VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ
FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY
INSTITUTE OF INFORMATICS

VYUŽITÍ UMĚLÉ INTELIGENCE NA KAPITÁLOVÝCH TRZÍCH
THE USE OF ARTIFICIAL INTELLIGENCE ON STOCK MARKET

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Michal Skočík**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Jan Budík, Ph.D.**
SUPERVISOR

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA PODNIKATELSKÁ

Zadání diplomové práce

Ústav:	Ústav informatiky
Student:	**Bc. Michal Skočík**
Studijní program:	Systémové inženýrství a informatika
Studijní obor:	Informační management
Vedoucí práce:	**Ing. Jan Budík, Ph.D.**
Akademický rok:	2016/17

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává diplomovou práci s názvem:

Využití umělé inteligence na kapitálových trzích

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postupy
Teoretická východiska práce
Analýza problému
Vlastní návrhy řešení
Závěr
Seznam použité literatury

Cíle, kterých má být dosaženo:

Cílem práce je zkoumání modelů pro predikci budoucího vývoje hodnot na kapitálových trzích a následné zlepšování modelů.

Základní literární prameny:

DOSTÁL, P. Pokročilé metody analýz a modelování v podnikatelství a veřejné správě. 1. vyd. Brno: CERM, 2008. 432 s. ISBN 978-80-7204-605-8.

GOLDBERG, D. Genetic Algorithms in Search, Optimization, and Machine Learning. 1. vyd. USA: Addison-Wesley, 1989. 412 p. ISBN 978-0-201157-67-3.

GRAHAM, B. Inteligentní investor. 1. vyd. Praha: GRADA, 2007. 504 s. ISBN 978-80-247-1792-0.

REJNUŠ, O. Finanční trhy. 1. vyd. Ostrava: KEY Publishing, 2008. 548 p. ISBN 978-80-87-8.

WILLIAMS, L. Long-Term Secrets to Short-Term Trading. 1. vyd. USA: Wiley-Interscience, 1999. 255 p. ISBN 0-471-29722-4.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2016/17

V Brně dne 28.2.2017

L. S.

doc. RNDr. Bedřich Půža, CSc. ředitel	doc. Ing. et Ing. Stanislav Škapa, Ph.D. děkan

ABSTRAKT

Práce je zaměřena na problematiku týkající se umělých neuronových sítí a jejich využití na kapitálových trzích. V práci je vytvořen program, pomocí kterého se načtou vstupní data a vytvoří se neuronová síť, která slouží k predikci vývoje cen akcií. Tento program je vytvořen pomocí výpočetního prostředí MATLAB. Vytvořená síť je testována na simulaci obchodního modelu. Na základě výsledku simulace jsou diskutovány výsledky.

ABSTRACT

Diploma thesis is focused on problematics of artificial neural networks and their usage on capital markets. There is a software created as a part of this diploma thesis which can load input data and create neural network that serves for share price forecast. This program is created in numerical computing environment MATLAB. Created neural network is tested under simulation of business model. Results are discussed upon examination of results of simulation.

KLÍČOVÁ SLOVA

Umělá inteligence, umělé neuronové sítě, kapitálové trhy, akcie, predikce časových řad, MATLAB, NAR, NARX, technická analýza.

KEYWORDS

Artificial intelligence, artificial neural networks, stock market, stocks, time series prediction, MATLAB, NAR, NARX, technical analysis.

BIBLIOGRAFICKÁ CITACE PRÁCE

SKOČÍK, M. *Využití umělé inteligence na kapitálových trzích.* Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2017. 85 s. Vedoucí diplomové práce Ing. Jan Budík, Ph.D..

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že jsem diplomovou práci zpracoval samostatně s použitím uvedené literatury a pod vedením svého vedoucího diplomové práce. Prohlašuji, že citace uvedených pramenů je úplná, a že jsem v práci neporušil autorská práva (ve smyslu zákona č. 121/2000 Sb. o právu autorském a o právech souvisejících s právem autorským).

V Brně dne .....

Michal Skočík

PODĚKOVÁNÍ

Rád bych poděkoval svému vedoucímu diplomové práce Ing. Janovi Budíkovi Ph.D. za jeho cenné rady a připomínky při zpracování diplomové práce.

OBSAH

ÚVOD **10**
CÍLE PRÁCE, METODY A POSTUPY **11**
1 TEORETICKÁ VÝCHODISKA PRÁCE **13**
1.1 Finanční trhy 13
1.1.1 Dělení finančních trhů 13
1.1.2 Trh NASDAQ 17
1.1.3 Fundamentální analýza 17
1.1.4 Psychologická analýza 18
1.1.5 Technická analýza 20
1.2 Časové řady 24
1.2.1 Hurstův exponent 25
1.3 Umělá inteligence 27
1.3.1 Fuzzy logika 27
1.3.2 Genetické algoritmy 28
1.3.3 Neuronové sítě 29
2 ANALÝZA PROBLÉMU **37**
3 VLASTNÍ NÁVRHY ŘEŠENÍ **43**
3.1 Popis systému 43
3.1.1 Metoda NAR a metoda NARX 44
3.1.2 Úprava časové řady 45
3.1.3 Trénování sítě 46
3.1.4 Model NAR 47
3.2 Optimalizace sítě 51
3.2.1 NETFLIX 52
3.2.2 APPLE 53
3.2.3 GOOGLE 54
3.2.4 AMAZON 55
3.2.5 YAHOO 56
3.2.6 Shrnutí výsledků 57
3.2.7 NETFLIX, APPLE, YAHOO 57
3.3 Simulace sítě 64

3.3.1 Nákup ... 65
3.3.2 Prodej ... 66
3.3.3 Realizování ztráty x zisku z nákupu ... 66
3.3.4 Realizování ztráty x zisku z prodeje ... 67
3.3.5 Výstup simulace ... 68
3.3.6 Běh simulace pro optimalizované sítě ... 68
3.4 Predikce ... 69
3.5 Shrnutí výsledků ... 75
3.5.1 Rozhodování ... 76
ZÁVĚR ... 78
SEZNAM POUŽITÉ LITERATURY ... 80
SEZNAM VZORCŮ ... 82
SEZNAM TABULEK ... 82
SEZNAM GRAFŮ ... 83
SEZNAM OBRÁZKŮ ... 83

ÚVOD

Umělá inteligence je známa už nějakou dobu. Počátky neuronových sítí lze datovat do 40. let 20 století. Jejímu rozvoji dříve bránily nedostatečné technické prostředky, v dnešní době je tomu však jinak. Obor umělé inteligence se neustále rozšiřuje a pro dnešní svět jsou charakteristické těžko řešitelné nedeterministické problémy, které se bez výpočetní techniky vyřešit nedají. Využití umělé inteligence můžeme najít v mnoha oborech, příkladem může být nově vyvinutý systém firmou Google, který vylepšil svůj překladač za použití neuronových sítí. Umělá inteligence se však hojně také využívá v dalších oborech jako je medicína, plánování a rozhodování. V práci bude předvedena ukázka využití umělé inteligence na kapitálovém trhu.

Touha po znalosti budoucnosti je velmi vysoká, je však nemožné s naprostou určitostí tvrdit, že dokážeme odhadnout, co se v danou dobu stane. Moderní technologie jako je i umělá inteligence se dokáží co nejvíce přiblížit k odhadu nejbližší budoucnosti, je však zapotřebí využít zdravého rozumu a faktu, že ani umělá inteligence nám nedá stoprocentní předpověď, pouze se nám pokusí nastínit, co by se s největší pravděpodobností mohlo stát.

V této práci budou zkoumány takové modely, které se budou snažit předpovídat budoucnost pomocí časových řad, které budou představovat akcie velkých společností. V první části práce bude vymezení cíle. Další částí práce pak seznámení se s finančním trhem a umělou inteligencí z teoretického hlediska. V praktické části bude popsán návrh a implementace predikčního nástroje podle stanovených cílů. V poslední části následuje zhodnocení práce, dosažené výsledky a splnění cílů.

CÍLE PRÁCE, METODY A POSTUPY

Hlavním cílem diplomové práce je optimalizace modelu vytvořeného pomocí umělých neuronových sítí. V práci je vytvořeno více modelů a na základě zkoumání je vybrán model, který nejlépe vystihuje daný problém a nejlépe predikuje vývoj akciového trhu. Tyto modely jsou vytvářeny na časových řadách, které odpovídají vývoji akciového trhu velkých společnosti jako je NETFLIX, APPLE, GOOGLE, AMAZON, YAHOO.

K optimalizaci modelu slouží především vhodný výběr počtu neuronů a velikosti zpoždění. Tohoto cíle bude dosaženo díky využití tzv. *for* cyklu, který několikrát projde většinu možností vícekrát po sobě. Vhodný počet neuronů a vhodný počet zpoždění jsou zvoleny podle nejmenší kvadratické chyby (Mean Square Error).

Pro ještě lepší optimalizaci je vytvořen algoritmus, pomocí kterého se počítá odhad úspěšnosti trendu na historických datech. Nejlepší model je pak vybírán kompromisem mezi nejmenší kvadratickou chybou a nejlepším odhadem úspěšnosti trendu.

V další fázi optimalizace je opět aplikován *for* cyklus, avšak již se zvoleným počtem neuronů a zpoždění. Díky tomu, že rozdělení synaptických vah je rozděleno náhodně, je potřeba síť několikrát přetrénovat, abychom dostali co nejlepších výsledků.

Na tomto modelu je pak využito testování na historických datech pomocí automatického backtestu. K tomu je vytvořena obchodní simulace, která podle stanovených pravidel nakupuje a prodává, a která využívá síti již známá data. Ve výsledku je tak znám zisk/ztráta, počet dobrých obchodů a počet špatných obchodů.

Po dokončení optimalizace následuje pokus o predikování budoucích hodnot akciového trhu a jeho porovnání s reálnými výsledky trhu. Tento test probíhá také na historických datech. K tomu je použita opět obchodní simulace, která funguje stejně jako na historických datech. Pro naučenou neuronovou síť jsou však tato nová data neznámá. Neuronová síť vždy nejdříve predikuje jednu hodnotu a až potom je jí odhalena skutečná hodnota proto, aby mohla predikovat další hodnotu.

Dílčím cílem je aplikování všech optimalizačních kroků pro vytvoření modelu, který je schopen sám rozpoznat ideální složení neuronů a zpoždění s minimalizací kvadratické chyby. Na základě tohoto modelu je uživateli predikována hodnota časové řady a rozhodnutí, zda na trh vstoupit do pozice buy nebo sell.

1 TEORETICKÁ VÝCHODISKA PRÁCE

V této kapitole budou popsána teoretická východiska diplomové práce. Nejdříve bude popsán finanční trh jako takový, pak jeho funkce a dělení. Budou popsány analýzy zkoumající trh, a to psychologická, fundamentální a technická. Největší část bude věnována poslední zmíněné analýze. V další podkapitole se podíváme na problematiku umělé inteligence, kde si řekneme základní pojmy k fuzzy logice, genetickým algoritmům a neuronovým sítím.

1.1 Finanční trhy

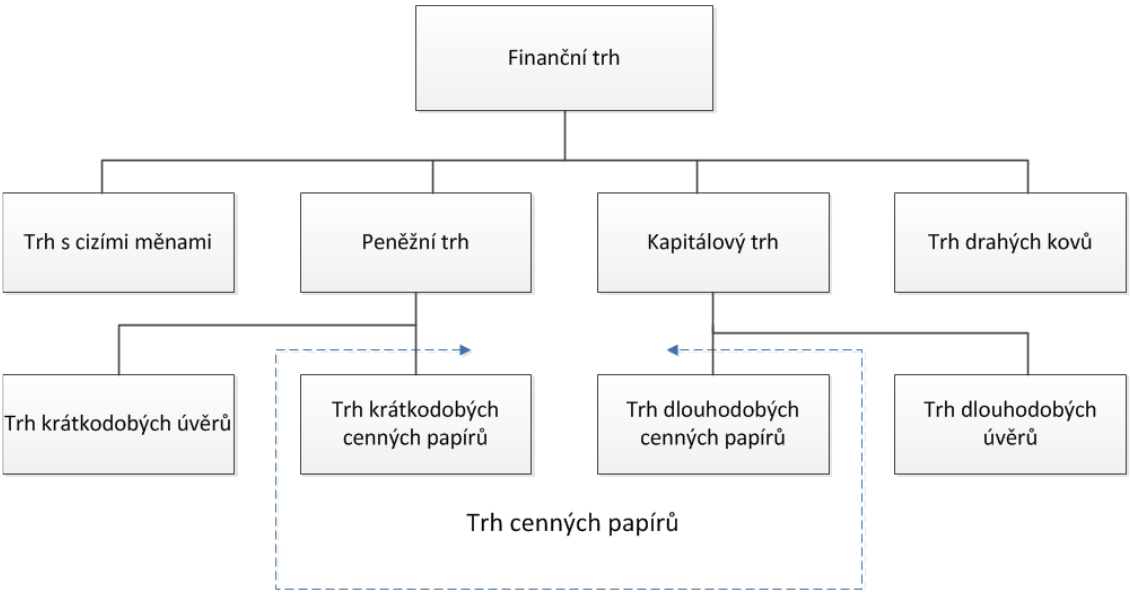
Finanční trhy jsou velmi obsáhlé, a proto se podíváme jen na stručné vymezení nezbytné k porozumění.

Finanční trh můžeme chápat jako systém institucí a instrumentů, zabezpečující pohyb peněz a kapitálu (nabízeného ve formě cenných papírů) ve všech jeho formách mezi různými ekonomickými subjekty; a to na základě poptávky a nabídky (1).

Hlavní principem finančních trhů v dnešní době je umožnit přesun peněžních prostředků od subjektů, které mají nadbytek peněžních prostředků (nabídková strana), k subjektům, které potřebují více peněžních prostředků, než jaký zrovna mají (poptávková strana). Tímto trh plní alokační funkci. Finanční trhy také plní funkci likvidity, což znamená, že umožňují přeměnu různých druhů finančních aktiv na peněžní hotovost. Funkcí finančního trhu je financování provozního kapitálu podniků, poskytování krátkodobých finančních úvěrů, a také poskytování financí pro spekulativní nákupy cenných papíru a komodit (2, 3).

1.1.1 Dělení finančních trhů

Finanční trhy se zpravidla dělí podle druhu finančních instrumentů, který se v daném segmentu trhu obchoduje. Na obrázku níže, je zobrazeno obvyklé členění.



Obrázek 1: Členění finančních trhů

Zdroj: [2, str. 61]

Trh s cizími měnami

Do trhu s cizími měnami můžeme zařadit valuty, hotovost, devize, bezhotovostní pohledávky (peníze na účtech v bankách, šeky, směnky a další). Kurzy jednotlivých měn se tvoří na devizovém trhu klasickým způsobem nabídky a poptávky. Valuty jsou z těchto cen odvozeny připočtením nákladů vznikajících z operací s hotovostí (2).

Peněžní trh

Peněžní trh se dělí na krátkodobé úvěry a krátkodobé cenné papíry. U peněžních trhů je tedy důležitá krátkodobost. Ta se vztahuje na splatnost úvěru či cenného papíru a její časové vymezení je zpravidla do jednoho roku (2).

Kapitálové trhy

Kapitálové trhy se dělí stejně jako peněžní trhy na úvěry a cenné papíry, u kapitálových trhů se však nejedná o krátkodobé, ale naopak dlouhodobé úvěry a dlouhodobé cenné papíry. Dlouhodobé znamená, že mají splatnost delší než jeden rok. Je zde zřejmá návaznost na peněžní trhy, protože pokud bude pohledávka splacena za dobu kratší než jeden rok, tak je převedena do peněžního trhu. Kapitálové trhy mají zpravidla větší výnosnost než peněžní trhy, ovšem podléhají vyššímu riziku kvůli jejich dlouhodobosti (2).

Dlouhodobé cenné papíry vs. Dlouhodobé úvěry

Dlouhodobé cenné papíry, nejčastěji akcie a obligace, mají oproti dlouhodobým úvěrům hlavně tu výhodu, že jsou převoditelné. Investor je po zakoupení nemusí držet po celou dobu jejich splatnosti, ale může je dále prodat. Dlouhodobé úvěry jsou zpravidla zabezpečovány jediným subjektem, většinou se jedná o bankovní subjekt, a tím vzniká velké riziko spojené s účastí pouze dvou stran v daném obchodu. Cenné papíry jsou naopak nakupovány velkým množstvím investorů, čímž se riziko rozloží (3).

Trh drahých kovů

Nejčastějšími obchodovanými drahými kovy na trhu drahých kovů jsou zlato, stříbro, platina a palladium.

Důvodem proč tyto drahé kovy nepatří do běžných komodit je, že zejména zlato a stříbro plnilo historicky jiné funkce (2).

Primární trh

Na tento trh se umisťuje prvotní kapitál. Patří sem poprvé obchodovatelné cenné papíry. Zejména se tu objevují velcí investoři, kteří mohou investovat značné objemy do méně známých instrumentů (2).

Sekundární trh

Jak je zřejmé z předchozího tvrzení, na sekundárním trhu tak budou cenné papíry, které byly dříve již alespoň jednou zakoupené. Cenné papíry se na tomto trhu můžou nakoupit a prodat mnohokrát, kdežto na primárním můžou být pouze jedenkrát (2).

Veřejné trhy

Na veřejných trzích se po emitentovi (tj. vydavateli finančního instrumentu) požaduje přísné plnění požadavků, a to informačních, administrativních, právních a jiných. Je to z důvodu ochrany investorů, kteří na veřejném trhu nemají přímý kontakt s emitentem tak jako na neveřejném trhu (4).

Neveřejné trhy

Na neveřejných trzích se z důvodu náročnosti na informace a právní zajištění vyskytují převážně velcí investoři, jelikož si dokážou smluvní náležitosti obchodu zajistit s relativně nižšími náklady (4).

Akcie

Akcie je majetkový cenný papír definovaný obchodním zákoníkem a zákonem o cenných papírech. Akcie svému majiteli přináší určité výhody. Nejdůležitější je právo na dividendy, což je podíl na zisku společnosti. Výše dividendy je určena valnou hromadou společnosti složenou z akcionářů. Valná hromada může rozhodnout například i o tom, že dividenda nebude vyplacena vůbec. Dividenda například nesmí být vyplacena, pokud by to mělo způsobit platební neschopnost podniku. Vlastník akcie má právo účastnit se valných hromad, může předkládat svoje návrhy a získává také hlasovací právo. Při likvidaci podniku má akcionář právo na podíl na likvidačním zůstatku (3, 5).

Pojmy spojené s akciemi:

- Autorizované akcie (počet akcií, které může společnost podle stanov vydat).
- Základní kapitál (počet emitovaných akcii násobených nominální hodnotou akcie).
- Emisní ážio (Rozdíl mezi nominální a tržní cenou akcie (součást kapitálových fondů).
- Zadržené akcie (společnost emitovala a později sama odkoupila)

Členění akcií

Akcie můžeme rozdělit podle více kritérií.

Dělení z pohledu práv, která jsou spojena s vlastnictvím:

- **Kmenové akcie:** Trvalá reziduální forma vlastnictví, výnosy jsou až po uspokojení státu, věřitelů a majitelů prioritních akcií.
- **Prioritní akcie:** Trvalý zdroj financování, zpravidla fixní příjem. Majitelé nemají hlasovací právo a mají prioritu při výplatě dividend a podílu na likvidačním zůstatku.

Dělení z hlediska formy:

- **Akcie na doručitele:** Jsou volně převoditelné, Je uzavřena smlouva, která nemusí být písemného charakteru. Právo má vždy aktuální držitel akcie.
- **Akcie na jméno:** Převoditelnost je vázána na souhlas emitenta, převod je možný rubopisem. Vykonávat práva spojená s držením akcie smí jen osoba, která je uvedena v seznamu akcionářů (5).

1.1.2 Trh NASDAQ

Trh NASDAQ je třetím největším akciovým trhem na světě. Název NASDQA je zkratkou pro anglický název National Association Securities Dealers Automatic Quotation a vznikl v roce 1971. NASDAQ je decentralizovaný elektronicky řízený trh a je financován účtováním poplatků firmám, které na něm kótují své akcie.

Společnost, která chce na tento trh vstoupit musí splňovat minimální požadavky vstupu, jedná se například o minimální hodnotu čistých aktiv. Ta musí dosahovat nejméně čtyři miliony dolarů a musí mít alespoň pět set tisíc veřejně obchodovatelných akcií.

1.1.3 Fundamentální analýza

Fundamentální analýza se snaží najít vnitřní hodnotu akcie, s cílem zjistit, zda je cena akcie nadhodnocená, podhodnocená nebo zda je ohodnocení odpovídající. Pokud je akcie podhodnocená znamená to, že se očekává její růst v budoucnosti, je-li nadhodnocená, předpokládá se její pokles. Opírá se o veřejně dostupné ekonomické údaje, jako jsou cena ropy, kurzy měn, nezaměstnanost, inflace, DPH, peněžní zásoba a vývoj úrokových měr. Fundamentální analýza se dělí na tři úrovně a to globální, odvětvové a firemní (6).

Makroekonomická fundamentální analýza (globální analýza) zahrnuje do hodnoty akcie vliv ekonomiky jako celku. Spadá do ní, peněžní zásoba, inflace, vývoj úrokových měr, HDP. Je však důležité počítat také s vlivem politické situace (6).

Odvětvová fundamentální analýza se zaměřuje spíše na situaci v daném oboru, v jakém společnost působí. V odvětvové analýze počítáme s ukazateli jako citlivost odvětví na hospodářský cyklus, tržní struktura odvětví, způsoby státní regulace odvětví, perspektivy budoucího vývoje odvětví. Odvětvová analýza se tedy zaměřuje na identifikaci

charakteristických znaků odvětví a předpovídání vývoje v jednotlivých odvětvích. Z hlediska citlivosti se fundamentální analýza rozděluje na:

Cyklická odvětví

Příkladem zde může být stavební průmysl a automobilový průmysl. V období expanze tyto odvětví dosahují dobrých hospodářských výsledků, v případě recese se však potýkají s problémy.

Neutrální odvětví

Příkladem pro toto odvětví je potravinářský průmysl nebo farmaceutický průmysl, jedná se totiž o odvětví, kde se nacházejí nezbytné statky.

Anticyklická odvětví

Pro anticyklická odvětví platí opak toho, co v cyklických, a tedy v době recese tyto společnosti vykazují lepší výsledky jak v době expanze. Jedná se například o levné zábavní průmysly.

