



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

FRAMEWORK PRO BACKTESTOVÁNÍ STRATEGIÍ ALGORITMICKÉHO OBCHODOVÁNÍ NA BURZE VČETNĚ PODPORY PRO VYLEPŠOVÁNÍ STRATEGIÍ S POMOCÍ EVOLUČNÍCH ALGORITMŮ.

FRAMEWORK FOR BACKTESTING OF ALGORITHMIC TRADING INCLUDING THE STRATEGY

IMPROVEMENT USING THE EVOLUTIONARY ALGORITHMS.

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MARTIN KMENTA

VEDOUcí PRÁCE

SUPERVISOR

Ing. IGOR SZÓKE, Ph.D.

BRNO 2023

Zadání diplomové práce



156541

Ústav: Ústav počítačové grafiky a multimédií (UPGM)
Student: **Kmenta Martin, Bc.**
Program: Informační technologie a umělá inteligence
Specializace: Softwarové inženýrství
Název: **Framework pro backtestování strategií algoritmického obchodování na burze včetně podpory pro vylepšování strategií s pomocí evolučních algoritmů.**
Kategorie: Zpracování signálů
Akademický rok: 2023/24

Zadání:

1. Prostudujte dostupné techniky technické analýzy. Seznamte se s teorií evolučních algoritmů. Určete relevantní data ze směnárů pro sběr a následný backtesting.
2. Prozkoumejte možnosti získávání dat u jednotlivých směnárů, získejte a zpracujte tato data. Vytvořte vizualizační nástroj pro jejich zobrazení.
3. Vytvořte systém pro zápis strategií a indikátorů vycházejících ze zpracovaných dat. Implementujte indikátory a strategie použitelné pro backtesting.
4. Navrhněte a implementujte nástroj pro backtestování strategií na získaných datech a vylepšování parametrů indikátorů ve strategiích s pomocí evolučních algoritmů.
5. Zhodnoťte výsledky a navrhněte směry dalšího vývoje.
6. Vytvořte A2 plakátek a cca 30 vteřinové video prezentující výsledky vaší práce.

Literatura:

- Larry Harris, "Market Microstructure for Practitioners", 2002, ISBN: 9780195144703
- Schwager Jack, "Getting Started in Technical Analysis", 1999, John Wiley & Sons Inc, ISBN: 9780471295426
- Anthony Brabazon, Michael O'Neill, "Biologically Inspired Algorithms for Financial Modelling", 2006, ISBN: 9783540262527
- Dále dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:
Body 1 až 3 ze zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Szőke Igor, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, prof. Dr. Ing.
Datum zadání: 1.11.2023
Termín pro odevzdání: 17.5.2024
Datum schválení: 9.11.2023

Abstrakt

Tato diplomová práce se soustředí na vývoj pokročilého frameworku pro backtestování algoritmických obchodních strategií, přičemž klade důraz na optimalizaci strategií pomocí evolučních algoritmů. Zabývá se analýzou a aplikací technické analýzy v kontextu obchodování na burze. Dále se zaměřuje na návrh a vývoj modulů pro efektivní získávání, zpracování, vizualizaci a analýzu různých typů tržních dat, což umožňuje uživatelům vytvářet a backtestovat své vlastní indikátory a obchodní strategie s využitím robustního frameworku.

Abstract

This thesis focuses on the development of an advanced framework for backtesting algorithmic trading strategies, emphasizing the optimization of strategies using evolutionary algorithms. It deals with the analysis and application of technical analysis in the trading context. It also focuses on the design and development of modules for efficient retrieving, processing, visualization, and analysis of various types of market data, allowing users to create and backtest their indicators and trading strategies using a robust framework.

Klíčová slova

Burza, framework, algoritmické obchodování, obchodní strategie, technická analýza, evoluční algoritmy, optimalizace parametrů

Keywords

Exchange, framework, algorithmic trading, trading strategy, technical analysis, evolutionary algorithms, parameter optimization

Citace

KMENTA, Martin. *Framework pro backtestování strategií algoritmického obchodování na burze včetně podpory pro vylepšování strategií s pomocí evolučních algoritmů..* Brno, 2023. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Igor Szóke, Ph.D.

Framework pro backtestování strategií algoritmickeho obchodování na burze včetně podpory pro vylepšování strategií s pomocí evolučních algoritmů.

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Igora Szóke, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Martin Kmenta
13. května 2024

Poděkování

Děkuji panu Ing. Igorovi Szóke, Ph.D. za vedení a odbornou pomoc v průběhu řešení této diplomové práce.

