VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ALGORITMICKÉ OBCHODOVÁNÍ NA BURZE S VYUŽITÍM STROJOVÉHO UČENÍ

ALGORITHMIC TRADING USING MACHINE LEARNING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **KAREL ČERVÍČEK**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. IGOR SZŐKE, Ph.D.**
SUPERVISOR

BRNO 2018

Abstrakt

Obchodování na burze ve spojení s automatizací je široce probírané téma. V této práci je snahou využití optimalizačních metod a prostředků strojového učení pro efektivní a obecné zpracování finančních časových řad. Je navržen a otestován systém, který zpracuje signál a generuje optimální strategii.

Abstract

Automatization is higly used in stock traiding. The thesis try to exploid optimalization principles and machine learning. Developed and tested stock traiding system proces financial time series and generate optimal strategy

Klíčová slova

Strojové učení, genetické algoritmy, Forex, obchodní systém, obchodování

Keywords

Machine learning, genetic algorithm, Forextraiding, traiding system

Citace

ČERVÍČEK, Karel. *Algoritmické obchodování na burze s využitím strojového učení.* Brno, 2018. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Igor Szőke, Ph.D.

Algoritmické obchodování na burze s využitím strojového učení

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Igora Szőkeho, Ph.D. Další informace mi poskytl Prof. Ing. Lukáš Sekanina, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Karel Červíček
23. května 2018

Poděkování

Chtěl bych poděkovat vedoucímu za podporu s teorií neuronových sítí a celkovým vedením, jak při práci postupovat. Také děkuji profesoru Sekaninovi za konzultace ohledně genetických algoritmů.

Obsah

Kapitola 1

Úvod

V úvodu je definován účel, cíl a popis, ale také zdůvodnění proč práce vznikla. Vysvětluje hlubší souvislosti návrhu postupu, návrhu architektury a výběru zkoumané oblasti, kterou je obchodování na Forexu. Otevírá témata, která přímo ovlivňují celkový návrh. Pokud čtenář není obeznámen s obchodováním na burze, pak by nemusel zcela pochopit některé kroky v návrhu či jistá omezení.

1.1 Obchodování

Díky moderním informačním technologiím se principy obchodování na burze mění již několik let. Z fyzicky řešených objednávek se přešlo na elektronické příkazy. Dříve měl broker na burze obchodníka, který objednávku vyřizoval. Dnes jsou veškeré objednávky vyřizovány elektronicky. Broker přijímá objednávky svých klientů prostřednictvím platformy. Obchodování na trzích je dnes prováděno jak, lidmi tak automaticky. Od devadesátých let vzniklo několik firem, které zavádí vysokofrekvenční obchodování [3]. Také je dnes velká část obchodníků, kteří trh analyzují fundamentálně nebo technicky za pomoci podpůrných programů a objednávky zadávají samostatně.

Zadání příkazu obchodníkem nebo obchodním programem je ovlivněno vývojem ceny. Cena je tvořena nabídkou a poptávkou. Jednoduše, pokud je poptávka po dané komoditě či jiné hodnotné entitě vyšší než nabídka, pak její cena roste. Stejně tak, pokud je nabídka vyšší než poptávka, pak cena dané komodity klesá. Za tímto jednoduchým principem stojí komplexní systém ekonomiky celého světa. Trh je ovlivňován mnoha faktory, jako jsou počasí, války, zprávy ze světa, zprávy z ekonomiky, ale také objemné objednávky bank, obchodní algoritmy a mnoho dalších.

Vysokofrekvenční obchodování je založeno na algoritmech, které jsou schopny v rámci sekund provést obchod například na základě různých cen z několika burz nebo na základě stochastického modelování. Pro odvětví vysokofrekvenční obchodování je nutné mít co nejrychlejší přístup k aktuálním informacím a také co nejkratší dobu přístupu přímo na servery burzy. Plnit tyto podmínky není snadné a je zapotřebí relativně velký kapitál. Tato práce se zaměřuje na intradenní obchodování prostřednictvím brokera 2. Výhodou je možnost obchodování s malým kapitálem a nízkonákladový přístup na burzu. I s tímto přístupem je možné zavést automatické obchodování nebo zavést automatického asistenta, který dohlíží na trh a upozorňuje na obchodní příležitosti v rámci minut, hodin a dnů. Většina brokerů poskytuje programové rozhraní pro nasazení obchodního systému.

1.2 Hlavní cíl

Text se zabývá vytvořením systému, který bude zpracovávat a analyzovat časové řady za účelem automatického sestavení řešení, jež pro časovou řadu hledáme. Řešený problém bude ohodnocen funkcí, kterou zadavatel určí. Sestavení řešení bude optimalizační problém minimalizující nebo maximalizující funkci pro ohodnocení. I přesto, že primárně se práce zabývá obchodováním na Forexu, je snahou systém vytvořit obecný a případně použitelný i pro jiné druhy signálu.

Vzhledem k tomu, jak se obchodování na burzách vyvíjelo 1.1, je dnes pro obchodníka takřka nemožné zpracovat všechna data, která trh potenciálně ovlivňují. Obchodník se snaží na základě aktuálních informací a historického vývoje trhu určit, zda uvažovanou entitu koupí nebo zda je čas na prodej. Obchodník využívá fundamentální 2.0.2 a technickou analýzu 2.0.1. Motivací automatizace je tato data zpracovávat strojově. Fundamentální data jsou většinou v textovém formátu, který je také možné zpracovávat strojově [28]. Získávání dat z textu a zpracování přirozeného jazyka je samostatná obsáhlá disciplína informatiky. Vedle zpracování fundamentálních dat je možné zakládat obchodní strategie na analýze historických dat. Z historického vývoje ceny je možné statisticky ohodnotit, zda obchod uskutečnit 2.0.2. Pro ziskový obchod je nutné správně vyhodnotit mnoho dat, která mohou i nemusí být relevantní vzhledem k obchodnímu záměru. To otevírá otázku, jaké přístupy zvolit. Historická data jsou popsána matematickými modely z oblasti finančního kalkulu. Na jejich základě je možné pochopit charakter burzovních dat a vyvodit přístupy pro jejich zpracování. Tato tématika je také popsaná v mnoha knihách zaměřených na technickou analýzu [12]. Většina obchodních systémů bývá parametrizována, a tak je také možné brát v úvahu optimalizační techniky. Obchodí systémy tvoří množinu ziskových a ztrátových obchodů, které je možné klasifikovat pomocí prostředků strojového učení [1].

Existují práce, které se touto problematikou zabývají. Jak bylo zmíněno, například práce [26] je kolekcí metod pro zpracování fundamentálních dat za účelem predikce kurzu. Forexový kalendář je jedním z kanálů, kde jsou aktuální fundamentální data. Práce [25] se zabývá zpracováním právě takovýchto dat. Některé práce se také zabývají optimalizací technické analýzy pomocí genetických algoritmů. Například je zde [22] optimalizován obchodní systém založený na pěti parametrech, které určují vstup do pozice, a pěti parametrech pro jejich ukončení. Podstatným zdrojem informací je kniha [31] řešící predikci časových řad na zakladě gramatik. V této knize jsou popsány možnosti generování jazyka pro zpracování časové řady. Kniha využívá i možnosti strojového učení, konkrétně support vector machines. Na podobném principu je založena práce [2], která využívá gramatickou evoluci popsanou i v této práci. Zmíněné strojové učení je také často skloňovaným nástrojem a je několik článků a knih, které toto téma popisují. Zajímavé je například využití dopředné neuronové sítě, které je popsáno autorem Robertem J.Eydenem [10]. Posuzuje vliv objemu dat na výsledky predikce pomocí neuronové sítě. Vstupem jsou různorodá data, jako například indexy, ekonomické statistiky, zprávy či technické indikátory. V závěru jsou posouzeny výsledky založené na různých datových zdrojích v různém objemu. Posouzení ukázalo, že 20 a 63 zdrojů nemělo úměrně rozdílné výsledky. Robustnost a objem dat neměly přímý vliv na lepší predikci. Tento výsledek opět pokládá otázku, jak určit, která data jsou relevantní pro správnou generalizaci problému. Dopředná neuronová síť se jeví jako vhodný nástroj pro zpracování finančních dat. Například konkrétně článek [36] používá tento návrh pro predikci. Jsou však i další architektury umělých neuronových sítí. Architektura LSTM (Long short-term memory) je v této práci také zmíněna. Podle článku [27] je tato síť schopna velmi dobře zpracovávat sekvenční data. Její vyžití je vhodné například pro časové řady.

Genetické algoritmy mohou být využity i pro nastavení parametrů neuronové sítě, optimalizaci architektury nebo optimalizaci zpracování dat. V tomto článku [8] využili autoři právě genetické algoritmy pro optimální nastavení parametrů, dále analýzu hlavních komponent za účelem snížení dimenzionality, kterou vyřešili odstranění nerelevantních dat.

Zmíněné informace spolu s knihou [4] stanovily směr této práce. Cílem je vhodně navrhnout architekturu obchodního systému, která bude efektivně určovat relevanci dat pro technickou analýzu. Určovat ziskové strategie a rozhodování pozic bude potenciálně podpořeno neuronovou sítí. Vzhledem k možným variacím, jak historická data vyhodnocovat a stavět možné obchodní strategie, je nutné efektivně generovat množinu výpočtů, obecně optimalizovat potenciálně výdělečné strategie a určit, kdy a jak bude využito strojové učení. Vzhledem k mnoha faktorům a možným přístupům, jak tento problém řešit, by měla architektura umožňovat eventuální rozšíření. Výsledný systém bude testován na měnovém páru a bude zaměřen na získání obchodních strategií. Avšak plánovaná obecnost by měla systém postavit do pozice, ve které bude stačit jeho minimální rozšíření pro zpracování i jiných dat.

1.3 Struktura práce

Práce testuje a předpokládá pouze forexová data. Výběr studovaných materiálů odpovídá cíli primárně vytvořit systém, který bude přínosem pro obchodování na Forexu. V první kapitole 2 uvádím, jak celý proces obchodování funguje. Popisuji rizika, která obchodování přináší spolu s nástrahami, které přináší obchodování přes brokera. Další kapitoly jsou teoretickým podkladem k samotnému systému. Kapitola 3.1 uvádí princip primárně dopředné, ale i Long short-term memory neuronové sítě. Po přečtení by mělo být jasné zasazení neuronové sítě do systému. Stejně tak kapitola popisuje, jak genetické algoritmy fungují. Každá teoretická část je zakončena shrnutím vzhledem k cíli práce. Kapitola 3.3 pouze okrajově popisuje některé modely pro stochastický popis finančních dat.

Vzhledem k tomu, že tato práce by měla mít především praktický dopad, je největší pozornost věnována kapitole popisující vytvořenou architekturu 4 s testy z různých pohledů. Čtenář, který je obeznámen s genetickými algoritmy, gramatikami a umělými neuronovými sítěmi, může cílit přímo na tuto kapitolu.

Kapitola 2

Forex

Volba finančních trhů byla provedena na základě charakteristiky dat 3 a také obecných vlastnostech. Forex je zkratkou pro Foreign Exchange. Jde o trh pro směnu měnových párů. Pro nás je i vzhledem k této práci zajímavé, že jde o nejlikvidnější a největší trh na světě, na kterém participují a ve kterém směňují aktiva a deriváty banky, fondy a jiné finanční instituce. Příkladem samotným je postavení našeho systému v roli investora, kdy jeho úkolem bude spekulace o budoucím pohybu různých měn za účelem zisku 1.2. Trh je tedy ovlivňován ekonomikami celého světa a odráží jejich charakter. Obchodníci, brokeři jako tvůrci trhu zveřejňují svoji nabídku a poptávku, která musí být na obou stranách potvrzena, neboť trh nemá centrální burzovní knihu.

Je zajímavé, co stojí za vznikem Forexu. Vývoj vzniku přesně vysvětluje, na čem je cena dané měny založena. Dříve byla měna daného státu podložena zlatem. Kurz mezi měnami byl pak určen poměrem zlata, které daný stát měl. Tomuto finančními principu se říká zlatý standard. S příchodem první světové války byl tento standard zrušen. V období války byly vytvářeny tlaky na ekonomiku. A zlatý standard ekonomiku brzdil a státy byly nuceny peníze tisknout bez jejich podkladu. Po válce došlo k usnesení, že hlavní podkladovou měnou se stane americký dolar, který byl vyrovnán zlatem. Postupem času byly nároky na rezervy zlata neúnosné. Byl zaveden systém plovoucí měnové kurzy, který je do dnes aktuální. Měna je tak podložena důvěrou držitelů a její cena je určena nabídkou a poptávkou [19].

Forex je dostupný i pro obchodníky s malým kapitálem a bez přímého přístupu na burzu. Forex broker ve smyslu finančních trhů je společnost, která poskytuje přístup na Forex. Zajišťuje software a server pro správu obchodů svých klientů a také finanční páku. Finanční páka dovoluje obchodovat s malým kapitálem a dosahovat velkých zisků resp. ztrát. Broker podkládá kapitál obchodníka v násobku finanční páky. Pokud obchodník využívá páku 1:100, pak je jeho kapitál stonásobně větší pro otevíraní obchodních pozic. Pro otevření pozice je nutné disponovat alespoň takovým kapitálem, který stačí pro složení jistiny. Jistina slouží jako krytí ze strany brokera z hlediska úvěrového rizika. Pokud obchodník neodhadne situaci a ponechá běžet pozice ve ztrátě tak velké, že se vyčerpá jeho volná jistina, pak bude vyzván k doplnění prostředků na účet, jinak jsou jeho pozice automaticky uzavřeny. Tento krok je ochranou proti zadlužení.

Dalšími skutečnostmi, které ovlivňují výsledný profit obchodování, je *spread* neboli rozdíl mezi nákupní a prodejní cenou. Rozdíl je proměnlivý, ale průměr je nutné brát v potaz. Další důležitou hodnotou je swap. Jde o rozdíl úrokových sazeb mezi obchodovanými měnami.

Při zadávaní pozice je nutné zvážit několik možností, které otevření přinese. Pokud pozici otevřeme, pak spekulujeme o cenové hladině v budoucím čase a stanovíme hodnotu ceny,

na které pozici uzavřeme, takzvaně určíme *take profit*. Také stanovíme rizika, která jsme schopni podstoupit. Hladina ceny, která určuje zavření pozice v případě, že se hladina ceny pohybuje směrem proti otevřené pozici, se nazývá *stop loss*. Obchodník má generalizovaně dva přístupy, jak trh analyzovat 2.0.1, 2.0.2. Pokud je v práci zmíněna obchodní pozice, je tím myšlena jedna z dvou variant:

Long - obchodník spekuluje na růst ceny. Pokud po uzavření kontraktu dojde k růstu ceny, pak obchodník nabývá zisku. Pokud dojde k poklesu, pak nabývá ztrát.

Short - obchodník spekuluje na pokles ceny. Pokud po uzavření kontraktu dojde k poklesu ceny, pak obchodník nabývá zisku. Pokud dojde k vzestupu, pak nabývá ztrát.

Příklad otevření pozice je uveden v technické analýze 2.0.2. V práci také bude dále používáno několik pojmů:

Tick - přijmutí aktualizované ceny z Forexu. Časová řada ticků je nejmenší časové rozlišení, které lze získat.

Bod - pohyb na burze je udáván v bodech. Jde o nejmenší možné vyjádření vertikálního pohybu. Pokud bude v textu udáván jakýkoliv zisk, pak bude uveden v bodech. Bodový zisk je pak možné přepočítat na zisk podle objemu otevřené pozice [1].

Spread - náklad spojený s otevřením pozice. Rozdíl mezi nákupní a prodejní cenou udává velikost spreadu.

Lot - je standartní jednotkou, která udává objem obchodované pozice. Jeden *lot* představuje 100 000 jednotek základní měny.

Příklad otevření pozice je uveden v technické analýze 2.0.2

2.0.1 Fundamentální analýza

Obchodník hodnotí události a data, které trh přímo i nepřímo ovlivňují. Jde nejen o makroekonomické informace týkající se daného ekonomického sektoru, ale také politické situace. Sleduje ekonomickou situaci ostatních sektorů a hodnotí jejich případný vliv na samotný trh. V rámci Forexu je nutné sledovat ekonomiky států, kterých se obchodovaný pár týká.

Vhodným příkladem obecné fundamentální analýzy je sledování otevírání světových burz. V těchto časech je možné uvažovat o vstupu do pozice. Také existují forexové kalendáře, které seskupují ekonomická data relativně k měnám [18].

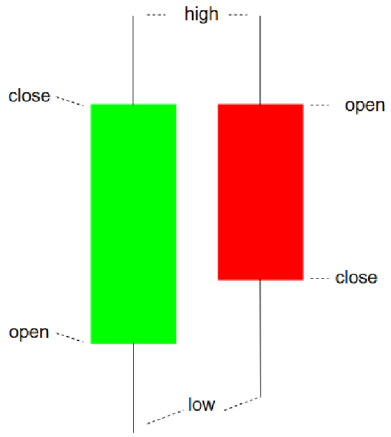
2.0.2 Technická analýza

Obchodník pracuje s daty přímo odvozenými z historické a aktuální ceny. Jde tedy o data tvořená přímo trhem. Je zveřejněno mnoho technických indikátorů, které transformují události na trhu a indikují možné situace. Popis a výčet technických indikátorů použitých v této práci je uveden ve výsledné architektuře 4.