Firemní analýza se snaží určit, zda jsou hodnoty akcii nadhodnoceny, podhodnoceny nebo jsou oceněny správně, také pomocí finanční analýzy určit kvalitu finančního hospodaření a prognózu do budoucích let (6).

1.1.4 Psychologická analýza

Psychologická analýza vychází z předpokladu, že pohyb cen akcii je ovlivněn psychikou investorů. Masová psychologie působí na burzovní publikum a tím ovlivňuje úroveň kurzů. Jednodušeji to jde tedy chápat tak, že budoucí vývoj akcií je závislý na impulzech, které investoři sledují a nechají se jimi ovlivnit buďto k nákupu nebo k prodeji.
Keynes ve své době tvrdil následující tvrzení:

„Nemá žádného smyslu zaplatit 25 peněžních jednotek, za investici, o níž podle jejího perspektivního výnosu důvodně soudíte, že má sice hodnotu 30, avšak předpokládáte, že za tři měsíce ji trh ocení pouze na 20“ (4, str. 209).

Ve své podstatě z toho jde vyvodit závěr, že investor musí věnovat značnou pozornost budoucímu vývoji, či spekulativním odhadům a vhodným způsobem je využívat ve svůj prospěch (4).
Na psychologickou analýzu můžeme nahlížet z více různých teoretických koncepcí, nejvlivnější investiční teorie jsou:

Keynesova investiční psychologie

Zásadní vliv na vývoj akciových kurzů mají faktory jako:

- Struktura vlastníků akcií – Zvyšuje se podíl vlastníků akcii, kteří nemají dostatečné znalosti k jejich ohodnocení.
- Akciové kurzy – Nadměrně reagují na události, které na akciový trh nemají skutečný dopad, což může způsobit vyšší volatilitu kurzů.
- Chování investorů je ovlivněno kolektivní psychologií neinformovaných jednotlivců.
- Investiční rozhodování je zaměřeno na odhadování budoucích reakcí investorů.

Kostolanyho psychologický přístup

V krátkém období jsou akciové kurzy ovlivňovány psychologickými reakcemi investorů na různé události. Ve středním a dlouhém období se za hlavní kurzotvorné faktory předpokládají fundamentální ukazatele.

Podle trhu jsou účastníci rozděleny na dvě skupiny:

Hráči – ti nejednají na základě fundamentálních faktoru, ale pouze na základě nových událostí. Chování hráčů je iracionální a reagují davově, neboli nakupují a prodávají podle toho, jak to dělá většina ostatních investorů. Snaží se dosáhnout zisku co nejrychleji a tvoří přibližně 90 % investorů.

Spekulanti – na rozdíl od hráčů, realizují transakce spíše dlouhodobého charakteru, chovají se racionálně a většinou nereagují davově, ale jdou spíše proti davu. Obchodují pouze na základě podložených prognóz a jsou úspěšnější než hráči.

Díky těmto dvěma účastníkům trhu, tato teorie tvrdí, že se budoucí kurz bude měnit podle toho, která z těchto dvou skupin drží většinu akcií, proto je důležité identifikovat, která ze skupin se na konkrétním trhu nachází (4).

Teorie spekulativních bublin

Snaží se popsat situace, kdy kurzy akcií po určitý čas rostou a potom dojde k náhlému poklesu. Pro tyto změny není možné nalézt fundamentální vysvětlení. Tyto fundamentálně nevysvětlitelné skoky se nazývají jako spekulativní bubliny. Podle této teorie je to díky jevům masové psychologie, která souvisí s reakcemi na různé události. Díky tomuto z pohledu investorů hrozí velká ztráta, protože ti můžou v určitém období očekávat nárůst a začít nakupovat, a tím vznikne spekulativní bublina a investor se později dostane do ztráty (4).

1.1.5 Technická analýza

Technická analýza zkoumá historická data, na základě kterých se pomocí časové analýzy snaží určit budoucí vývoj daného instrumentu. Práce s historickými soubory je výpočetně náročná a je tak závislá také na rozvoji počítačů a informačních technologií.
Technická analýza se tedy zabývá, jakým směrem se hodnota akcie vydá. Pokud trh či akcie roste, nazýváme trh či akcii „býčí“. Naopak pokud trh či akcie klesá, hovoříme o „medvědím“ trhu či akcii. Můžeme takhle označit i investory, pokud investor očekává růst, říkáme mu „býk“, pokud pokles je to pro nás „medvěd“ (2).
Východiskem pro technickou analýzu je tak zvaná Dowova teorie.
Charles H. Dow vycházel z pozorování, že většina akcií na trhu má podobné chování. Domníval se tedy, že vývoj trhu lze popsat pomocí akciových indexů. Murphy tuto teorii shrnul do následujících šesti výroků (7).

1. Akciové indexy v sobě mají zahrnuty již všechny známé a důležité informace
2. Pohyb na trhu se skládá ze tří druhů trendů – primárního, sekundárního a terciálního.
3. Primární tržní trend má tři fáze – akumulační, participační a distribuční.

4. Akciové indexy se musí navzájem potvrzovat, aby se jednalo o spolehlivý tržní signál.
5. Objem obchodů musí potvrzovat trend.
6. Trend je považován za platný, dokud nedá jasný signál o svém obratu.

Dále Murphy uvádí tři předpoklady, které jsou základními stavebními kameny technické analýzy (7).

1. Trh diskontuje všechno – V ceně akcie jsou zahrnuty již všechny známé faktory, které ji mohou ovlivnit.
2. Ceny se pohybují v trendech
3. Historie se opakuje – Analýzou historických dat, byl objeven výskyt určitých vzorců ve vývoji cen. Tyto vzorce se v čase opakují. Díky tomu, že vzorce odráží vliv lidské psychiky a chování v určitých situacích dá se předpokládat, že se vzorce budou nadále opakovat. Cílem je najít vzorec a vhodným způsobem na něj reagovat (7).

Metody technické analýzy:

Grafické metody:

Grafickými metodami se myslí čtení grafů, pomocí kterých můžeme identifikovat různé trendy, hledat známé vzorce či formace. Je-li nějaká formace v grafu objevena, můžeme na ní reagovat a předpovědět další možný vývoj.

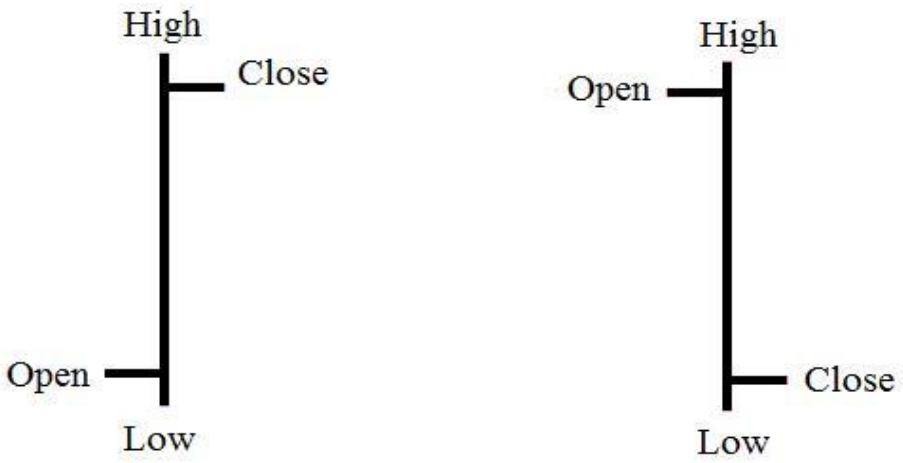
Můžeme rozlišit 3 druhy trendu. Rostoucí trend (uptrend), klesající trend (downtrend) a boční trend, někdy nazýván jako horizontální trend.

Dowova teorie klasifikuje trendy do 3 kategorií podle doby vlivu. Primární trend působí delší dobu než jeden rok, sekundární trend trvá od tří týdnu až po jeden měsíc a terciální trend je krátkodobý trend nejvýše v délce tři týdny.

Při studii grafů se dále setkáme s pojmy jako je hladina podpory (support) a rezistence (resistance). Tyto pojmy hrají při obchodovaní významnou psychologickou roli. Hladina podpory označuje úroveň cen, pod kterou většina investorů nechce prodávat. Hladina rezistence naopak nejvyšší cenu, za kterou většina investorů je ochotna zaplatit. Situaci, kdy se cena akcie dostane nad hranici rezistence nebo klesne pod hranici podpory, se říká průlom.

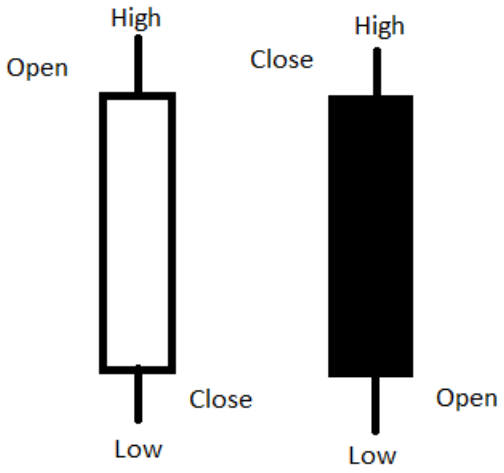
Pro obchodování a sledování vývoje akcií se nejvíce využívají dva typické způsoby konstrukce grafů. Jedná se o OHLC neboli sloupcový graf a svícový graf.

OHLC graf zobrazuje na vodorovné ose x čas ve zvolených jednotkách a na ose y jsou ceny akcií pomocí svislých úseček. Na těchto úsečkách jsou naznačeny ceny Open, High, Low, Close. Kde Open je cena akcie, se kterou byl trh daný den otevřen. Close je uzavírací cena daného dne. High označuje nejvyšší dosaženou cenu v daném dni a Low nejnižší cenu daného dne. Někdy se v grafu muže vyskytovat hodnota Volume, což je obchodovaný objem daného dne (7).



Obrázek 2: OHLC graf
Zdroj: (Vlastní)

Svícový graf zobrazuje stejné informace jako OHLC graf, ale využívá k tomu graficky lépe čitelný způsob. Hodnoty Open, High. Low, Close jsou zobrazeny na takzvané svíci, která je rozdělená na tělo a stíny. Pro tělo se používají dvě barvy, jedna je pro svíci klesající a druhá pro rostoucí. Tělo svíce představuje rozpětí mezi hodnotami otvírací a zavírací ceny. Stíny zobrazují rozpětí mezi minimální a maximální cenou (7).



Obrázek 3: Svícový graf
Zdroj: (Vlastní)

Technické indikátory:

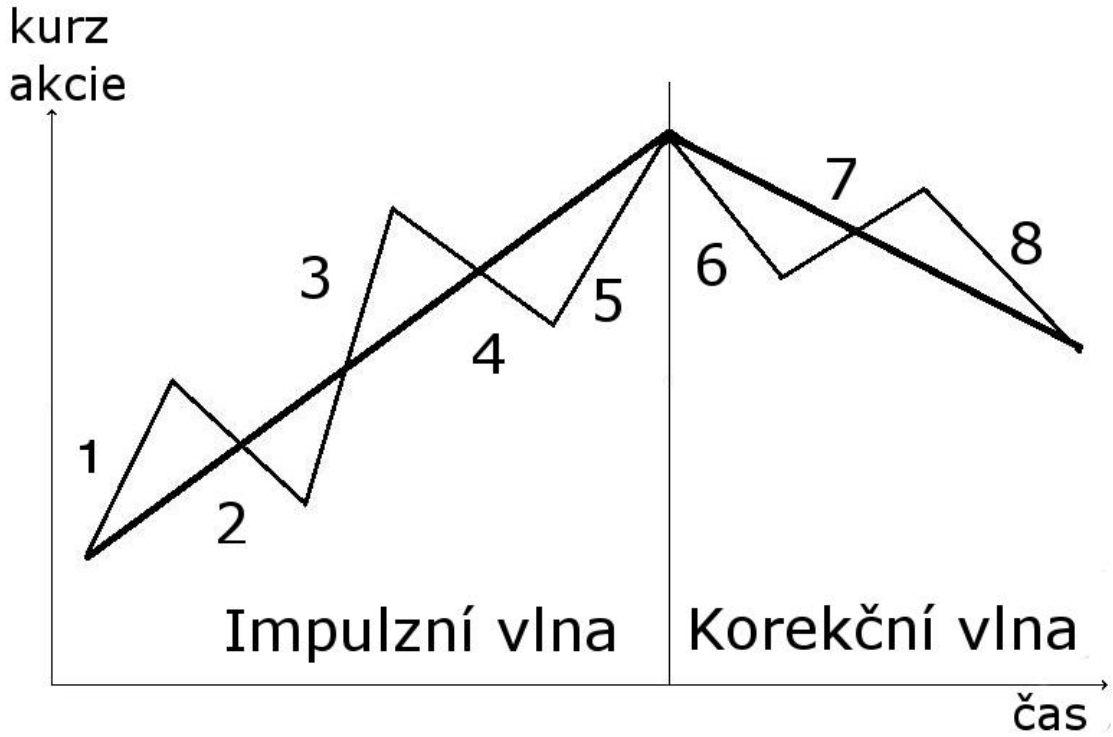
Klouzavé průměry (moving average) – Jednoduchá pomůcka pro mechanické vyrovnání časové řady. Můžeme pro ně použít buď jednoduchý aritmetický průměr, nebo používat průměry vážené či různé další výpočty průměru. Zvýrazňují trend a odstraňují stochastickou složku. Do analýzy vstupuje velké množství historických dat. Podle toho kolik těchto dat do analýzy zahrneme, tak se mění reakční doba změny. Tento problém můžeme eliminovat pomocí Exponencial Moving Average. Klouzavé průměry můžeme použít jako samostatný indikátor, pomocí kterého se investor může rozhodnout o tom, zda uskuteční obchod či nikoliv (8, 9).

Bolingerova pásma – Indikátor umožňující zobrazit volatilitu a cenovou úroveň instrumentu. Index tak zobrazuje relativní cenu instrumentu k současné situaci na trhu. Bolingerova pásma jsou tvořena klouzavými průměry, které tvoří střed a horní, respektive spodní linie. Hodnoty jsou vytvořeny přičtením nebo odečtením určitého počtu směrodatných odchylek (9).

MACD (Moving average Convergence Divergence) – Používá se pro identifikaci nákupních a prodejních signálů na trhu nebo potvrzení současného trendu. Používá se nejčastěji ve formě dvanácti nebo dvaceti šesti denních klouzavých průměrů s nulovou křivkou, vodorovnou osou x a procházejícím bodem rovnosti dvanácti a šestnácti denních klouzavých průměrů (7).

Stochastický oscilátor – Indikátor hybnosti, který popisuje umístění aktuální zavírací ceny akcie, s ohledem na rozpětí nejvyšší a nejnižší ceny v průběhu určitého časového období. Výhodou metody je schopnost předpovídání zvratů investičních nástrojů několik dní dopředu. Je tedy možné vytvořit vrchol a následný očekávaný pokles ještě před tím, než takováto situace nastane v reálném prostředí (7).

Elliottovy vlny – Hledají opakující se vzory v časové řadě na finančním trhu. Základním principem Elliottových vln je, že v impulzivní vlně, složené z pěti podvln, trh výrazně roste. Vlivem opravné vlny, složené ze tří podvln trh mírně poklesne. Celkem se tak Elliottova vlna skládá z osmi vln a dvou fází. Takovéto Elliottovy vlny pak ve skutečné časové řadě vytvářejí opakující se vzory větších a větších velikostí (2, 8).



Obrázek 4: Elliottovy vlny
Zdroj: [2, str. 287]

1.2 Časové řady

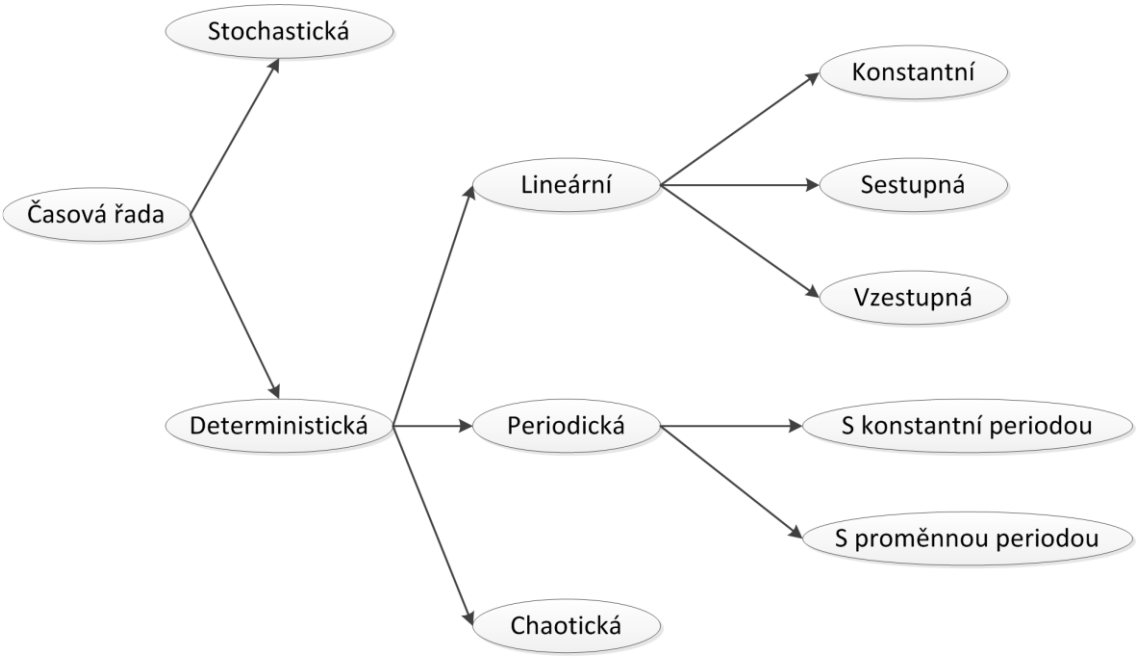
Historická data na kapitálových trzích tvoří časové řady. Proto je důležité je brát v potaz a bude zde nastíněna jejich problematika.

Časová řada je posloupnost číselných hodnot x v závislosti na čase t. Můžeme ji matematicky zapsat jako vektor:

$$\vec{x} = (x_1, x_2, \dots, x_N) \tag{1.1}$$

Indexy jednotlivých hodnot představují diskrétní vzorkování v čase t, kde t náleží množině <1, N>. Úsek mezi vzorky časové řady je vždy konstantní. Frekvence vzorkování může být v řádech minut, hodin, dnů, měsíců nebo i let, podle požadované přesnosti měření.

Časové řady se mohou skládat z deterministické nebo ze stochastické složky. Ty se dále můžou dělit. Rozdělení časové řady je vidět na následujícím obrázku (10).



Obrázek 5: Dělení složek časové řady

Zdroj: [10, str. 114]

1.2.1 Hurstův exponent

Hurstův exponent H do určité míry určuje chaotičnost časové řady. Dokáže rozlišit chaotickou a náhodnou časovou řadu, a také je schopen u chaotické časové řady najít dlouhodobý paměťový cyklus.

Výpočet Hurstova exponentu:

$$\overline{x}(r) = \frac{1}{r}\sum_{i=1}^{r} x_i \tag{1.2}$$

$$X(t,r) = \sum_{i=1}^{t}[x_i - \overline{x}(r)] \qquad t = 1,2,\dots,r \tag{1.3}$$

$$S(r) = \sqrt{\frac{1}{r}\sum_{i=1}^{r}[x_i - \overline{x}(r)]^2} \tag{1.4}$$

$$R(r) = \min_{1\le t\le r} X(t,r) - \min_{1\le t\le r} X(t,r) \tag{1.5}$$

$$H(r) = \frac{\log\frac{R(r)}{S(r)}}{\log(r)} \tag{1.6}$$

Kde $x = (x_{1,\dots,}x_t)$ je časová řada o r hodnotách. Podle hodnot $H(r)$ lze charakterizovat časovou řadu následovně:

- H ≈ 0,5 představuje náhodnost časové řady, hodnoty mají normální rozdělení a neexistuje korelace mezi současnými a budoucími hodnotami. Tyto časové řady jsou velice těžko predikovatelné.
- H = $\langle 0\ ;\ 0{,}5)$ představuje časovou řadu, která obsahuje dlouhodobý paměťový cyklus a vykazuje antiperspektivní chování. To znamená, že vzestupný trend bude následován sestupným a naopak.
- H = $\langle 0{,}5\ ;\ 1)$ představuje časovou řadu, která obsahuje dlouhodobý paměťový cyklus a vykazuje persistentní chování. To znamená, že vzestupný trend bude následován opět vzestupným trendem a naopak.

Hurstův exponent poskytuje ohodnocení prediktability časové řady v krátkém období. Často se při predikčních aplikacích používá v kombinaci s Lyapunovým exponentem pro potvrzení prediktability časové řady a ohodnocení spolehlivosti predikce.

1.3 Umělá inteligence

Inteligencí se rozumí schopnost učit se věci a být schopný intuitivně řešit problém, kterému je jedinec vystavený. Jako vědní obor umělá inteligence simuluje inteligentní řešení problému za pomocí strojů a výpočetní techniky. Tento obor se snaží napodobovat lidské chování, vyhodnocovat situace, analyzovat podněty z prostředí, případně vytvářet tvořivou činnost. Původní snaha byla zaměřena na zkopírování lidského chování a jeho následné zdokonalování (11).