Obsah

1	Úvod	4
1.1	Algoritmické obchodování	4
1.2	Motivace	4
1.3	Dosažené výsledky	5
1.4	Struktura práce	5
1.5	Dodatek	6
2	Analýza obchodování na burze	7
2.1	Obchodování na burze	7
2.1.1	Jak fungují trhy	7
2.1.2	Identifikace klíčových účastníků trhu	8
2.1.3	Terminologie	8
2.1.4	Finanční nástroje	9
2.2	Burzy a směnárny	10
2.2.1	Obchodní hodiny	11
2.2.2	Makléři (Brokers)	12
2.2.3	Data ze směnáren	12
2.2.4	Typy transakcí	15
2.2.5	Poplatky a další náklady	15
2.3	Typy obchodních strategií	16
2.4	Analýza trhu	16
2.4.1	Technická analýza (TA)	17
2.4.2	Indikátory pro TA	17
2.5	Shrnutí v kontextu s cílem práce	20
2.5.1	Finanční nástroje	20
2.5.2	Potřebná data	20
2.5.3	Zpracování dat	20
2.5.4	Vizualizace dat	20
2.5.5	Indikátory	20
2.5.6	Reprezentace strategií	21
3	Analýza existujících řešení	22
3.1	Open-source frameworky	22
3.1.1	Freqtrade	22
3.1.2	Backtrader	23
3.2	Platformy	24
3.3	Knihovny pro technickou analýzu	24
3.3.1	TA-Lib	24

3.3.2	Technical Analysis Library in Python	25
3.3.3	Technical Analysis (ta.py)	25
3.3.4	Shrnutí analýzy knihoven pro TA	25
3.4	Další užitečné knihovny	25
3.5	Shrnutí v kontextu s cílem práce	26
3.5.1	Možnosti integrace	26
3.5.2	Testování a validace	26
3.5.3	Inspirace v backtestingu	26
3.5.4	Vyhodnocování strategií	26
3.5.5	Konkurenceschopnost	27
4	Evoluční algoritmy (EA)	28
4.1	Co jsou EA	28
4.1.1	Reprezentace jedinců	29
4.1.2	Velikost populace a počet potomků	29
4.1.3	Selekce a výpočet zdatnosti (fitness)	30
4.1.4	Křížení a mutace	30
4.1.5	Ukončovací podmínka	30
4.2	Typy EA	31
4.3	Shrnutí v kontextu s cílem práce	31
4.3.1	Vylepšování parametrů strategií	31
4.3.2	Fitness funkce strategie	32
5	Návrh a architektura frameworku	33
5.1	Volba programovacího jazyka	34
5.2	Stahování dat	34
5.3	Zpracování dat	36
5.4	Ukládání dat	37
5.5	Zobrazování dat	38
5.6	Backtesting	40
5.6.1	Indikátory	41
5.6.2	Strategie a jejich reprezentace	41
5.6.3	Virtuální směnárna	43
5.6.4	Data v simulaci	45
5.6.5	Návrh modulu	46
5.6.6	Vlastní backtesting	49
5.7	Vylepšování parametrů strategií s pomocí evolučních algoritmů	49
5.7.1	Parametry	50
5.7.2	Genetický algoritmus	51
5.7.3	Fitness metoda	52
6	Testování, experimenty a validace	56
6.1	Testování stahování dat	56
6.2	Testování, zpracování a ukládání dat	59
6.3	Validace dat pomocí vizualizace	62
6.4	Testování funkcionalit Backtestovacího modulu	65
6.5	Testování vlastního backtestingu	68
6.6	Porovnání výsledku s Freqtrade frameworkem	68

6.7	Experimenty s optimalizací parametrů strategií	69
6.8	Experiment se zpožděním reakce směnární	73
7	Závěr	74
7.1	Návrhy na vylepšení frameworku	75
	Literatura	77
A	Další indikátory pro TA	78
A.1	Klouzavý průměr konvergence divergence (MACD)	78
A.2	Index relativní síly (RSI)	78
A.3	Swingové body	80
B	Tabulka směnáren	82
C	Strategie vygenerované pomocí generativních modelů	86
C.1	Instrukce pro modely:	86
C.2	Odpověď od modelu GPT-3.5	86
C.3	Odpověď od modelu claude-3-opus-20240229	87
C.4	Odpověď od modelu GPT-4	88
C.5	Odpověď od modelu Gemini Pro	89
C.6	Odpověď od modelu Gemini 1.5 Pro	89
D	Ukázka Freqtrade backtestingu	93
E	Podklady na konferenci Excel@FIT 2024	95

Kapitola 1

Úvod

Tato diplomová práce se soustředí na vývoj pokročilého frameworku pro backtestování algoritmických obchodních strategií a jejich optimalizace. Zabývá se analýzou a aplikací technické analýzy v kontextu obchodování na burze, přičemž se snaží detekovat co nejvíce faktorů ovlivňující výkonnost strategií. Práce obsahuje rozsáhlý průzkum existujících řešení za účelem zjištění konkurenceschopnosti frameworku, možné inspirace a případné integrace těchto řešení. Taktéž se snaží o efektivní optimalizace obchodních strategií převážně s pomocí evolučních algoritmů.

