je následně přidána do seznamu entit.

```
StockMarketProduct Product = new Futures(  
    "Corn", 10, 10, 10, 1, 1,10);
```

```
TimeSeries.Add(Product);
```

Parametry konstruktoru entity vyjadřují vlastnosti dané entity. Například pro komoditní futures musí být zadána velikost zálohy, poplatků za nákup a další. Každá z entit poté obsahuje dva seznamy dat. `InSampleData` a `OutOfSampleData`. První z nich vyjadřuje data načtená třídou dědicí od `InSampleData` a druhý obsahuje testovací data. Do těchto seznamů jsou načítány záznamy třídy `InputDataRecord`. `InputDataRecord` obsahuje hodnoty OHLC, objem obchodů a čas záznamu. Jejich načtení má zajistit právě metoda `loadData()`. Pokud by bylo potřeba více dat v záznamu, musí být navržena vlastní třída.

K dispozici je několik parserů, které dokáží načíst data ze souborů. Ty je možno nalézt ve složce `InputDataHandler/TextFile/`. Každý z parserů dokáže načíst jiný typ CSV souboru. Implementované parsery jsou určeny pro načtení dat ze souborů, které jsou dodány současně s prací.

```
//vytvorime parser  
CSVFileLoader Loader = new CSVFileLoader();  
  
//pouziti parseru pro nacteni dat  
Loader.loadProductFromCSVFile(nasdaq.OutOfSampleData,  
    files[0].Name,files[0].PercentFrom,files[0].PercentTo);
```

### D.2.2 Model pro předpověď budoucího vývoje trhu

Model pro předpověď budoucího vývoje je třída dědicí od třídy `PredictionModel`. Tato třída nemusí implementovat žádnou metodu, protože některé obchodní systémy predikce

nevyžadují. Pokud potřebuje model provést nějaké výpočty před samotným testováním, budou provedeny v metodě `train()`, kterou musí třída pro tento účel přepsat. K dispozici má tato metoda data načtená metodou `loadData()` třídy dědící od `InSampleData`.

Dále může nová třída přepsat metodu `getDataForPrediction()`. Tato metoda provádí načtení jednoho vzorku dat pro předpověď. Tento vzorek může obsahovat údaje z entit ze seznamu entit nebo i jiná, vypočtená data. Takový vzorek je poté předán metodě `predict()`.

Metoda `predict()` provede předpověď modelu na základě dat, která jsou jí předána jako parametr. Tato data jsou reprezentována seznamem hodnot typu `decimal`. Metoda vrací třídu, která dědí od třídy `Prediction`. Lze si tedy vytvořit vlastní třídu pro předávání předpovězených hodnot. Základní třída `Prediction` obsahuje pouze seznam hodnot typu `decimal`.

Žádná z těchto metod nemusí být přepsána, pokud model neprovádí předpovědi a v obchodním systému je využito jiného přístupu k rozhodování. Metody jsou v rodičovské třídě implementovány jako virtuální a může tedy být provedena jejich implicitní implementace.

### D.2.3 Rozhodovací strategie

Pro tvorbu vlastní rozhodovací strategie je potřeba vytvořit třídu, která dědí od třídy `TradingStrategy`. Tato nová třída musí přepisovat metodu `decide()`. V této metodě je přístupný seznam entit a také případná predikce předpovědovacího modelu. Ze seznamu entit lze získat například předcházející hodnoty některé z entit nebo zjistit, zda máme některou z entit aktuálně nakoupenou nebo prodanou. Na základě těchto informací je ve strategii implementován algoritmus pro rozhodování o chování obchodního systému.