1.3.1 Fuzzy logika

Fuzzy logika se objevila v roce 1965, kdy byla L. Zadechem, definována jako fuzzy množina. Slovo fuzzy můžeme přeložit jako neostrý, matný, mlhavý, neurčitý. Vzhledem k zaměření oboru je tento překlad vystihující. Fuzzy logika se snaží pokrýt realitu v její nepřesnosti a neurčitosti (11).
Ve fuzzy logice uvažujeme celou množinu čísel <0;1>, což je rozdíl od klasické logiky, která využívá pouze stavy Pravda a Nepravda, respektive 1 a 0. Fuzzy logika tak umožňuje lépe popsat prvky ležící mezi krajními extrémy. Samotný proces fuzzy logiky je na následující:
Fuzzifikace -> Fuzzy inference -> defuzzifikace

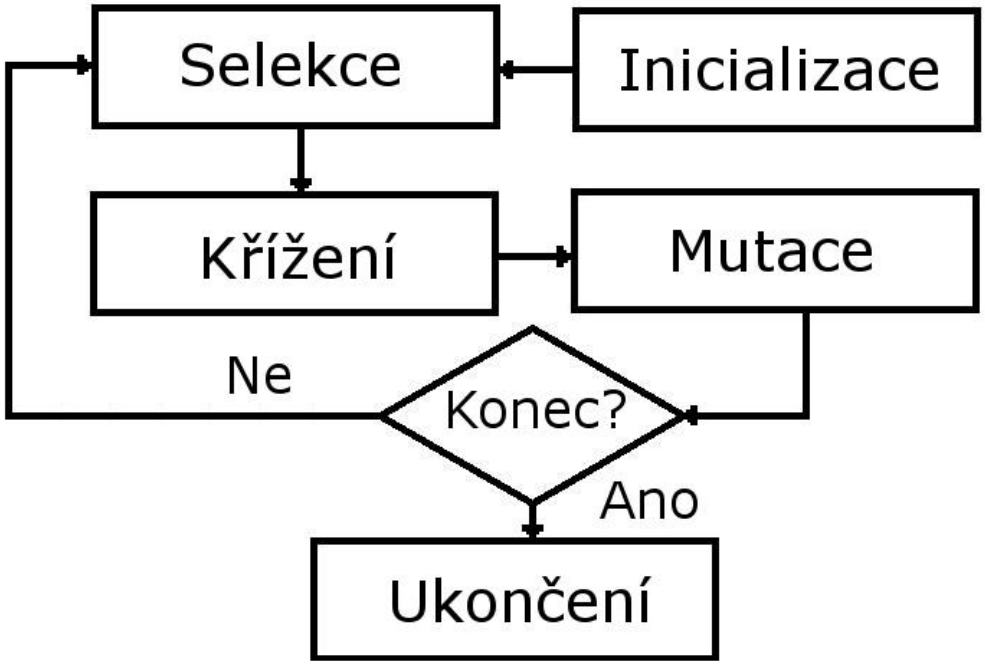
Fuzzifikace – transformuje vstupní proměnné do fuzzy podoby (lingvistická podoba), čím jsou například fuzzy prvek, fuzzy množina a další. Přiřazení do proměnných se provede podle funkcí jednotlivých fuzzy množin (11).

Fuzzy inference – až jsou všechny proměnné přeměněny, je možné s nimi pracovat. Ve fuzzy inferenci mohou být počítány algebraické operace nebo použity řídící pravidla a podobně. Nejčastěji využívané jsou podmínkové věty typu, KDYŽ, POTOM, S VÁHOU. Tyto podmínky vytváří samotný uživatel systému (12).

Defuzzifikace – Převede výsledky fuzzy inference zpět na srozumitelné proměnné, respektive reálné hodnoty (12).

1.3.2 Genetické algoritmy

Genetické algoritmy jsou odvozeny z přírodních zákonů. Používají se především pro složité operace, kde je velké množství parametrů a jejich kombinací. Takovýto výpočet by byl běžným způsobem příliš náročný a neefektivní. Pomocí genetického algoritmu vybereme nejvýznamnější parametry a ty méně významné vyřadíme, díky tomu relativně rychle najdeme lokální extrém nebo přijatelný výsledek. Díky vyřazení parametru však nebyly vyzkoušeny všechny možnosti, nemůžeme tedy říci, zda výsledné řešení je optimální. Musíme tedy ještě zvážit výpočet hrubou silou a významnost nalezení absolutního extrému (13, 14).



Obrázek 6: Proces reprodukce

Zdroj: [10, str. 87]

Inicializace je zavedení startovní populace, buďto pomocí náhodně zvolené sekvence parametrů, nebo již podle v minulosti použitých řešení. Selekce roztřídí vytvořenou populaci na základě účelové funkce a vyřadí jedince s nejhoršími výsledky a ty s nejlepšími výsledky namnoží. Jedinci, kteří nebyli vyřazeni, se pak mezi sebou kříží. Vznikají potomci, kteří mají část parametrů od každého z rodičů. Pomocí mutace zavedeme náhodné změny, které zvýší rozmanitost populace o parametry, které by pouhým křížením nevznikly.

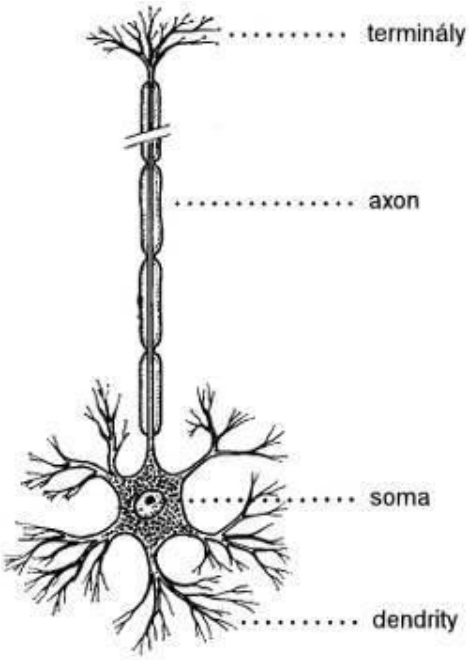
Pokud bude výsledná populace přijata, proces se ukončí, jestliže je populace zamítnuta, celý cyklus se opakuje, dokud není výsledná populace přijatelná. Běžně se do algoritmu vkládá ukončovací podmínka, která proces ukončí, pokud se výstup po několik generací nedaří zlepšit. V takové situaci se musí změnit systém, změnit způsob křížení, četnost mutací velikost populace a podobně (13, 14).

1.3.3 Neuronové sítě

Neuronové sítě používáme u deterministických závislostí, které jsou příliš provázané.
Pro pochopení neuronové sítě je potřeba se podívat na biologický základ, respektive biologickou strukturu mozku, ze které neuronová síť vychází (14).

Biologický základ

Biologická struktura mozku je základem k rozvoji zkoumání umělých neuronů. Umělé neurony, i když poslední době prošli velkým pokrokem, stále velice zaostávají za neurony biologickými. Neuron, jak biologický, tak umělý má vstupní a výstupní část. U biologického neuronu rozpoznáváme dendrin (což je výběžek nervových buněk který vede vzruch směrem do buňky), Axion (což je výstup z nervové buňky), Synapse (což je spojení buněk nervovými vlákny) a Soma (což je tělo biologického neuronu) (14, 15).



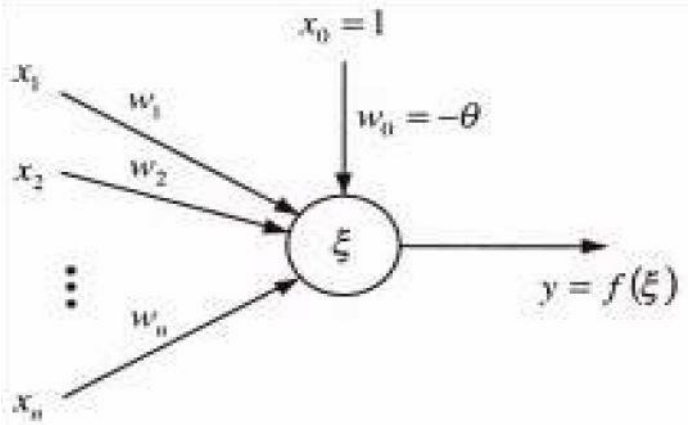
Obrázek 7: Biologický neuron
Zdroj: [16, str. 74]

Biologický neuron má oproti umělým neuronům mnohem větší počet neuronů, neboť řeší velký počet problémů, které spolu nijak nemusí souviset a jsou tedy mnohem komplexnější než umělé neurony, které jsou navrženy pro řešení specifických problémů, či úkolů (14, 15).

Umělé neurony

„Umělý neuron je založen na analogických principech. Vstupní informace jsou váženými vahami (weights). Odečítá se prahová hodnota (treshold) a pomocí aktivační funkce (activation function, transfer function) se signál transformuje na výstupní signál, který je předán následujícím umělým neuronům ve skryté (hidden), resp. výstupní vrstvě.“ (12, str. 97).

Umělý neuron sice vychází z biologického vzoru, ale tento model je velmi zjednodušený. Základem neuronových sítí je tzv. perceptron. Perceptron má *n* vstupů, které tvoří vstupní vektor neuronu. Tento vstupní vektor přivádí informace buď z vnějšího okolí, nebo z jiných neuronů. Jednotlivé vstupy představují reálné čísla a zároveň jsou váženy tzv. synaptickými váhami *w*, které ohodnocují důležitost daného vstupu (13).



Obrázek 8: Formální neuron
Zdroj: (13, str. 204)

Pro neuron je také stanoven tzv. práh citlivosti *b*. Tyto charakteristiky se dají zapsat do rovnice, která nám udává tzv. potenciál neuronu (14).

$$\varepsilon = \sum_{i=1} nw_i * x_i - b \tag{1.7}$$

Aktivační funkce:

- Hardlim - Tvrdá nelineární přenosová funkce. Tato funkce na výstupu neuronu vrací hodnotu 0, když vstup nepřesáhne daný práh hodnot. Pokud funkce na výstupu přesáhne daný práh hodnot je výstupem 1. Matematické vyjádření:

$$n = 0 \ pro\ a < 0; n = 1\ pro\ a \geq 0 \tag{1.8}$$

- Purelin – Lineární přenosová funkce, výstupní hodnota se rovná vstupní hodnotě. Matematické vyjádření:

$$n = a \tag{1.9}$$

- Logsig – Logaritmická signoida, vstup se převádí na výstup v intervalu od 0 po 1. Matematické vyjádření:

$$n = \frac{1}{1 + e^{-a}} \tag{1.10}$$

- Tansig – Hyperbolická tangentoida, vstupní hodnota se převádí na výstupní v intervalu od -1 do 1. Matematické vyjádření:

$$n = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}} \tag{1.11}$$

Umělá neuronová sít je složena ze vstupní vrstvy, jedné či více skrytých vrstev a výstupní vrstvy. Skrytým vrstvám říkáme „černá skříňka“ neuronové sítě. Mezi těmito vrstvami najdeme přechodové funkce, které jsou různých tvarů, respektive mají různé průběhy. Tyto funkce mají vliv na sílu reakce neuronu, na extrémní a středové hodnoty. Umělá síť se od pouhého matematického zápisu liší schopností učit se. Metody učení umělých

neuronových sítí se dělí do dvou skupin, z nichž jedna je Učení bez učitele, a druhá Učení s učitelem (15, 16).

Učení bez učitele

Tyto sítě neberou v potaz cílové hodnoty, ale hledají strukturu nebo shluky ve vstupech, ze kterých pak určují výsledek. Sítě se řídí vlastními pravidly, která určují úspěšnost. Některé sítě organizují strukturu podle vstupních dat. Jako příklad síě, která se učí bez učitele, může být Kohenova síť (13).

Učení s učitelem

Tyto sítě naopak potřebují ke vstupním hodnotám přiřazené výstupní hodnoty. Na těchto spojených hodnotách se síť učí, mělo by se tedy jednat o typické případy, které bude síť po naučení řešit. Pro síť tedy budou připravena data pro učení a sada pro validaci úspěšnosti testování. Pokud nebudeme schopni vybrat ideální data pro učení, použijeme datový soubor, ze kterého náhodným rozdělením určíme data, která se použijí k učení, validaci a testování. Validační data jsou ta, které využíváme pro sledování chyby učení, která by se měla snižovat. Testovací data jsou ta, která jsou mimo učení, a na kterých je kontrolován výstup, používají se k úpravě vah neuronu. Data můžeme také očistit o neúplné vstupy nebo o data které se opakují. Samotný proces učení může probíhat dvojím způsobem, buď dopředným nebo rekurentním (17).

Dopředné

Jedná se o nejjednodušší způsob, kdy se průchodem sítě vstupní neurony třídí, nebo se přibližují cílové hodnotě a proces jde pouze jedním směrem – vpřed. Příkladem této sítě je již zmíněná Kohenova síť. Neurony jsou zde zpravidla rozděleny do vrstev, a proto se jedná o vrstvenou sít.

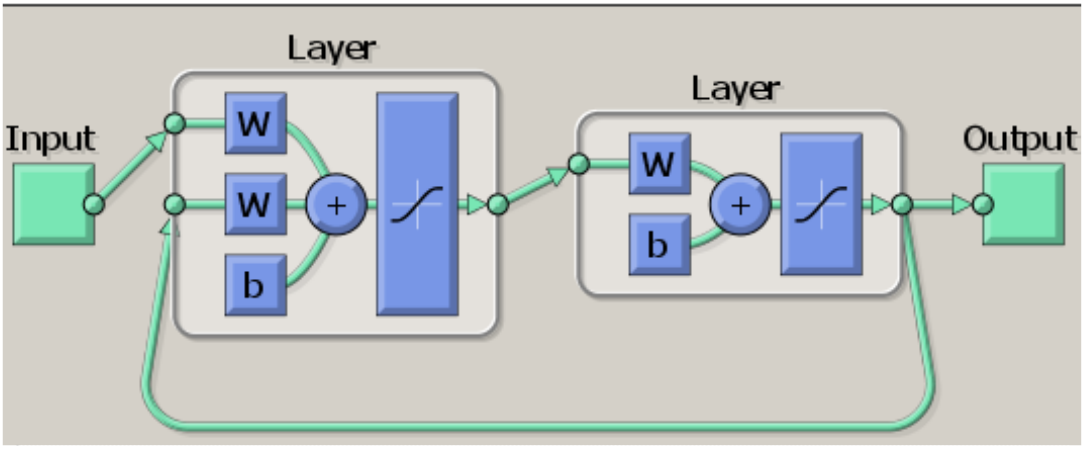
Vrstvená sít se skládá z jedné nebo více vrstev, kde poslední vrstvou je vrstva výstupní. U těchto vrstvených sítí je zásadní zvolit správný počet neuronů a vrstev tak, aby výsledná síť byla vhodná pro řešení zadané úlohy. Volba neuronů a jejich rozdělení do vrstev se řeší odhadem pomoci heuristických metod. Na první výstupní vrstvu se volí obvykle o něco větší počet neuronů, než je počet vstupů do dané sítě, a na druhou skrytou vrstvu se

volí počet neuronů odpovídající aritmetickému průměru ze součtu počtu perceptronů v první vrstvě a počtu výstupů (13).

Rekurentní

Do procesu vstupují výsledky nebo chyby předchozích kroků a upravují se počáteční parametry výpočtu. Proces běží ve smyčce do té doby, dokud není výsledek přijatelný, nebo dokud se nepřeruší z jiného důvodu. Proces se může ukončit například velkým počtem iterací či nesplnění časového limitu nebo v případě, že se výsledek se nepřibližuje k cílové hodnotě, ale je stále stejný nebo diverguje od cílové hodnoty.

Typickou metodou je zpětné šíření chyby (Back-propagation). Z celkové chyby je zpětně zjištěna chyba jednotlivých neuronů. Ta se použije k úpravě vah vstupů jednotlivých neuronů. Tento cyklus se opakuje do té doby, dokud se nepodaří dosáhnout tolerované míry chybovosti, nebo se nedaří chybu snižovat. Tímto způsobem učení můžeme nalézt lokální minimum chyby, ale minout globální minimum. Na následujícím obrázku můžeme vidět příklad rekurentní sítě (13).



Obrázek 9: Rekurentní síť

Zdroj: (Vlastní)

Neuronové sítě můžeme využít na celou řadu architektur jako je například:

- Multilayer Perceptron
- Radial Basis Function
- Hopfieldova síť
- Kohonenova síť

- Sítě ART
- Pravděpodobnostní sítě
- Generalized Regression Neural Network

Multilayer Perceptron

Vícevrstvá perceptronová síť je trénována učením s učitelem. K jejímu učení se používá backpropagation (algoritmus zpětného šíření). Důležitou roli hraje správné určení počtu neuronů. Pokud je počet neuronů zbytečně vysoký, tak to výrazně zvýší dobu trénování sítě. Zároveň se zhorší schopnost generalizace a dochází k přeučení sítě tzv.: overfitting. Taková síť má perfektní výsledky na známých datech, pokud jí však předložíme data nová, dochází k dramatickému zhoršení a zvyšují se chyby.

Největší využití perceptronové sítě je v predikci časových řad, kompresi dat nebo v klasifikaci. V současnosti patří mezi nejrozšířenější a nejpoužívanější, existuje tak spousta variant tohoto modelu, které se snaží odstranit jeho nevýhody a slabé stránky (18).

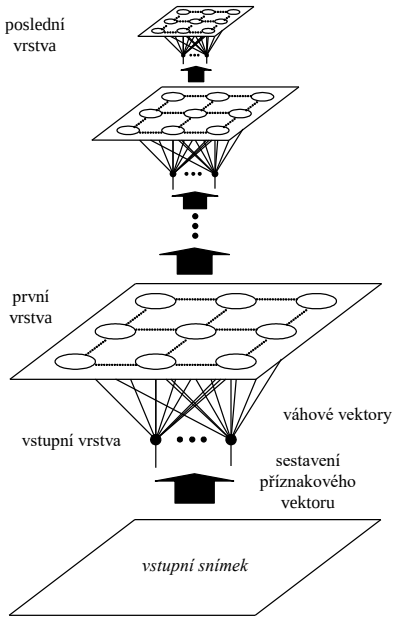
Hopfieldova síť

Pro Hopfieldovu síť je typické uspořádání, kdy počet neuronů je stejný jako počet vstupů. Každý z neuronů má jeden vnější vstup a výstup a zároveň jsou výstupy vedeny přes váhy *w* na vstupy ostatních neuronů, čímž je vytvořena uzavřená smyčka. Hopfield rozpracoval energetickou funkci, ze které jsou odvozena pravidla pro učení a vybavování sítě. Využívá zejména asociativní paměť, při rozpoznání obrazu nebo při řešení optimalizačních problémů. Nedostatkem sítě je omezená kapacita paměti, která je dána poměrem počtu trénovacích vzorů a počtu použitých neuronů. Dalším problémem je nutnost zvolit trénovací vzory, které si jsou pokud možno co nejméně podobné (13).

Kohonenova síť

Někdy je označována jako samoorganizační mapa. K učení nepotřebuje učitele. Funguje na principu vektorové kvantizace, kdy se síť pokouší aproximovat pravděpodobnostní rozdělení daných trénovacích dat. Síť obsahuje pouze jednu vrstvu neuronů. Do každého z nich jsou vedeny hodnoty ze všech vstupů. Výhodou je odolnost sítě proti náhodnému šumu, který se v trénovacích datech vyskytuje. Jako nevýhoda je velký počet kroků

nutných k natrénování sítě a náročnost trénování. Proto tuto metodu vznikla také modifikovaná metoda, která se dokáže učit i pomocí učitele. Kohonenova síť se nejčastěji využívá pro řešení problému obchodního cestujícího nebo k analýze dat (13).



Obrázek 10: Kohenova síť
Zdroj: [18, str. 38]

Síť ART

Adaptivní rezonanční teorie. K učení není potřeba učitel. Tato síť řeší problém proměnné stability. Řeší tak situaci, kdy se síť přetrénuje novými daty a díky tomu ztratí již naučené informace z předchozího tréninku. Síť ART má dvě vrstvy, první je porovnávací a druhá je rozpoznávací. Typické využití je rozpoznávání obrazu či řeči (19).

Algoritmy používané při trénování sítí:

Algoritmus zpětného šíření

Je to nejpoužívanější učící algoritmus, někdy se označuje jako zobecněné delta pravidlo. Cílem metody je minimalizovat chybu pomocí gradientní metody.

Trénování probíhá následujícím způsobem: Nejdříve se provede vybavovací fáze, to je fáze, kdy jsou na vstup sítě přivedena trénovací data. Data jsou dopředně šířena přes všechny vrstvy a na každém neuronu je vypočítán potenciál. Na poslední výstupní vrstvě může následovat výpočet chyby, protože známe celkový výstup sítě. Chyba je potom

zpětně šířena sítí, a nakonec jsou upraveny váhy neuronu ve snaze chybu snížit. Hodnota změny váhy v aktuálním kroku t je prováděna dle vztahu:

$$\Delta\omega_{ij}(t) = -n\frac{\partial E}{\partial \omega_{ij}} + \alpha\Delta\omega_{ij}(t-1) \tag{1.12}$$

Kde n je tzv. parametr učení (learning rate), pohybuje se v intervalu od 0 do 1. Parciální derivace chyby E představuje zpětné šíření chyby z výstupní vrstvy. Koeficient α je parametr momentu, který je násoben hodnotou změny váhy v předchozím kroku, a tím zajišťuje, že řešení minimalizace chyby nezůstane na lokálním minimu (18).

Levenberg-Marquardtův algoritmus

Modifikace algoritmu pro zpětné šíření je jeden z velmi oblíbených algoritmů, zejména proto, že je rychlý. Jeho nevýhodou je velká paměťová náročnost, používá se proto spíše pro menší sítě. Výpočet změny váhy můžeme vypočítat jako:

$$\Delta\omega_k = -[J^T(\omega_k)J(\omega_k) + \mu_k I]^{-1}J^T(\omega_k)v(\omega_k) \tag{1.13}$$

Kde J je Jacobian derivací chyb podle vah a *v* je vektor chyb. Parametr μ upravuje chovaní algoritmu, pokud je μ rovno nule, chová se algoritmus jako Gaussova-Newtonova metoda. Pokud je μ zvětšován, tak má algoritmus vlastnosti metody gradientového sestupu (18).