Dále se práce zaměřuje na samotný návrh a implementaci jednotlivých modulů pro efektivní získávání, zpracování, vizualizaci a analýzu různých typů tržních dat stejně jako o implementaci optimalizačního modulu se snahou vytvořit jednoduchý a snadno rozšiřitelný framework.

1.1 Algoritmické obchodování

Algoritmické obchodování, často označované jako automatizované obchodování, je charakterizováno využitím algoritmů a matematických modelů pro vytváření a realizaci obchodních příkazů na základě obchodních strategií. Tento přístup umožňuje rychlou a efektivní reakci na tržní změny, minimalizaci chyb způsobených lidským faktorem a využití složitých strategií, které by pro člověka byly obtížně realizovatelné.

Backtestování je klíčový proces, při kterém se používají historická data k ověření efektivity a výkonnosti obchodních strategií. Je to zásadní součást algoritmického obchodování, protože umožňuje obchodníkům simulovat jejich strategie a poskytuje důležité informace pro jejich optimalizaci a vylepšení. Tímto způsobem mohou obchodníci identifikovat potenciální slabiny ve svých obchodních strategiích a následně strategie přizpůsobit, aby více vyhovovaly jejich preferencím.

1.2 Motivace

Hlavním důvodem vzniku této práce byla neexistence veřejně dostupného komplexního přístupu k backtestování algoritmických obchodních strategií za použití nezkraslených dat. Všechny prozkoumané dostupné nástroje zkraslují výsledky samotného backtestingu tím, že používají nejčastěji vstupní data ve formátu OHLCV¹. Tento přístup neumožňuje pra-

¹OHLCV (zkratka pro Open, High, Low, Close, Volume) představuje základní informace o cenové aktivitě za určité časové období a je podrobněji popsán v podsekcí [2.2.3](#)

covat s jednotlivými transakcemi a ochuzuje analýzu o cenné informace, například o poměr objemu nákupu a prodeje, seskupením jednotlivých transakcí do jedné informace. Sice umožňuje pracovat s informacemi seskupenými po jednotlivých vteřinách, avšak například pro Skalповé obchodování², kde transakce mohou trvat jednotky vteřin, je tento přístup příliš omezující.

Dalším důvodem byly optimalizace strategií. Dostupné frameworky a platformy sice nabízí různé optimalizace parametrů, ať už jednoduchým prohledáváním všech permutací, nebo sofistikovanějšími metodami strojového učení a biologicky inspirovanými algoritmy. Tato řešení mají ovšem vždy nemalé limity kvůli omezením samotných nástrojů.

Cílem diplomové práce bylo vyvinutí pokročilého frameworku, který eliminuje zmíněná omezení a je konkurenceschopný s existujícími řešeními, ať už z pohledu funkcionality, uživatelské přívětivosti nebo rozšiřitelnosti.

1.3 Dosažené výsledky

Podařilo se implementovat framework, který splňuje veškeré stanovené požadavky. Přesněji bylo implementováno šest spolu spolupracujících modulů, kde se každý stará o jednu z těchto funkcionalit: získávání dat, zpracování dat, správu úložiště, vizualizaci, backtestování strategií a optimalizaci parametrů strategie. Celý framework je možné ovládat pomocí příkazové řádky nebo prostřednictvím instancí tříd přímo z Python kódu.

Backtestování probíhá formou simulace reálného prostředí, do kterého je zahrnuta směnárna a obchodníci. Během simulace jsou vyhodnocovány historické transakce jedna po druhé, což mnohonásobně zvyšuje přesnost backtestingu vůči všem veřejně dostupným nástrojům. Každý obchodník žádá směrnaru o vytváření pozic³ na základě vyhodnocení dostupných dat. Také žádá směrnaru o další manipulace s vytvořenými pozicemi, jako je například jejich předčasné zrušení. Virtuální směnárna přijímá požadavky od obchodníků a vytváří, nebo manipuluje s pozicemi, které v průběhu času také uzavírá podle jejich parametrů.