Pouze na základě technické analýzy je možné vytvořit obchodní strategie. Pro představu, jak může vypadat a být využita, uvedu příklad. Samotné zobrazení dat je základem

[1] https://bossa.cz/cs/vzdelani/forex-uvod/bod-spread-lot

technické analýzy. Nejpoužívanější jsou dva typy grafů zobrazení uzavírací ceny a graf svíčkový. Historická kurzovní data můžeme zobrazit v různém časovém rozlišení. V případě grafu s uzavírací cenou je zobrazena pouze cena v daném časovém intervalu. Svíčkový graf zobrazuje i informaci o ceně v mezičase. Svíčku tvoří čtyři hodnoty: open, high, low, close. Na obrázku 2.1 jsou vyznačené dvě možné varianty. Svíčka se zeleným tělem primárně vyznačuje pohyb ceny v intervalu 15 minut. Zobrazená svíčka je tvořena v 15 minutách. Na začátku cena začne na hladině open. V průběhu 15 minut potom cena dosáhne hodnoty high a low. Na konci intervalu se svíčka uzavře na hladině close. Obdobně je tomu u svíčky s červeným tělem, která vyznačuje pokles.



Obrázek 2.1: svíčky

Svíčkové formace jsou jednoduchým zpracováním ceny. Většina indikátorů je počítána nad hodnotami open, high, low a close. Pro příklad obchodní strategie jsem vybral spojení klouzavého průměru a indikátoru RSI.

Klouzavý průměr je základním `indikátorem`[2], počítá průměr svíček přes interval. Většinou je počítán přes hodnotu close. Klouzavý průměr MA a exponenciální klouzavý průměr EMA jsou počítány následovně:

$$MA_i(n) = \frac{1}{n}\sum_{i=1}^{n} P_{n-i} \tag{2.1}$$

$$EMA_i(n) = (P_i - EMA_{i-1})\frac{2}{n+1} + EMA_{i-1} \tag{2.2}$$

Index relativní síly se neustále pohybuje v rozmezí 0 - 100. RSI uvádím zejména kvůli jeho širokému využití. Jeho stavy jsou reprezentací síly, indikuje případné překoupení trhu a možnost uzavření pozice. `Výpočet`[3] je opět uvedený níže.

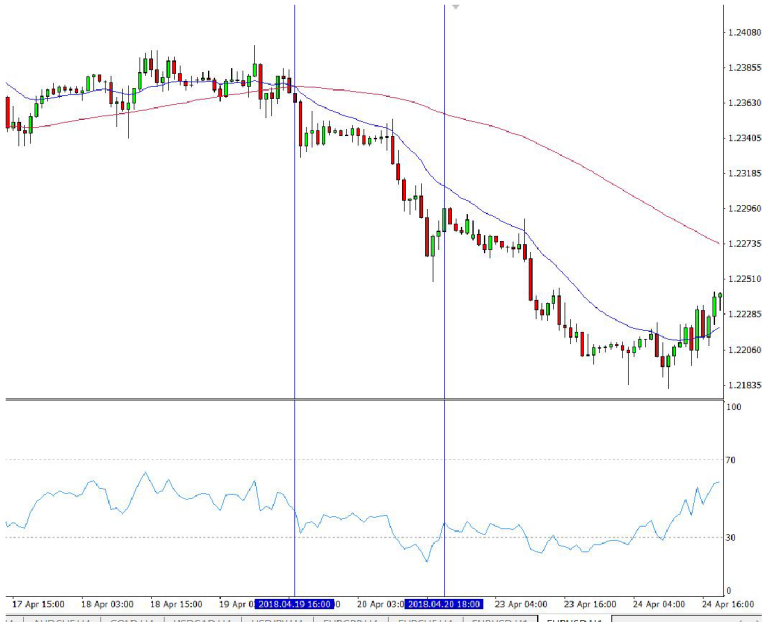
$$RSI_t(n) = 100 - \frac{100}{1 + \frac{U(n)}{D(n)}} \tag{2.3}$$

U(n) - součet kladných cenových změn za období délky n

D(n) - součet záporných cenových změn za období délky n

[2]Více informací je možné zjistit zde: $https://en.wikipedia.org/wiki/Moving\_average$
[3]Více informací je možné zjistit zde: $https://en.wikipedia.org/wiki/Relative_strength_index$

Strategie spojující tyto indikátory by mohla vypadat následovně: klouzavé průměry počítané přes rozdílný interval budou indikovat možné otevření pozice. Křivka modré barvy vyznačuje EMA počítané přes 18 uzavřených cen close. Červeně značená křivka je klasický MA počítaný přes 80 uzavřených cen close. V dolní části je zobrazený indikátor RSI s kritickými hladinami 70 % a 30 %. Překřížení EMA směrem dolů přes MA určí zadání pozice short. Situace překřížení je označena modrou vertikální čárou v čase 16:00. Při vstupu do pozice zajistíme obchod pomocí *stop lossu* a určíme, kdy pozici ukončit. Ukončení může být stanoveno pevně. Já jsem pro ukončení pozice zakomponoval indikátor RSI. Konkrétně pozice short bude ukončena, pokud RSI překoná dolních 30 % a opět se vrátí. Modrá vertikální čára v čase 18:00 ukazuje uzavření pozice.



Obrázek 2.2: Ukázka exponenciálního a normálního klouzavého průměru

2.0.3 Diskuze vzhledem k systému

Systém v pozici tradera bude podléhat výše uvedeným principům trhu, a tak je třeba do návrhu zahrnout řízení rizik spojené s finanční pákou a volnou jistinou. Využití vysoké finanční páky bude možné pouze za předpokladu otevření pozice s rychlou realizací a důkladnou analýzou o budoucím vývoji trhu nejen v čase předpokládaného ukončení. Uvedu příklad pro demonstraci situace, která by byla při vysoké finanční páce a velkém objemu pozice kritická.

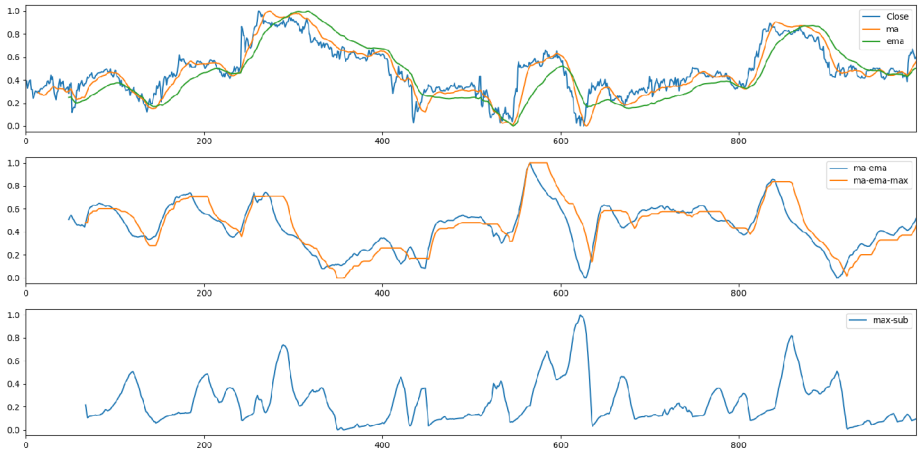
Na obrázku 2.3 jsou označeny oblasti, kde pomyslně mohlo dojít ke vstupu do pozice short, tedy spekulace na pokles ceny. Čas 8:15 označuje začátek červené oblasti, kdy předpokládáme, že v následujících přibližně třech hodinách dojde k poklesu. Pokud bychom otevřeli pozici o objemu, který spotřebuje veškerou jistinu, budeme v čase červené zóny nuceni jistinu doplnit. Pokud tento krok neuděláme, naše pozice bude uzavřena ve ztrátě i přes to, že v čase zóny označené zeleně by naše predikce uspěla.

Obrázek 2.3: Ukázka pohybu proti směru spekulace, který může zapříčinit ztrátu

Systém bude vhodné vytvořit primárně pro technickou analýzu. Je nespočet možností, jak indikátory využít. Navržených strategií je mnoho a všechny mají za jistých podmínek svůj potenciál. Od automatického obchodního systému požaduji, aby se pokusil vytvořit různou kombinaci a různé další přepočty nad množinou technických indikátorů. Podle charakteru dat a modelů, které se snaží finanční data popsat, je zřejmé, že různé kombinace a výpočty mohou fungovat pouze určitý časový úsek.

Pro představu, jak bude přepočet fungovat, uvedu krátký příklad, který jsem vytvořil v začátcích implementace systému jako základ překladače 4.1.2. Na vstupu bude číselná řada, konkrétně časová řada s hodnotami *close*. Nad touto řadou se pokusí systém provést náhodné výpočty za účelem nalezení výpočtu vhodného pro ziskovou strategii.

Na obrázku jsou zobrazeny signály, které byly odvozeny z hodnot *close*. Systém nejdříve provedl výpočet klouzavých průměrů *ma* a *ema*, které jsou zobrazeny v horní části spolu s hodnotami *close*. V dalším kroce provedl odečtení $ma - ema$ a následně nad hodnotami z předchozího kroku aplikoval okenní funkci pro nalezení maxima. Obě časové řady jsem opět zobrazil společně v druhém okně. Poslední operací bylo odečtení časových řad z druhého okna $(ma - ema - max) - (ma - ema)$. Takto nalezený výpočet může participovat na výsledné strategii.



Obrázek 2.4: Ukázka aplikace výpočtů nad časovou řadou

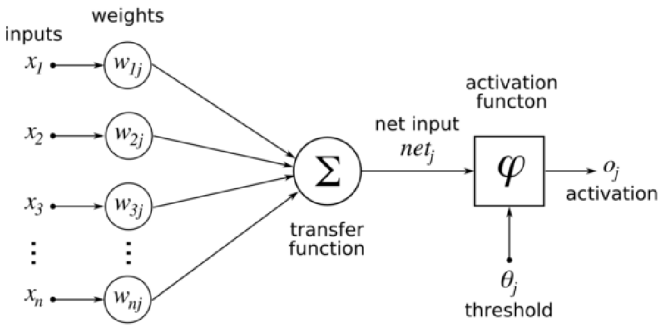
Kapitola 3

Prostředky, techniky a poznatky pro návrh systému

Vzhledem k tomu, že výsledný návrh se opírá o širokou škálu teorií a témat, jsou v této kapitole nastíněny informace, které jsem do hloubky prostudoval. Například hned první kapitola 3.1 uvádí základní principy neuronové sítě. Stejně tak kapitola 3.2 uvádí základní informace o tom, jak genetické algoritmy fungují. Mnoho zajímavých informací jsem se dozvěděl z knihy [33], kde je popsán matematický popis finančních dat. Kniha uvádí mnoho teorií, jak trh popsat. Kapitola 3.3 čerpá právě z této knihy. V závěru 6 zhodnocuji, jak tyto informace byly využity, a uvádím možnosti dalšího vývoje podle této kapitoly.

3.1 Neuronové sítě

Jde o matematický model odvozený z principu funkce neuronů v mozku. Model umělé neuronové sítě vznikl již v padesátých letech, kdy se výzkumníci snažili o simulaci biologického neuronu. Základem umělé neuronové sítě je umělý neuron:



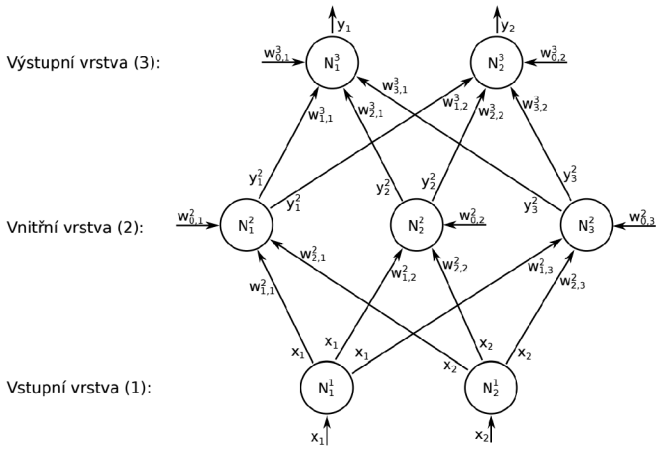
Obrázek 3.1: Umělý neuron, zdroj http://www.zive.cz

Na obrázku vektor X značí vstupní hodnoty, které jsou dále násobeny příslušnou váhou W_{nj}. Hodnoty jsou sečteny a výsledek je testován prahem neboli aktivační funkcí. Pokud signál projde, je poslán na výstup. Model neuronu je schopný klasifikovat lineárně separovatelné problémy. Umělá neuronová síť je pak tvořena spojením umělých neuronů do N vrstev. Neurony jsou spojeny do topologie navržené architektem.

3.1.1 Dopředná neuronová síť

Správné navržení topologie není triviální úkol. Dopředná neuronová síť má každý neuron výstupem propojený se všemi neurony vrstvy vyšší. Pouze výstupní nejvyšší vrstva vrací přímo hodnoty jako výstup.

Dopředná neuronová síť je vhodná pro řešení nelineárních problémů, pro aproximaci složitých funkcí, kde funguje jako obecný aproximační model. Podle práce [20] není pravidlem, že více vrstev zaručí lepší výsledky. Je možné použít architekturu jedné vstupní, vnořené a výstupní vrstvy [20], pro představu je možné se podívat na ukázku:

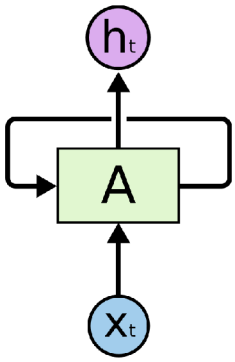


Obrázek 3.2: Příklad topologie

Správný počet neuronů je klíčovým faktorem pro dosažení dobrých výsledků. Při velkém počtu neuronů má neuronová sít problémy s přetrénováním. Při malém počtu není schopna problém generalizovat. Nelze definovat obecný postup, jak počet neuronů volit. Často používaným postupem je určení počtu neuronů aplikací nějakého optimalizačního algoritmu. Přetrénování neuronové sítě je časté, jedná se o problém, kdy neuronová síť vykazuje velmi dobré výsledky na trénovacích datech, ale selže při reálném nasazení. Tedy přetrénování neuronové sítě má za následek selhání na testovacích datech i přes to, že trénování vykazovalo dobrou generalizaci. Dopředná neuronová síť není jediným používaným typem.

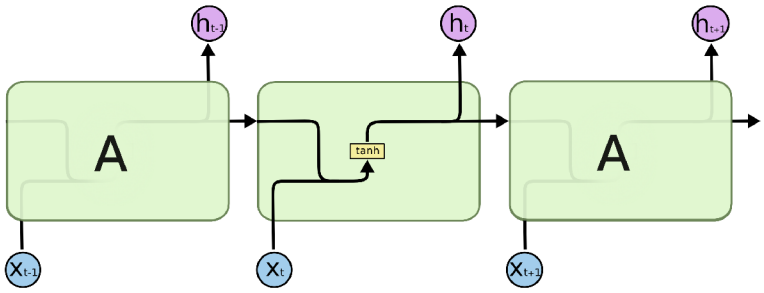
3.1.2 LSTM neuronová síť

LSTM znamená Long Short Term Memory. LSTM neuronová síť je speciálním případem sítě rekurentní. Rekurentní sítě jsem studoval ze zdroje [27], kde je i kvalitní grafické zobrazení. Proto jsem použil jejich vyobrazení. Na obrázku 3.3 je zobrazen jeden článek rekurentní sítě. X_t značí vstup a H_t výstup. Rozdíl oproti dopředné neuronové síti spočívá v tom, že článek má navíc smyčku, která vrátí výslednou hodnotu v druhém kroku zpět. Jde tedy o jednokrokovou paměť.



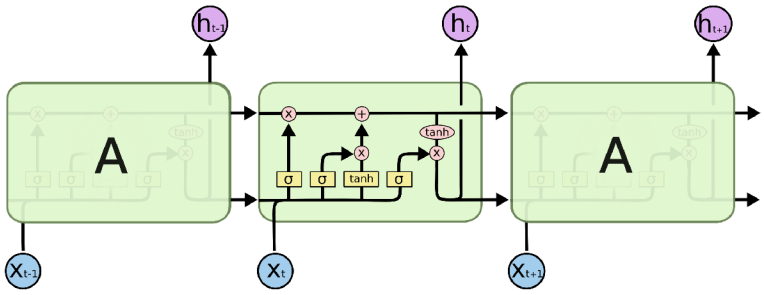
Obrázek 3.3: Rekurentní článek

Obrázek 3.4 ukazuje skrytou vrstvu běžné rekurentní sítě. Zde je vidět, že do neuronu je přivedena jak vstupní hodnota, tak hodnota neuronu z pozice $t-1$. Obě hodnoty jsou vstupem do aktivační funkce a případně předány na výstup a na pozici sousedního neuronu v čase $t+1$.

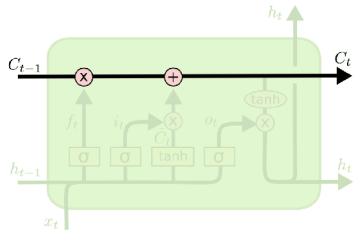


Obrázek 3.4: Rekurentní neuronová síť

Z obrázku 3.5 je patrné, že princip LSTM sítě je složitější. Skrytá rekurentní vrstva obsahuje vstupní hradla, která určují, jaké hodnoty budou předány dále. Hradla jsou vázána na stavový vektor c, který je zobrazen samostatně 3.6. Slouží pro aktualizaci stavových proměnných.