2 ANALÝZA PROBLÉMU

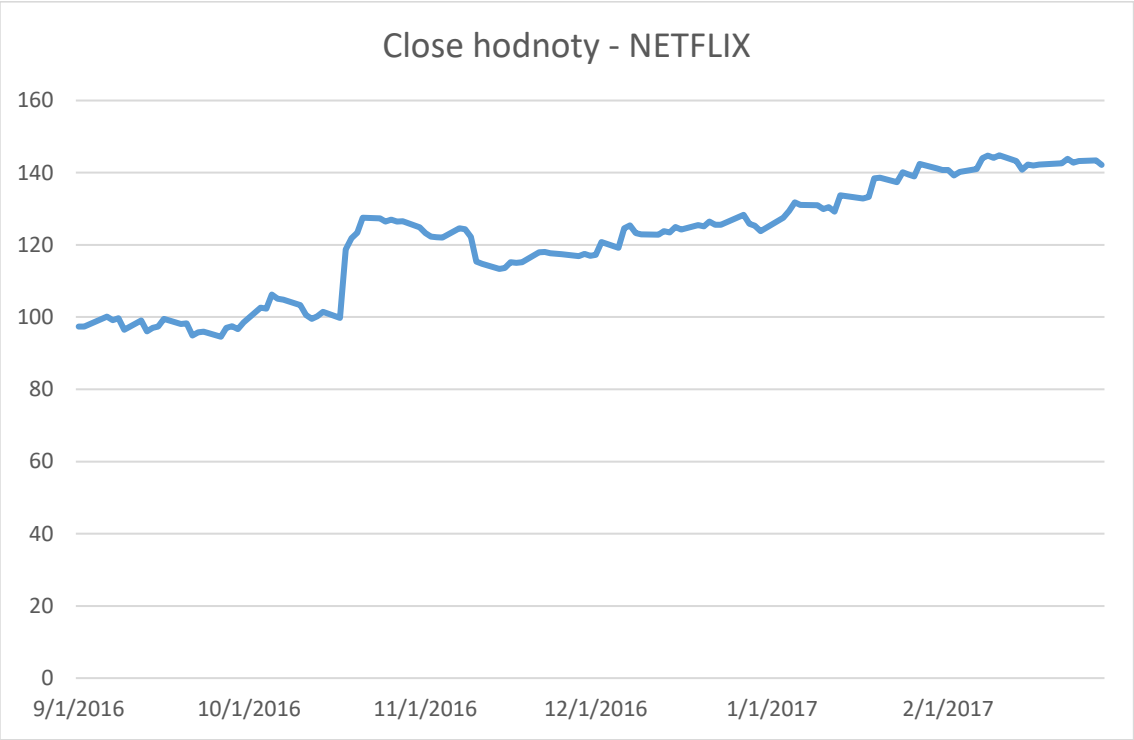
V úvodní kapitole byla popsána teoretická východiska, která jsou nezbytná pro pochopení praktické části. Bylo ukázáno, že vývoj cen akcií má charakter časové řady. Byly popsány metody, kterými lze predikovat časovou řadu. Úspěšnost predikce časové řady do značné míry ovlivňují její vlastnosti a její prediktabilitu můžeme určit pomocí Hurstova exponentu.

Pro predikci byl zvolen nelineární autoregresivní model implementovaný v prostředí MATLAB. Pro to, abychom mohli vyhodnotit výsledky predikce, je také potřeba vytvořit zjednodušený obchodní simulátor, který podle predikované hodnoty bude nakupovat a prodávat akcie. Ten pak bude dosahovat zisku nebo ztráty a výsledky obchodního simulátoru pak budou předány uživateli pro podporu rozhodnutí, zda vytvořený model dosahuje uspokojivých výsledků.

Na trhu existuje velké množství modelů zabývajících se obchodováním na kapitálovém trhu pomocí umělé inteligence. Před jakýmkoliv použitím modelu je však nutno provést rozsáhlé testy, jelikož výpočet je prováděn podle černé skřínky, čímž dostáváme pokaždé různé výsledky i při použití stejných dat. Ke skutečnému používání modelu je tedy potřeba mít dostatečně znalosti a modely si před použitím řádně odzkoušet na starých datech.

Před samotným učením neuronové sítě je potřeba zanalyzovat časovou řadu a ohodnotit tak prediktabilitu, jak již bylo zmíněno, toho se dá docílit pomocí Hurstova exponentu. Také je vhodné se na charakter jednotlivých časových řad podívat obecně z hlediska grafu. V analytické části se tak zabývám vývojem časové řady každé společnosti, pro kterou byl model tvořen.

NETFLIX

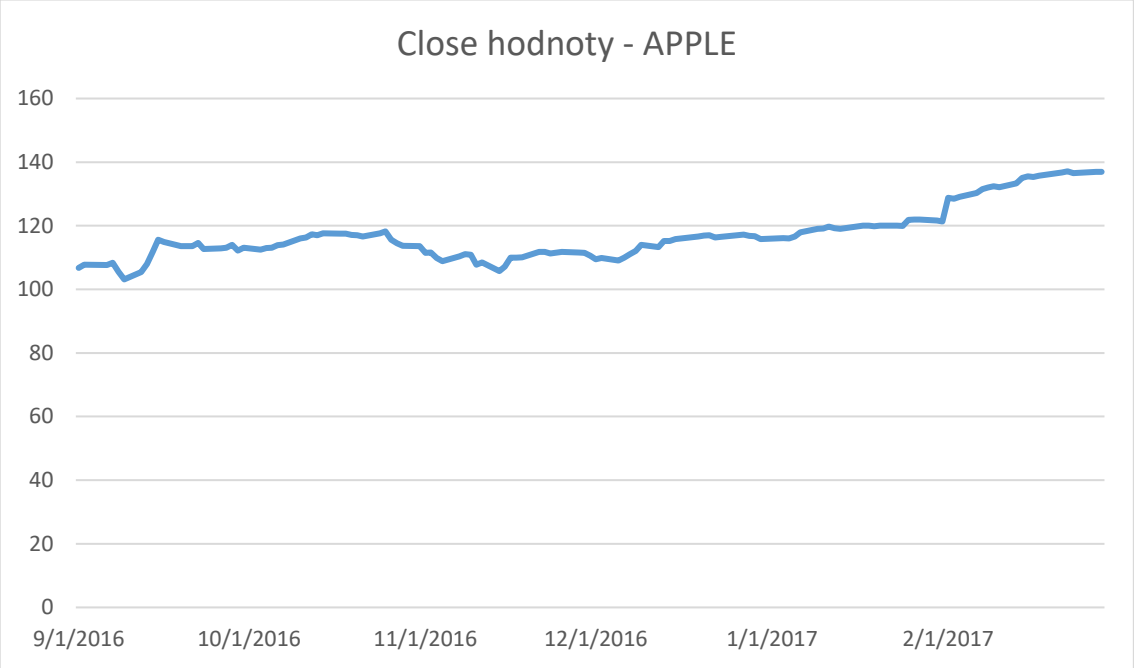


Graf 1: Close hodnoty – NETFLIX

Zdroj: (Vlastní)

Podle vývoje hodnot lze předpokládat dobrou prediktabilitu. V grafu nevidíme žádné výrazné skoky, nejvyšším skokem, který nastal, byl skok na konci října, kdy akcie společnosti vstouply z částky přibližně 100$ za akcii na hodnotu přibližně 120$ za akcii. V grafu je patrný rostoucí trend, který se na konci zkoumaného období mírně zpomalil a hodnoty se ustálili na ceně přibližně 140$ za akcii. Pro ověření prediktability byl vypočítán Hurstův exponent v hodnotě 0,8019, což značí, že časová řada je vhodná k predikci a k dalšímu zkoumání.

APPLE

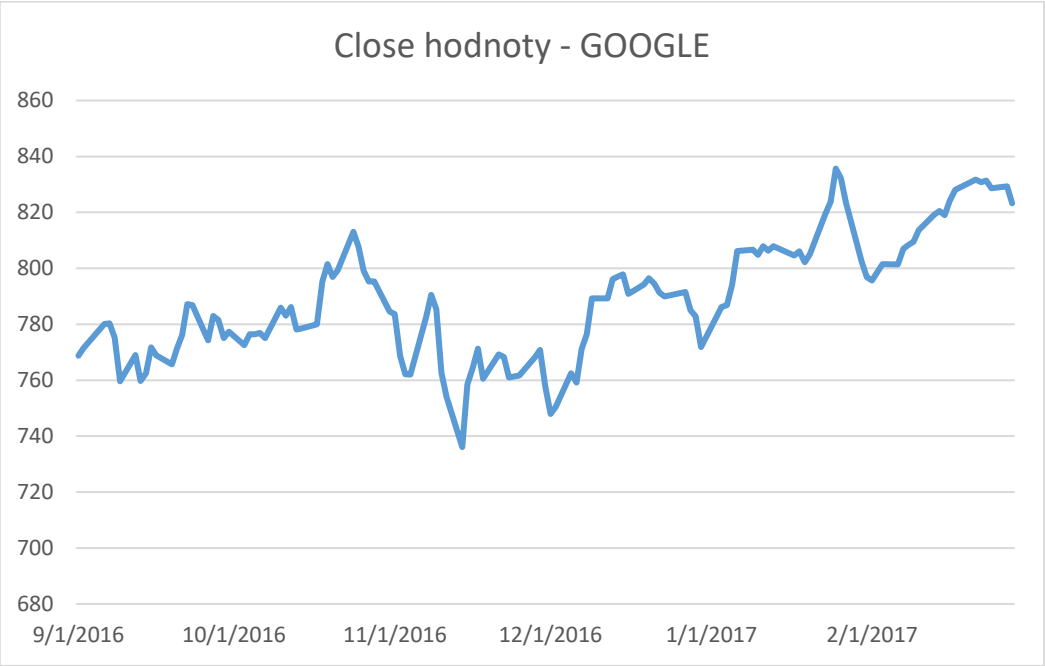


Graf 2: Close hodnoty – APPLE

Zdroj: (Vlastní)

V grafu, který znázorňuje close hodnoty akciového trhu pro společnost APPLE, nevidíme žádný skok jako tomu bylo u společnosti NETFLIX. Jsou zde zřejmé dvě změny trendu, kdy se ke konci října hodnota akcii dostala ke svému lokálnímu maximu 120$ za akcii. Trend se tak změnil z rostoucího na klesající. Ke konci měsíce listopadu se však tento vývoj změnil zpátky na trend rostoucí. V době zhruba okolo konce ledna a začátkem února, kdy hodnota opět nabyla částky 120$ za akcii, vznikl menši skok. Tentokrát však pokořil lokální minimum a došel na hodnotu přibližně 130$, kde dále pokračoval rostoucí trend. Hurstův exponent pro tuto časovou řadu je 0,7903 a i pro tuto řadu je tak vhodné zkoumat predikci.

GOOGLE



Graf 3: Close hodnoty – GOOGLE

Zdroj: (Vlastní)

Pro společnost GOOGLE se akciové close hodnoty pohybovaly chaotičtěji než v předchozích dvou případech. V grafu je vidět značné množství skoků, hodnoty se tedy pohybovaly. Minimální hodnota za sledované období byla zhruba 740$, kdežto maximální hodnota dosahovala k 840$ za akcii. Pro tuto časovou řadu však Hurstův exponent vyšel také pozitivně k další predikci, a to 0,7872.

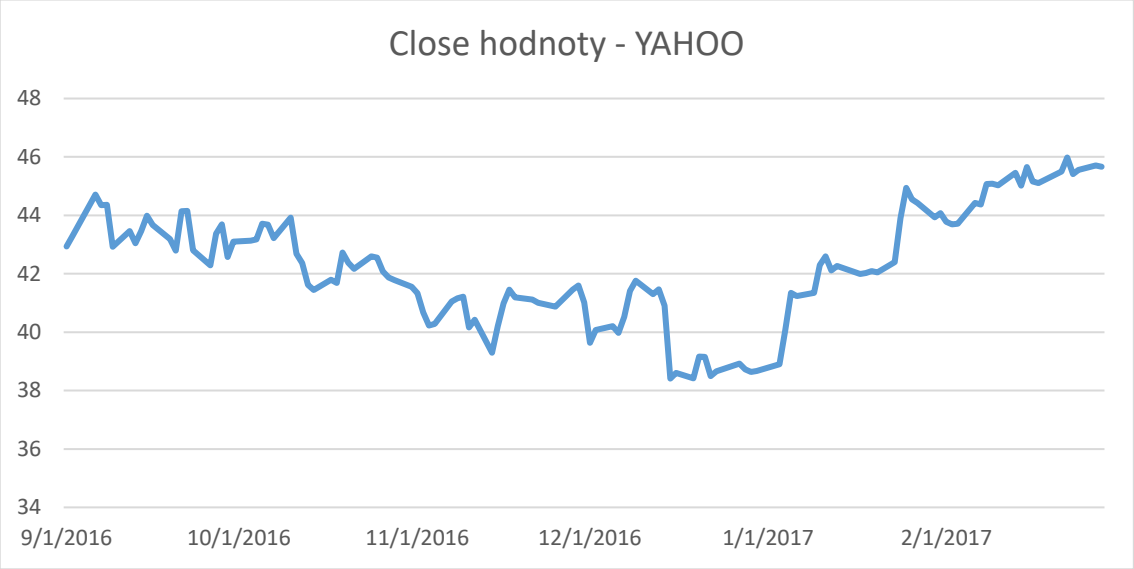
AMAZON

Graf 4: Close hodnoty – AMAZON

Zdroj: (Vlastní)

Vývoj cen akcií pro AMAZON má velké cenové rozpětí, hodnoty se ve sledovaném období výrazně mění. Cena akcie na začátku sledovaného období byla přibližně 775$, po rostoucím trendu se dostala téměř na hranici 850$, což byla hranice lokálního maxima. Potom však opět cena výrazně klesla a za zhruba jeden měsíc se cena akcie dostala do lokálního minima přibližně k ceně 725$. Od tohoto dna se opět odrazila a na konci sledovaného období končí cena na přibližně na 850$. Vývoj ceny akcie pro AMAZON je tedy velmi kolísavý. Hodnota Hurstova exponentu však vyšla 0,7936, můžeme tedy tvrdit, že tato časová řada je vhodná k predikci a bude zahrnuta v praktické části práce.

YAHOO



Graf 5: Close hodnoty – YAHOO

Zdroj: (Vlastní)

V tomto grafu také vidíme skoky stejně jako to bylo u společnosti GOOGLE a AMAZON, zde se však nejedná o tak velké výkyvy a hodnota akcie se pohybuje kolem 42$. Zpočátku je zřejmý klesající trend, kdy se cena akcie snížila z 42$ na 38$, což bylo lokální minimum v měsíci lednu. Z tohoto minima se z klesajícího trendu stal trend rostoucí a na konci sledovaného období, tedy na konci měsíce února, cena dosahuje 46$ za akcii. Hodnota Hurstova exponentu vyšla 0,8115, proto i tato časová řada je v hodná k predikci a je tedy zahrnuta do praktické části práce.

3 VLASTNÍ NÁVRHY ŘEŠENÍ

3.1 Popis systému

K predikci vývoje akcií je použita technická analýza, konkrétně metoda umělých neuronových sítí. Tato metoda vychází z analýzy časové řady, kde se pomocí historického vývoje kurzu snaží predikovat kurz budoucí. Postupně je vytvořen model v prostředí MATLAB, na základě, kterého se investor může rozhodovat, zda na daném trhu vstoupí do pozice krátké, nebo do pozice dlouhé.

Jako prvním krokem je potřeba nahrání dat do programu v prostředí MATLAB. K tomu je vytvořen modul, který data stahuje ze stránky yahoo.finance.com. V defaultním nastavení se stahuje akciový kurz společnosti YAHOO s půlroční historií od aktuálního okamžiku spuštění. Program určí datum podle počítačového nastavení a odečte půl roku. S ohledem na stahování dat z posledního půl roku program predikuje vývoj kurzu na následující jeden den. Posledními parametry, které jsou defaultně nastaveny, jsou maximální počet neuronů, který je nastaven na hodnotu třicet a velikost zpoždění, které je nastaveno na hodnotu pět.

Všechny parametry se dají nastavit libovolně. V následujících krocích bude popsán postup výběru dat.

Symbol

Uživatel si vybere symbol společnosti, pro kterou chce model tvořit.

Datum

Pro stažení historických dat je potřeba zadat dvě data. První datum značí, od kdy se data budou stahovat. Pro správnost je nutné uvést den, měsíc a rok. Druhý datum ukazuje, po jaké období chci historická data stáhnout. Pro predikci je potřeba uvést současné datum. Můžeme také predikovat data již minulá a sledovat odchylku predikce od reálných dat.

Počet neuronů

Uživateli je umožněno vybrat horní a dolní hranici počtu neuronů. Počet neuronů ovlivňuje schopnost učení sítě a její generalizaci. Čím více neuronů, tím se předpokládá kvalitněji natrénovaná síť. Není to však úplná pravda, proto byl vytvořen model, který na základě intervalu prohledá prostor a vybere nejlepší možnost. Může se však stát, že výběrem velkého počtu neuronů dojde k přeučení sítě.

Velikost zpoždění

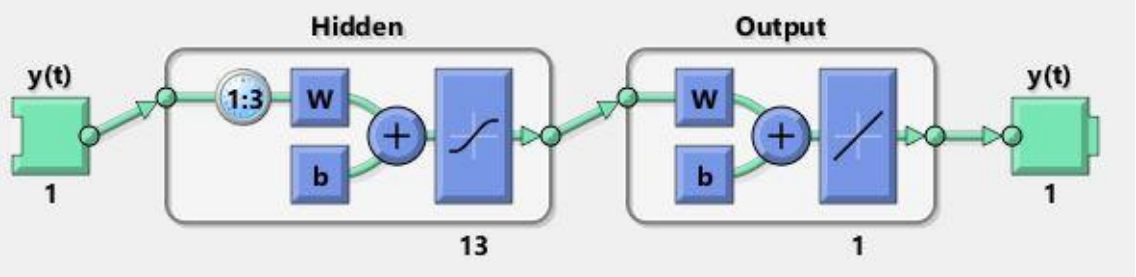
Podobně jako u výběru neuronů i tady je uživateli umožněno zadat horní a dolní hranici zpoždění vstupu, na kterém bude probíhat optimalizace modelu.

3.1.1 Metoda NAR a metoda NARX

Model NAR *(Nonlinear Autoregressive)*, nelineární autoregresní model. Tento model je založen na neuronových sítích a využívá se pro predikci dat. Na základě vstupních dat (historických dat) model odhaduje budoucí vývoj. Model se dá použít jak na predikci jednoho kroku, tak o na predikci více kroků. Čím více kroků však chceme zjistit, tím větší musíme uvažovat riziko chyby. Matematicky se model dá zapsat jako:

$$y(t) = f(y(t-1), \dots y(t-d)) \tag{1.14}$$

Y(t) představuje časovou řadu, *d* představuje počet hodnot časové řady. Schéma modelu NAR je znázorněno na následujícím obrázku:



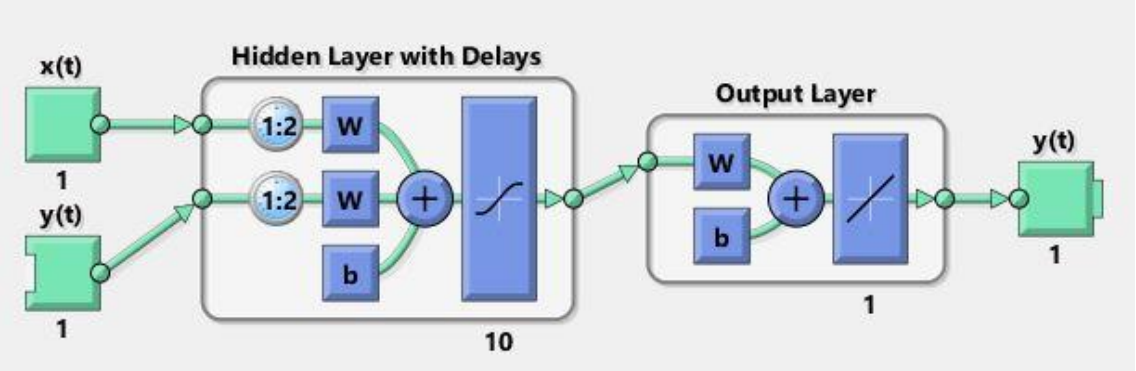
Obrázek 11: Neuronová síť NAR vytvořena v MATLABu

Zdroj: (Vlastní)

Model NARX *(Nonlinear Autoregressive network with exogenous inputs)* – oproti NAR modelu je model NARX rozšířen o další vstup. Dalším vstupem je myšlena další časová řada, která s původní časovou řadou nějak souvisí, což mohou být například indikátory časové řady. Může se jednat jak o technické indikátory, tak i fundamentální. Problémem je vhodný výběr přídavných vstupů. Nové vstupy jsou sice předpokladem pro kvalitnější učení a lepší odhad predikované hodnoty, avšak může se stát, že se díky příliš velkému počtu vstupů predikce zhorší. Rovněž se tak může stát, pokud na časovou řadu připojíme ukazatel, který nemá s časovou řadou vhodnou spojitost. Matematicky se model dá zapsat jako:

$$y(t) = f(x(t-1), \dots x(t-d), y(t-1), \dots, y(t-d)) \tag{1.15}$$

Y(t) představuje hlavní časovou řadu, x(t) jsou hodnoty přídavného vstupu. Všechny časové řady musí mít stejnou délku *d*. Schéma modelu NARX je znázorněno na následujícím obrázku:



Obrázek 12: Neuronová síť NARX vytvořena v MATLABu

Zdroj: (Vlastní)

3.1.2 Úprava časové řady

Časovou řadu, kterou použijeme, za nás automaticky upraví MATLAB při spuštění trénovacího procesu, jedná se o tzv. *preprocessing* dat. Pro pochopení fungování celého procesu se podíváme na to, co s našimi daty MATLAB udělá.

První funkce je *mapminmax***.** Tato funkce data normalizuje, řadí je do intervalu od -1 do 1, což umožňuje rychlejší trénování sítě. Další funkcí je funkce *removecostantrow***s**, která z dat odstraní opakující se řádky, a to z toho důvodu, že tréninku nepřidávají žádnou novou informační hodnotu. Poslední funkcí, která je provedena na časové řadě je funkce *fixunknowns*, která převádí nečíselné hodnoty vyskytujících se v datech (NaN hodnoty) na hodnoty číselné, aby díky těmto hodnotám nedošlo k přerušení tréninku.

Dále před samotným trénováním je potřeba data rozdělit na trénovací, validační a testovací. Trénovací data jsou použita k samotnému trénování sítě neboli úpravě vah jednotlivých neuronů sítě. Validační data kontrolují generalizaci sítě a zjišťují, jestli nedochází k přeučení neuronové sítě. Nemění se již váhy neuronů, ale data zkoumají, jestli se při snížení chyby sítě u trénovacích dat sníží také chyba u dat, která ještě nebyla ukázána. Generalizace zde znamená, zda je síť schopna řešit úlohy bez nutnosti onu síť znovu trénovat. Pokud je úroveň generalizace sítě nízká, jedná se o přetrénování sítě. Naopak pokud je úroveň generalizace sítě vysoká, jedná se o podtrénování sítě. Tento problém se řeší pomocí přidání šumu do vstupních dat, úpravou počtu neuronů, a změnou poměru trénovacích, validačních a testovacích dat. Posledními daty jsou testovací data, což jsou data, se kterými se síť ještě nesetkala, a ověřuje se jimi, jak je daná síť naučená.