Kvůli časové náročnosti simulace a nemožnosti projít množinu všech možných permutací parametrů komplexnějších strategií v reálném čase, byl vytvořen modul pro učení parametrů s pomocí genetického algoritmu. Framework byl testován a ověřován. Při experimentování se podařilo optimalizovat parametry několika strategií. Efektivnost optimalizace závisela na účinnosti poskytnuté strategie, rozsahu dat, počtu iterací a nastavení algoritmu. Také byly navrženy další zajímavé možnosti pro následný vývoj frameworku.

1.4 Struktura práce

Práce zachycuje postup vývoje frameworku od samotné analýzy tématu až po testování jeho schopností. Text je strukturován následujícím způsobem: Kapitola 2 obsahuje detailní popis tématu obchodování na burze včetně důkladné analýzy faktorů ovlivňujících úspěšnost strategií. Následující kapitola 3 popisuje existující řešení a shrnuje získané informace ovlivňující vývoj frameworku a jeho konkurenceschopnost. Kapitola 4 vysvětlí podstatu Evolučních algoritmů a popíše možnosti jejich využití pro tuto práci. Následuje kapitola 5

²Skalповé obchodování je obchodní strategie, která využívá krátkodobého vstupu a výstupu z pozic k zisku z malých cenových pohybů na trhu a je podrobněji popsán v sekci 2.3

³Pozicí je myšleno spekulace na pohyb cen aktiva, bez nutnosti jeho vlastnictví. Jak to funguje, je vysvětleno v podsekcí 2.1.4.

o analýze a návrhu frameworku. Poslední obsáhlá kapitola 6 je o testování simulace, validaci její funkčnosti a různých experimentech s optimalizací strategií. Práce končí kapitolou 7 shrnující dosažené výsledky a popisující možnosti vylepšení frameworku.

1.5 Dodatek

V průběhu vývoje tohoto frameworku a psaní textu práce bylo využito řady nástrojů, které napomohly ke zefektivnění a zjednodušení práce. Specificky, během programování byl často využíván nástroj `GitHub Copilot`⁴, jenž poskytl asistenci a přispěl ke zvýšení produktivity v psaní zdrojového kódu.

Také byly využity nástroje: `ChatGPT`⁵ (GPT-3.5, GPT-4), `Claude`⁶ (Claude 3 Opus), `Gemini`⁷ (Gemini Pro, Gemini 1.5 Pro) a `Bing AI Chat`⁸ pro získávání specifických informací týkajících se problémů, které se v průběhu práce objevovaly. Tyto informace byly následně dále ověřovány pomocí důvěryhodných zdrojů. Příkladem užití může být generování algoritmických obchodních strategií k následným experimentům s frameworkem.

Při formulaci textu byly tyto nástroje využity k vylepšení textu a opravě gramatických chyb. Je důležité zdůraznit, že tyto nástroje byly využívány jako pomocníci a podpora vlastní práce, nikoli jako její primární autoři.

⁴<https://github.com/features/copilot>

⁵<https://chat.openai.com/>

⁶<https://claude.ai/>

⁷<https://gemini.google.com/>

⁸<https://www.bing.com/chat>

Kapitola 2

Analýza obchodování na burze

Burza je organizovaný trh, kde se obchoduje s širokým spektrem finančních nástrojů. Studium a pochopení fungování burzy a věcí s tím spojených je prvním cílem této práce. To se ovšem neobejde bez seznámení se s tématem a jednotlivými pojmy.

V této kapitole je prozkoumáno prostředí obchodování na burze. Nejprve jsou analyzovány základní principy a mechanismy, které řídí trhy. Dále je poskytnut podrobný přehled různých finančních nástrojů. Zkoumání, jak jsou trhy ovlivňovány nabídkou a poptávkou a jak reagují na rozmanité různé události, je taktéž součástí této části.

Další část je zaměřena na směnárny a obchodní platformy, které představují zásadní prvky infrastruktury umožňující obchodování. Jsou popsány podstatné informace, jako například obchodní hodiny, data poskytovaná směnárnami, typy transakcí a poplatky, které je potřeba znát pro pochopení kontextu celé této práce.

Kapitola taktéž obsahuje detailní pohled na analýzu trhu, kde je zdůrazněna technická analýza. Technická analýza poskytne přehled o cenových grafech, trendech a různých indikátorech, které mohou být využity k posílení obchodních strategií.

Na konci se pak nachází shrnutí informací z této kapitoly a jejich uvedení do kontextu této práce spolu s upřesněním některých cílů.