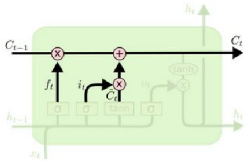
Obrázek 3.5: LSTM síť



Obrázek 3.6: Stavový vektor c

Následující obrázek 3.7 a výpočet ukazují průchod statového vektoru, který vyhodnotí hodnoty hradel a určí stav:

$$c_t = c_{t-1} f_t + g_t i_t$$



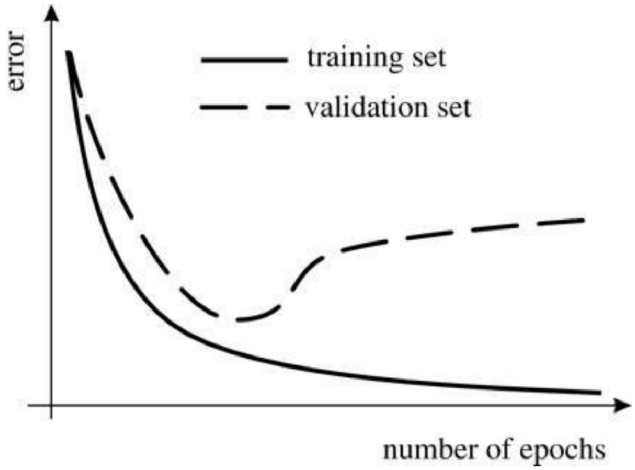
Obrázek 3.7: Stavový vektor c

3.1.3 Trénování neuronové sítě

Pod pojmem generalizace je možné si představit odchylku a varianci dat. Na začátku trénování je velká odchylka s minimální variancí. Během trénování se odchylka zmenšuje a variance by neměla růst. Ideálním koncem je nejmenší součet odchylky a variance podle [35].

Trénování je rozděleno na dopředné šíření signálu a následnou zpětnou propagaci chyby a adaptaci vah 3.1.4. Tento úkon je prováděn v cyklech, kdy se data na výstupu ohodnotí a určí se chyba. Trénování je děleno na učení s učitelem a bez učitele. Bez učitele je využíváno, pokud předem nevíme, co chceme klasifikovat, nebo nemáme data pro porovnání na výstup. Příkladem může být clustering. V naší práci bude využíván přístup učení s učitelem, kdy předem připravíme referenční data pro výstup.

Datovou sadu je vhodné rozdělit na část pro trénování a část validační. Validační sada slouží k dynamickému ukončení trénování. Díky validační sadě, kterou neuronová síť nevyužívá pro samotné trénování, je možné předejít přetrénování. Obrázek 3.8 ukazuje snižování chyby v průběhu trénování pro sadu trénovací a validační. Pokud se chyba validační sady nesnižuje, tak je vhodné proces trénování ukončit.



Obrázek 3.8: Validace, převzato z [23]

Trénování je založeno na funkci $E\left(D^i, P(Z^i, W)\right)$. Funkce představuje porovnání dat na výstupu neuronové sítě a dat referenčních, vyjadřuje závislost D^i a $P(Z^i, W)$. Funkce obecně představuje metodu pro ohodnocení vhodnou pro konkrétní problém. Parametry funkce $P(Z^i, W)$ jsou vektor vstupních i-tých dat Z^i a vektor vah W. Váhy jsou upravovány během zpětné propagace v závislosti na funkci $E\left(D^i, P(Z^i, W)\right)$.

Cílem trénování je minimalizace funkce $Etrain$. Průměrem funkce E v rámci dat pro trénování ($Etrain$) dostaneme hodnotu určující stav trénování.

3.1.4 Zpětná propagace chyby

Tento algoritmus bude vysvětlen pomocí jednoduchého vícevrstvého modelu založeného na učení z gradientu.

Vrstvy jsou definovány funkcí $F_n\left(W_n, X_{n-1}\right)$. X_n je výstupem vrstvy a X_{n-1} je vstupem. W je vektorem proměnlivých vah. Z^i 3.1.3 je vstupním vzorkem dat a je ekvivalentem X_0. Tím je dáno, že je známa parciální derivace funkce E^p podle proměnné X_n. A pokud lze spočítat parciální derivaci funkce E^p podle W_n za pomoci diferenčních rovnic, pak je možné spočítat X_{n-1}.

$$\frac{\partial E^p}{\partial W_n} = \frac{\partial F_n}{\partial W}\left(W_n, X_{n-1}\right)\frac{\partial E^p}{\partial X_n} \tag{3.1}$$

$$\frac{\partial E^p}{\partial X_{n-1}} = \frac{\partial F_n}{\partial X}\left(W_n, X_{n-1}\right)\frac{\partial E^p}{\partial X_n} \tag{3.2}$$

Kde $\frac{\partial F_n}{\partial W}$ je Jakobian W podle proměnné W v bodě daném $((W_n, X_{n-1}))$ a $\frac{\partial F_n}{\partial X}$ je Jakobian podle proměnné X. Jakobian vektorové funkce obsahuje parciální derivace všech výstupů s ohledem ke všem vstupům. Reverzním výpočtem pro každou vrstvu spočítáme všechny parciální derivace vzhledem ke všem parametrům. Tento postup je uveden podle knihy [35].

3.1.5 Stochastické a dávkové trénování

Podle [35] je možné použít dva způsoby trénování. Prvním je stochastické trénování. V této práci je použito trénování po dávce. K ustanovení gradientu dochází až po zpracování celé dávky dat. U stochastického trénování je vybrán jeden prvek.

3.1.6 Nastavení parametrů

Rychlost minimalizace chyby je závislá na učícím faktoru, který určuje velikost kroku ve směru gradientu. Příliš velký učící faktor může způsobit neoptimální vyhodnocení minima. Váhy neuronové sítě tvoří chybový prostor, hledáme minimum. V případě algoritmu gradient descent se může stát, že budeme v blízkosti minima. Pokud bude krok příliš velký, tak nebudeme schopni detekovat minimum.

Dále jsem se také soustředil na parametry dávky určující ustálení gradientu. Pokud dávka bude příliš malá, bude průběh trénování velmi chaotický a nestálý, také výpočetní náročnost bude nepřiměřeně velká. Velikost dávky ovlivňuje výpočetní náročnost a průběh trénování.

3.1.7 Aktivační funkce

Aktivační funkce dávají neuronové síti možnost nelineárně klasifikovat. Zde uvedu dvě základní: *sigmoida*

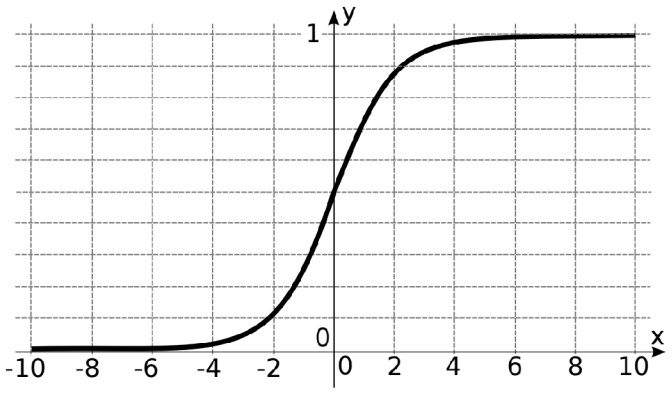
$$X_i = \frac{1}{1 - e^{-x}} \tag{3.3}$$

a *tanh* :

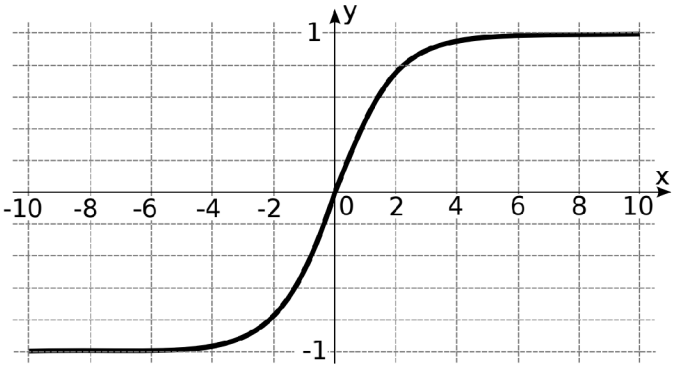
$$X_i = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{3.4}$$

Vzhledem k normalizaci je vhodné používat *tanh*. Na obrázku 3.10 je vidět rozsah oboru hodnot mezi -1,1. Sigmoida má obor hodnot pouze v kladných hodnotách, tedy pokud jsou zkoumaná data vhodná pro normalizaci do kladných hodnot, pak je možné ji využít. Charakter dat odvijí, jakou aktivační funkci použít. Funkce jsou vhodné, neboť jejich derivace je snadná, a tak jsou použitelné v případě výpočtu zpětné propagace.

Funkce se často drží v plochých zónách a učení je zpomaleno. V úvahu je možné vzít upravenou funkci: $f(x) = 1.7159tanh(\frac{2}{3}x)$ [35]. Hlavní výhodou je, že hodnota funkce v 1 je 1.



Obrázek 3.9: Funkce sigmoida [20]



Obrázek 3.10: Funkce tanh [20]

3.1.8 Redukce dimenze

Využívanou transformací je metoda PCA (Principal Component Analysis). Metoda určí lineární transformaci ortogonálních bází. Její aplikací jsou data dekorelována a je snížena jejich dimenzionalita. V transformovaných datech je zachována největší variabilita.

Vlastní vektory kovarianční matice určují bázové vektory PCA. Matice je počítaná takto [24].

$$C = \frac{1}{N}\sum_{i=1}^{N}(x_i - m)(x_i - m)^T \tag{3.5}$$

Kde C značí kovarianční matici. Matice je počítána na základě N vstupních trénovacích vektorů. Dále m je odhadovaný střední vektor a x_i je i-tý trénovací vektor.

$$m = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{3.6}$$

Vlastní vektory, které odpovídají největším vlastním hodnotám, jsou použity jako báze pro účel redukce. Vlastní hodnota s odpovídajícím vlastním vektorem je určena množstvím variability získané projekcí vstupního vektoru do vektoru vlastního.

Pokud tuto transformaci využijeme, je nutné myslet na fakt, že není možné transformovat testovací data do nově vypočítaného prostoru, ale je nutné využít parametry pro transformaci dat trénovacích.

3.2 Optimalizace

Optimalizace je jedním z přístupů, které systém využívá. Za předpokladu, že systém zpracovává stochastické signály, které nelze popsat deterministicky, jsem uznal za vhodné při modelování využít principů optimalizace.

Optimalizace je proces nad systémem, který vede k co nejlepším výsledkům podle daného kritéria. Optimalizaci je možné uvažovat na několika úrovních. Pojetí této kapitoly je zaměřeno na optimalizaci z hlediska matematiky, kde jde o proces nad parametry funkce, tak aby funkce nabývala maximálních, případně minimálních hodnot. Tedy pro funkci $A \to \mathbb{R}$ v případě maximalizace hledáme $x_h \iff A$, kde $f(x_h) \geq f(x)$ a $x \iff A$.

Na základě studie několika vybraných algoritmů jsem se rozhodl použít genetické algoritmy. V úvahu jsem bral Simulované žíhání, které využívá heuristiku procesu žíhání, kdy

se látka zahřeje a atomy se vyvážou z krystalické mřížky. Při chlazení se atomy vážou v krystalickém uspořádání s co nejmenším výdejem energie. Algoritmus tak vnáší možnost vymanit se z lokálního minima. Symbolický parametr teploty je pravděpodobnost, že krok bude přidán mezi plánované kroky.

Genetické algoritmy jsem vybral zejména díky možnostem jejich nasazení v mém systému.

3.2.1 Genetické algoritmy

Na základě heuristiky odvozené z evolučních principů algoritmus postupně vylepšuje parametry systému. Napodobuje procesy, na kterých je evoluce založena. Algoritmus postupuje v generacích, které obsahují různá řešení. Každé řešení v generaci je ohodnoceno fitness funkcí, která reprezentuje přirozený výběr. Fitness funkce udává kvalitu řešení v generaci neboli kvalitu jedince. V další generaci jsou vybráni nejlepší jedinci a nad nimi je aplikováno křížení a mutace, takto vznikne generace nová.

Algoritmus je možné zapsat v krocích:

1. Inicializuj první populaci z náhodných jedinců.
2. V cyklu pomocí ohodnocovací funkce vyber několik nejlépe ohodnocených jedinců.
3. Z vybraných jedinců vytvoř novou populaci za pomoci operátorů pro křížení, mutaci a reprodukci. Křížení vymění náhodně části jedinců mezi sebou. Mutace náhodně změní část jedince. Reprodukce jedince beze změny zkopíruje.
4. Vypočti ohodnocení nových jedinců.
5. Konec cyklu, který kontroluje ukončovací podmínku.
6. Řešení reprezentuje jedinec z poslední populace s nejvyšším ohodnocením.

Výše popsaný postup algoritmu jsem vyvodil z knihy [11]. V literatuře [5] jsou nejčastěji jedinci uváděni jako binární kombinace o délce N a operace nad nimi také. Proto dále rozvedu myšlenku, jak budu s tímto faktem pracovat a zda zachovám vlastnosti algoritmu 3.2.3.

1. Jednobodová mutace - je zvolen náhodně index i s podmínkou $1 \leq i \leq N$, kde N je délka jedince a nejnižší index je 1. Na daném indexu i je změněna hodnota. Například opět pro index $i = 3$:

jedinec	010111
jedinec po mutaci	011111

2. Jednobodové křížení - je zvolen náhodně index i s podmínkou $1 \leq i \leq N$, kde N je délka jedince a nejnižší index je 1. Nový potomek je pak složen z části obou rodičů. Podle indexu i je vybráno z rodiče A prvních X genů s indexem $0 \leq x \leq i$ a z rodiče B je vybráno zbylých Y genů s indexem $i + 1 \leq x \leq N$. Možné křížení dvou jedinců by mohlo vypadat takto, pokud by $i = 3$:

rodič A	010111
rodič B	111000
potomek	010000

3. Přirozený výběr - selekce může být realizována mnoha způsoby za podmínek, že jedinci s lepším ohodnocením budou s větší pravděpodobností reprodukováni a páry křížení budou vybrány náhodně. Konkrétní algoritmus turnaje je přímo využíván v systému.

3.2.2 Gramatická evoluce

Vzhledem k systému zmíním z oblasti genetických algoritmů gramatickou evoluci. Pro vyřešení problému je stanovena bezkontextová gramatika. Gramatika $G = (N, \Sigma, P, S)$ je bezkontextovou gramatikou, pokud pravidla z množiny P mají tvar:

$$A \to \alpha; A \in N; \alpha \in (N \bigcup \Sigma)^*$$

Gramatická evoluce využívá překladač jazyka generovaného gramatikou. Musí pracovat s proměnlivou délkou jedinců. Pravidla gramatiky jsou zapsána ve tvaru:

$$< symbol >:=< option >$$

$< symbol >$ náleží do množiny neterminálů a $< option >$ je posloupnost jednotlivě oddělených terminálů a neterminálů. Gramatika tak slouží ke generování syntakticky správné posloupnosti. Genotypem je posloupnost, která určuje pořadí jednotlivých operací.

3.2.3 Aplikace genetických algoritmů

Pro účely systému je nutné principy genetických algoritmů vhodně uchopit. Vzhledem k celkové architektuře je nutné uvažovat, jak celkově budou pojaty pojmy jedinec, generace, křížení, mutace a jak bude navržena fitness funkce. Nad časovými řadami budou počítány uvedené funkce a technické indikátory. Půjde o posloupnost matematických operací s následným ohodnocením fitness funkcí, zda daná posloupnost vedla k očekávaným výsledkům. Jedinec bude reprezentován řetězcem a operace genetických algoritmů jako operace nad řetězcem.

3.3 Finanční kalkul

Kapitola komentuje některé finanční modely. Uvádí pravděpodobnostní rozložení, která se týkají finančních dat. Také celkově popisuje charakter dat a vysvětluje, v jakém formátu je vhodné uvádět změnu ceny. V knize [33] je tento přepočet zmíněn spíše ve smyslu reprezentace zhodnocení investice.

Účelem je uvést základní pohled na finanční matematiku a vyvodit poznatky pro cíl této práce 1.2. Pro hlubší pochopení je nutné nastudovat kontext dále uvedených informací [33].

3.3.1 Reprezentace cenové změny

V knize [33] popisuji jednoduchou reprezentaci změny ceny z času $t-1$ na t jako poměr dvou cen. Cenu v daném čase značí P_t, kde t je časový index.

$$1 + R_t = \frac{P_t}{P_{t-1}} \tag{3.7}$$

$$R_t = \frac{P_t - P_{t-1}}{P_t} \tag{3.8}$$

V rozšíření pro k změn z času $t-k$ na t je změna reprezentována takto:

$$1 + R_t[k] = \frac{P_t}{P_{t-k}} = \frac{P_t}{P_{t-1}} \times \frac{P_{t-1}}{P_{t-2}} \times ... \times \frac{P_{t-k+1}}{P_{t-k}} = (1 + R_t)(1 + R_{t-1})...(1 + R_{t-k+1}) \quad (3.9)$$

Celková změna pro periodu o délce k může být zapsána jako:

$$1 + R_t[k] = \frac{P_t}{P_{t-k}} \quad (3.10)$$

V knize [33] je pak dále vysvětlen vliv úroku na dané aktivum. Tento princip je založen na přirozeném růstu, a pokud chceme vyjádřit změnu ceny s ohledem na tento fakt, musíme poměr změny logaritmovat:

$$ln(1 + R_t) = ln(\frac{P_t}{P_t - P_{t-1}}) \quad (3.11)$$

Dělení je vzhledem k logatirmu možné převést na odčítání:

$$ln(1 + R_t) = P_t - P_{t-1} \quad (3.12)$$

3.3.2 Vlastnosti rozložení pravděpodobnosti finančních dat

Pro základní pochopení finančních časových řad a utvoření obecné představy o jejich vlastnostech uvedu pravděpodobnostní rozložení, které s nimi souvisí.

Budu mluvit o spojité, marginální a podmíněné pravděpodobnosti. Mějme n-dimensionální Euklidovský prostor R^n, kde bod $x \in R^n$ a $y \in R^m$. Uvažujme vektory $X = (X_1, ..., X_i)$ a $Y = (Y_1, ..., Y_i)$ a uvažujme pravděpodobnost $P(X \in A, Y \in B)$, že X je v podprostoru $A \in R^n$ a Y je v podprostoru $B \in R^n$.