Rozdělení dat do těchto tří množin je závislé na velikosti časové řady a na povaze a parametrů řešené úlohy.

3.1.3 Trénování sítě

Pro učení neuronové sítě bylo vytvořeno mnoho algoritmů. Neznámější jsou *Black propagation, Levenberg-Marquardt, Kohenova adaptace, volba vstupu podle genetických algoritmů,* a jiné. V programovém prostředí MATLAB je přímo implementována Levenberg-Marquardt metoda, která má pro naše použití vyhovující výsledky jak z hlediska rychlosti zpracování, tak z hlediska dobrých výsledků. Princip metody byl popsán v teoretické části práce.

Po natrénování sítě jsou k dispozici její charakteristiky a grafy, podle kterých lze posoudit úspěšnost učícího procesu.

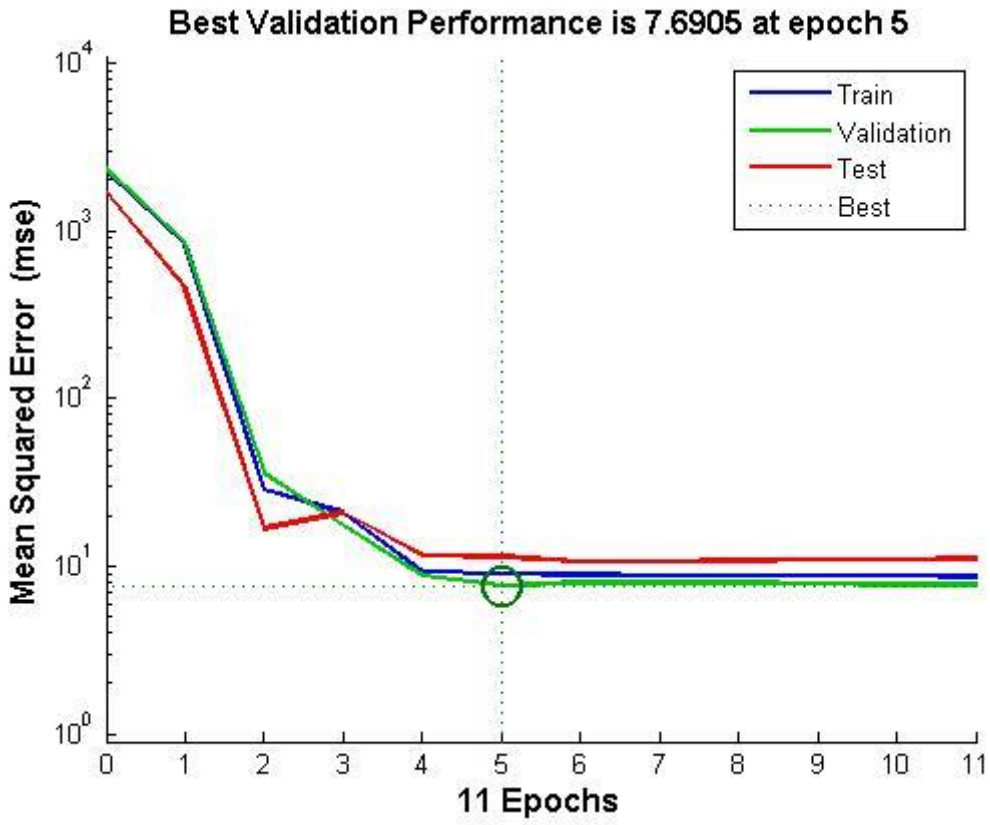
3.1.4 Model NAR

Časovou řadu pro model NAR již máme zpracovanou pomocí předcházejících kroků, nezbývá nic jiného než samotnou metodu spustit a podívat se na výsledky. Prvním krokem je volba dat. Jako hodnotu pro časovou řadu budeme používat data z uzavření obchodního dne, neboli hodnotu Close. K získání této hodnoty je použit modul *stazeni_dat*, který je defaultně nastaven na stažení půl ročního období akcií YAHOO. Poté je potřeba procentuálně rozdělit data na trénovací, validační a testovací. Tento krok jsem ponechal defaultně, a tedy trénovací data činí 70 %, validační data činí 15 % a testovací data činí 15 %.

Obrázek 13: Rozdělení dat

Zdroj: (Vlastní)

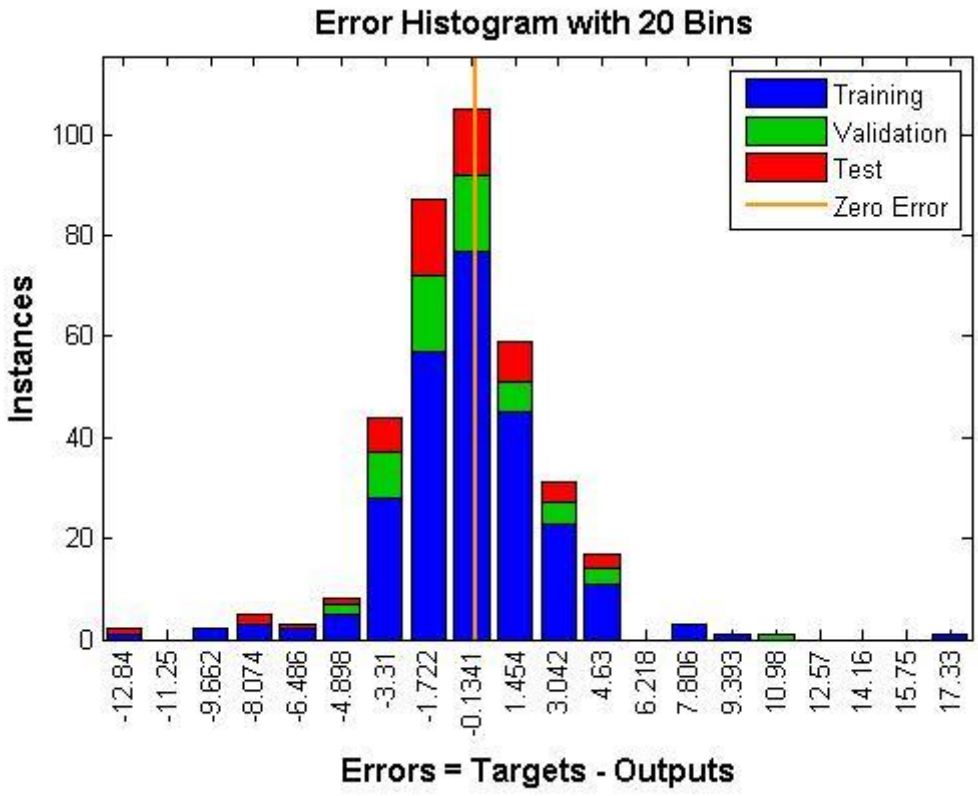
Dalším krokem je nastavení počtu neuronů a délky zpoždění. Tady nezbývá o mnoho více možností než zkoušet vhodný poměr těchto dvou veličin. Záleží i na tom, jak velký se použije počet historických dat (vstupu). Počet neuronů pro tento pokus jsem nastavil na hodnotu 13 a počet zpoždění na hodnotu 3. Samostatné učení je spuštěno pomocí funkce *train*. Výstupem je datová struktura pojmenovaná *net*. Tato struktura představuje naučenou neuronovou síť. O tom, jak se nám povedlo síť správně naučit, se můžeme přesvědčit pomocí různých charakteristik a grafů, které MATLAB přímo nabízí. Nejvýznamnějším ukazatelem je metoda střední kvadratické chyby MSE (*Mean Squared Error).* Na ose x jsou vždy zobrazeny epochy tréninku a na ose Y je vypočtena kvadratická chyba MSE. Křivky zobrazují vývoj chyby v datech, konkrétně modrá křivka představuje trénovací data, zelená křivka představuje validační data a červená křivka představuje testovací data. Je zřejmé, že během procesu učení dochází ke zlepšování sítě. Jak lze však vidět, není to kontinuální proces, protože nejlepším výsledkem je minimum chyby, která se nenachází vždy na konci učícího procesu. Po nějaké době v učení totiž začne narůstat chyba na validačních datech a síť tak ztrácí svoji generalizaci.



Graf 6: Kvadratická chyba MSE

Zdroj: (Vlastní)

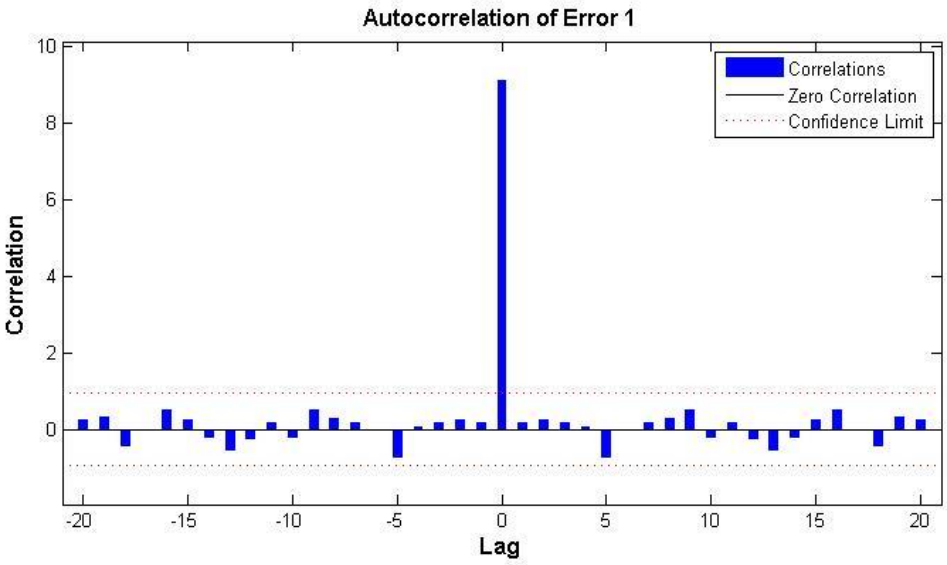
V záložce Error Histogram najdeme, jakých chyb se síť nejčastěji dopouští. Barevné rozložení chyb je stejné jako v předchozím grafu. Graf má tvar normálního rozložení. Rozsah chyb je od -12 do 17, kde záporná část nám ukazuje, kdy síť odhadovala nižší cenu, než ve skutečnosti byla. Kladná část zobrazuje, kdy síť odhadovala vyšší cenu, než ve skutečnosti byla. Nejvíce chyb je v intervalu od -4 do 5. Jelikož se tento interval pohybuje kolem 0 a je přiměřeně velký, tak je tento výsledek uspokojivý.



Graf 7: Histogram chyb

Zdroj: (Vlastní)

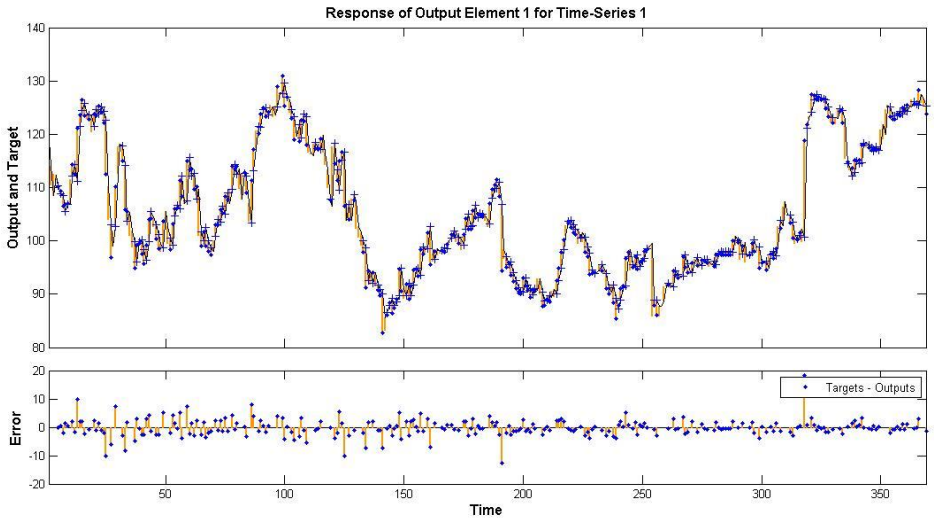
Další charakteristikou, kterou nám MATLAB pro kontrolu chyby nabízí, je Error autocorrelation plot. Ten zobrazuje korelaci chyby v závislosti na časovém posunu (lag). Maximum by mělo být v bodu, kde je hodnota Lag rovna nule a při vyšších hodnotách by měla klesat. Červené přerušované přímky zobrazují tzv. interval jistoty. Všechny hodnoty korelace kromě té, kde je lag, musí být v tomto intervalu. Pokud by tyto podmínky graf nesplňoval, bylo by nutné celou síť přeučit a zvolit jiný počet neuronů nebo upravit zpoždění sítě.



Graf 8: Error autocorrelation

Zdroj: (Vlastní)

Dalším grafem, který nám ukáže úspěšnost učení neuronové sítě, je graf odezvy časové řady, tzv. Time-Series Response. Na tomto grafu můžeme vidět odchylky cílových a vstupních hodnot pro všechna data. Pro nás je však nejdůležitější odchylka trénovacích dat, proto jsou tyto data o ostatní data očištěna a na grafu jsou zobrazeny jen hodnoty vstupní a hodnoty trénovací. Pokud je v trénovacích datech velká odchylka, vypovídá nám to o tom, že model není dostatečně kvalitní.



Graf 9: Time-Series Response

Zdroj: (Vlastní)

Tímto krokem je již celá síť vytvořena a můžeme se podle ní řídit. V praxi však tento model nemusí fungovat úplně na sto procent. Proto je nutné některé kroky projít znovu a síť co nejlépe optimalizovat a zlepšovat tak její prediktabilitu.

3.2 Optimalizace sítě

Prvním krokem k optimalizaci je vybrat správná data. Bylo tedy vybráno více společností, na kterých bylo použito vytvoření neuronových sítí. Je to z toho důvodu, že vytvoření neuronové sítě potřebuje stabilní data. Protože hledáme co nejlepší optimalizaci, je nutné najít společnost, u které budou dosažené výsledky pro naši potřebuju dostačující. K testování neuronových sítí bylo vybráno pět společností, které se zdají být jako stabilní, jedná se o NETFLIX, APPLE, GOOGLE, AMAZON, YAHOO. Pro všechny společnosti byla stažena data za poslední půlrok.

Trénovací, validační a testovací data byla nechána podle defaultního rozložení tedy v poměru 70 %, 15 % a 15 %. Tento krok totiž nemá příliš veliký význam a změna parametrů jen minimálně změní výsledný model.

Dalším optimalizačním krokem je tedy pohled na počet neuronů, které budou pro vytvoření sítě použity. Zároveň je také možné dívat se na velikost zpoždění vstupu, které je na vytvoření použito.

K optimalizaci počtu neuronů a velikosti zpoždění na datech jsem vytvořil program, který na dané síti vytváří neuronové sítě s různým počtem neuronů (od jednoho neuronu po padesát neuronů) a s různou velikostí zpoždění (od jednokrokového zpoždění po pětikrokové zpoždění). Toho je docíleno za použitím dvou vnořených *for* cyklů. Před dokončením jednoho cyklu je pokaždé zaznamenána kvadratická chyba MSE, která nám ukazuje chybovost sítě na datech. Na základě této chyby pak rozhodneme, která síť vykazuje nejlepší výsledky. Jelikož každé vytvoření sítě má jiné výsledky (pokud vytvoříme neuronovou síť za použití jednoho neuronu s jednokrokovým zpožděním pětkrát po sobě, dostaneme pokaždé jiné výsledky), je potřeba celý proces opakovat vícekrát. Proto jsou dva *for* cykly zabaleny do třetího cyklu, který již automaticky vyhledává minimální chybu a tu zapisuje do tabulky. Do této tabulky se kromě minimální

chyby MSE zapisuje úspěšnost prováděného pokusu. Tato úspěšnost je počítána algoritmem zobrazeným na obrázku č. 14. Je tedy vytvořen *for* cyklus pro délku sítě, který kontroluje, zda se skutečná hodnota další den zvýšila/snížila. Tento trend pak porovnává s hodnotou neuronové sítě. Pokud se obě hodnoty pohnuly stejným směrem, je to zaznamenáno jako kladný pokus. Na konci se počet kladných pokusů vydělí počtem celkových pokusů a vynásobí stem, z čehož dostaneme celkovou úspěšnost neuronové sítě. Jelikož je tato operace časově náročná, bylo provedeno dvacet opakování.

Data by se také dala třídit pouze podle odhadu úspěšnosti, pro nás je však více důležitější snažit se minimalizovat kvadratickou chybu a dosáhnout největší úspěšnosti s nejmenší možnou chybou.

```
for i=1:(length(today_Price)-1)
 if (today_Price(i+1)>=today_Price(i) && predicted_Price(i)>=today_Price(i) || (today_Price(i+1)<today_Price(i) && predicted_Price(i)<today_Price(i)))
     u=u+1;
 end
end

uspesnost=(1-u/(length(today_Price)))*100
```

Obrázek 14: Kód. který počítá úspěšnost sítě

Zdroj: (Vlastní)

Tento proces byl aplikován na všechny společnosti.

3.2.1 NETFLIX

Výsledky testu pro společnost NETFLIX

Neurony	Zpoždění	MSE	Úspěšnost		Neurony	Zpoždění	MSE	Úspěšnost
36	3	3,433	59 %		48	3	4,668	54 %
42	5	3,965	57 %		34	2	4,687	56 %
39	4	4,444	50 %		30	2	4,71	57 %
31	2	4,453	57 %		8	4	4,822	54 %
25	3	4,474	56 %		25	3	4,84	56 %
42	3	4,531	53 %		43	2	4,928	57 %
19	2	4,568	55 %		38	3	5,035	56 %
23	2	4,61	55 %		20	3	5,081	58 %
41	3	4,632	58 %		30	2	5,085	58 %
24	5	4,634	64 %		14	5	5,219	58 %

Tabulka 1: Výsledky optimalizace neuronů a zpoždění pro NETFLIX

Zdroj: (Vlastní)

Nejnižší kvadratická chyba MSE tedy dosahuje hodnoty 3,433 s 59% úspěšností odhadu při použití třiceti šesti neuronů a tříkrokovém zpoždění. Síť dosahuje nejlepších výsledků při použití většího množství neuronů. Pouze jedenkrát dosahovala síť nejlepších hodnot při použití méně než deseti neuronů, konkrétně při použití osmy neuronů s čtyřkrokovým zpožděním, přičemž dosahovala síť kvadratické chyby 4,822 a 54 % úspěšnosti. Nejčastější hodnoty se blíží k číslu třiceti pěti neuronů. Můžeme si však všimnout, že síť dosahuje dostačujících výsledků nehledě na to, jaký zvolíme počet neuronů nebo zpoždění. Dle výsledků však není vhodné volit menší zpoždění jak dva kroky, protože jednokrokové zpoždění se v síti totiž nevyskytlo ani jednou.

Jako druhou volbu bych, s ohledem na úspěšnost sítě, volil dvacet čtyři neuronů s pětikrokovým zpožděním, sít zde dosahovala 64 % úspěšnosti s kvadratickou chybou 4,634. Podle těchto hodnot se síť jeví jako vyhovující k dalšímu zkoumání, nejdříve si však ověříme další společnosti.

3.2.2 APPLE

Výsledky testu pro společnost Apple

Neurony	Zpoždění	MSE	Úspěšnost		Neurony	Zpoždění	MSE	Úspěšnost
49	5	0,717	51 %		8	5	0,872	50 %
28	5	0,728	55 %		24	5	0,917	56 %
14	5	0,77	53 %		26	5	0,923	52 %
16	5	0,799	53 %		38	5	0,931	54 %
10	5	0,801	47 %		16	5	0,959	57 %
24	5	0,818	53 %		28	5	0,975	53 %
9	5	0,819	53 %		44	2	0,986	55 %
47	2	0,82	55 %		16	5	1,037	53 %
7	5	0,854	53 %		19	5	1,043	57 %
46	5	0,869	53 %		23	5	1,047	58 %

Tabulka 2: Výsledky optimalizace neuronů a zpoždění pro APPLE

Zdroj: (Vlastní)

Pro společnost Apple dosahuje nejnižší kvadratická chyba hodnoty 0,717 s úspěšností 51 % za použití čtyřiceti devíti neuronů s pěti krokovým zpožděním. Stejně jak u akcii společnosti NETFLIX ani zde nemůžeme jednoznačně určit nejideálnější výběr neuronů, výsledky se hodně rozcházejí, můžeme však s největší pravděpodobností určit, že pro

vytváření této neuronové sítě bude nejlepší použít pěti krokové zpoždění. Nejlepší úspěšnost dosahuje 58 % s kvadratickou chybou 1,047.

3.2.3 GOOGLE

Výsledky testu pro společnost GOOGLE

Neurony	Zpoždění	MSE	Úspěšnost		Neurony	Zpoždění	MSE	Úspěšnost
44	3	33,844	54 %		38	3	37,679	50 %
45	4	35,15	54 %		29	2	38,268	54 %
7	5	35,499	49 %		23	5	38,407	53 %
15	4	35,812	55 %		11	5	38,602	54 %
41	4	35,857	54 %		37	2	38,637	54 %
41	5	36,048	56 %		8	5	39,2	53 %
21	4	36,597	52 %		14	4	39,535	49 %
27	4	37,015	51 %		37	1	40,261	49 %
29	4	37,373	46 %		47	4	40,434	51 %
29	2	37,624	56 %		34	1	40,763	54 %

Tabulka 3: Výsledky optimalizace neuronů a zpoždění pro GOOGLE

Zdroj: (Vlastní)

Pro společnost GOOGLE dosahovaly nejnižší kvadratickou chybu 33,844 sítě tvořené čtyřiceti čtyřmi neurony s tříkrokovým zpožděním a úspěšností 54 %. Ani u společnosti GOOGLE nemůžeme určit nejlepší možný počet neuronů a hodnoty se hodně podobají při použití jakéhokoliv počtu, nejčastěji se však vyskytuje čtyřkrokové zpoždění. Je tedy namístě držet se tohoto faktu, a to i přesto, že nejnižší chyby dosahovalo tříkrokové zpoždění. Pro další manipulaci se sítí je vhodné volit zpoždění čtyřkrokové. U neuronových sítí tvořených pro společnost GOOGLE si však můžeme všimnout, že jsme dostávali horší výsledky, co se týče hlediska úspěšnosti, vyskytují se nám tady i hodnoty nesplňující pomyslnou hranici 50%. Nejvyšší úspěšnosti však dosahují dvě hodnoty s 56 %, což je pro nás stále vyhovující, jedna z hodnot má však nižší kvadratickou chybu a to 36,048 oproti 37,624, a proto je volena tato hodnota, a to s počtem čtyřicet jedna neuronů a pětikrokovým zpožděním.