2.1 Obchodování na burze

Obchodování na burze zahrnuje nákup a prodej finančních nástrojů. Cílem je spekulace na cenové pohyby nebo zajištění proti finančním rizikům. Trhy poskytují likviditu a transparentnost cen.

2.1.1 Jak fungují trhy

Trhy fungují na základě nabídky a poptávky, kde obchodníci obchodují s různými finančními nástroji. Cena těchto nástrojů je určena vzájemnou interakcí kupujících a prodávajících. Ceny jsou také ovlivněny nejrůznějšími událostmi, ať už ekonomickými, politickými, či událostmi spojenými s výsledky firem. Bohužel je velké množství prvků ovlivňující vývoj trhu, které se kvůli komplexnosti a neustálému vývoji nedají jednoduše zmapovat, a tak je nemožné přesně předpovídat vývoj trhu.

2.1.2 Identifikace klíčových účastníků trhu

Na trzích vystupuje mnoho účastníků, kteří se dělí na makléře, obchodníky, institucionální investory, market makery a regulační orgány. **Makléři** (Brokers) umožňují obchodníkům přístup na trhy a zajišťují provedení obchodů. **Obchodníci** obchodují s různými finančními nástroji, může se jednat o jednotlivce nebo instituce. **Institucionální investoři** (Institutional Investors) jsou pak velké organizace jako penzijní fondy, pojišťovny a investiční fondy. **Market Makers** jsou firmy nebo jednotlivci, kteří udržují likviditu trhu tím, že jsou stále připraveni koupit nebo prodat určitá aktiva. **Regulační orgány** dohlížejí na spravedlnost a transparentnost trhu [8].

2.1.3 Terminologie

V oblasti obchodování je zásadní terminologie pro jasnou komunikaci. V této části jsou uvedeny definice klíčových pojmů, které budou v této práci často používány.

- **Aktivum** je v obchodování jakýkoli zdroj s ekonomickou hodnotou, který lze vlastnit nebo s ním lze obchodovat. Patří sem například akcie, komodity, dluhopisy, měny.
- **Leverage** (Pákový efekt) v obchodování znamená použití vypůjčených prostředků ke zvýšení potenciálního výnosu pozice. Sice může zvýšit zisky, ale také zvyšuje možnost výrazných ztrát a vzniká riziko případné likvidace celého účtu.
- **Volatilita** označuje míru kolísání ceny finančního nástroje v čase. Vysoká volatilita naznačuje, že se cena může během krátké doby dramaticky změnit na oba směry.
- **Spread** je rozpětí a označuje rozdíl mezi nákupní a prodejní cenou aktiva. Představuje náklady na obchodování s tímto aktivem a je rozhodujícím faktorem ziskovosti krátkodobých obchodů. (Na směnárnách často nezahrnuje poplatky.)
- **Likvidita** je ukazatel vypovídající o tom, jak moc velké pozice lze na trhu vytvářet a jak rychle aniž to ovlivnilo jeho cenu. Vysoká likvidita je často spojena s nižším spreadem a menšími skoky ceny.
- **Slippage** (skluz) je pojem označující sklouznutí ceny na jinou cenu, než bylo očekáváno, při realizaci transakce. Ke Slippage může dojít například při vytvoření pozice, která překračuje dostupné prostředky na trhu za aktuální cenu, a proto je pozice naplněna za horší cenu, než byla očekávána.
- **Take Profit** slouží k braní profitu. Jedná se o pokyn k uzavření otevřené pozice se ziskem, jakmile aktivum dosáhne určité cenové úrovně. Tuto úroveň lze poslat na některé směnárny spolu s dalšími informacemi pro automatické braní zisku. V závislosti na směnárně a jejímu vyhodnocování se můžou k Take Profitu vztahovat různé poplatky.
- **Stop Loss** funguje obdobně jako Take Profit s rozdílem, že omezuje ztráty. Úroveň Stop Lossu lze poslat na některé směnárny spolu s dalšími informacemi pro automatické uzavření otevřené pozice pro předejití větších ztrát. Také jako u Take Profit se může jednat o různý typ pozice s různými poplatky a může dojít ke Slippage.
- **Trailing Stop Loss** je speciální typ Stop Lossu, který nastavuje úroveň pro uzavření pozice na určitou částku pod tržní cenou. Například u dlouhé pozice s růstem ceny Trailing Stop zvyšuje svoji úroveň, ale pokud cena klesá, úroveň se nemění.