Spojitá pravděpodobnost je popsána funkcí $F(x, y, \Theta)$. Funkce charakterizuje vlastnosti X a Y, kde $x \in R^n$ a $y \in R^m$:

$$F_{X,Y}(x, y, \Theta) = P(X \leq x, Y \leq y; \Theta) \quad (3.13)$$

Pokud jde o spojité hodnoty a funkce hustoty spojité pravděpodobnosti $f_{x,y}(x, y, \Theta)$, existuje pro X a Y:

$$f(x, y, \Theta) = \int_{-\inf}^{x} \int_{-\inf}^{y} f_{x,y}(w, z, \Theta) dz dw \quad (3.14)$$

Marginální pravděpodobnost je dána funkcí pro X:

$$F_X(x, \Theta) = F_{X,Y}(x, \inf, \ldots, \inf; \Theta) \quad (3.15)$$

Pak můžeme ziskat marginální pravděpodobnost pouze pro X. Stejně může být získána pro Y. Distribuční funkcí je tedy [33]:

$$F_X(x) = P(X \leq x, \Theta) \quad (3.16)$$

Podmíněné rozdělení pravděpodobnosti pro veličinu X podmíněnou y je pravděpodobnostní rozdělení X za podmínky, že náhodná veličina Y nabude určené hodnoty y. Podmíněné rozdělení je dáno podílem rozdělení spojitého a marginálního. Tedy pro X za podmínky $Y \leq y$ je dáno:

$$F_{X|Y\leq y}(x,\Theta) = \frac{P(X \leq x, Y \leq y; \Theta)}{P(Y \leq y)}$$

Podmíněná hustota pravděpodobnosti je potom:

$$F_{X|Y}(x,\Theta) = \frac{f_{x,y}(x,y;\Theta)}{f_y(y;\Theta)}$$

Stěžejní je vztah těchto tří pravděpodobnostních rozdělení:

$$f_{x|y}(x,y,\Theta) = f_{x|y}(x;\Theta) \times f_y(y;\Theta)$$

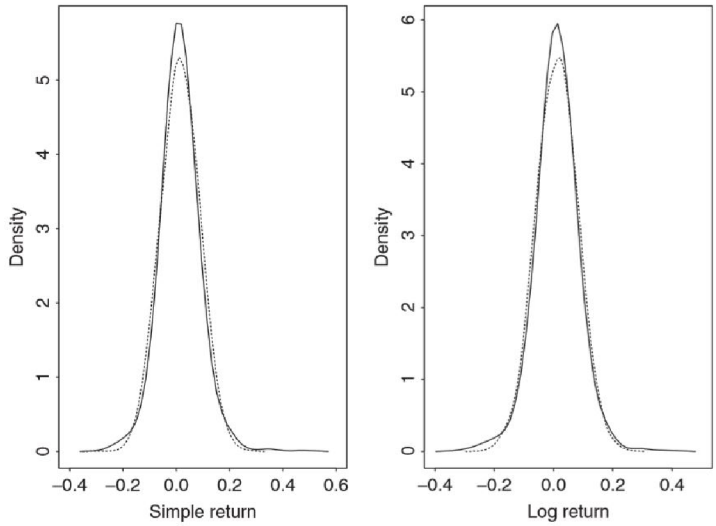
Náhodné vektory X a Y jsou nezávislé právě tehdy, když:

$$f_{x|y}(x,y,\Theta) = f_x(x,y,\Theta)$$

Dalším důležitým termínem je **moment náhodné proměnné** [33]. Prvními dvěma momenty jsou variance a směrodatná odchylka, které popisují normální rozložení. Jsou i další momenty pro popis dalších pravděpodobnostních rozložení. Třetí centrální moment určuje symetrii vzhledem k průměru a čtvrtý centrální moment udává míru, s jakou jsou data vychýlena.

Čtvrtý centrální moment by měl být pro přesné normální rozložení nulový. Pokud nabývá pozitivních hodnot, pak značí takzvaně *heavy tails* rozložení pravděpodobnosti. Z praktického hlediska to znamená, že datová množina obsahuje silné výkyvy. V knize [33] jsou také zveřejněny testy na základě těchto momentů, kde vychází, že finanční data mají silné heavy tails rozložení. Z textu je také zřejmé, že podmíněné rozložení pravděpodobnosti může být podmíněno časem $F(r_{it}|(r_{i,t-1}))$. Jedna z teorií založených na modelu náhodné procházky říká, že podmíněná pravděpodobnost časem $F(r_{it}|(r_{i,t-1},\ldots,(r_{i,1}))$ je rovna marginální pravděpodobnosti $F(r_{it})$. V tomto případě jsou data kontinuelně nezávislá a nepredikovatelná.

Závěrem je přiložen obrázek, který ukazuje výše popsané z praktického hlediska. V knize [33] autoři zpracovali cenová data akcií firmy IBM a zobrazili rozložení pravděpodobnosti normálního rozdílu a logaritmovaného rozdílu cen. Zobrazeny jsou měsíční rozdíly v letech od roku 1926 až 2008. Tečkami je zobrazeno odpovídající normální rozložení. Je tedy vidět, že reálné rozložení je mírně jiné. Také je vidět, že logaritmovaný rozdíl cen nemá takové výkyvy hodnot.



Obrázek 3.11: Ukázka rozložení pravděpodobnosti měsíčních rozdílů cen akcií firmy IBM od roku 1926 do roku 2008

Dále jsem se rozhodl zmínit některé informace ze zdroje [30]. Pokud bychom uvažovali pravděpodobnostní rozložení ceny samotné, bude spadat do rozložení exponenciálního. `Centrální limitní věta` [1] není u tohoto typu rozložení splněna. Normální rozložení splňuje podmínku centrální limitní věty.

Centrální limitní věta 1 *Nechť $\bar{X}$ je průměr náhodného výběru z rozdělení se střední hodnotou μ a rozptylem $\sigma > 0$. Pak*

$$W = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt[2]{n}}} \tag{3.17}$$

má rozdělení $N(0,1)$ pro $n \Rightarrow \infty$.

Aplikace: Pokud je n dost velké, pak:

$$W = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt[2]{n}}} = \frac{\frac{1}{n}\sum \bar{X} - \mu}{\sqrt[2]{n}} = \frac{\frac{n}{n}\sum \bar{X} - \mu * n}{n * \frac{\sigma}{\sqrt[2]{n}}} = \frac{\sum \bar{X}_i - \mu * n}{\sqrt[2]{n} * \sigma} \tag{3.18}$$

má přibližně standardní normální rozložení $N(0,1)$.

Centrální limitní věta říká, že pokud náhodná veličina V z dané množiny může být vyjádřena jako součet náhodných proměnných složených z variancí, tak distribuce V je přibližně normální [30].

3.3.3 Binomický strom

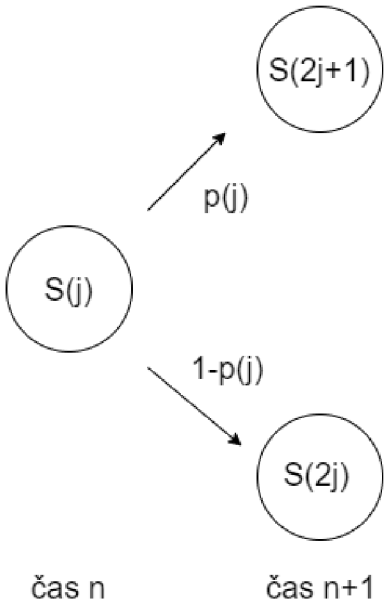
Dále uvedený model se používá pro ohodnocování opcí. Pro přesné pochopení, co jsou opce, je vhodné pročíst [34]. Pro rozhodnutí, proč prostudovat model Binomický strom, je právě

[1] Více informací je možné zjistit zde: *http : //vychodil.inf.upol.cz/kmi/pras/pr09.pdf*

jeho účel oceňování opcí. Opce dle [34] lze chápat jako druh spekulace. Jde o podmíněný termínovaný obchod, kdy v čase t_0 se jedna strana zaváže druhé, že v čase t_n, kde je $n > 0$, jedna strana od druhé koupí aktivum za cenu sjednanou v čase t_0, respektive že aktivum za určitou cenu prodá. Modely pro oceňování opcí jsou v této práci vhodné pro pochopení, jak finanční trhy fungují. Co je aktivum a co jsou opce pro tuto práci není důležité, ale souvislosti je opět možné pochopit zde [34].

Binomický strom reprezentuje vývoj ceny a předpokladem je, že cena může klesat nebo růst s určitou pravděpodobností. Vyhodnocení pro danou větev je možné chápat takto: S jako binomický proces s počáteční hodnotou s_1 v čase t_0, s_2 pro hodnotu menší a s_3 pro hodnotu větší vzhledem k s_1 v čase t_1. Pak očekáváme s pravděpodobností p, že v čase t_1 bude hodnota větší takto:

$$P(S_1) = (1-p)s_2 + ps_3$$



Obrázek 3.12: Jeden krok Binomického stromu

Od základů větvení je možné sestavit model Binomického stromu. Trh není tak jednoznačný, aby stačilo definovat výše zmíněné. Z informací o trhu [6] je možné vyvodit, že informace o ceně jsou přijímány po jistém časovém intervalu 2. Rozdíl mezi aktuální cenou a cenou předchozí je také určen časových rozdílem. Cenová aktualizace je jedním *tickem*. Model Binomického stromu pro n ticků reprezentuje pravděpodobnostní model, který popisuje 2^n cenových hladin daného trhu. Předpokladem konstrukce Binomického stromu jsou tyto podmínky:

1. Jak je uvedeno výše, předpokládáme, že se cena mění diskrétně.
2. V čase t_n je cena S_k. V čase t_{n+1} může cena nabývat hodnot uS_{k+1} nebo dS_{k+2}. Parametry u a d určují růst respektivě pokles.
3. Cena vzroste s pravděpodobností p a klesne s pravděpodobností $1-p$.
4. $d < 1 < u$

5. Absence arbitráže, $d < e^r < u$

Podle [34] zde uvedu poznatky, které plynou z vlastností Binomického stromu. Pravděpodobnost koncových stavů má charakter binomického rozložení pravděpodobnosti s hustotou:

$$p[X = k] = \binom{n}{k} p^k (1-p)^{n-k}$$

Pro koncové uzly pak platí:

$$c_t = e^{-(T-t)} \sum_{z}^{k} \binom{n}{k} p^k (1-p)^{n-k} (u^z d^l - z S_t - K)$$

$$z > \frac{\log \frac{K}{S_t d^k}}{\log \frac{u}{d}}, z$$

Pro práci jsem z uvedených informací vyvodil, že Binomický model podléhá nastavení parametrů. Dále je také popsána jejich kalibrace. Hodnotu parametrů je nutné vyvodit z aktuálních a historických dat. Z práce [34] bych chtěl uvést shrnutí postupu kalibrace parametrů, odvození parametru p vyjadřujícího pravděpodobnost nárůstu podkladového aktiva. Uvažuje nákup X kusů podkladového aktiva o ceně S_{T-t} a opce call za cenu c_{T-t}. Vše je financováno půjčkou ve výši L za bezrizikovou úrokovou míru. Autor pokládá:

$$L = X S_{T-t} - c_{T-t}$$

Autor získáním zajišťovacího poměru a předpokladem rizikově neutrálního prostředí vyjadřujě c_{T-t}. Díky stejnému tvaru, jako je vzorec pro jednokrokový Binomický model, získá pravděpodobnost p. Odvození pravděpodobnosti jsem uvedl pro ucelení, aby informace důležité pro práci nebyly vytrženy z kontextu jejich použití.

Následně odvozuje parametry u a d. Určuje podmínku hodnoty podkladového aktiva. Po jednom kroku Binomického modelu je nutné, aby cena odpovídala ve střední hodnotě zhodnocení aktiva v rizikově neutrálním prostředí. Také určuje podmínku pro rozptyl z knihy [17]. Tyto podmínky jsem vyhodnotil jako přínos:

$$pu + (1-p)d = e^t$$

$$pu + (1-p)d = e^{2r+\sigma^2 t}$$

Pro určení parametrů uvádí ještě třetí podmínku jako model Cox-Ross-Rubinstain. Tento model je publikován v článku [7]. S těmito podmínkami potom jednoduše odvodí u a d, a tedy i hledanou pravděpodobnost:

$$p_c \approx \frac{-(e^{-\sqrt[2]{t\sigma^2}}) + e^{rt}}{(e^{\sqrt[2]{t\sigma^2}}) - (e^{-\sqrt[2]{t\sigma^2}})}$$

S těmito podmínkami potom autor dokazuje konvergenci k Black-Scholes modelu podle knihy [15]. V tomto postupu je v praxi možné vidět aplikaci centrální limitní věty. V souladu s touto větou autor aproximuje binomické rozdělení normálním. Centrální limitní větu jsem se rozhodl v této práci uvažovat.

3.3.4 Lineární modely pro popis časových řad

Jako první bych chtěl uvést metodu, jakou je možné analyzovat závislost dat v časové řadě. Uvažujme slabě stacionární řadu. Pokud jsou v časové řadě závislé hodnoty r_t a r_{t-i}, pak mluvíme o autokorelaci. Pokud mluvíme o závislosti v jistém rozmezí, pak ji nazýváme i-lag autokorelation. Výpočet autokorelačního koeficientu je veden takto:

$$ac = \frac{Cov(r_t, r_{t-i})}{\sqrt[2]{Var(r_t)Var(r_{t-i})}}$$

Čím vyšší autokorelační koeficient je, tím vyšší je nalezení závislosti v datech.

Autoregresivní model slouží pro stochastické lineární modelování časové řady. Podmínkou pro aplikaci modelu je slabá stacionarita signálu. AR je počítán [33]:

$$x_t = \phi_0 + \phi_1 r_{t-1} + a_t \tag{3.19}$$

Kde a_t je pouze náhodný šum. Nad tímto polynomem je počítána autokorelace, která vyjadřuje závislost dat v čase. Tomuto jevu se říká kauzalita. Podle výše popsaných informací je nasazení tohoto modelu nemožné.

Složitějším rozšířením je model ARMA. Což je spojení autoregresivního modelu a klouzavého průměru. Vzorec nejjednoduššího modelu $ARMA(1,1)$:

$$r_t - \phi_1 r_{t-1} = \phi_0 + a_t - \theta_1 a_{t-1} \tag{3.20}$$

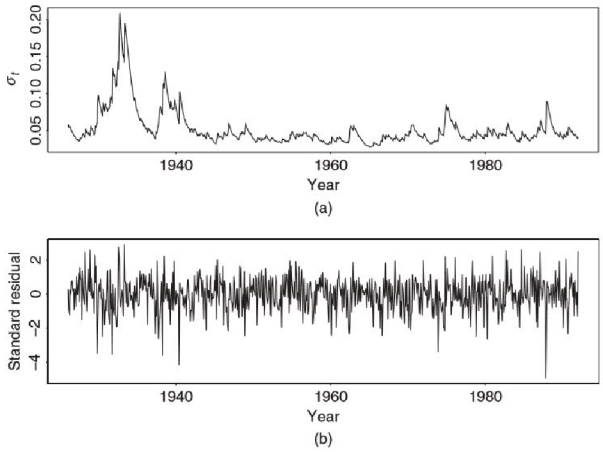
U systému musí platit podmínka stability lineárního systému, pokud chceme vyjadřovat náhodný proces jako kauzální, respektive pokud nad tímto modelem počítáme autokorelaci.

3.3.5 Autoregressive conditional heteroskedasticity

Posledním popsaným modelem bude GARCH neboli generalized autoregressive conditional heteroskedasticity. Jde o model pro stochastické modelování volatility časové řady. GARCH má dva parametry a je definován takto GARCH(m,s):

$$a_t\epsilon_t, \omega_t^2 = \alpha_0 + \sum_{i=1}^{m} + \alpha_i a_{t-i}^2 + \sum_{j=1}^{s} \beta_j \omega_{t-j}^2 \tag{3.21}$$

Kde ϵ_t je sekvence náhodných čísel s jednotkovou variancí a nulovým průměrem. $\alpha_0 > 0; \alpha_i \geq 0; \beta_j \geq 0$ a $\sum_{n=1}^{max(s,m)}(\alpha_i + \beta_i) < 1$. Z podmínek lze vyvodit, že variance a_t je konečná. V knize je psáno, že GARCH model dobře popisuje heavy-tailed rozložení pravděpodobnosti. Pro ukázku je v knize praktický příklad, kde na obrázku 3.13 je vidět využití GARCH modelu pro odhad směrodatné odchylky a standardizace cenové změny.



Obrázek 3.13: (a) ukazuje modelovanou směrodatnou odchylku finančního indexu S&P 500 pomocí GARCH modelu, (b) ukazuje standardizované měsíční změny indexu S&P 500. [33]

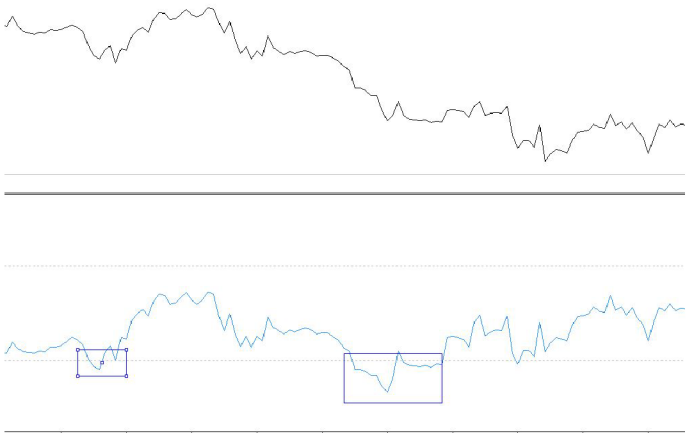
3.3.6 Diskuze vzhledem k obchodnímu systému

Pro návrh obchodního systému 4 jsou důležité podkapitoly 3.3.1 a 3.3.2. Informace o cenovém pohybu v čase je přínosem z důvodu nezávislosti od relativního měřítka vzhledem k danému aktivu. Také informace o charakteru dat je přínosem například pro případné strojové učení 3.1. Zbytek kapitol je hrubý popis pro ukázku, co se od stochastického modelování ve finančnictví čeká. Systém by měl být navrhnut tak, aby podobné modelování případně uměl také.