3.2.4 AMAZON

Výsledky testu pro společnost AMAZON

Neurony	Zpoždění	MSE	Úspěšnost		Neurony	Zpoždění	MSE	Úspěšnost
14	3	57,769	48 %		49	4	65,458	45 %
24	3	59,205	54 %		30	3	65,464	46 %
35	3	59,704	52 %		46	3	65,547	45 %
31	3	60,381	50 %		31	3	66,002	57 %
41	3	60,865	52 %		32	3	66,114	53 %
21	4	61,348	45 %		15	4	66,966	44 %
25	5	61,894	48 %		15	5	67,467	43 %
21	5	63,536	49 %		30	3	68,389	51 %
28	3	64,102	46 %		15	3	69,338	48 %
45	3	65,303	49 %		31	4	69,652	42 %

Tabulka 4: Výsledky optimalizace neuronů a zpoždění pro AMAZON

Zdroj: (Vlastní)

Ani pro společnost AMAZON nemůžeme po dvaceti krocích určit vhodné množství neuronů na sestavení neuronové sítě, hodnoty se hodně rozcházejí i při použití tohoto vstupu. Můžeme si však opět všimnout, že síť nejčastěji dosahovala nejlepších výsledků při tříkrokovém zpoždění. Nejlepší výsledky dosahovala síť při použití čtrnácti neuronů s tříkrokovým zpožděním, kvadratická chyba zde dosahovala hodnoty 57,769 s úspěšností 48 %. Kvůli tomu, že síť nesplňuje úspěšnost vyšší jak 50 %, bude použita síť následující, která tento fakt splnila, a to s úspěšností 54 % s kvadratickou chybou MSE 59,205 a rozložením dvaceti čtyř neuronů s tříkrokovým zpožděním. Z hlediska úspěšnosti dosahovala nejlepších výsledků síť, která má třicet jedna neuronů a zpoždění tři kroky. Tato síť dosahovala kvadratické chyby 66,002 s 57 % úspěšností.

3.2.5 YAHOO

Výsledky testu společnosti YAHOO

Neurony	Zpoždění	MSE	Úspěšnost		Neurony	Zpoždění	MSE	Úspěšnost
37	3	0,194	52 %		24	4	0,229	53 %
42	3	0,203	52 %		49	3	0,239	57 %
42	4	0,204	46 %		20	5	0,241	51 %
49	5	0,217	50 %		33	3	0,243	54 %
47	5	0,218	47 %		19	3	0,247	52 %
29	3	0,222	58 %		18	4	0,249	55 %
31	3	0,223	57 %		40	3	0,249	53 %
41	3	0,225	57 %		31	5	0,254	52 %
23	5	0,225	52 %		30	5	0,26	54 %
49	4	0,229	52 %		27	2	0,271	60 %

Tabulka 5: Výsledky optimalizace neuronů a zpoždění pro YAHOO

Zdroj: (Vlastní)

Nejnižší kvadratická chyba MSE pro vstupní data společnosti YAHOO dosahovala hodnoty 0,194 s úspěšností 52 % za použití třiceti sedmi neuronů se tříkrokovým zpožděním. Nejčastější zpoždění při vytváření neuronové sítě pro YAHOO je tři, mnohokrát se však vyskytují i jiné hodnoty. Nemůžeme tak přesněji určit, které zpoždění by bylo pro tato data nejvhodnější. Stejný efekt platí pro počet neuronů, jelikož i tady se hodnoty různě rozcházejí. Podle nejvyšší úspěšnosti je na tom je nejlíp dvoukrokové zpoždění s použitím dvaceti sedmi neuronů a to 60 % úspěšností s kvadratickou chybou o velikosti 0,271.

3.2.6 Shrnutí výsledků

V tabulce níže jsou vybrány pokaždé dvě hodnoty pro každou společnost. První hodnota je vždy vybrána na základě nejnižší kvadratické chyby, druhá hodnota je vybrána podle nejvyšší úspěšnosti odhadu trendu.

	Neurony	Zpoždení	MSE	Úspěšnost
NETFLIX	36	3	3,433	59 %
NETFLIX	24	5	4,634	64 %
APPLE	49	5	0,717	51 %
APPLE	23	5	1,047	58 %
GOOGLE	44	3	33,844	54 %
GOOGLE	41	5	36,048	56 %
AMAZON	24	3	59,205	54 %
AMAZON	31	3	66,002	57 %
YAHOO	37	3	0,194	52 %
YAHOO	27	2	0,271	60 %

Tabulka 6: Výsledky optimalizace neuronů a zpoždění

Zdroj: (Vlastní)

Z výsledku testu vidíme, že nejhůře, co se týče z hlediska učení neuronové sítě, na tom byly společnosti GOOGLE a AMAZON, proto pro tyto společnosti nebude tvořena další optimalizace modelu. Další tři společnosti však dosahují poměrně slušných výsledků a pro rozhodnutí, která data pro nás budou nejlepší, je potřeba dalších kroků.

3.2.7 NETFLIX, APPLE, YAHOO

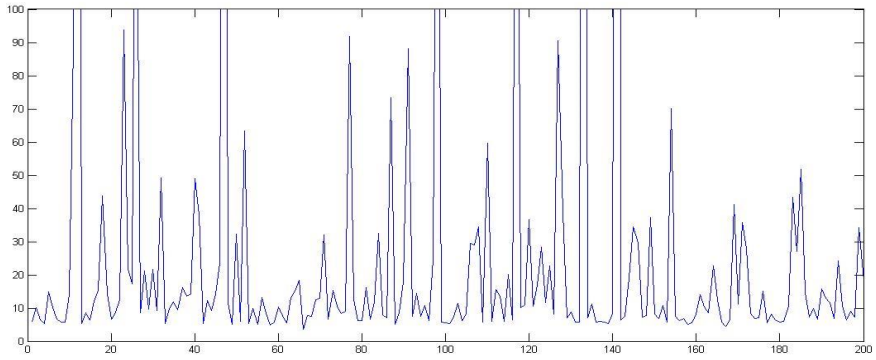
K dalšímu testování bylo použito rozložení neuronů a zpoždění dle tabulky výše. Tento test spočívá ve vytvoření dalších sítí pro každou společnost a zjištění, jak moc byly výsledky testy náhodné, a jestli bude síť dosahovat podobných výsledků i v dalších pokusech.

Doposud jsme sice našli nejmenší chybu pomocí prohledání různého počtu neuronů, pro to, aby byl model optimalizován správně, je však potřeba prohledat i prostor s již optimalizovanými neurony. Pro tento test je vytvořen algoritmus, který prohledá prostor na základě dvoustovek opakování a zobrazí rozložení učení sítě do grafu ukazujícího,

jakých chyb daná síť dosahuje při každém učení. Celkové výsledky jsou pak zobrazeny v grafu, ke kterému je přidána tabulka minimálních chyb podle počtu pokusů.

NETFLIX

- **Minimální hodnota MSE**



Graf 10: Nejmenší možná kvadratická chyba MSE pro NETFLIX

Zdroj: (Vlastní)

Maximální hodnota chyby je v grafu ořezána, aby byl graf přehlednější. Vidíme, že hodnota chyby hodně kolísá, minimální kvadratická chyba proto není jednoznačná, ale na základě tohoto pokusu se k ní můžeme alespoň přiblížit.

Opakování	66	167	58	88	158
MSE	3,634	4,402	4,988	5,036	5,164
Úspěšnost	55 %	59 %	61 %	60 %	55 %

Tabulka 7: Vyhodnocení pěti nejmenších kvadratických chyb MSE pro NETLIX

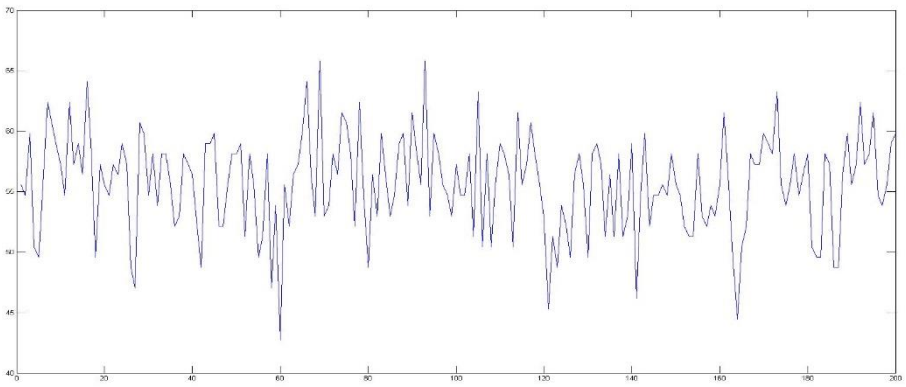
Zdroj: (Vlastní)

Nejnižší chyby bylo dosaženo po šedesáti šesti pokusech, nejedná se však o minimální chybu, pouze o přibližnou hodnotu minimální chyby. Díky tomu, že se síť pokaždé učí jinak, nemůžeme přesně určit, jaké je minimum. Kdybychom místo dvou set pokusů, zkusili třeba tisíc, chyba by se mohla, ale také nemusela o něco snížit. Rozdíly však už budou mizivé a prohledávat větší prostor nebude mít na naučenou síť nikterak zásadní vliv. Pro další pokusy je dokonce nevhodné počítat s chybou, která byla nalezena, ale je spíše doporučeno volit chybu o něco vyšší. Vzhledem velkému nárůstu chyb v některých oblastech není ani vhodné volit celkový průměr ze všech dvou set testovaných hodnot. Pro nejlepší řešení jsem volil průměr z pěti nejnižších hodnot. Pro nalezení vhodného modelu je tedy postačující podmínka splnění kvadratické chyby menší jak 4,645. I tak se

může stát, že nalezení takové chyby může trvat více jak dvě stě pokusů, a proto je zvolena podmínka maximálního počtu pokusů.

- **Maximální úspěšnost s minimální MSE**

Stejný krok provedeme také pro nejvhodnější rozložení neuronů, které znamenaly nejvyšší úspěšnost s nejnižší možnou chybou.



Graf 11: Nejvyšší úspěšnost s minimální kvadratickou chybou MSE pro NETFLIX

Zdroj: (Vlastní)

V grafu, který hledá maximální úspěšnost, vidíme velké výkyvy. Nejvyšší úspěšnost se vyskytuje kolem 60 %, pět nejlepších hodnot si napíšeme do tabulky níže.

Opakování	69	93	16	66	105
Úspěšnost	66 %	66 %	64 %	64 %	63 %
MSE	8,965	6,481	7,662	5,845	5,374

Tabulka 8: Vyhodnocení pěti nejvyšších úspěšností s nejmenší kvadratickou chybou MSE pro NETLIX

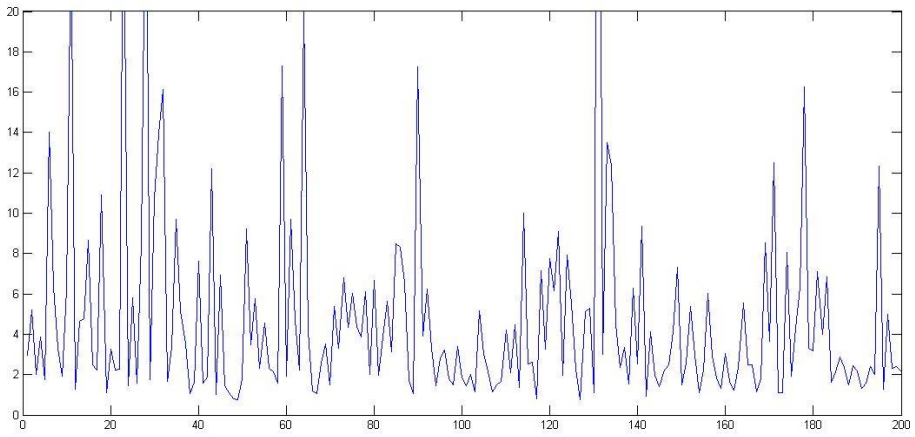
Zdroj: (Vlastní)

Nejvyšší úspěšnost dosahovala dvakrát 66 %, jednou to bylo v šedesátém devátém opakování a podruhé v devadesátém třetím opakování. Ve druhém zmíněném však dosahovala kvadratická chyba nižší hodnoty, a to konkrétně 6,481. Pro porovnání, nejnižší možná chyba dosahovala 3,634, avšak s úspěšností 55 %. Pro to, abychom mohli určit minimální možnou chybu kombinovanou s maximální úspěšností, nemůžeme použít stejně jak v minulém případě aritmetického průměru pěti posledních hodnot, může se totiž stát, že nejmenší možná chyba nebude splňovat maximální úspěšnosti, proto jsou

vybrány hodnoty, kdy je úspěšnost snížena o jednotku a požadovaná chyba je vybrána podle odhadu. Konkrétně tedy trénování modelu musí splňovat podmínku 64 % úspěšnosti s kvadratickou chybou menší jak 7,5.

APPLE

- **Minimální MSE**



Graf 12: Nejmenší možná kvadratická chyba MSE pro APPLE

Zdroj: (Vlastní)

V grafu vidíme podobný vývoj jako u NETFLIXU, všechny hodnoty různě kolísají, ale vždy se vracejí ke spodní hranici. Pro lepší přehled si vybereme pět nejnižších hodnot, které si zobrazíme v tabulce níže.

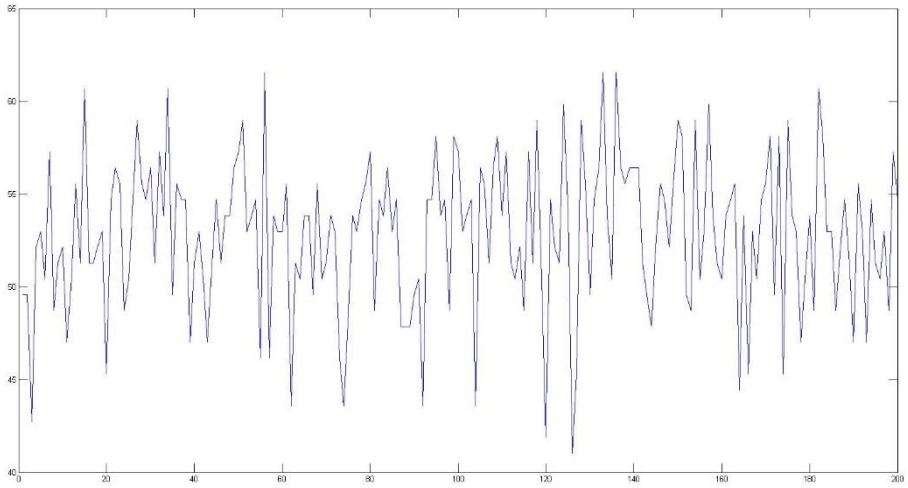
Opakování	49	127	48	117	142
MSE	0,753	0,761	0,794	0,801	0,921
Úspěšnost	58 %	55 %	53 %	56 %	56 %

Tabulka 9: Vyhodnocení pěti nejmenších kvadratických chyb MSE pro APPLE

Zdroj: (Vlastní)

Nejnižší kvadratická chyba byla nalezeno po 49 opakováních, a to 0,753 s 58 % úspěšností. Průměrem pěti hodnot, a tedy hodnotou, která pro nás bude dostačující podmínkou, je 0,777.

- **Maximální úspěšnost s minimální MSE**



Graf 13: Nejvyšší úspěšnost s minimální kvadratickou chybou MSE pro APPLE

Zdroj: (Vlastní)

Úspěšnost odhadu s nízkou kvadratickou chybou se pohybuje okolo hranice 60 % stejně jako u akcií NETFLIXU. Hodnoty však opět hodně kolísají a nás ještě zajímá, jaký podíl na tom má kvadratická chyba. Pět nejlepších hodnot si zobrazíme v tabulce níže.

Opakování	56	133	136	15	34
Úspěšnost	62 %	62 %	62 %	61 %	61 %
MSE	2,023	3,07	3,569	1,169	1,121

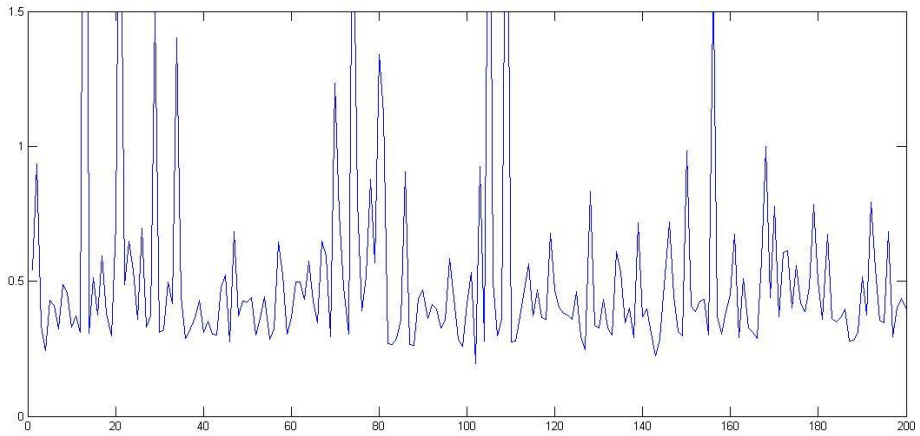
Tabulka 10: Vyhodnocení pěti nejvyšších úspěšností s nejmenší kvadratickou chybou MSE pro APPLE

Zdroj: (Vlastní)

Nejvyšší úspěšnost s nejmenší chybou dosahovala 62 % a byla dosažena hned třikrát, v padesátém šestém pokusu však dosahovala nejmenší kvadratickou chybu, a to 2,023. Pro porovnání, nejnižší možná kvadratická chyba byla 0,753 s úspěšností 58 %. Pro úspěšné natrénování modelu musí být úspěšnost vyšší jak 61 % a s kvadratickou chybou, která není vyšší jak 2,5.

YAHOO

- **Minimální MSE**



Graf 14: Nejmenší možná kvadratická chyba MSE pro YAHOO

Zdroj: (Vlastní)

Znovu je ukázáno, že se kvadratická chyba vzhledem náhodnosti hodně mění. V tabulce níže si opět zobrazíme posledních pět hodnot.

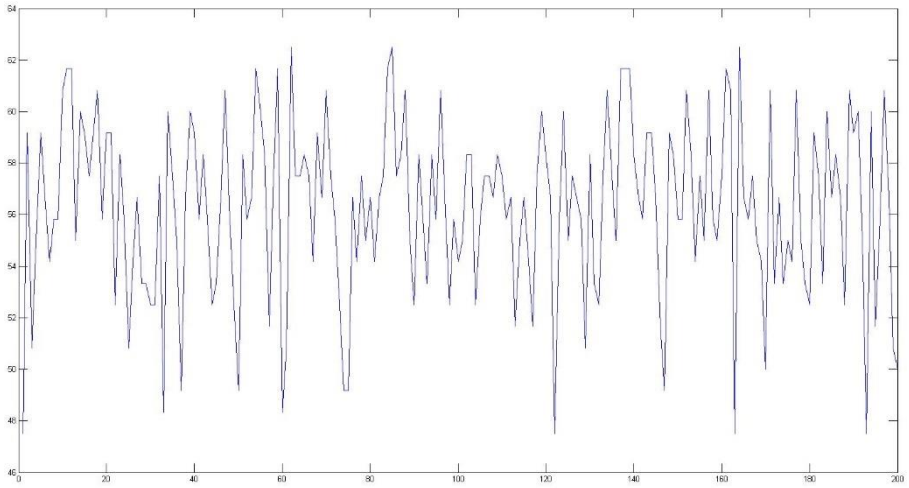
Opakování	102	143	4	127	99
MSE	0,194	0,225	0,243	0,245	0,259
Úspěšnost	55 %	57 %	51 %	56 %	53 %

Tabulka 11: Vyhodnocení pěti nejmenších kvadratických chyb MSE pro YAHOO

Zdroj: (Vlastní)

Nejnižší možná chyba byla až ve sto druhém opakování, a to 0,194. Průměrem pěti hodnot je hodnota 0,243. Tuto hodnotu si opět určíme jako minimální chybu, kterou bude síť při učení pro společnost YAHOO používat.

- **Maximální úspěšnost s minimální MSE**



Graf 15: Nejvyšší úspěšnost s minimální kvadratickou chybou MSE pro YAHOO

Zdroj: (Vlastní)

Hodnota nejlepší úspěšnosti se zde pohybuje mezi padesáti pěti až šedesáti procenty. Pět nejvyšších úspěšností odhadu si zobrazíme v tabulce níže.

Opakování	62	85	164	11	12
Úspěšnost	63 %	63 %	63 %	62 %	62 %
MSE	0,441	0,461	0,26	0,402	0,648

Tabulka 12: Vyhodnocení pěti nejvyšších úspěšností s nejmenší kvadratickou chybou MSE pro YAHOO

Zdroj: (Vlastní)

Nejvyšší úspěšnost 63 % byla dosažena dvakrát, avšak v šedesátém druhém pokusu byla nižší kvadratická chyba, a to 0,441. Pro porovnání, nejmenší možná kvadratická chyba byla 0,194 s úspěšností 55 %. Pro společnost YAHOO jsou hodnoty pro úspěšné natrénování modelu nastaveny na úspěšnost vyšší jak 61 % a s kvadratickou chybu 0,5.

Souhrn

Vzhledem k náhodnosti učení není možné určit minimální hodnotu kvadratické chyby. Na základě pokusů jsme však získali informace pro každou neuronovou síť, které nám postačí k tomu abychom měli síť lépe optimalizovanou a mohli pokračovat v dalším

zkoumání. Pro hledání minimální kvadratické chyby není brána v potaz úspěšnost sítě, proto jsou pro nás tyto údaje spíše informativní. V tomto kroku se snažíme najít co nejmenší možnou chybu a pak ji porovnat s nejvyšší dosaženou úspěšností. V tabulce níže je zaznamenána vždy nejmenší možná chyba a vedle ní je hodnota, která pro nás bude pro naučení sítě dostačující.