- **Trend** v obchodování označuje směr, kterým se pohybuje cena finančního nástroje. Trendy mohou být vzestupné (Bullish Trend), sestupné (Bearish Trend) nebo boční / neutrální (Sideways Trend / Flat Trend).

S trendem se také pojí označení pro různá období trhu. Je možné detekovat býčí trh (Bull Market), medvědí trh (Bear Market) a konsolidační trh (Consolidation Market). Býčí trh označuje období, kdy vybrané finanční trhy jako celek rostou a očekává se pokračování pozitivního trendu. Medvědí trh je opakem býčího a je obvykle spojen s pesimismem a očekáváním ekonomického poklesu. V konsolidačním trhu ceny finančních nástrojů kolísají v určitém cenovém rozmezí bez výrazného směru nahoru nebo dolů.

- **Support** je pojem označující cenovou úroveň, kde je pohyb ceny směrem dolů zastaven nebo zbrzděn koncentrací likvidity v této úrovni. Většinou za tvorbu takové úrovně mohou početně nastavené Take Profity a Stop Lossy.
- **Resistance** je doba supportu a označuje cenovou úroveň, která ovlivňuje cenový pohyb směrem nahoru.

2.1.4 Finanční nástroje

Je celá řada finančních nástrojů, se kterými se obchoduje. Každý z těchto nástrojů má své specifické vlastnosti a rizika. Některé z nich jsou: [6, 7, 8, 12]

- **Akcie** (Stocks), které představují vlastnický podíl ve společnosti. Držitel akcií má nárok na část zisku společnosti a hlasovací práva v některých otázkách. Příkladem mohou být akcie společností Apple, Inc. (AAPL) a Tesla, Inc. (TSLA).
- **Kryptoměny** (Cryptocurrencies), jako je Bitcoin nebo Ethereum, jsou digitální nebo virtuální měny používající kryptografii pro zabezpečení transakcí.
- **Dluhopisy** (Bonds), tyto dluhové cenné papíry vydávají společnosti nebo vlády za účelem získání kapitálu. Obchodník, kupující dluhopis, půjčuje své peníze za cílem získání pravidelných úrokových plateb.
- **Devizové produkty** (Forex - Foreign Exchange), kde je zahrnut nákup a prodej měn. V závislosti na pohybech směnných kurzů, kde jedna měna získává na hodnotě oproti jiné, se pak dá profitovat. Tyto páry se pak dále dělí do tří hlavních kategorií: hlavní, menší a exotické.
 - **Hlavní měnové páry** (Majors) zahrnují nejsilnější světové měny a mají největší objem obchodů. Obsahují americký dolar (USD) a další silnou měnu, například: EURUSD (euro a americký dolar).
 - **Menší měnové páry** (Minors/Crosses) jsou tvořeny taktéž silnými měnami, ale nezahrnují americký dolar (USD). Příkladem může být: EURGBP (euro a britská libra).
 - **Exotické měnové páry** (Exotics) obsahují jednu hlavní měnu mající menší likviditu a větší spread jako například USDINR (americký dolar a indická rupie).
- **Indexy** (Indices), které jsou ukazatelé výkonnosti skupin aktiv, jako jsou akciové indexy. Populárním indexem je S&P 500 zahrnující 500 nejvýznamnějších světových

společností obchodujících na amerických burzách. Přímé obchodování s indexy není na burze možné. Jedná se pouze o statistické ukazatele. Lze ale obchodovat s **Indexovými fondy** (ETFs), kde příkladem, kopírující index S&P 500, je SPDR S&P 500 ETF Trust.

- **Komodity** (Commodities), což jsou fyzické suroviny jako je zlato a ropa.
- **Deriváty** (Derivatives), které jsou finanční nástroje, jejichž hodnota je odvozena od hodnoty jiného aktiva nebo indexu. Zahrnují různé typy kontraktů:
 - **Opce** (Options) jsou smlouvy, které dávají kupujícímu právo, ale ne povinnost koupit (call opce) nebo prodat (put opce) určité množství aktiva za předem stanovenou cenu do určitého data. Opce se používají pro zajištění proti pohybům cen.
 - **Termínové kontrakty** (Futures) jsou kontrakty, které zavazují kupujícího koupit a prodávajícího prodat specifikované množství aktiva za předem stanovenou cenu k určitému budoucímu datu.
 - **Perpetual Contracts** (Smlouvy na dobu neurčitou) jsou podobné termínovým kontraktům, ale nemají expirační datum.
 - **Swapové kontrakty** (Swaps) jsou smlouvy, v rámci kterých si dvě strany vyměňují peněžní toky nebo finanční nástroje. Nejčastějšími typy jsou úrokové swapy a měnové swapy.
 - **CFD** (kontrakty na rozdíl) (Contracts for Difference) umožňují obchodníkům spekulovat na pohyb cen různých aktiv, bez potřeby vlastnictví samotného podkladového aktiva. CFD reflektují vývoj cen podkladového aktiva, což znamená, že zisky nebo ztráty jsou založeny na rozdílu ceny aktiva mezi otevřením a uzavřením obchodu.