Rozhodovací algoritmus může využívat metod třídy `StockMarketProdukt`, která dědí od třídy `StockMarketEntity` a jejíž instance jsou uloženy ve zmiňovaném seznamu entit třídy `TradingSystem`. Od třídy `StockMarketProduct` dále dědí třídy `Forex`, `Futures` a `Stock`, které představují instrumenty finančního trhu. Pro nákup a prodej instrumentu slouží metody `openNewTransactionBuy()`, `openNewTransactionSell()`, `closeExistingTransaction()`, `closeAllTransactions()`. Jsou vnitřně implementovány tak, aby reflektovaly zákonitosti skutečného obchodování v závislosti na obchodovaném instrumentu. Metody provádějí například výpočet stavu účtu, zaplacení poplatků nebo blokování zálohy. Mezi parametry těchto metod je například množství instrumentu, které má být nakoupeno nebo prodáno. Dále je k dispozici metoda pro položení obchodního příkazu `placeOrder()`. Tato metoda má několik přetížení pro různé typy příkazů.

```
//prodej instrumentu Product, 10 ks
StockMarketTransaction tr = Product.openNewTransactionSell(
    GlobalTime.Time, 10m, 1);

//polozeni obchodního prikazu movingstoploss s paramerty
tr.PlaceOrder(ORDERTYPE.MOVINGSTOPLOSS, Product.getCurrentValue().ClosePrice,
    Product.getCurrentValue().ClosePrice + 21m, ACTION.BUY, 10m, -20m, -20m);

//polozeni prikazu last-trading-day order
tr.PlaceOrder(ORDERTYPE.LTD, Product.getCurrentValue().ClosePrice,
    GlobalTime.Time.AddDays(7), ACTION.BUY, 10m);
```

Pro vyhledání aktuální hodnoty kurzu je implementována metoda třídy `StockMarketEntity` nazvaná `getCurrentValue()`. Pro získání předcházející hodnoty je připravena metoda `getNextValue()`. Obě metody vrací záznam `InputDataRecord`.



```

//vybereme trh, se kterym budeme obchodovat ze seznamu entit
StockMarketProduct Product = (StockMarketProduct) Products[0];

//predikce predpovedniho modelu
EBPNeuralNetworkPrediction EBPNNPred = (EBPNeuralNetworkPrediction)Pred;

//ziskame predchozi hodnotu (-2 by byla o dve predchozi, atp.)
InputDataRecord rec = Product.getNextValue(
    Product.OutOfSampleData,GlobalTime.Time,-1);

//ziskame soucasnou hodnotu (ekvivalentni k GetNextValue(,0);)
rec = Product.getCurrentValue();

```

Dále byla zmíněna možnost zjistit, zda máme některou z entit aktuálně nakoupenou nebo prodanou. Toho docílíme následovně.

```

//vybereme trh, se kterym budeme obchodovat ze seznamu entit
StockMarketProduct Product = (StockMarketProduct) Products[0];

//seznam otevrenych obchodu daného trhu
StockMarketTransaction trans = Product.OpenedPositions[0];

//počet otevrenych transakci
Int count = Product.OpenedPositions.Count;

```

Tyto transakce jsou typu `StockMarketTransaction`. Lze z nich zjistit informaci, zda se jednalo o nákup nebo prodej, čas a množství.

#### D.2.4 Výpočet chyby

Poslední třídou, která musí být obchodnímu systému při jeho tvorbě předána, je třída pro výpočet chyby. V této třídě, jejíž rodičovskou třídou bude `ResidualError`, mohou být přepsány metody `ComputeResidualAverage()` a `ComputeResidualForEach()`. První z těchto metod vypočítá jednotlivě chybu každé z předpovědí modelu pro předpověď budoucího vývoje. Druhá z nich vypočítá průměrnou chybu předpovědi. Tyto metody je třeba implementovat na základě použitého modelu pro předpověď. Výpočet chyb je proveden pro každý testovací vzorek.

```

Decimal err = Error.computeResidualsAverage(
    Pred, this.ParticipatedMarketEntities, GlobalTime.Time);

List<decimal> ress = Error.computeResidualsForEach(
    Pred, ParticipatedMarketEntities, GlobalTime.Time);

```

Pokud v aktuálním čase nelze některou z chyb vypočítat, je zachycena výjimka. Vypočtené chyby poté slouží ke tvorbě statistik