	Neurony	Zpoždění	Minimální MSE	Stanovené MSE
NETFLIX	36	3	3,634	4,645
APPLE	49	5	0,753	0,777
YAHOO	37	3	0,194	0,243

Tabulka 13: Optimalizované modely s minimální kvadratickou chybou MSE

Zdroj: (Vlastní)

Z pohledu úspěšnosti s minimální chybou bude rozhodovat jak úspěšnost, tak minimální chyba podle tabulky níže.

	Neurony	Zpoždění	Stanovené MSE	Stanovená úspěšnost
NETFLIX	24	5	7,5	64 %
APPLE	23	5	2,5	61 %
YAHOO	27	2	0,5	61 %

Tabulka 14: Optimalizované modely s nejvyšší úspěšností a minimální MSE

Zdroj: (Vlastní)

Zjistili jsme tedy optimální nastavení sítě pro stanovená data. V dalším kroku provedeme simulaci sítě na základě obchodní simulace, které se věnuje další kapitola.

3.3 Simulace sítě

K simulaci sítě je vytvořený demo program obchodování, na základě kterého můžeme vyhodnotit zisk na minulých datech a zjistit tak využitelnost predikovaných hodnot. Demo model je však hodně zjednodušený a nelze se jím řídit na reálném trhu. Pro širší využití programu v praxi by bylo nutné jeho detailnější zpracování.

Pro zapnutí dema je potřeba nahrát data, aby program fungoval. První krokem je nahrání reálných dat, tedy těch dat, na kterých se síť učila. Dalším parametrem je velikost intervalu, na kterém se obchodní simulace bude provádět. Výchozí nastavení simulace je sto dní. Tato hodnota se však dá změnit dle uvážení. Posledním parametrem, který je nutno zadat, je již naučená neuronová síť.

Po zadání potřebných dat pak probíhá samotná simulace. Tento proces probíhá pomocí algoritmu, kdy se porovnávají hodnoty, které jsou reálné, respektive je porovnávána současna cena akcie proti predikované hodnotě, která představuje hodnoty vytvořené neuronovou sítí.

3.3.1 Nákup

Při nákupu se nejdříve zkontroluje podmínka, zda je současná cena nižší než cena predikovaná. Pokud je tato podmínka splněna, musíme zajistit, zda na trhu nejsme v krátké pozici. Pokud jsme v krátké pozici, musíme z ní nejdříve vystoupit, jelikož očekáváme, že cena akcie se bude v následujícím dnu zvyšovat. Z krátké pozice vystoupíme nákupem akcií, a to tak, že od hodnoty, kterou momentálně máme, *balance* odečteme dnešní cenu akcie *today_Price* vynásobenou počtem nakoupených akcií. Poté již můžeme přijít k samotnému obchodu. Tudíž se provede funkce *buy* neboli nákup jedné akcie. Pokud tedy nakoupíme jednu akcii, odečteme si z momentální hodnoty *balance* současnou cenu akcie *today_Price* a připíšeme si ke svému vlastnictví jednu akcii, nakoupenou funkcí *buy*. Dále se v tomto kroku nastaví funkce nastavení ztrát *stop_Loss* na stejnou cenu, za kterou jsme akcie nakupovali.

```
if today_Price(1,i) < predicted_Price(1,i)                  %Predpokladam rust akcie - nakupuji
    if sell>0                                               %Pokud vlastním akcie formou sell, musím vystoupit z trhu
            balance=balance-(sell*today_Price(i)*pocetAkcii);   %Realizování zisku
            sell=0;                                         %Již nevlastním žádnou akcii formou sell
            good=good+1;                                    %Realizován dobrý obchod
            mi=0;                                           %Nedržím žádné akcie
    end
    balance=balance-(today_Price(1,i)*pocetAkcii);          %Nákup akcie formou buy
    stop_Loss=today_Price(1,i);                             %Pri teto cene vystupuji z trhu
    buy=buy+1;                                              %Nakup jedne akcie
    process=0;                                              %Funkce pro dalsi rozhodování - 0=nákup
```

Obrázek 15: Ukázka kódu nákupu akcií

Zdroj: (Vlastní)

3.3.2 Prodej

Prodej bude realizován stejným způsobem jako byl nákup, nejdříve se tedy zkontroluje podmínka, zda je současná cena akcie vyšší než predikovaná cena. Po splnění této podmínky musíme zajistit, že na trhu nejsme v dlouhé pozici. Pokud jsme v dlouhé pozici, musíme z ní nejdříve vystoupit, jelikož očekáváme, že cena akcie se bude v následujícím dnu snižovat. Z dlouhé pozice vystoupíme prodejem akcií, a to tak, že od hodnoty, kterou momentálně máme *balance* odečteme současnou cenu akcie *today_Price* vynásobenou počtem nakoupených akcií. Poté již můžeme přejít k samotnému prodeji. Tudíž se provede funkce *sell*, neboli prodej jedné akcie. Pokud tedy prodáme jednu akcii, přičteme si k momentální hodnotě *balance* dnešní cenu akcie *today_Price* a připíšeme si ke svému vlastnictví jednu akcii prodanou funkcí *sell*. Dále se v tomto kroku nastaví funkce nastavení ztrát *stop_Loss* na stejnou cenu, za kterou jsme akcie prodávaly.

```
elseif today_Price(1,i)>predicted_Price(1,i)                %Predpokladam pokles akcie - prodavam
            if buy>0                                        %Pokud vlastním akcie formou buy, musím vystoupit z trhu
            balance=balance+(buy*today_Price(i)*pocetAkcii); %Realizování zisku
            buy=0;                                          %Již nevlastním žádnou akcii formou buy
            good=good+1;                                    %Realizován dobrý obchod
            mi=0;                                           %Nedržím žádné akcie
            end
    balance=balance+(today_Price(1,i)*pocetAkcii);          %Nákup akcie formou buy
    stop_Loss=today_Price(1,i);                             %Pri teto cene vystupuji z trhu
    sell=sell+1;                                            %Nakup jedne akcie
    process=1;                                              %Funkce pro dalsi rozhodování - 0=nákup
```

Obrázek 16: Ukázka kódu prodeje akcií

Zdroj: (Vlastní)

3.3.3 Realizování ztráty x zisku z nákupu

Nejdříve musíme zjistit, zda jsme v minulém kroku nakupovali. Proto, jestli je potřeba vystoupit z trhu a realizovat ztrátu, musíme porovnat dnešní skutečnou hodnotu s hodnotou, na kterou jsme si stanovili *stop_Loss*, neboli se jedná o skutečnou včerejší hodnotu. Pokud je dnešní skutečná hodnota nižší jak hodnota zastavení ztráty *stop_Loss*, musíme vystoupit z trhu, jelikož se jedná o špatnou předpověď. Očekávali jsme, že hodnota poroste, jenže ve skutečnosti klesala. Tedy je to pro nás špatně realizovaný obchod a jedná se o ztrátu. Z trhu vystoupíme prodejem všech akcií, které jsme nakoupili. K momentální hodnotě, kterou máme *balance* přičteme současnou cenu akcie vynásobenou počtem akcií, které momentálně vlastníme.

Pokud jsme odhadli trend správně, můžeme realizovat zisk. V tomto kroku je však pouze nastaveno zastavení ztrát *stop_Loss* na vyšší hodnotu. Realizování zisku bude až

v případě, že nakoupené akcie vlastníme déle, než je stanovený interval *dobaDrzeniAkcie*, která je defaultně nastavena na hodnotu dva.

```
if process==0
    if today_Price(1,i+1) < stop_Loss                    %Byl realizovan nakup
        balance=balance+(buy*today_Price(1,i+1)*pocetAkcii);  %Pokud jsem nakoupil za today_Price a zitrejsi hodnota klesla
        buy=0;                                           %Prodej akcie, ktere mam formou buy
        bad=bad+1;                                       %Nemam zadne akcie formou buy
        mi=0;                                            %Realizovan spatny obchod
        stop_Loss=0;                                     %Nevlastnim zadne akcie
    elseif today_Price(1,i+1)>=stop_Loss                 %Nejsem na trhu, neni stop Loss
       stop_Loss=today_Price(1,i+1);                     %Pokud jsem nakoupil za today_Price a zitrejsi hodnota vstoupla
    end                                                  %Zmena stop Loss

    if mi>=dobaDrzeniAkcie && buy>0                      %Pokud drzim akcie pozadovanou dobu
       balance=balance+(buy*today_Price(i)*pocetAkcii);  %Realizovani zisku
       buy=0;                                            %Jiz nevlastni akcie formou buy
       mi=0;                                             %Nevlastnim zadne akcie
       stop_Loss=0;                                      %Nejsem na trhu, neni stop Loss
       good=good+1;                                      %Realizovan dobry obchod
    end
end
```

Obrázek 17: Ukázka kódu prodeje po nákupu

Zdroj: (Vlastní)

3.3.4 Realizování ztráty x zisku z prodeje

Tento krok je opět stejný jako předchozí, akorát se jedná o hodnotu, kterou budeme realizovat z prodeje. Prvně tedy musíme zjistit, zda se jedná o ztrátu nebo o zisk. O ztrátu se jedná v případě, když současná skutečná hodnota je vyšší než hodnota zastavení ztráty *stop_Loss*. Jelikož jsme vstoupili na trh s očekáváním, že hodnota akcie bude klesat, ale ona vystoupala, je to pro nás špatně realizovaný obchod a musíme tak z krátké pozice vystoupit. Z trhu vystoupíme nákupem akcií, které jsme prodávali. Toho docílíme tak, že z hodnoty, kterou máme *balance* odečteme současnou cenu akcie vynásobenou počtem akcií, které jsme prodávali.

Pokud jsme odhadli trend správně, můžeme realizovat zisk, stejně jako při nákupu však pouze změním hodnotu zastavení ztráty a zisk realizujeme až v případě, že nakoupené akcie vlastníme déle, než je stanovený interval *dobaDrzeniAkcie*.

```
if process==1                                          %Byl realizovan prodej
    if today_Price(1,i+1) > stop_Loss                  %Pokud jsem prodaval za today_Price a zitrejsi hodnota vstoupla
        balance=balance-(sell*today_Price(1,i+1)*pocetAkcii);  %Nákup akcie, ktere mam formou sell
        sell=0;                                        %Nemam zadne akcie formou sell
        bad=bad+1;                                     %Realizovan spatny obchod
        mi=0;                                          %Nevlastnim zadne akcie
        stop_Loss=0;                                   %Nejsem na trhu, neni stop Loss
    elseif today_Price(1,i+1) <=stop_Loss              %Pokud jsem prodal za today_Price a zitrejsi hodnota klesla
        stop_Loss=today_Price(1,i+1);                  %Zmena stop Loss
    end

        if mi>=dobaDrzeniAkcie && sell>0               %Pokud drzim akcie pozadovanou dobu
            balance=balance-(sell*today_Price(i)*pocetAkcii);  %Realizovani zisku
            sell=0;                                    %Jiz nevlastni akcie formou buy
            mi=0;                                      %Nevlastnim zadne akcie
            good2=good2+1;                             %Realizovan dobry obchod
    end
            stop_Loss=0;                               %Nejsem na trhu, neni stop Loss
        end
end
```

Obrázek 18: Ukázka kódu nákupu po prodeji

Zdroj: (Vlastní)

3.3.5 Výstup simulace

Výstup simulace nám vypíše, jakého bychom dosahovali zisku podle naučené neuronové sítě. Také nám vypíše, kolik obchodů bylo odhadnuto správně, tedy kolikrát byl realizovaný zisk a kolik obchodů bylo chybných, neboli byla realizována ztráta.

3.3.6 Běh simulace pro optimalizované sítě

Postupně tedy zapneme simulace pro všechny vytvořené a optimalizované modely a zjistíme, jak na tom jsou z pohledu ziskovosti. Předtím, než se na výsledky podíváme, bych rád připomenul, že zisky jsou uváděné pro nákup jedné akcie společnosti. Pokud bychom nakoupili více akcií než jen jednu, zisk by se násobil požadovaným nákupem. Výsledky běhu pro všechny modely optimalizované podle minimální kvadratické chyby jsou zavedeny to tabulky níže.

	Dobrý obchod	Špatný obchod	Zisk
NETFLIX	42	42	50,24
APPLE	39	47	18,49
YAHOO	41	45	-6,71

Tabulka 15: Zisk po běhu simulace pro model s minimální kvadratickou chybou MSE

Zdroj: (Vlastní)

Nejlepšího výsledku bylo dosaženo modelem vytvořeným pro NETFLIX, a to zisku 50,24$. Přitom bylo realizováno čtyřicet dva dobrých a čtyřicet dva špatných obchodů, zbývající dny, které jsou do počtu sta obchodních dnů, si aplikace držela více než jednu akcii a prodala ji společně s dobře, nebo špatně realizovaným obchodem. Nejhorší výsledek byl dosažen pro firmu YAHOO, pro tento model vznikla dokonce ztráta 6,71$.

Stejně tak se podíváme na ziskovost pro optimalizovanou síť z hlediska nejvyšší úspěšnosti s nejmenší kvadratickou chybou.

	Dobrý obchod	Špatný obchod	Zisk
NETFLIX	41	36	64,13
APPLE	37	40	19
YAHOO	44	37	17,55

Tabulka 16: Zisk po běhu simulace pro modely optimalizované dle nejvyšší úspěšnosti s minimální MSE

Zdroj: (Vlastní)

Nejlepší model opět tvoří model vytvořený pro NETFLIX, který dosahuje dokonce vyššího zisku než model, který byl tvořen pro nejmenší možnou kvadratickou chybu, a to 64,13$.

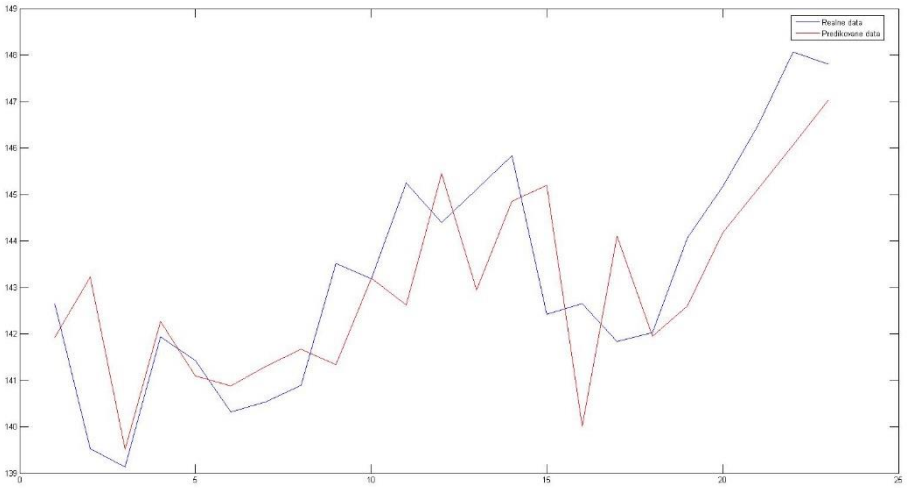
3.4 Predikce

Předchozí kroky, včetně ziskovosti modelu, byly uskutečněny na síti již známých datech. V dalším korku budeme testovat, jak model dokáže správně predikovat hodnoty. Vzhledem k půlročním datům, na kterých se síť učila, je vhodné predikovat vždy pouze pro každý jeden následující den. Nemůžeme tedy v testu očekávat nikterak překvapivé výsledky. K tomu, abychom mohli predikovat více dnů, můžeme použít dvě metody. Jedna spočívá v přeučení neuronové sítě pro následující den. Zde však může vzniknout komplikace z pohledu optimalizace sítě, jelikož síť by se pokaždé učila na nových datech a nemusela by splňovat stanovené podmínky optimalizace. Druhým způsobem je předávat optimalizované síti nová data a na základě nich predikovat novou hodnotu.

Může se však stát, že si data s některou novou hodnotou nebudou moci poradit. Přistoupil jsem tedy ke druhému způsobu.

Data jsou pro všechny modely zakreslena do níže uvedených grafů. Stejně tak je na datech zapnuta výše zmíněná simulace obchodování, která je nastavena na délku 20 dnů a je vypočítána úspěšnost predikce.

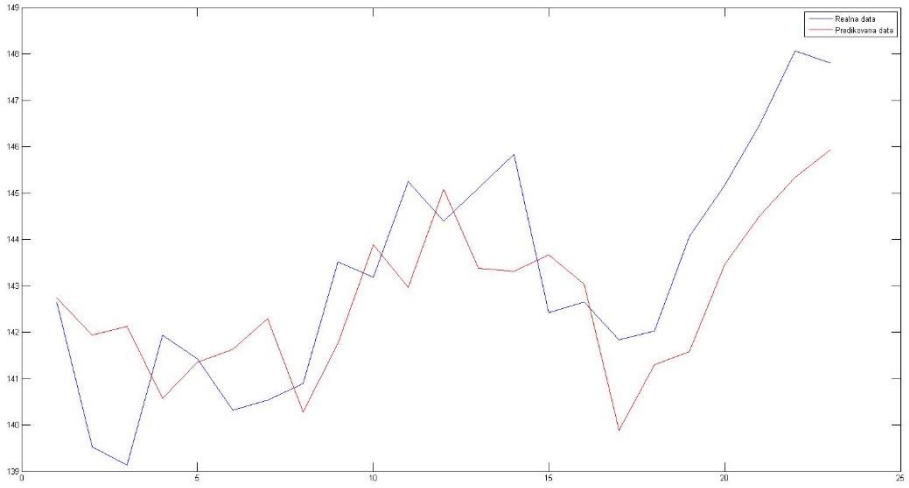
NETFLIX



Graf 16: Predikce 20 dnů pro NETFLIX podle minimální MSE

Zdroj: (Vlastní)

Predikovaná hodnota celkem věrně kopíruje hodnoty nových reálných dat. Úspěšnost třiadvaceti denní predikce pro optimalizovanou síť NETFLIX pomocí nejmenší kvadratické chyby činní 68 % se ziskem 5,91$. Můžeme tak říci, že model pro firmu NETFLIX je vhodně optimalizovaný, jelikož výsledky jsou uspokojivé.



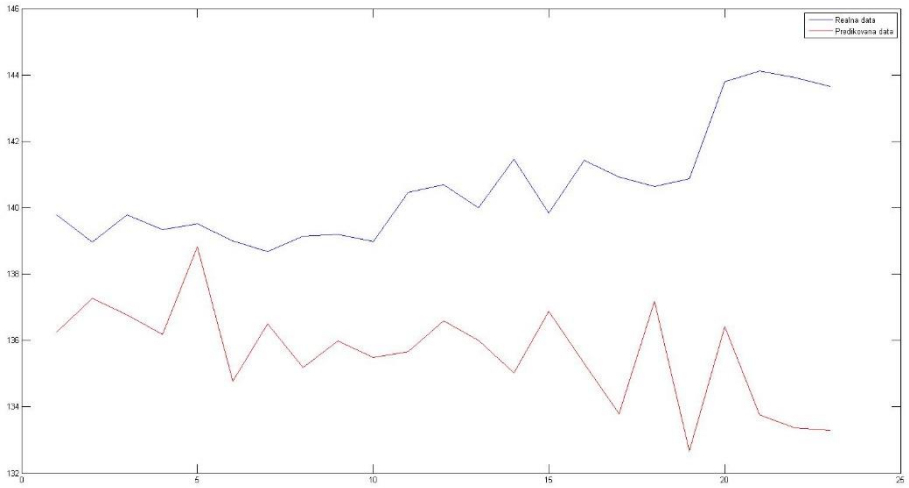
Graf 17: Predikce pro 20 dnů pro NETFLIX podle maximální úspěšnosti a minimální MSE

Zdroj: (Vlastní)

Predikované hodnoty i v tomto případě celkem věrně kopírují reálná data. Úspěšnost pro síť, která byla optimalizována na základě nejvyšší úspěšnosti s minimální kvadratickou chybou, je 55 % se ziskem 3,13$.

Vidíme, že tedy lepší výsledky dosahovala síť optimalizovaná pro nejmenší kvadratickou chybu.

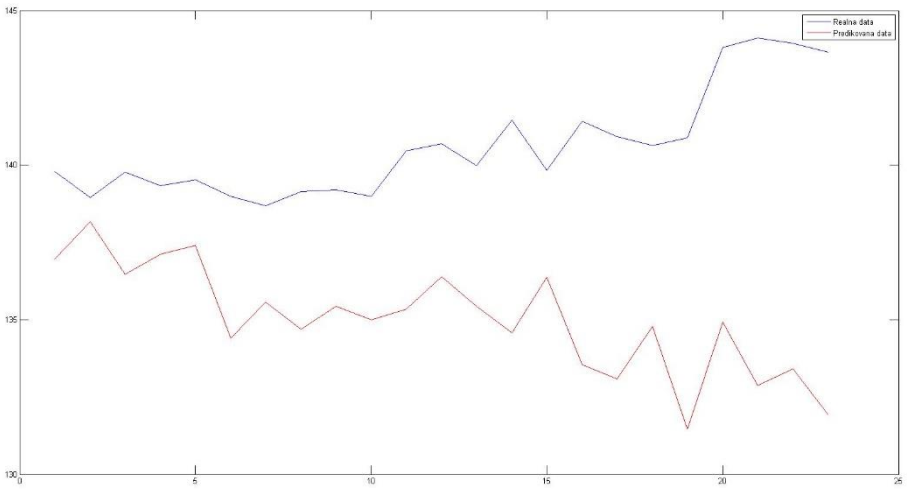
APPLE



Graf 18: Predikce 20 dnů pro APPLE podle minimální MSE

Zdroj: (Vlastní)

V tomto grafu už se nám hodnoty rozcházejí a predikovaná data se od skutečných celkem významně vzdalují, a to i přes dostačující výsledky v předchozích krocích. Pokud však provedeme tyto data simulací obchodování, dostaneme i přesto celkem uspokojivé výsledky. Je to z toho důvodu, že nekontrolujeme přesné hodnoty naučené sítě, ale rozhodujeme se na základě trendu. Ten naše naučená síť odhadla správně v 50 %. Pomocí obchodního dema bychom se dostali do ztráty 9,45$. Problém může být přeučení sítě, jelikož síť správně pracuje na starých datech a má vyhovující výsledky, pokud však síť dostane data nová, nedokáže si s nimi poradit tak, jak bychom očekávali. Proto, aby byl tento model lépe optimalizován, museli bychom volit jiný počet neuronů nebo zpoždění.