V kapitole 3.3.2 je uvedena zmínka o teorii náhodné procházky, která tvrdí, že cena není predikovatelná na základě historické ceny, tedy že náhodné vektory X a Y jsou nezávislé právě tehdy, když:

$$f_{x|y}(x, y, \Theta) = f_x(x, y, \Theta) \tag{3.22}$$

Od systému očekávám, že bude hledat vektory na základě historické ceny, které tuto podmínku poruší. Pokud jsem návrh finančních modelů pochopil správně, tak z širšího pohledu přesně to dělají. Příkladem, jak takový vektor vytvořím, může být například opět výpočet RSI. Vektor náhodných hodnot X jako ceny close a vektor Y jako hodnoty RSI budou podmínku nezávislých proměnných porušovat. Pokud jsem vše správně pochopil, tak je zanedbatelné, že vektor Y není náhodný. Důležité je, že podmíněná pravděpodobnost náhodného jevu x je rozdílná, pokud nastane jev $y \leq 30\%$ v čase i od pravděpodobnosti marginální. Na obrázku 3.14 je vidět, že systém se může pokusit stanovit podmínky, které najdou korelaci dvou jevů. A Určí jejich podmíněnou pravděpodobnost. Konkrétní případ, kde systém optimalizuje a hledá nejvhodnější hladinu pro indikátor RSI, je uveden v 4.



Obrázek 3.14: Obrázek ukazuje korelaci dvou jevů. Za podmínky, že modrá funkce vrací hodnoty menší než 30, pak je vyšší pravděpodobnost jevu, že černá funkce je ve svém lokálním minimu.

Také jsem vyvodil závěr, že pokud budu chtít aplikovat nějaké analýzy variance, případně posuzovat korelaci různých měnových párů, pokusím se pracovat se slabě stacionárním signálem. Stacionaritu se pokusím zajistit odečtením klouzavého průměru od uzavírací ceny.

Kapitola 4

Návrh systému

Možnosti návrhu musí podléhat základním podmínkám. Těmi jsou obecnost, rozšiřitelnost, možnost testování modulů. Vzhledem k tomu, že neexistuje obecný předpis, jak navrhovat obchodní systémy, je možné vzít v potaz více přístupů. Je možné na problém pohlížet obecněji a snažit se využít co nejvíce informací. Vzhledem k charakteru dat je možné na problém pohlížet jako na teorii chaosu [32], je možné využít technik pro zpracování signálu [21]. Kapitola 3 je tedy kompendiem informací, které jsem vyhodnotil jako podstatné pro návrh. Je otázkou, jak tyto informace vhodně využít. Možností je mnoho, vzhledem k prostudovaným materiálům [9] je možné vytvořit obchodní systém založený na technické analýze. Je možné postavit obchodní systém na principech strategií z [18]. V mnoha přístupech je využito strojového učení [14].

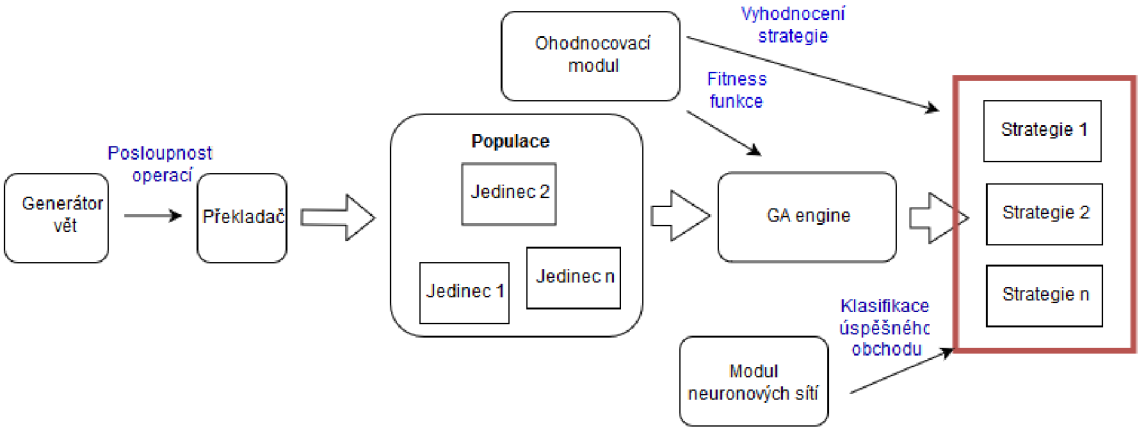
V úvahu jsem bral možnost rozhodovacího stromu, případně konstrukci konečného automatu [13] nebo systém založený na optimalizaci.

Vzhledem k obecnosti chceme, abychom mohli systém rozšiřovat o parametry, které budeme chtít do obchodování zahrnout. Opět se zde otevírá otázka, jak parametry zobecnit. V úvahu připadalo sdílené pole hodnot, které by však bylo pevně dané a případná optimalizace by byla aplikována pouze na tyto parametry. Obchodní systém může brát v potaz širokou škálu možností. Pokud by měl využívat výše zmíněné, je nutné navrhnout, jak obecně pracovat se všemi prostředky. Jde o využití technických indikátorů, nastavení obchodů, zakomponování strojového učení.

Výsledkem této úvahy je architektura popsaná dále v 4.1. V návrhu jsem se snažil vzít v potaz možnost přidávání nových výpočtů a heuristik. Pomocí gramatiky jsem schopen zajistit inicializaci systému s velkou počáteční rozmanitostí různých strategií. Systém by měl být dobře rozšiřitelný o nové prvky, například techniky strojového učení.

4.1 Architektura systému

Systém je založen na gramatice generující jazyk, jehož věty reprezentují danou obchodní strategii. Systém generuje strategie reprezentované řetězcem pro popis posloupnosti výpočtů. Řetězce zpracuje překladač. Výsledkem je matice obsahující vypočtené hodnoty a popis vstupů do obchodní pozice. Ohodnocovací modul přijímá tuto matici a vyhodnotí průměrné zhodnocení obchodních pozic. Kooperace modulů a architektura systému je vyznačena na obrázku 4.1.



Obrázek 4.1: Architektura systému

4.1.1 Generátor vět

Podstatnou částí systému je generátor vět, který inicializuje první populaci. Vytváří množinu vět, které se přeloží a tvoří první populaci. Předpis gramatiky ovlivňuje celkový výsledek. V rámci testování jsem použil gramatiku, která využívá všechny výpočty. Gramatika je parametrem systému. Zápis gramatiky má jistá omezení. Pro správnou funkcionalitu je nutné po výpočtu technických indikátorů provést normalizaci celé tabulky, například takto:

$$S \rightarrow iS \mid iA$$
$$A \rightarrow nB$$
$$B \rightarrow sB \mid dB \mid C$$
$$C \rightarrow bC \mid k$$

Normalizaci jsem zavedl pro správnou kontrolu výpočtů prováděných nad maticí po vygenerování indikátorů. Indikátory počítám s hodnotami trhu, které jsou na vstupu. Pokud od systému chceme, aby výsledné strategie byly vyhodnoceny podle pozic, je nutné gramatiku navrhnout s generováním neterminálu b pro spuštění algoritmu *označení pozice* v překladači.

Tabulka 4.1: Význam neterminálů

Neterminál	význam
i	výpočet a přidání indikátoru
n	(srcnorm) normalizace generované matice
s	(sub) odečtení dvou náhodně vybraných sloupců
d	(div) vydělení dvou náhodně vybraných sloupců
f	(diff) pro náhodně vybraný sloupec se aplikuje funkce pro výpočet rozdílu dvou po sobě jdoucích hodnot
l	(lag) pro náhodný sloupec se aplikuje posunutí směrem do budoucna
g	(log) pro náhodný sloupec se aplikuje logaritmus
p	(oneprops) generování vlastnosti sloupce
x	(windowmax) výpočet maxima v klouzavém okně pro náhodný sloupec
w	(windowmin) výpočet minima v klouzavém okně pro náhodný sloupec
v	(onediv) pro náhodný sloupec se náhodně vybere jednočíslo, které všechny hodnoty vydělí
e	(oneexp) pro náhodný sloupec se náhodně vybere celé jednočíslo, které všechny hodnoty umocní
m	(onemul) pro náhodný sloupec se náhodně vybere jednočíslo, které všechny hodnoty vynásobí
b	(bool) pro náhodný sloupec se náhodně vybere hodnota jako hranice, kde dojde k možnému nastavení pozice

Například pro předpis:

$S \rightarrow iS \mid iA$
$A \rightarrow nB$
$B \rightarrow sB \mid dB \mid C$
$C \rightarrow fC \mid lC \mid gC \mid D$
$D \rightarrow pD \mid E$
$E \rightarrow xE \mid wE \mid F$
$F \rightarrow vF \mid eF \mid mF \mid G$
$G \rightarrow bG \mid k$

Může být vygenerována množina vět ve formátu (operace |)+

- i|srcnorm|sub|div|div|sub|div|sub|sub|div|div|div||div|div|div|div|div||lag|lag|||||bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|bool|final
- i|srcnorm||||div|sub|div|div|sub|div|sub|sub||log|diff|lag|lag|lag|log|log|diff|diff|diff|lag|lag|lag|lag|log|diff|diff|diff|log|diff|lag|log|||windowmin|windowmin||onemul|oneexp||bool|bool|bool|bool|bool|bool|final

Generátor vět je základem celého konceptu a tvoří jeden z parametrů systému. Podle stavu gramatiky můžeme ovlivnit výsledné chování. Vynecháním některých pravidel můžeme

zcela pozměnit charakter strategií. Návrh gramatiky jsem využil především pro testování a vyhodnocení, jaké výpočty a heuristiky jsou zásadním přínosem.

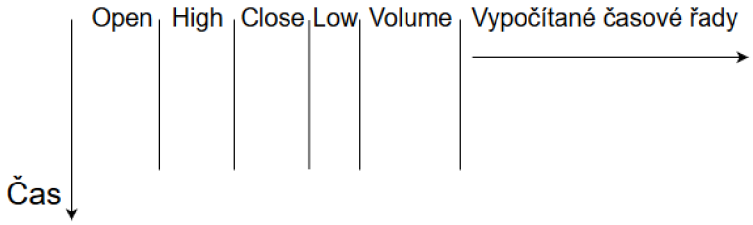
4.1.2 Překladač

Komponenta překladače vytváří na základě vět kompletní jedince jako objekt strategie, který drží několik parametrů pro popis strategie. Parametry reprezentující strategii jsou vyznačeny dále v 4.1.2.

Věta je jediným neprázdným parametrem před tím, než objekt strategie zpracuje překladač.

Předpis výpočtu je celkový záznam výpočtu matice.

Hlavní matice je záznam všech vypočtených kroků. Uchovává také časový index pro označení obchodních pozic. Matice slouží pro celkové zpracování a vyhodnocení strategie evaluačním modulem, který poskytuje fitness funkci. Matice je vyznačena na obrázku. Zpočátku obsahuje pouze informace o ceně pro daný timeframe.



Obrázek 4.2: Formát hlavní matice

Matice vlastností drží doplňující informace pro výpočet hlavní matice

Při zpracování jedince překladač pracuje ve dvou módech, pokud je prázdný parametr *Předpis výpočtu* pak překládá větu a počítá matice. V opačném případě počítá hlavní matici pouze na základě *Předpisu výpočtu*. Při překladu věty jsou postupně z leva do prava zpracována všechna pravidla. Zpracováním je myšleno provedení požadované funkce podle příkazu, jejíž výsledek je případně přidán do hlavní matice nebo do matice vlastností. Funkce jsou popsány v kapitole 4.1. V této kapitole bude podrobněji popsán jejich výpočet.

Při zpracování *věty* jsou funkcím přidány náhodně parametry. Pro provedení výpočtu je výsledek uložen do jedné z matic a název funkce i s parametry je přidán do *Předpisu výpočtu*. V případě neterminálu i je náhodně vybrán jeden z indikátorů pro výpočet.

MA

$$MA_t(n) = \frac{1}{n}\sum_{t=1}^{n} P_{n-t}$$

EMA

$$EMA_t(n) = (P_t - EMA_{t-1})\frac{2}{n+1} + EMA_{t-1}$$

Momentum

$$M_t(n) = P_t - P_{t-n}$$

ROC

$$\frac{P_t - P_{t-n}}{P_{t-n} \times 100} \tag{4.1}$$

ATR

$$EWMA(max(Phigh_t, Pclose_{t-1}) - min(Plow_t, Pclose_{t-1}) \tag{4.2}$$

kde EWMA je exponentially weighted moving average počítaný takto:

$$S_t = aP_t + (1-a)S_{t-1} \tag{4.3}$$

a - Koeficient reprezentuje stupeň snížení váhy. Vyhlazovací faktor mezi 0 a 1.

S_t - exponentially weighted moving average v daném čase.

STO

$$S_t = \frac{Pclose_t - Plow_t}{Phigh_t - Plow_t} \times 100 \tag{4.4}$$

$$STO_t = \frac{1}{n}\sum_{t=1}^{n} S_{n-t} \tag{4.5}$$

Trix

$$Trix_t = 3 \times EWMA_t - 3 \times EWMA(EWMA_t) - EWMA(EWMA(EWMA_t))$$

Vortex

$$TR_t = (max(Phigh_t, Pclose_{t-1}) - min(Plow_t, Pclose_{t-1}) \tag{4.6}$$

$$VM_t = (abs(Phigh_t - Plow_{t-1}) - abs(Plow_t - Pclose_{t-1}) \tag{4.7}$$

$$Trix_t = \frac{\sum_{k=t}^{n} TR_k}{\sum_{k=t}^{n} VM_k} \tag{4.8}$$

RSI

$$RSI_t(n) = 100 - \frac{100}{1 + \frac{U(n)}{D(n)}} \tag{4.9}$$

U(n) součet kladných cenových změn za období délky n

D(n) součet záporných cenových změn za období délky n

OBV

$$L_t = \begin{cases} volume_t & \text{if } Pclose_t - Pclose_{t-1} > 0 \\ 0 & \text{if } Pclose_t - Pclose_{t-1} = 0 \\ volume_t \times -1 & \text{if } Pclose_t - Pclose_{t-1} < 0 \end{cases} \tag{4.10}$$

$$OBV_t(n) = \frac{1}{n}\sum_{k=(t-n)}^{n} L_{t-k} \tag{4.11}$$

EoM

$$L_t = (Phigh_t - Phigh_{t-1}) + (Plow_t - Plow_{t-1}) \times \frac{(Phigh_t - Plow_{t-1})}{2 \times volume_t} \tag{4.12}$$

$$EoM_t(n) = \frac{1}{n}\sum_{k=(t-n)}^{n} L_{t-k} \tag{4.13}$$

CCI

$$CCI_t(n) = \frac{P_t - MA_t}{0.015 \times \sigma} \tag{4.14}$$

kde σ je směrodatná odchylka MA

Copp

$$Copp_t(n) = EWMA(ROC(11) + ROC(14), n) \tag{4.15}$$

Výpočet indikátorů je heuristikou, od které očekávám rychlejší nalezení ziskových strategií. Seznam indikátorů 4.1.2 jsem uvedl zejména pro ukázku, jaké matematické operace obsahují. V podstatě je možné, že by systém k těmto výpočtům došel pouze s ostatními operacemi 4.1. Provedení operace *srcnorm* představuje normalizaci, která nenaruší varianci, tedy pouze časovou řadu transformuje do oboru hodnot mezi 0 a 1.Myslel jsem na problém s přenesením informace z budoucnosti. Normalizací, která nezmění varianci, nepřenesu data do budoucnosti a nalezené prahy budou pouze v jiném měřítku. Tato operace také zajistí, že všechny nové vypočítané řady budou normalizovány předtím, než jsou přidány do hlavní matice.

$$X_i = \frac{X_i - min(X)}{max(X) - min(X)} \tag{4.16}$$

Operace *sub, div, mul, diff, log, lag, oneprops, windowmax, windowmin,onediv, oneexp, onemul* náhodně vyberou jeden nebo dva operandy mezi již existujícími sloupci hlavní matice. Náhodně vygenerují parametr. Zvolí název sloupce a provedou danou operaci. Výsledkem je nová časová řada, která projde případnou normalizací a je přidána do hlavní matice. Tvar matice se tedy po každém výpočtu změní. Provede se nejen přidání sloupce, ale také může dojít k odebrání některých řádků. Například pokud je aplikovaná operace v klouzavém okně, pak jsou v matici hodnoty, pro které není provedenou operací vypočítána hodnota. Je tedy nutné odstranit celý řádek. Pokud počítáme například klouzavý průměr přes deset hodnot, pak popsaná situace nastane pro prvních devět hodnot.

Operace *bool*, která je gramatikou umístěna na konec, určuje časový index v hlavní matici pro otevření pozice. Náhodně vybere jeden ze sloupců a provede algoritmus *označení pozice*. Pro vybraný sloupec, tedy časovou řadu, je vybrána hladina z oboru hodnot, kde protnutí bude určovat vstup do pozice:

```
threshold = setThreshold();
 columm = getRandomCol();
 while value_i not null do
   if value_i >threshold then
      if threshold >value_{i-1} then
         set to index i 1
      end
   end
   if threshold >value_i then
      if value_{i-1} >threshold then
         set to index i -1
      end
   end
end
```

Algorithm 1: position set

Algoritmus nastaví v časovém indexu 1 nebo −1, pokud dojde k překřížení hladiny shora nebo zespoda. Operace *bool* opět po provedení přidá operaci i s parametrem do *předpisu výpočtu*.