Pro účely této práce budeme považovat CFD za klíčový typ finančního nástroje, neboť se zaměřuje pouze na spekulaci o růstu či poklesu cen, aniž by bylo nutné řešit vlastnictví samotných aktiv. Jinými slovy, obchodník nemusí řešit, zda má co prodávat, jelikož obchodování s CFD se točí kolem sázek na cenové změny, nikoli kolem fyzického držení aktiv.

2.2 Burzy a směnárny

Burzy mohou to být fyzická místa jako například NYSE (New York Stock Exchange), nebo elektronické platformy, jako je NASDAQ (National Association of Securities Dealers Automated Quotations). Nabízejí obchodování s širokou škálou finančních nástrojů. Na burze se shromažďují kupující a prodávající k obchodování s různými druhy finančních nástrojů. Poskytují infrastrukturu pro provedení obchodů mezi různými účastníky trhu.

V oblasti tradingu se pojmem **směnárna** často myslí kryptoměnová směnárna, jako je Binance a Bybit. V této práci je pojem směnárna brán jako kryptoměnová směnárna. Tyto směnárny umožňují uživatelům obchodovat s převážně různými kryptoměnami. Některé směnárny také nabízí deriváty na kryptoměny, například opce, termínové kontrakty a CFD.

Hlavním důvodem, proč se tato práce zabývá směnárnami, a ne jinými burzami, jsou data. Ke všem finančním nástrojům existují poskytovatelé dat, avšak ne vždy jsou data dostupná v dostatečné kvalitě. Také získání dat se dost často neobejde bez poplatku a případné

smlouvy. Během zkoumání různých poskytovatelů dat se podařilo najít několik směnár, které poskytují vlastní data o obchodování na těchto směnárnách. Vzhledem k dostupnosti dat je tato práce postavena na datech ze směnár, avšak nebude tímto faktorem nijak limitována pro následné rozšíření [8].

2.2.1 Obchodní hodiny

Obchodní hodiny jsou závislé na typu burzy, typu obchodovaného finančního nástroje a geografická lokaci. Některé obchodní hodiny mají globální dopad na více finančních nástrojů než jen na ty, na které se vztahují, a to zejména na začátku a na konci těchto hodin. Příkladem mohou být obchodní hodiny pro Forex [6, 7].

Forex seance (Sessions)

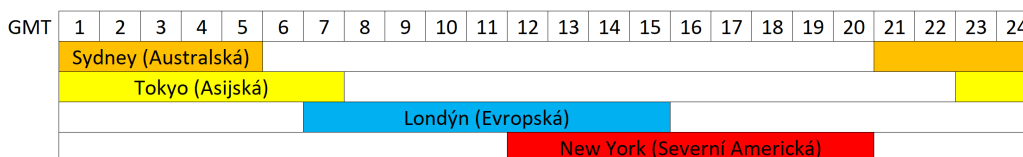
Forexový trh je jedinečný svou vysokou likviditou a rozsahem, a také tím, že je obchodován nepřetržitě 24 hodin denně, 5 dní v týdnu. Tato charakteristika umožňuje obchodníkům vstupovat na trh kdykoliv během dne. Například webová stránka Market24hClock¹ má zajímavou vizualizaci obchodních hodin pro Forex.

Máme několik seancí, které shromažďují jednotlivé obchodní hodiny do skupin za účelem zjednodušení. Tyto seance nejsou oficiálně stanovené. Existuje mnoho zdrojů, které tyto seance popisují, avšak časová rozmezí se liší o pár hodin.

Tyto seance se dají používat v analýze trhu. Během obchodních seancí je typicky vyšší likvidita, protože je mnoho obchodníků aktivních současně. S vyšší likviditou se pojí i větší objem obchodů, což může znamenat menší spread. Některé seance mohou přinést větší volatilitu, zejména když se překrývají, jako například evropská a americká seance. Také ekonomické události a zprávy zveřejněné v různých částech světa mohou mít výrazný dopad na trhy během příslušných obchodních seancí.

Pro demonstraci, jak takové seance na Forexu vypadají, můžeme stanovit jejich hodiny následovně. Jejich vizuální zobrazení je pak na obrázku 2.1 [6, 7].