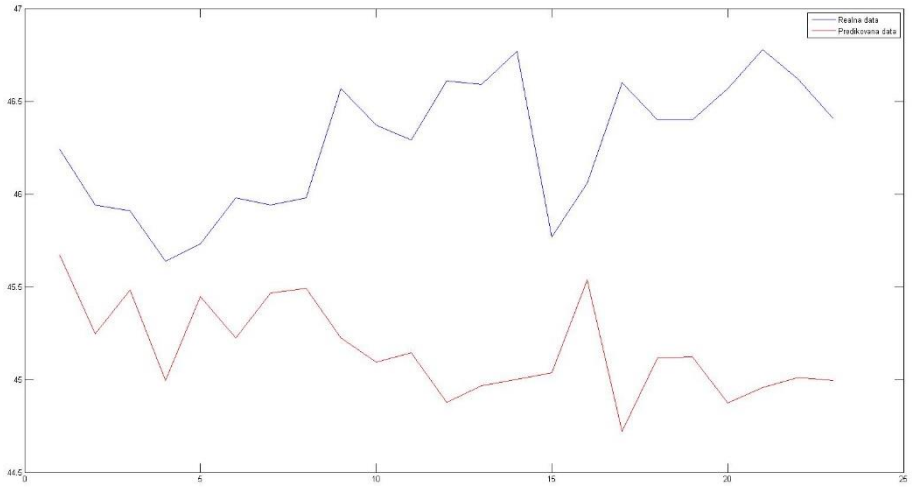


Graf 19: Predikce pro 20 dnů pro APPLE podle maximální úspěšnosti a minimální MSE

Zdroj: (Vlastní)

Také v případě optimalizovaného modelu pomocí maximální úspěšnosti s minimální kvadratickou chybou se data významně rozcházejí a model tedy nedokáže zareagovat na nová data. Trend byl však odhadnut stejně jak v minulém případě také s 50% úspěšností a ztráta tak činí 9,45$.

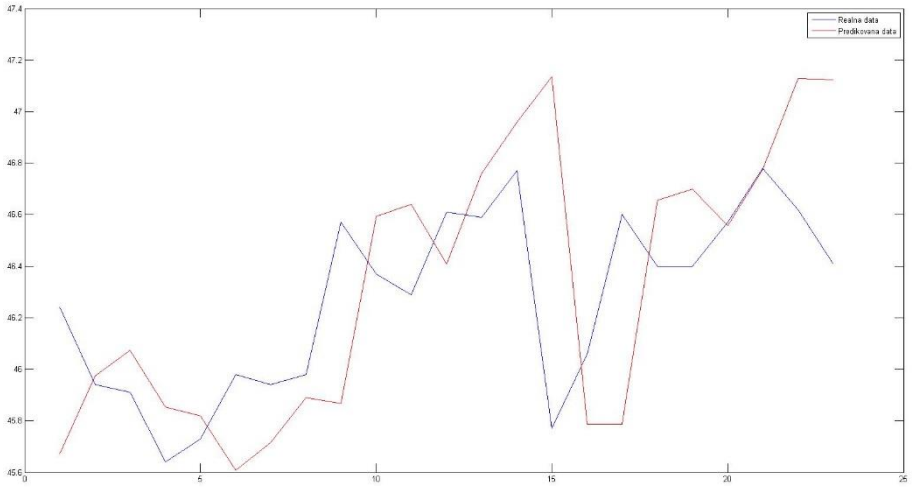
YAHOO



Graf 20: Predikce 20 dnů pro YAHOO podle minimální MSE

Zdroj: (Vlastní)

Ani pro YAHOO nevypadá predikce ideálně a vznikl tak stejný problém jako v případě APPLE, a to ten, že síť je přeučená a je potřeba upravit počet neuronů. Úspěšnost odhadu trendu však dosahuje opět k 50 % a ztráta po obchodní simulaci je 1,39$.



Graf 21: Predikce pro 20 dnů pro YAHOO podle maximální úspěšnosti a minimální MSE

Zdroj: (Vlastní)

Predikce pro YAHOO pomocí optimalizace maximální úspěšnosti s nejmenší chybou dosahuje viditelně lepších výsledků než predikce pouze pomocí minimální kvadratické chyby. Predikované hodnoty se blíží reálným hodnotám. Úspěšnost odhadu trendu této predikce je 59 %, i přesto je však pro obchodní model realizována ztráta ve výši 2,14 $.

3.5 Shrnutí výsledků

V této části bude popsáno shrnutí výsledků vytvořených modulů.

Prvně se podíváme na vytvořené modely pomocí metody minimalizace kvadratické chyby. Ty jsou zaneseny do tabulky níže.

	NETFLIX	APPLE	YAHOO
Stanovené MSE	3,843	0,777	0,243
Zisk - minulá	50,24	18,49	-6,71
Úspěšnost - minula	58	54	56
Zisk - predikovaná	5,91	-9,45	-1,39
Úspěšnost - predikovaná	68	50	50
HURST	0,8019	0,7908	0,8115

Tabulka 17: Shrnutí výsledku pro modely optimalizované pomocí minimální kvadratické chyby MSE

Zdroj: (Vlastní)

Vzhledem k prediktabilitě na tom byl nejlépe model vytvořený pro časovou řadu tvořenou akciemi od firmy NETFLIX. Hodnota Hurstova exponentu u všech společností dosahuje hodnoty kolem 0,8, což značí, že časové řady mají paměťový cyklus a jsou dobře predikovatelné. Predikce pro firmu APPLE i YAHOO dopadly o něco hůře. Model pro tyto společnosti odhadl trend pouze s 50 % úspěšností a tyto dva modely po zakreslení do grafu nevykazovaly dobré výsledky a hodně se rozcházeli se skutečnou hodnotou. Nejlépe na tom tedy po celkovém hodnocení je model vytvořený pro NETFLI. Ten vykazoval již na minulých datech nejlepší úspěšnost, a to 58 % se ziskem 50,24 $. Když jsme tomuto modelu zadali predikovaní dat na dalších dvacet dní, ukázalo se, že dokáže predikovat s 68 % úspěšností, což je uspokojivý výsledek. Za dvacet obchodních dní byl vygenerován zisk 5,91$, při nákupu a prodeji vždy jedné akcie. Můžeme tedy říct, že optimalizace modelu s nejmenší kvadratickou chybou dopadla nejlépe pro NETFLIX a jeho predikční výsledky dosahují uspokojivých výsledků.

V druhém bodu je popsán model vytvořený pomocí metody nejmenší kvadratické chyby, avšak společně s nejvyšší úspěšností odhadu.

	NETFLIX	APPLE	YAHOO
Stanovené MSE	7,5	2.5	0.5
Stanovená Úspěšnost	64	61	61
Úspěšnost - minulá	66	62	64
Zisk - minulá	64,13	19	17,55
Úspěšnost - predikovaná	55	50	59
Zisk – predikovaná	3,13	-9,45	-2,14
HURST	0,8019	0,7908	0,8115

Tabulka 18: Shrnutí výsledků pro modely optimalizované podle nejvyšší úspěšnosti s nejmenší kvadratickou chybou

Zdroj: (Vlastní)

Modely optimalizované pro nejvyšší úspěšnost dosahovaly na minulých datech úspěšnosti vyšší jak 60 %. Nejlépe na tom byl model optimalizovaný pro společnost NETFLIX, který dosahoval úspěšnosti 66 % a ziskovosti 64,13$. Model tvořený pro společnost YAHOO dosahoval úspěšnosti 64 % se ziskem 17,55$ a model tvořený pro společnost APPLE dosahoval úspěšnosti 62 % se ziskem 19$. Pokud se však podíváme na predikci, na první pohled vidíme, že stejně jako při minimalizaci kvadratické chyby je na tom nejhůře APPLE, kde model dosahuje naprosto shodných výsledků, čili úspěšnosti na predikovaných datech 50 % se ztrátou 9,45$. Ovšem YAHOO zde dosahuje jiných výsledků a model dosahuje 59 % úspěšnosti odhadu trendu, ovšem se ztrátou 2,14$ a celkem věrně také kopíruje data na reálném trhu. Podobně tomu je u grafu pro model na základě dat časové řady NETFLIX, která sice měla menší úspěšnost odhadu trendu, a to 55 %, ale končí se ziskem 3,13$. V modelu vytvořeném podle úspěšnosti s nejmenší chybou, tak nemáme jednoznačného vítěze a o nejlepší model se dělí model NETFLIX společně s modelem YAHOO.

3.5.1 Rozhodování

V poslední části je vytvořen modul pro podporu rozhodování. Tento modul analyzuje časovou řadu pomocí Hurstova exponentu, který nám říká, jestli časová řada obsahuje paměťový cyklus a predikovatelnost časové řady. Výpočet Hurstova exponentu byl uveden v teoretické části práce. Byl vytvořen program, který hledá optimalizovaný model

na nových datech, tedy všechny výše zmíněné kroky a postupy jsou zautomatizovány. Uživateli tak stačí nahrát časovou řadu a program už ji zpracuje a vytvoří model, na základě kterého uživateli doporučí, zda vstoupit na trh a v jaké pozici. Kroky, jak nahrát data do programu, byly popsány výše. Níže je ukázka výpisu finálního kódu, který byl vytvořen také pro společnost NETLFIX.

```
Pocet neuronu byl vybran: 17
Velikost zpodeni je: 3
Uspesnost odhadu trendu je: 57.9832
Kvadraticka chyba MSE: 12.6597
----------------------------------------
Bylo realizovano: 20 dobrych obchodu
Bylo realizovano: 17 spatnych obchodu
Ziskovost modelu je: 31.9701$
----------------------------------------
Hodnota Hurstova exponentu je: 0.80187
Soucasna hodnota akcie je: 142.13
Predikovana hodnota akcie je: 142.8505
Vstoupit do pozice: buy
```

Obrázek 19: Výpis finálního kódu pro automaticky zoptimalizovaný model

Zdroj: (Vlastní)

ZÁVĚR

V diplomové práci jsem se zabýval využitím umělé inteligence na kapitálových trzích. V první části práce byla popsána teoretická východiska jakožto kapitálový trh, časové řady spojené s kapitálovým trhem a nástroje umělé inteligence jako je fuzzy logika, genetické algoritmy a v práci využívané neuronové sítě. V praktické části pak byla provedena ukázka tvorby modelu pomocí neuronové sítě NAR a v následujících krocích byla popsána její optimalizace.

Jako vstupní parametry časové řady bylo zvoleno pět časových řad známých společností, z nichž dvě nesplňovali požadovanou úspěšnost hned při prvním testování. Pro další tři společnosti byly vytvořeny modely. Pro každou společnost vždy zvlášť jeden model optimalizovaný pouze podle nejmenší možné kvadratické chyby a druhý pomocí nejvyšší dosažené úspěšnosti odhadu trendu.

Dále byl v práci vytvořen obchodní simulátor, kde bylo na vytvořených modelech zkoumáno, jak bychom si vedli, kdybychom se podle daných modelů na reálném trhu řídili. Jinými slovy, jestli bychom dosahovali zisku nebo ztráty.

Dalším testem pro vytvořené modely byla skutečná predikce. Podle ní jsme zjistili, že tři modely nemají v tomto směru uspokojivé výsledky, což bylo dáno zřejmě přeučením neuronové sítě. Proto, aby tyto modely měly lepší výsledky, museli by se opět upravit počty neuronů a zpoždění. Rovněž bychom mohli zkusit predikci pomocí přeučení modelu po každém vykonaném kroku. Další tři modely však dosahovali výsledky, které byly uspokojivé.

Nejlepší model, který se povedlo vytvořit, dosahoval padesáti osmi procentní úspěšnosti na historických datech se ziskem 36,46$, na predikovaných dvaceti dnech dosahoval až šedesáti osmi procentní úspěšnost se ziskem 5,91$. Jednalo se o model tvořený pro společnost NETFLIX, který byl optimalizovaný podle nejmenší kvadratické chyby, která činila 5,545. Druhý model, který vykazoval dobré výsledky, byl také tvořen pro společnost NETFLIX, ale jeho optimalizace byla na základě kompromisu mezi kvadratickou chybou a úspěšností na historických datech. V tomto modelu tedy kvadratická chyba dosahovala větší nepřesnosti, konkrétně 6,972, ale úspěšnost na historických datech dosahovala šedesáti šesti procent a zisk tvořil 64,13$. Na historických datech byl tak tento model ziskovější. Když jsme však dali modelu predikovat data na

dvacet dní, jeho výsledky dosahovaly horších hodnot než v předchozím případě. Tento model na predikovaných datech dosahoval padesáti pěti procentní úspěšnosti a jeho zisk v obchodním modelu byl 3,13$. Poslední model, který dosahoval dobrých výsledků, co se predikce týče, byl model tvořený pro firmu YAHOO, který fungoval na základě kompromisu mezi úspěšností odhadu trendu a minimální kvadratickou chybou. Tento model na historických datech dosahoval šedesáti čtyř procentní úspěšnosti se ziskem 17,55$. Na predikovaných datech sice vykazoval ztrátové hodnoty konkrétně 2,14$, ale úspěšnost jeho odhadu byla padesát devět procent.

V poslední části byl vytvořen rozhodovací modul, který sám zvládne všechny uvedené optimalizace a na konci predikuje novou hodnotu, která by se měla další den objevit. Na tomto základě se uživatel může rozhodnout, zda na trh vstoupí a do jaké pozice.

V práci tedy byly splněny všechny stanovené cíle, úspěšně byl vytvořen a optimalizován model, který na konci dosahoval šedesáti osmi procentní úspěšnosti predikce trendu na dvaceti dnech.

SEZNAM POUŽITÉ LITERATURY

BIBLIOGRAFICKÉ CITACE

1. SYNEK, M a E. KISLINGEROVÁ.: *Podniková ekonomika*. 5. přeprac. a dopl. vyd. Praha: C.H. Beck, 2010, 445 s. ISBN 978-80-7400-336-3.
2. REJNUŠ, O.: *Peněžní ekonomie: (Finanční trhy).* Podnikatelská fakulta - nakladatelství CERM, 2007, ISBN 978-80-214-3466-0.
3. POLÁCH, J.: *Peněžní a kapitálové trhy*. Svazek 2. Univerzita Tomáše Bati ve Zlíně, 2008, ISBN 978-80-7318-758-3.
4. REJNUŠ, O.: *Cenné papíry a burzy*. 2. přepracované vyd. Brno: CERM, 2013, 406 s. ISBN 978-80-214-793-703-1.
5. REŽNÁKOVÁ, M.: *Finanční management :(pro obor Podnikové finance a obchod): studijní text pro kombinované studium.* 1. vyd. Brno:Vysoké učení technické, Fakulta podnikatelská v nakl. Zdeněk Novotný, 2001, 183 s. ISBN 80-214-1968-7
6. VESELÁ, J.: *Analýzy trhu cenných papír·: Fundamentální analýza. II. díl.* Oeconomica, 2003, ISBN 80-245-0506-1.
7. MURPHY, J. J.: *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications.* New York Institute of Finance, 1999, ISBN 0-7352-0066-1.
8. SOJKA, Z., DOSTÁL, P.: *Elliottovy vlny*. 1. Vydání. Tribun EU. Brno, 2008. 272s. ISBN: 978-80- 7399-630-7.
9. KROPÁČ, Jiří. *Statistika A: náhodné jevy, náhodné veličiny, náhodné vektory, indexní analýza, rozhodování za rizika*. 1. vyd. Brno: Jiří Kropáč, 2006, 151 s. ISBN 80-214-3194-6.
10. DOSTÁL, P.: *Pokročilé metody analýz a modelování v podnikatelství a veřejné správě.* 1. vyd. Brno: CERM, 2008. 340 s. ISBN 978-80-7204-605-8.
11. DOHNAL, M.: *Metody investičního rozhodování: studijní text pro denní i kombinovanou formu studia*. Vyd. 1. Brno: Akademické nakladatelství CERM, 2006, 67 s. ISBN 80-214-3133-4.
12. FANTA, J.: *Psychologie, algoritmy a umělá inteligence na kapitálových trzích.* 1.vyd. Praha: Grada Publishing, 2001. 168 s. ISBN 80-247-0024-7.

13. MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J.: *Umělá inteligence*. 1. Vyd. Academia, 1993, ISBN 80-200-0502-1.
14. HAGAN, M. T., DEMUTH, H. B., BEALE, M. H.: *Neural Network Design*. 1996, ISBN 7-111-10841-8.
15. MYSLIVEČEK, J.: *Základy neurověd.* Triton, 2003, ISBN 80-7254-234-6.
16. ŠÍMA, J., NERUDA, R.: *Teoretické otázky neuronových sítí*. Matfyzpress. 1996, ISBN 80-85863-18-9.
17. DOSTÁL, P. *Advanced Decision Making in Business and Public Services.* CERM, Brno, 2011, 168 s., ISBN 978-80-7204-747-5.
18. KECMAN, V.: *Learning and soft computing: support vector machines, neuralnetworks, and fuzzy logic models.* MIT Press, 2001, ISBN 00-262-11255-8.
19. FAUSSET, L. V.: *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Upper Saddle River : Prentice Hall, 1994, ISBN 0-13-334186-0.

SEZNAM VZORCŮ

(1.1) Časová řada 25
(1.2) Výpočet Hurstova Exponentu 1 26
(1.3) Výpočet Hurstova Exponentu 2 26
(1.4) Výpočet Hurstova Exponentu 3 26
(1.5) Výpočet Hurstova Exponentu 4 26
(1.6) Výpočet Hurstova Exponentu 5 26
(1.7) Potenciál neuronu 31
(1.8) Hardlim 31
(1.9) Purelin 31
(1.10) Logsig 31
(1.11) Tansig 31
(1.12) Hodnota změny váhy v aktualním kroku 36
(1.13) Hodnota změny váhy 36
(1.14) NAR 44
(1.15) NARX 45

SEZNAM TABULEK

Tabulka 1: Výsledky optimalizace neuronů a zpoždění pro NETFLIX 52
Tabulka 2: Výsledky optimalizace neuronů a zpoždění pro APPLE 53
Tabulka 3: Výsledky optimalizace neuronů a zpoždění pro GOOGLE 54
Tabulka 4: Výsledky optimalizace neuronů a zpoždění pro AMAZON 55
Tabulka 5: Výsledky optimalizace neuronů a zpoždění pro YAHOO 56
Tabulka 6: Výsledky optimalizace neuronů a zpoždění 57
Tabulka 7: Vyhodnocení pěti nejmenších kvadratických chyb MSE pro NETLIX 58
Tabulka 8: Vyhodnocení pěti nejvyšších úspěšností s nejmenší kvadratickou chybou MSE pro NETLIX 59
Tabulka 9: Vyhodnocení pěti nejmenších kvadratických chyb MSE pro APPLE 60
Tabulka 10: Vyhodnocení pěti nejvyšších úspěšností s nejmenší kvadratickou chybou MSE pro APPLE 61
Tabulka 11: Vyhodnocení pěti nejmenších kvadratických chyb MSE pro YAHOO 62
Tabulka 12: Vyhodnocení pěti nejvyšších úspěšností s nejmenší kvadratickou chybou MSE pro YAHOO 63
Tabulka 13: Optimalizované modely s minimální kvadratickou chybou MSE 64
Tabulka 14: Optimalizované modely s nejvyšší úspěšností a minimální MSE 64
Tabulka 15: Zisk po běhu simulace pro model s minimální kvadratickou chybou MSE 68
Tabulka 16: Zisk po běhu simulace pro modely optimalizované dle nejvyšší úspěšnosti s minimální MSE 69

Tabulka 17: Shrnutí výsledku pro modely optimalizované pomocí minimální kvadratické chyby MSE .. 75
Tabulka 18: Shrnutí výsledků pro modely optimalizované podle nejvyšší úspěšnosti s nejmenší kvadratickou chybou 76

SEZNAM GRAFŮ

Graf 1: Close hodnoty – NETFLIX 38
Graf 2: Close hodnoty – APPLE 39
Graf 3: Close hodnoty – GOOGLE 40
Graf 4 Close hodnoty – AMAZON 41
Graf 5: Close hodnoty – YAHOO 42
Graf 6: Kvadratická chyba MSE 48
Graf 7: Histogram chyb 49
Graf 8: Error autocorrelation 50
Graf 9: Time-Series Response 50
Graf 10: Nejmenší možná kvadratická chyba MSE pro NETFLIX 58
Graf 11: Nejvyšší úspěšnost s minimální kvadratickou chybou MSE pro NETFLIX 59
Graf 12: Nejmenší možná kvadratická chyba MSE pro APPLE 60
Graf 13: Nejvyšší úspěšnost s minimální kvadratickou chybou MSE pro APPLE 61
Graf 14: Nejmenší možná kvadratická chyba MSE pro YAHOO 62
Graf 15: Nejvyšší úspěšnost s minimální kvadratickou chybou MSE pro YAHOO 63
Graf 16: Predikce 20 dnů pro NETFLIX podle minimální MSE 70
Graf 17: Predikce pro 20 dnů pro NETFLIX podle maximální úspěšnosti a minimální MSE 71
Graf 18: Predikce 20 dnů pro APPLE podle minimální MSE 71
Graf 19: Predikce pro 20 dnů pro APPLE podle maximální úspěšnosti a minimální MSE 72
Graf 20: Predikce 20 dnů pro YAHOO podle minimální MSE 73
Graf 21: Predikce pro 20 dnů pro YAHOO podle maximální úspěšnosti a minimální MSE 73

SEZNAM OBRÁZKŮ

Obrázek 1: Členění finančních trhů 14
Obrázek 2: OHLC graf 22
Obrázek 3: Svícový graf 23
Obrázek 4: Elliottovy vlny 24
Obrázek 5: Dělení složek časové řady 25
Obrázek 6: Proces reprodukce 28
Obrázek 7: Biologický neuron 29

Obrázek 8: Formální neuron 30
Obrázek 9: Rekurentní síť 33
Obrázek 10: Kohenova síť 35
Obrázek 11: Neuronová síť NAR vytvořena v MATLABu 44
Obrázek 12: Neuronová síť NARX vytvořena v MATLABu 45
Obrázek 13: Rozdělení dat 47
Obrázek 14: Kód. který počítá úspěšnost sítě 52
Obrázek 15: Ukázka kódu nákupu akcií 65
Obrázek 16: Ukázka kódu prodeje akcií 66
Obrázek 17: Ukázka kódu prodeje po nákupu 67
Obrázek 18: Ukázka kódu nákupu po prodeji 68
Obrázek 19: Výpis finálního kódu pro automaticky zoptimalizovaný model 77

A Obsah CD

- Zdrojové soubory
- Tabulky s výsledky