Po zpracování věty je tedy *předpis výpočtu* ve tvaru $(operace-operandA(,operandB)?-(parametry-)*|)*$. Podle *předpisu výpočtu* může být strategie dále zkoumána nebo nasazena, pokud ji systém vyhodnotí jako ziskovou. Příklad překladu *věty* na *předpis výpočtu* je dále v 4.1.3.

Věta $i|i|i|i|srcnorm|div|div|div|onediv|bool|bool|bool|final$

Předpis výpočtu $MA-33-MA\_XTMFZE|ATR-46-ATR\_JQEVIM|$
$ATR-52-ATR\_EREYFL|CCI-7-CCI\_JQWPFV|$
$srcnorm|div-Low,CCI\_JQWPFV-div\_QUTZGY|$
$div-Volume,div_QUTZGY-div\_DTQWRI|$
$div-ATR\_EREYFL,div\_DTQWRI-div\_WHWWQA|$
$onediv-Close-0.33973084225923667-onediv\_OBAIRF|$
$onediv-div\_DTQWRI-0.017998461092171847-onediv\_IMFFJV|$
$bool-Volume-0.6469928741458684-bool\_WFOFCY|$
$bool-div\_WHWWQA-0.6457190703788932-bool\_GXVDRZ|$
$bool-div_QUTZGY-0.6158056574393338-bool\_DVGMUT$

Funkcionalita překladače je dále rozšířena o možnost výpočtu hlavní matice pomocí *předpisu výpočtu*. Překladač je pro přepočítání využíván při optimalizaci a také je možné separátně strategii přepočítat, například na datech z jiného časového období.

4.1.3 Optimalizační modul

Modul optimalizace využívá generátor vět s překladačem pro inicializaci první generace. Překladač je dále využíván pro přepočítání matice po operaci mutace a křížení. Využívá evaluační modul, který poskytuje fitness funkci.

Optimalizace je inspirována genetickými algoritmy 3.2. Reprezentaci jedince jsem navrhl jako objekt s parametry uvedenými v kapitole překladače 4.1.2. Pro optimalizaci je důležitým parametrem *předpis výpočtu*. Nad tímto řetězcem jsou počítány operace mutace a křížení.

Mutace by měla být postavena následovně: prostor jedinců, který se optimalizace snaží prohledat a najít v něm optimální strategie, je třeba projít celý i přesto, že jsme našli lokální maximum. Pokud by byla délka řetězce omezena, pak by měla být mutace sama schopna projít celý uvažovaný prostor.

Tedy implementace mutace projde celý řetězec a náhodně vymění buď funkci, nebo pouze její parametr. Je třeba kontrolovat, zda je výměna možná vzhledem k tomu, že jednotlivé operace mohou být unární nebo binární. Pro přehlednost je níže popis algoritmu.

```
value_i = rozdeleniRetezce();
while value_i not null do
    hodnotaProVymenu = separujParametrNeboFunkci();
    if hodnotaProVymenu == funkce then
        if hodnotaProVymenu == unární operace then
            Na pozici v řetězci je umístěna nová unarní operace
        end
        if hodnotaProVymenu == unární operace then
            Na pozici v řetězci je umístěna nová binární operace
        end
    end
    if hodnotaProVymenu == parametr then
        Na pozici v řetzci je umístěna nová hodnota
    end
end
```

Algorithm 2: Mutace

Příklad takovéto mutace je následující:

Předpis výpočtu $MA-33-MA\_XTMFZE|ATR-46-ATR\_JQEVIM|$
$ATR-52-ATR\_EREYFL|CCI-7-CCI\_JQWPFV|$
$srcnorm|div-Low,CCI\_JQWPFV-div\_QUTZGY|$
$div-Volume,div\_QUTZGY-div\_DTQWRI|$
$div-ATR\_EREYFL,div\_DTQWRI-div\_WHWWQA|$
$onediv-Close-0.33973084225923667-onediv\_OBAIRF|$
$onediv-div\_DTQWRI-$
$0.017998461092171847-onediv\_IMFFJV|$
$bool-Volume-0.6469928741458684-bool\_WFOFCY|$
$bool-div\_WHWWQA-0.6457190703788932-bool\_GXVDRZ|$
$bool-div_QUTZGY-0.6158056574393338-bool\_DVGMUT$

Mutovaný předpis výpočtu $STO-33-MA\_XTMFZE|ATR-46-ATR\_JQEVIM|$
$ATR-52-ATR\_EREYFL|CCI-7-CCI\_JQWPFV|srcnorm|$
$sub-Low,CCI\_JQWPFV-div\_QUTZGY|$
$div-Volume,div\_QUTZGY-div\_DTQWRI|$
$div-ATR\_EREYFL,div\_DTQWRI-div\_WHWWQA|$
$onediv-Close-0.33973084225923667-onediv\_OBAIRF|$
$onediv-div\_DTQWRI-0.017998461092171847-onediv\_IMFFJV|$
$bool-Volume-0.6469928741458684-bool\_WFOFCY|$
$bool-div\_WHWWQA-0.6457190703788932-bool\_GXVDRZ|$
$bool-div\_QUTZGY-0.752-bool\_DVGMUT$

Návrh křížení byl složitý. Bylo vyzkoušeno několik přístupů, jak bych mohl provést náhodné křížení v bodě. Po konzultaci s autorem přednášky [29] prof. Sekaninou jsem tento problém uzavřel s cílem křížit jedince v pevném bodě. Za výběrem bodu stojí opět mnou stanovená heuristika, že optimální bude křížit jedince v bodě normalizace. Již efektivní kombinace indikátorů s již efektivní kombinací dalších výpočtů u daného jedince může být zkombinována s jedincem s podobnými vlastnostmi. Příklad křížení je následující:

Jedinec A $MA-33-MA_XTMFZE|ATR-46-ATR\_JQEVIM|$
$ATR-52-ATR\_EREYFL|$
$CCI-7-CCI\_JQWPFV$
$srcnorm|div-Low,CCI\_JQWPFV-div\_QUTZGY|$
$div-Volume,div\_QUTZGY-div\_DTQWRI|$
$div-ATR\_EREYFL,div\_DTQWRI-div\_WHWWQA|$
$onediv-Close-0.33973084225923667-onediv\_OBAIRF|$
$onediv-div\_DTQWRI-0.017998461092171847-onediv\_IMFFJV|$
$bool-Volume-0.6469928741458684-bool\_WFOFCY|$
$bool-div\_WHWWQA-0.6457190703788932-bool\_GXVDRZ|$
$bool-div\_QUTZGY-0.6158056574393338-bool\_DVGMUT$

Jedinec B $Copp-95-Copp\_KCLEFO|STO-54-STO\_SHEZSF|ROC-33-ROC_HMLBFS|$
$srcnorm|div-Copp\_KCLEFO,Low-div\_KQQCZL|$
$div-STO\_SHEZSF,ROC\_HMLBFS-div\_ITSUME|$
$div-High,STO\_SHEZSF-div\_AXNOLJ|$
$div-div\_KQQCZL,Volume-div\_IYWEOL|$
$div-Close,High-div\_KATWYM|$
$sub-div\_KQQCZL,ROC\_HMLBFS-sub_JOVHLZ$
$sub-Open,div\_ITSUME-sub\_ALVBVZ|$
$sub-div\_KQQCZL,sub\_ALVBVZ-sub\_IJFUKS|$
$div-Open,Volume-div\_WYCNDX|$
$sub-sub_IJFUKS,div\_KATWYM-sub\_EBPPKQ|$
$sub-ROC\_HMLBFS,div\_IYWEOL-sub\_HKGEGK|$
$div-div\_ITSUME,div\_KQQCZL-div\_GPTAYG|$
$div-sub\_HKGEGK,STO\_SHEZSF-div\_HTLZDJ|$
$sub-div\_ITSUME,sub\_HKGEGK-sub\_QOBCVC|$
$div-Low,div\_KQQCZL-div\_XSIQEJ|$
$div-div\_ITSUME,div\_IYWEOL-div\_KJHZJZ|$
$bool-sub\_IJFUKS-0.6096366953175696-bool_QPIUHA|$
$bool-div\_KQQCZL-0.8278115654647632-bool\_IBAFGO$

Potomek jako výsledek křížení $MA-95-Copp\_KCLEFO|ATR-54-STO\_SHEZSF|$
$ATR-33-ROC\_HMLBFS|srcnorm|$
$div-Copp\_KCLEFO,Low-div\_KQQCZL|$
$div-STO\_SHEZSF,ROC\_HMLBFS-div\_ITSUME|$
$div-High,STO\_SHEZSF-div\_AXNOLJ|$
$div-div\_KQQCZL,Volume-div\_IYWEOL|$
$div-Close,High-div\_KATWYM|$
$sub-div\_KQQCZL,ROC\_HMLBFS-sub\_JOVHLZ|$
$sub-Open,div\_ITSUME-sub\_ALVBVZ|$
$sub-div\_KQQCZL,sub\_ALVBVZ-sub\_IJFUKS|$
$div-Open,Volume-div\_WYCNDX|$
$sub-sub\_IJFUKS,div\_KATWYM-sub\_EBPPKQ|$
$sub-ROC\_HMLBFS,div\_IYWEOL-sub\_HKGEGK|$
$div-div\_ITSUME,div\_KQQCZL-div\_GPTAYG|$
$div-sub\_HKGEGK,STO_SHEZSF-div\_HTLZDJ|$
$sub-div\_ITSUME,sub\_HKGEGK-sub\_QOBCVC|$
$div-Low,div\_KQQCZL-div\_XSIQEJ|$

$$div - div\_ITSUME, div\_IYWEOL - div\_KJHZJZ|$$
$$bool - sub\_IJFUKS - 0.6096366953175696 - bool\_QPIUHA|$$
$$bool - div\_KQQCZL - 0.8278115654647632 - bool\_IBAFGO$$

Vyměněno bylo tolik indikátorů, kolik obsahoval jedinec s menším počtem indikátorů. Zbytek indikátorů byl ponechán nezměněn. Výsledkem křížení může být i potomek s opačnou výměnou. Na příkladu je červeně vyznačeno místo křížení. Modře je potom vyznačeno předání indikátorů. Zamýšlenou heuristikou je především myšlenka, že indikátory tvoří základ technické analýzy, tedy obchodních strategií. Pokud byl jistý přepočet nad indikátory úspěsný, je možné, že s jinými indikátory bude také úspěsný.

Optimalizační modul je možné nastavit, tak aby provedení operací mutace a křížení proběhlo pouze s jistou pravděpodobností. Pro selekci potomků v další generaci je využit algoritmus turnajový výběr [16]. Jde o populární přístup, ve kterém jsou jedinci náhodně vybráni a ohodnoceni fitness funkcí. V daném kole vyhrává jedinec s nejlepším ohodnocením a postupuje dále. Výherce postupuje do dalšího kola turnaje.

4.1.4 Evaluační modul

Obsahuje několik užitečných funkcí pro zpracování dat. Tento modul je využíván modulem pro strojové učení a modulem optimalizačním. Zastupuje jakékoliv funkce pro vytvoření statistik, například počet ziskových a ztrátových obchodů. Poskytuje pole indexů, na kterých jsou vypočítané pozice v *hlavní matici*.

Nejdůležitějším článkem je poskytování funkcí pro ohodnocení jedince, tedy poskytuje fitness funkce pro genetickou optimalizaci. Důležitost návrhu ohodnocení jedince je patrná z teorie 3.2. Je žádoucí, aby ohodnocení odpovídalo podmínkám při reálném nasazení.

V této práci beru konkrétně v potaz, že ohodnocení bude závislé na čase a opět není vhodné využívat informace v čase $t + n$, pokud poslední přijatou cenou je cena v čase t. Dále je popsána funkce pro ohodnocení pozice long. Ohodnocení pozice short je postavené na stejných principech.

Otevření pozice přináší dva problémy. Prvním je správný výběr času pro výstup z pozice neboli prodání měny za co největší cenu v případě pozice *long*. Problém správného výstupu by mohl být opět řešen systémem samotným. Ovšem rozšiřování a obecné nasazení systému je nad rámec této práce. Pro začátek jsem se řídil základním postupem při návrhu strategie 2.0.2. Tedy určil jsem pevnou hladinu pro výstup z pozice, která bude dosahovat vyšších zisků než potenciální ztráta určená hladinou pro ukončení pozice v případě rostoucích ztrát.

Také je možné aplikovat metodu posouvání stoplossu zvanou *trailing stop* 2.0.2. Metoda *trailing stop* byla implementována jako první a bylo provedeno několik testů 5.0.2.

Dále systém poskytuje vyhodnocení pozice podle dosaženého maxima v případě zisku a ztrát, opět podle určené hladiny *stoploss*. Použití této funkce vysvětluje, proč jsem použil výraz, že není vhodné využívat informace z budoucnosti. Přímo jsem však tento přístup nevyloučil. Určení lokálního maxima v aktuálním čase bez znalosti budoucích hodnot není možné. Funkci jsem implementoval pro porovnání, zda takto přesné ohodnocení nepřinese lepší výsledky v průběhu optimalizace. Nasazení optimalizované strategie touto funkcí by podléhalo omezení nemožnosti hledané maximum určit. Ziskový výstup z pozice by mohl být stanoven jako *take profit* na cenové hladině průměru mezi maximem a minimem z trénovací sady.

Poslední implementovanou a výsledně používanou ohodnocovací funkcí je stanovení pevné hladiny *take profit* a *stop loss*. Řešení *take profitu* a *stop lossu* bere v úvahu, že zkoumaný signál je diskrétní. Zaznamenání hladiny tak nemusí být přesné a je nutné tuto

skutečnost modelovat. Uvedená nepřesnost spolu s vyhodnocením pozice je uvedena na obrázku 4.3

Černá vertikální čára v čase 11:45 značí vstup do pozice. Funkce sleduje budoucích 100, a pokud v tomto období není zaznamenán průraz *take profitu* nebo *stop lossu*, pak je připsána cena konečná. Tedy v čase $t = 11 : 45$ začne simulace vyhodnocení na historických datech. Horizontální zelená čára na cenové hladině $1,19743$ vyznačuje *take profit* a fialová na hladině $1,19545$ *stop loss*. Máme tedy nějaký *money management*, který určuje možnou ztrátu přibližně 50 bodů. Mezi přijetím uzavírací ceny v čase 14:45 a 15:00 dojde z propadu přibližně 200 bodů, a *stop loss* je tak zaznamenán s přibližnou ztrátou 200 bodů. K této situaci může dojít na libovolném časovém rozlišení včetně tickových dat.

Obrázek 4.3: Ukázka nepřesného překročení hladiny

4.1.5 Modul neuronových sítí

Modul poskytuje klasifikátor založený na neuronových sítích, který vyhodnocuje, zda pozici určenou strategií otevřít. Obsahuje dopřednou i LSTM síť. Modul je založen na teorii popsané v kapitole 3.1. Obsahuje tedy metody pro *preprocessing*, pro trénování sítí a jejich vyhodnocení na testovacích datech.

Systém byl navržen tak, aby přepočty, které se opírají o teorii a případně by proces generalizace zlepšily, byly zahrnuty ve výpočtech strategie. Tedy například transformace dat do normálního *heavy tailed* rozložení je zanesena přímo v matematických operacích strategie. Dalším krokem před samotným trénováním je redukce dimenze pomocí PCA.

Trénování sítě je provedeno nad daty, která byla použita při optimalizaci. Modulu je předána celá hlavní matice s indexy pro otevření pozic. Modul neuronových sítí pomocí evaluačního modulu ohodnotí jednotlivé indexy jako ziskové nebo ztrátové a opět tato data přidá do hlavní matice. Na základě hlavní matice sestaví trénovací sadu. Sada je sestavena

vybráním dat, která souvisí s indexem obchodní pozice. Sestavení je založené na myšlence, že pokud v čase (indexu) t je označeno otevření pozice, pak v intervalu $0 - t$ jsou data relevantní pro správné vyhodnocení, zda je pozice zisková. Konečné nastavení je takové, že modul jako jeden vstup přijímá interval $(t - 150) - t$. V tomto intervalu jsou všechny sloupce zaneseny jako vstup do neuronové sítě kromě sloupců *bool* a značky, zda je obchod ziskový. Na základě informace značky je sestaven referenční vektor pro porovnání výstupu neuronové sítě. Vstupní data jsou normalizována a normalizační konstanty jsou uloženy pro normalizaci dat testovacích.

Natrénovaný model sítě je uložen a připraven pro spuštění na trénovacích datech. Testování je závislé na testování samotné strategie, neboť nejdříve je nutné na testovacích datech vypočítat hlavní matici. Nasazení modulu je uvedeno v testech 5.

Konkrétní architektura LSTM sítě obsahuje dvě rekurentní vrstvy o šířce 64 neuronů a výstupní dopřednou vrstvu s aktivační funkcí *textit*. Architekturu dopředné neuronové sítě tvoří 4 vrstvy o velikosti dané velikostí trénovacího vektoru s aktivační funkcí *tanh*. Faktor učení je nastaven na hodnotu 0.001.

Kapitola 5

Testování systému

V následujících podkapitolách jsou popsány testy trénování a vyhodnocení na testovacích datech v různé konfiguraci. Kapitola je přínosem i pro čtenáře, který si chce udělat přehled o experimentech, jejichž výsledky stanovily další směr a vedly ke konečnému návrhu, neboť jsou také obsahem kapitoly.

Pro testování a trénování jsem stáhl historická data měnového páru *EUR/USD* v časovém rozlišení patnácti minut. Celkově je k dispozici 12 tisíc záznamů o ceně za rok 2017 a začátek roku 2018. Z pohledu dní jde přibližně o data ze 125 historickcých dní. Trénování je prováděno v rámci jednoho měsíce. Test je proveden na navazujícím období sedmi dnů. Historická data pro tuto práci poskytla banka Dukascopy [1].

Obecně napříč testy byly testována různá období. Žádný časový úsek nebyl vyhodnocen jako extrém. Vyvodil jsem tak závěr, že systém pracuje rovnoměrně napříč časovými úseky.

5.0.1 Test základního principu optimalizace

Vyhodnocení chování genetických algoritmů je nutné provést na dostatečném počtů pokusů [29]. Cílem této podkapitoly je uvést několik možností a konstatování chování navrženého systému vzhledem k optimalizaci pomocí genetických algoritmů. Genetické algoritmy jsou založeny na operacích 3.2.1, které mohou být pojaty různě. V kapitole 4.1 je popsáno jejich konečné nastavení. Při návrhu systému jsem uvažoval i jiné možnosti.

Než přejdu k testům samotné architektury, zmíním test, který testuje základní princip a posuzuje nasazení genetických algoritmů na jednoduchém případu. Pokusím se vyhledat hladiny *RSI*, které nejlépe odpovídají indikaci, že dojde k obratu kurzu. Test jsem provedl na datech páru EUR/USD s intervalem patnácti minut. V rámci testu jsem uvažoval pouze uzavírací cenu a indikátor *RSI*. V rámci testu byl prověřen algoritmus *označení pozice*, jak je uvedený zde 4.1. Cílem bylo nalézt optimální hodnotu *RSI*, která bude indikovat vhodnou situaci pro otevření pozice long a short.

První populace byla množina jedinců s přepočtem *RSI* v intervalu 5 až 100 a náhodnou hladinou pro vstup do pozice v intervalu oboru hodnot *RSI*. Jedinec je tedy reprezentován dvěma parametry. Mutace byla implementována jako náhodné přičtení k aktuální hodnotě parametrů v intervalu 1 až 100 s validací, že výsledná hodnota parametru je kladná. Nastavení a funkcionalita ostatních faktorů genetického algoritmu byla stejná jako výsledná architektura. Provedl jsem třicet spuštění s množinou 100 jedinců pro první populaci. Průměrné zhodnocení výsledků je uvedeno tabulce 5.1.

[1] https://www.dukascopy.com/

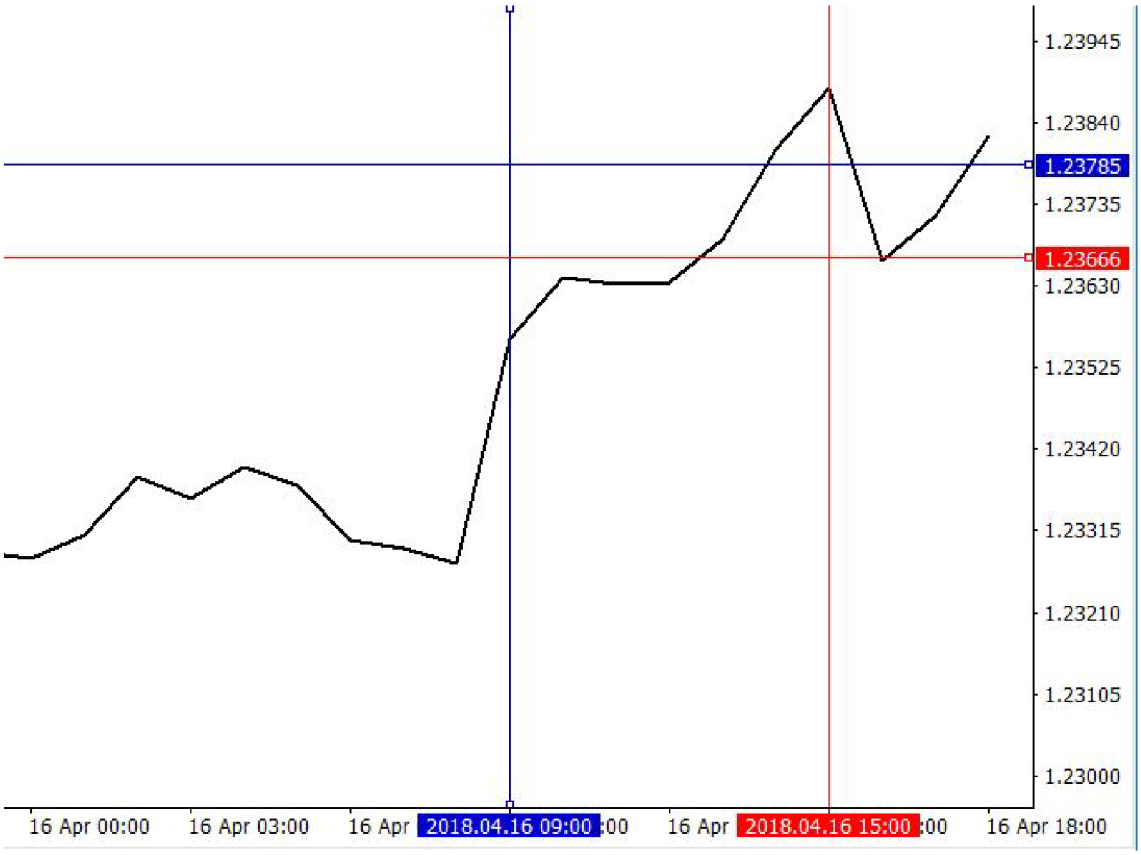
Tabulka 5.1: Test základního principu systému na indikátoru RSI

	průměr	variance
Hladina RSI pro pozici long	21	8
Hladina RSI pro pozici short	83	5
Období pro výpočet RSI	28	4

Z výsledků jsem vyvodil, že na tomto základním testu návrh optimalizace uspěl, neboť jednotlivé výsledky došly k podobnému závěru. Výsledky se shodují s teoretickým popisem charakteru indikátoru RSI.

5.0.2 Testy fitness funkce trailing stop

Z velkého počtu možností, jak implementovat fitness funkci, byla vybrána jako první simulace trailing stop 2.0.2. Překážkou nastavení byl především charakter dat. Vzhledem k výpočetní náročnosti optimalizace jsem zvolil vstupní data v rámci daného timeframu. Pokud by systém zpracovával tickova data, výpočet by byl paměťově náročný a zpomalení by bylo neúměrné vzhledem k přínosu, který by tento charakter dat měl. Například pro testovací data pro timeframe 15 minut je rozdíl mezi cenou v čase P_t a P_{t-1} značný, a tak stop loss může být aktivován v propadu větším než je nastavený limit. Následující příklad situaci vysvětluje 5.0.2.



Obrázek 5.1: Architektura systému

Případný vstup do pozice long v čase 9:00 je označen modrou vertikální čárou. Následuje vzestup až do lokálního maxima v čase 15:00. Horizontální modrá čára vyznačuje nastavený trailing stop loss přibližně na 50 bodů. Pokud cena poklesne, očekává se ukončení pozice na hladině horizontální modré čáry. Pokles je však zaznamenán až na hladině horizontální červené čáry, kde je přijata následující cena.

Test 1

data - 15minutové svíčky páru EURUSD.

fitness funkce - simulace trailing stop s omezením 15 min.

počet spustění pro každé nastavení - 30

použitá gramatika -

$$
\begin{aligned}
&S \rightarrow iS \mid iA\\
&A \rightarrow nB\\
&B \rightarrow sB \mid dB \mid C\\
&C \rightarrow fC \mid lC \mid gC \mid D\\
&D \rightarrow pD \mid E\\
&E \rightarrow xE \mid wE \mid F\\
&F \rightarrow vF \mid eF \mid mF \mid G\\
&G \rightarrow bG \mid k
\end{aligned}
$$

význam neterminálů je vysvětlen zde 4.1

Sloupec A ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 100. Jsou použity operace **mutace** a **křížení**. Mutace je založena pouze na náhodné **změně číselných parametrů**, jak je uvedeno v popisu architektury 4.1. Stejně tak křížení.

Sloupec B ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 500, také jsou použity operace **mutace** a **křížení**, a to ve stejném nastavení jako test pro sloupec A.

Sloupec C ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 500. Využívá operaci mutace založenou pouze na náhodné **změně číselných parametrů**.

Sloupec D ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 500. Využívá mutaci, která zaměňuje číselné parametry, operace a operandy.

Sloupec E ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 1000. Jinak je nastavení stejné jako pro sloupec D.

Z výsledků jsem vyvodil, že výsledek je závislý na velikosti první populace, která je generována uvedenou gramatikou. Gramatika samotná ovlivňuje výsledek. Křížení v normalizaci je přínosem a bude dále využíváno. Mutaci je opravdu nutné navrhnout tak, aby byla schopna vytvořit jakéhokoliv jedince z uvažovaného prostoru jedinců.

Tabulka 5.2: Tabulka výsledků

	A	B	C	D	E
Průměr zlepšení (v násobku:)	3	3	1,8	3	4
Průměrná směrodatná odchylka	0,5	0,6	0,2	2	3
Průměr generací, kde docházelo ke zlepšení	5	8	9	9	11
Průměr nejlepšího výsledku	0,1	0,21	0,26	0.3	0.58

5.0.3 Test finální optimalizace

V další části jsem provedl sérii testů 5.0.3 zaměřených na finální nastavení genetických algoritmů. Nastavení a popis jednotlivých operací je uveden v architektuře systému 4.1. Kapitola také obhajuje, proč některé varianty uvedené v 4.1 byly pozměněny do konečné podoby *testu 3*. *Test 2* bere v potaz obchodní pozice všech sloupců *bool*. V závěru popisu *testu 2* je vysvětleno, proč byla dále nasazena fitness funkce s výběrem pouze nejlepšího sloupce určujícího pozici.

Testování bylo obtížné vzhledem k výpočetní náročnosti, která je nevýhodou genetických algoritmů obecně. Ovšem systém by měl být schopný reagovat na změny trhu v rámci obchodovaného timeframu. Vzhledem k faktu, že systém testuji na měnovém páru EUR/USD v rámci patnáctiminutového intervalu, jsem určil, že adekvátní doba výpočtu bude v rámci hodin, maximálně jednoho dne.

S nastavením finální fitness funkce, která stanovuje přesný *take profit* a *stop loss*, jsem v průběhu testování dosahoval lepších výsledků než v případě fitness funkce z prvního testu. Ale vzhledem k nedostatku výpočetního výkonu jsem byl nucený omezit počet generací na 50. Toto omezení však z 90% nemělo vliv na ukončení. Pokud se výsledek nezlepšil po 4 generace, pak byla optimalizace ukončena.

Test 2

data - 15minutové svíčky páru EUR/USD.

fitness funkce - stanovuje přesný *take profit* a *stop loss*.

počet spuštění pro každé nastavení - 30

význam neterminálů je vysvětlen 4.1

Sloupec A ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 100. Jsou použity operace **mutace** a **křížení**.

Sloupec B ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 500. Také jsou použity operace **mutace** a **křížení**.

Sloupec C ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 500. Využívá pouze operaci mutace.

Sloupec D ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 1000. Jsou použity operace **mutace** a **křížení**.

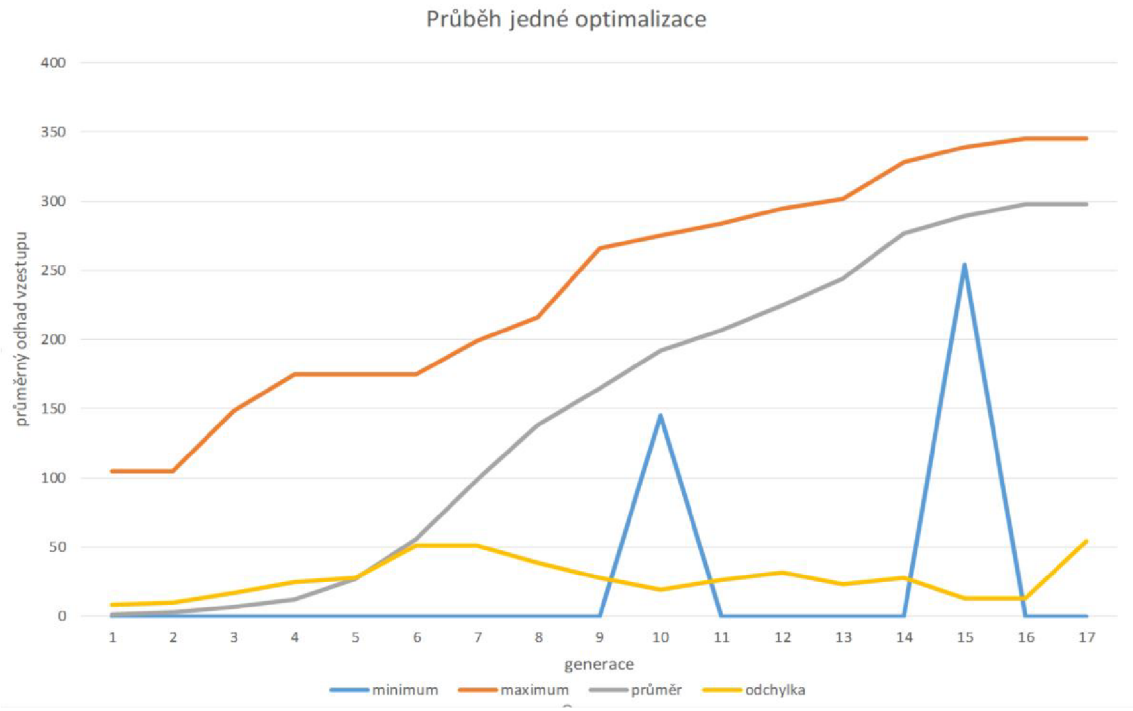
Z testů jsem zjistil, že velikost počáteční populace zvyšuje pravděpodobnost rychlejšího nalezení lepšího řešení. Operace křížení také celý proces urychluje, ale není nutná.

Tabulka 5.3: Tabulka výsledků pro test 2

	A	B	C	D
Průměr zlepšení (v násobku:)	8	9	5	3
Průměrná směrodatná odchylka	54	161	42	197
Průměr generací, kde docházelo ke zlepšení	25	28	16	31
Průměr nejlepšího výsledku	212	374	286	523

Na grafu jsem zobrazil průběh jednoho spuštění 5.0.3. Průběh ukazuje, že postupem evoluce se neustále přibližuje průměrný zisk a zisk maximální. Pokud se stále zlepšovala průměrná kvalita populace a pokud i odchylka byla relativně vysoká, pak průběh evoluce indikoval nalezení ještě lepšího jedince. Optimalizaci jsem ukončil ve chvíli, kdy rozdíl 4 předchozích maxim byl zanedbatelný.

Tento test sice ukazoval velmi dobrý průměrný násobný výsledek, ale po důkladném prozkoumání, jak strategie vypadají, jsem zjistil, že závislost jednotlivých sloupců *bool* je minimální, a tak byl počet vygenerovaných sloupců úměrný zisku, což plně nereprezentovalo kvalitu daného výpočtu. Proto jsem v dalším testu upravil fitness funkci pro hodnocení strategie pouze podle nejlepšího sloupce *bool*.



Obrázek 5.2: Průběh optimalizace

5.0.4 Ohodnocení podle nejlepšího sloupce *bool*

Tento test prezentuje konečné nastavení. Komentuje jak průběh trénování, tak nasazení na testovacích datech.

Test 3

data - 15minutové svíčky páru EURUSD.

fitness funkce - pevný *take profit* a *stop loss*.

počet spuštění pro každé nastavení - 30

význam neterminálů je vysvětlen v 4.1

Sloupec A ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 100. Jsou použity operace **mutace** a **křížení**.

Sloupec B ukazuje výsledky pro nastavení, kde je počátečční populace o velikosti 500. Také jsou použity operace **mutace** a **křížení**.

Sloupec C ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 500. Využívá pouze operaci mutace.

Sloupec D ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 1000. Jsou použity operace **mutace** a **křížení**.

Tabulka 5.4:

	A	B	C	D
Průměr zlepšení (v násobku:)	2	4	3	3
Průměrná směrodatná odchylka	5	11	4	13
Průměr generací, kde docházelo ke zlepšení	19	24	18	25
Průměr nejlepšího výsledku	20	26	17	37

Na vyhodnocení je vidět, že výsledky nejsou zkreslené počtem sloupců *bool* v jedné strategii. Tedy násobek zlepšení není tak vysoký jako v testu 2. Test přinesl pozitivní výsledky, že při trénování je možné najít ziskové strategie. Vzhledem k tomu, že toto je finální test, tak dále přikládám výsledek poslední optimalizace. Nejlepší průměrný výsledek dosahoval průměrného zisku 20 bodů. Průměrný zisk populace byl 2 body. V další části je výsledek této optimalizace na testovacích datech, který pokračuje testem trénování neuronové sítě, a opětovné vyhodnocení na testovacích datech.

Důležité je zmínit nasazení na reálných datech. Test byl vždy proveden na následujícím období, které bylo poloviční oproti trénovací sadě. Optimalizace byla z 70 % procent přetrénovaná a v testu převažovaly ztráty. Proto jsem provedl další test, ve kterém jsem přidělil jistá testovací data na validaci a testoval, zda strategie bude výdělečná, pokud bude zvalidována. Pokud strategie prošla validací, pak byla z 85 % procent zisková i na testovací sadě. Zisk se lišil průměrně o 20 % od zisku při trénování.

5.0.5 Nasazení strategie na testovacích datech

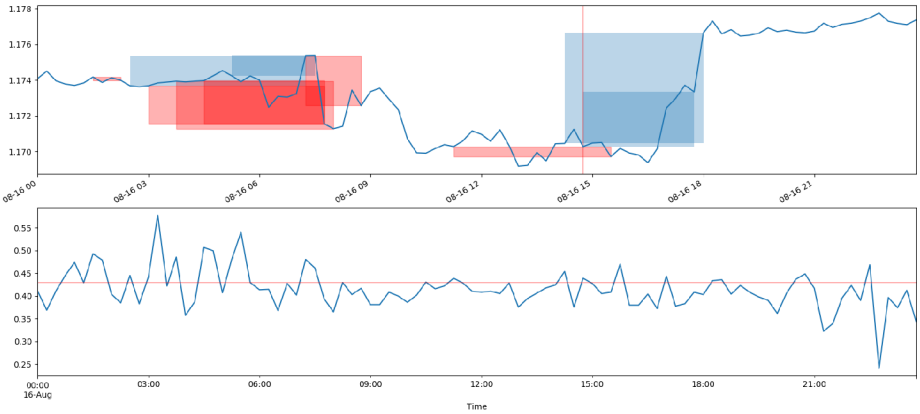
Provedl jsem optimalizaci s nejlepší strategií o průměrném výdělku 20 bodů. Do výsledku je nutné započítat náklady na otevření pozice pro zcela přesnou představu 2 o tom, zda je možné uvažovat o nasazení strategie. Další sledované hodnoty na trénovacích datech také vykazovaly dobrý výsledek. Počet ztrátových pozic byl 186 a počet ziskových 328 s celkovým ziskem 5163 bodů. Velikost trénovacích dat byla 1418.

Následně byla strategie nasazena na testovacích datech, kde bylo simulováno reálné nasazení. Testovací sada měla velikost 680. Průměrný zisk činil 11 bodů a celkový zisk byl

1131 bodů. Ztrátových pozic bylo pouze 67 oproti ziskovým, kterých bylo 130. Strategie obsahovala následující výpočet:

$Momentum - 3 - STO\_DDGLPN|srcnorm|sub - High, Volume - sub\_VCQSGG|$
$sub - Close, Open - sub\_SVCZSP|sub - High, Close - sub\_EPINNX|$
$lag - Low - 84 - lag\_WKLQJK|lag - High - 161 - lag\_HKKZFS|$
$lag - Close - 53 - lag\_RDEJTM|windowmin - STO\_DDGLPN - 25 - windowmax\_WSGAXA|$
$bool - sub\_SVCZSP - 0.436 - bool\_KIWZJJ$

Odečtením hodnot ze sloupce *close* a *open* dostáváme ziskovou strategii při vstupu na hladině 0.43. Pro představu, jak vypočítaná data vypadají, jsem přiložil obrázek 5.3. Obchod je spuštěn, pokud je hodnota překročena. Zobrazeny jsou jen obchody, které jsou v zobrazeném rozmezí ukončeny. Obchody po čase vyznačeném červenou vertikální čárou nejsou ukončeny, proto na grafu zobrazeny nejsou.



Obrázek 5.3: Průběh optimalizace

5.0.6 Nasazení neuronové sítě

v poslední části byla testována podpora strategie pomocí klasifikátoru založeném na neuronové síti. Byla testována dopředná i LSTM síť. Na data bylo pohlíženo jako na sekvenci v čase. Jako první byla testována síť LSTM. Topologie byla následující: síť obsahovala dvě LSTM vrstvy o výstupní šířce 64 a výstupní dopřednou vrstvu s aktivační funkcí *tanh*.

Průměrný počet epoch (4) vypovídal o špatné generalizaci. Dále bylo testováno přidání další rekurentní vrstvy. Průběh trénování se však nezlepšil.

Nasazení na testovacích datech bylo opět neuspokojivé. Strategie nebyla zlepšena. Nejvyššího zlepšení síť dosáhla v oblasti stoupajícího trendu, kde všechny obchody vyhodnotila jako ziskové.

Nasazení dopředné neuronové sítě vykazovalo mnohem lepší výsledky. Testoval jsem topologii uvedenou v kapitole 4.1.5. Zpracování dat jsem také aplikoval tak, jak je popsáno v návrhu modulu 4.1.5. Zásadním parametrem pro správnou generalizaci problému byla velikost intervalu pro jeden trénovací vektor. Celkově jsem provedl pět testů, kdy jsem otestoval interval o délce 10, 50, 150, 200 a 300. Vyjmul jsem z *hlavní matice* hodnoty všech sloupců v daném intervalu, jak je popsáno v návrhu. Konečná velikost vstupní vrstvy pak

byla zavislá i na počtu sloupců v dané matici. Průběh trénování byl nejlepší pro velikost 150 a dále na trénovací sadě nedocházelo k lepším výsledkům.

Provedl jsem trénování na 10 různých strategiích: pro interval 10 byl průměrný počet epoch 4, pro interval 50 pak průměrem bylo 6 epoch. U velikosti intervalu 150 a více síť provedla průměrně 21 epoch učení, kdy stále docházelo ke zlepšení na validační sadě. Druh dat měl také vliv na průběh učení, čímž jsem si ověřil závislost mezi charakterem dat vypočítaných danou strategií a učením neuronové sítě.

Konkrétní příklad bude pokračováním předchozí části 5.0.5. Nad vypočítanou maticí strategie natrénujeme popsanou dopřednou neuronovou síť. Průběh trénování byl po dobu 24 epoch. Počáteční přesnost klasifikace na validační sadě byla 54 % s konečnou přesností 86 %. Vzhledem k tomu, že některé pozice jsem nemohl zanést do trénovací sady kvůli zmíněnému intervalu, neodpovídá počet pozic počtu pozic z testu, kde je nasazena strategie bez neuronové sítě.

Po natrénování síť zapříčinila 119 ztrát a rozhodla o 254 ziskových pozicích. Bez neuronové sítě by došlo k 170 ztrátám a 263 ziskům. Na zbytek pozic rozhodování nemělo vliv.

Na testovacích datech bylo bez neuronové sítě 47 ztrátových pozic a 81 pozic ziskových. S nasazením natrénovaného klasifikátoru pak bylo klasifikováno 71 zisků a 3 ztráty.

Kapitola 6

Závěr

Systém vykazuje předpokládané výsledky. Navržené moduly plní svůj účel. Předpokládanými výsledky je myšleno nejen hledání ziskových strategií, ale také fungující návrh obecného systému pro stochastický popis časových řad. Systém je navržen tak, že v případě potřeby je možné gramatiku s překladačem rozšířit o výpočty, které by mohly funkčnost dále vylepšit.

Systém jsem se snažil implementovat podle teoretického základu. Díky gramatice je možné velmi efektivně nastavit charakter výpočtů. Řetězec, který gramatiku reprezentuje, je vhodný formát pro reprezentaci. Jednak z důvodu, že si uživatel vytvoří rychlou představu o charakteru strategie, jak a nad čím je počítána. Druhým důvodem je jednoduchost, pokud bychom chtěli strategii dále automaticky zpracovávat. Každý modul je možné použít samostatně i pro jiné účely, pokud bychom systém dále rozšiřovali.

Modul neuronových sítí je schopen strategii průměrně zlepšit o 30 %. Správná generalizace problému je závislá na vypočítaném trénovacím vektoru. Z testů jsem vyvodil závěr, že existují výpočty, které obsah matice přepočítaly do normálního rozložení. Následná normalizace tak byla použita ve správném kontextu. Navržená architektura s nastavením z *testu 3* je připravena k nasazení, pokud nad systémem dodržíme správné řízení rizik a nalezenou strategii nejdříve zvalidujeme.

Podle teorie 3 jsem naplánoval možnosti dalšího rozšíření. V kapitole 3.3 je prezentována analýza autokorelace. Systém by byl schopný tuto analýzu provést se základními funkcemi. Ovšem výpočet autokorelace, případně korelace s jinými signály, bychom do překladače zavedli vedle technických indikátorů jako další heuristiku.

V kapitole je také uvedena souvislost 3, proč není možné neuronovou síť trénovat na kurzovních historických datech v původním formátu. Jejich rozložení není normální a ani nesplňuje centrální limitní větu. Podle testů opravdu není možné mít data v jiném než normálním rozložení. Aktivační funkce *tanh* a *sigmoida* modelují distribuční funkci normálního rozdělení. Je tedy nutná transformace dat do normálního rozložení. Vzhledem k tomu, že i data vypočítaná pomocí výpočtů z kapitoly 3.3.1 mají *heavy-tailed* normální rozložení, jsem navrhl další možnost, jak výsledky neuronových sítí dále zlepšit. Podle *centrální limitní věty* bych zvolil práh určující, které extrémy z trénovací sady vynechám.

Celkový přístup k používání systému by mohl být širší. Vzhledem k mobecnosti návrhu by bylo možné systém obohatit o další ohodnocovací funkce a na vstupu zpracovávat nejen kurzovní data, ale již vypočítané strategie za účelem jejich dalšího vylepšení. Například optimální ukončení pozice. Aktuální nastavení uvažuje pevné ukončení, což je pouze ten nejjednodušší způsob. Myšlenka iterativního využití již navrženého systému by přinesla nové

možnosti řízení pozic a zavedení lepšího řízení rizik. Například vytvořením ohodnocovací funkce, která bude zohledňovat celkový pokles a nárůst majetku po realizaci pozice.

Vzhledem k formátu, jakým je strategie popsána, je systém dále možné přes programové rozhraní nasadit přímo na server burzy.

Literatura

[1] Bishop, C.: *Pattern recognition and machine learning*. New York: Springer, 2006, ISBN 978-0387310732.

[2] Brabazon, A.; O'Neill, M.: Evolving technical trading rules for spot foreign-exchange markets using grammatical evolution. *Computational Management Science*, ročník 1, č. 3, Oct 2004: s. 311–327, ISSN 1619-6988, doi:10.1007/s10287-004-0018-5.
URL https://doi.org/10.1007/s10287-004-0018-5

[3] BROGAARD, J. A.: High Frequency Trading and its Impact on Market Quality. [online]. 2010, [cit. 2018-05-03] http://www.clasesdebolsa.com/archivos/HTF.pdf.

[4] Carter, J.: *Mastering the trade : proven techniques for profiting from intraday and swing trading setups*. New York: McGraw-Hill, 2012, ISBN 978-0071775144.

[5] Colin Reeves, J. E. R.: *@articlesimulovanezihani, publisher = Springer US, year = 2002, note = ISBN: 978-0-306-48050-8, owner = Springer Science+Business Media New York*.

[6] Coulling, A.: *Forex For Beginners*. Anna Coulling, 2013, ISBN 1494753758.

[7] Cox, J. C.; Ross, S. A.; Rubinstein, M.: *Option pricing: A simplified approach*. 1979.

[8] Ding, Y.; Cai, Y.; Sun, P.; aj.: The Use of Combined Neural Networks and Genetic Algorithms for Prediction of River Water Quality. *Journal of Applied Research and Technology*, ročník 12, č. 3, 2014: s. 493 – 499, ISSN 1665-6423, doi:https://doi.org/10.1016/S1665-6423(14)71629-3.
URL http://www.sciencedirect.com/science/article/pii/S1665642314716293

[9] Durenard, E. A.: *Professional Automated Trading: Theory and Practice 1st Edition*. 2013.

[10] EYDEN: *The Application of Neural Networks in the Forecasting of Share Prices*. Finance and Technology Publishing, 1996, ISBN 978-0965133203.

[11] Goldberg, D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989, iSBN:0201157675.

[12] Grimes, A.: *The art and science of technical analysis : market structure, price action, and trading strategies*. Hoboken: John Wiley & Sons, Inc, 2012, ISBN 978-1118115121.

[13] Hopcroft, J.; Motwani, R.; Ullman, J.: *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley, 2nd ed., 2000, iSBN: 0-201-44124-1.

[14] Huf, I. P.: *TEXFundamentální analýza numerických dat pro automatický trading*. FACULTY OF INFORMATION TECHNOLOGY DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA.

[15] Hull, C.: *TEXJ Options, Futures, and Other Derivatives. Seventh Edition*. Englewood Cliffs, 2009.

[16] Hynek, J.: *Genetické algoritmy a genetické programování*. Grada Publishing, 2008, ISBN 978-80-247-2695-3.

[17] J. Dupačová, J. H.; Štěpán., J.: *Stochastic Modeling in Economics and Finance, volume 75*. 2002.

[18] Kathy, L.: *FOREX – Ziskové intradenní a swingové obchodní strategie*. FXstreet.cz s.r.o., 2011, iSBN: 978-80-904418-2-8.

[19] KAČER, P.: FOREXOVÝ AUTOMATICKÝ OBCHODNÍ SYSTÉM ZALOŽENÝ NA NEURONOVÝCH SÍTÍCH. [online]. 2010.

[20] KAČER, P.: *FOREXOVÝ AUTOMATICKÝ OBCHODNÍ SYSTÉM ZALOŽENÝ NA NEURONOVÝCH SÍTÍCH*. Diplomová práce, VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, 2015, vedoucí: doc. Ing. VÁCLAV JIRSÍK, CSc. https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=99820.

[21] Laxpati, S. R.: *Practical signal processing and its applications : with solved homework problems*. Singapore Hackensack, NJ: World Scientific Publishing Co. Pte. Ltd, 2018, ISBN 978-9813224025.

[22] Mendes, L.; Godinho, P.; Dias, J.: A Forex trading system based on a genetic algorithm. *Journal of Heuristics*, ročník 18, č. 4, Aug 2012: s. 627–656, ISSN 1572-9397, doi:10.1007/s10732-012-9201-y.
URL https://doi.org/10.1007/s10732-012-9201-y

[23] MIKULENČÁK, R.: PREDIKCE KURSŮ PRO OBCHODOVÁNÍ NA AKCIOVÝCH TRZÍCH. 2011, vedoucí: Ing. Igor Szőke, Ph.D.

[24] MIKULENČÁK, R.: PREDIKCE KURSŮ PRO OBCHODOVÁNÍ NA AKCIOVÝCH TRZÍCH. 2011, vedoucí: Ing. Igor Szőke, Ph.D.

[25] MIKULENČÁK, R.: Predikce kursů pro obchodování na akciových trzích. Diplomová práce[online]. 2015, [cit. 2018-05-03] http://www.fit.vutbr.cz/study/DP/DP.php?id=17801&y=2014.

[26] NASSIRTOUSSI, S. W. T. Y., A. K.; AGHABOZORGI: Text mining for market prediction: A systematic review. Expert Systems with Applications. 2014, [cit. 2018-05-03].

[27] Olah, C.: Understanding LSTM Networks. 2014, [cit. 2018-05-03] https://ssrn.com/abstract=1858626.

[28] Sebastiani, F.: Machine Learning in Automated Text Categorization. *ACM Comput. Surv.*, ročník 34, č. 1, Březen 2002: s. 1–47, ISSN 0360-0300, doi:10.1145/505282.505283.
URL http://doi.acm.org/10.1145/505282.505283

[29] Sekanina Lukáš, P., prof. Ing.: Evoluční design, Biologií inspirované počítače 3. [cit. 2018-04-20] http://www.fit.vutbr.cz/study/course-l.php.cs?id=12653.

[30] SHARPE, M. J.: LOGNORMAL MODEL FOR STOCK PRICES [online]. MATHEMATICS DEPARTMENT, UCSD, [cit. 2016-01-18] http://math.ucsd.edu/~msharpe/stockgrowth.pdf.

[31] Silva, A.: *Grammar-based feature generation for time-series prediction*. Singapore: Springer, 2015, ISBN 978-9812874108.

[32] Sprott, J.: *Chaos and time-series analysis*. Oxford New York: Oxford University Press, 2003, ISBN 978-0198508403.

[33] Tsay, R.: *Analysis of financial time series*. Hoboken, N.J: Wiley, 2010, ISBN 978-0470414354.

[34] VESKA, T.: *TEXKalibrace stromů úrokových měr a ocenění úrokových opcí. Praha, 2017. Bakalářská práce*. Univerzita Karlova, Matematicko-fyzikální fakulta, Katedra pravděpodobnosti a matematické statistiky. Vedoucí práce Witzany, Jiří.

[35] Yann, L.; Leon, B.; Genevieve, B. O.; aj.: Efficient BackProp [online]. Dec 1998, [cit. 2016-01-5] http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf.

[36] Zhang, G.; Patuwo, B. E.; Hu, M. Y.: Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting*, ročník 14, č. 1, 1998: s. 35 – 62, ISSN 0169-2070, doi:https://doi.org/10.1016/S0169-2070(97)00044-7.
URL http://www.sciencedirect.com/science/article/pii/S0169207097000447

Příloha A

Obsah přiloženého paměťového média

Obsahem CD jsou zdrojové kódy v jazyce python ve složce *dp-code*. Složka *dp-code* také obsahuje příklad testovacích dat *eurusd15.csv* a *eurusd153.csv* a složku *debug*, která ukládá objekt nalezené strategie. Složka *conda-env* obsahuje prostředí Anaconda, ve kterém byl systém testován. Obsahem je také demonstrační video a plakát.

Příloha B

Manuál

Program je postaven na jazyce Python 3. Využívá několik knihoven pro práci s daty, které je nutné instalovat:

ScyPy - programový balíček pro práci s daty

Pandas - knihovna poskytující operace a výpočty s maticí

Numpy - knihovna poskytující operace a výpočty s maticí

Matplotlib - knihovna pro zobrazení dat

Tensorflow - backend pro simulaci neuronové síťě

Keras - nadstavba nad knihovnou Tensorflow

Deap - knihovna pro práci s genetickými algoritmy

sklearn - knihovna pro PCA

Knihovna *pickle* by měla být nainstalována s*numpy*, pokud neni, je třeba jí také doinstalovat.Stejně tak knihovna *operator* by měla být v základu. Další knihovny by měly být v základu Python 3. Pokud nebudou, je také potřeba je doinstalovat. Na médiu je přiložené prostředí *Anaconda* ve kterém byla práce testována. Prostředí je možné nainstalovat.

Script *GAengine.py* slouží pro spustění systému v různých módech. Pro výpočet pozic optimalizované strategie je možné spustit *GAengine.py -o evolution-profit*. Pro test strategie je určený příkaz *GAengine.py -o test-ind*. Pro test strategie s natrénovanou neuronovou sítí slouží příkaz *GAengine.py -o result-ind-nnet*. Pokud bychom chtěli spustit samotnou optimalizaci, pak stačí spustit script bez příkazu *GAengine.py*. Optimalizace bude spuštěna s počáteční generací o velikosti 1000.