- **Australská** Seance: 21:00 GMT - 6:00 GMT.
- **Asijská** Seance: 23:00 GMT - 8:00 GMT.
- **Evropská** Seance: 7:00 GMT - 16:00 GMT.
- **Americká** Seance: 12:00 GMT - 21:00 GMT.



Obrázek 2.1: Vizualizace obchodních hodin seancí na Forexu v GMT zóně

¹<https://market24hclock.com>

2.2.2 Makléři (Brokers)

Makléři poskytují obchodníkům přístup k různým trhům, které mohou být reprezentovány různými směnárnami. Směnárna je místo, kde se skutečně uskutečňují obchody, zatímco broker zprostředkovává přístup k této směnárně pro své klienty.

Obchodní platformy

Většina obchodních platform patří makléřům, nebo je jimi provozována. Tyto platformy jsou primárním prostředkem, jakým makléři umožňují obchodníkům přístup k finančním trhům a provádění obchodů.

Někteří makléři vyvíjejí vlastní platformy, aby se odlišili od konkurence a poskytli specifickou funkcionalitu nebo lepší uživatelské rozhraní, které odpovídá potřebám jejich klientů. Tyto platformy pak mohou být specializované pro manuální obchodování a provádění rozsáhlých technických analýz nebo pro jednoduchou integraci v rámci algoritmického obchodování.

2.2.3 Data ze směnáren

Každá směnárna poskytuje klientům určitá data. Obsáhlost těchto dat, stejně jako jejich kvalita, je ovlivňována různými faktory. Většina směnáren poskytuje základní OHLC, případně OHLCV pro různé časové rámce. Z těchto dat již jde dělat běžná technická analýza, která pro účely této práce nestačí. Existuje i několik směnáren, převážně větších, které poskytují data v podobě jednotlivých transakcí, která jsou pro tuto práci klíčové.

Typy dat

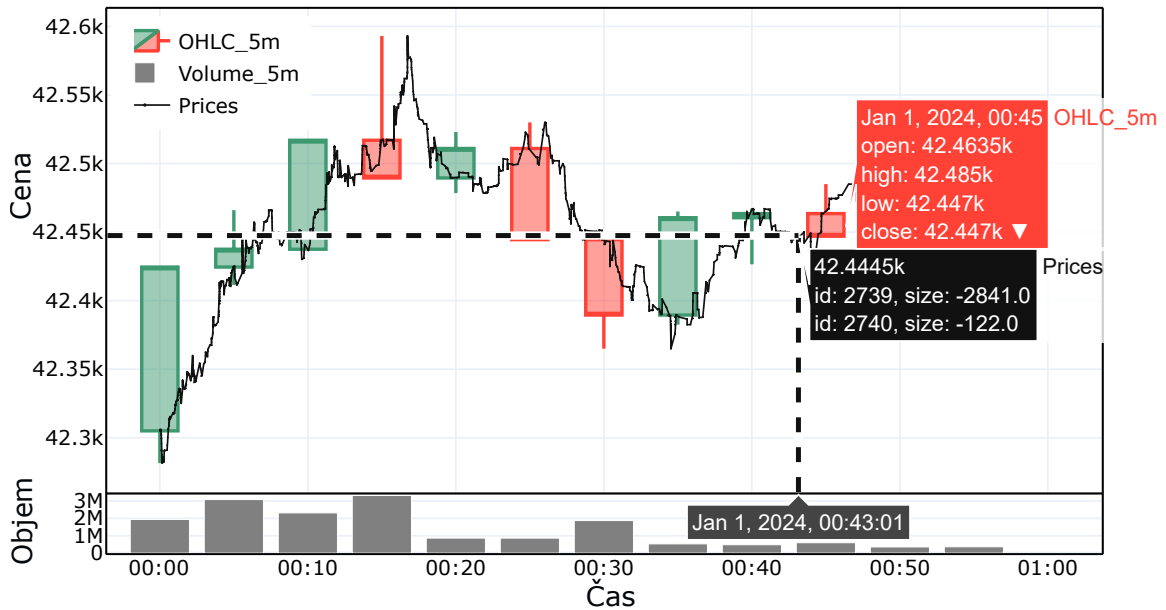
Následuje popis dat, která lze získat ze směnáren. Jedná se pouze o vybrané typy dat, které jsou zajímavé z pohledu algoritmického obchodování [6, 7, 12].

- **Transakce:** Informace o jednotlivých transakcích. Jedna transakce odpovídá jednomu záznamu získaného ze směnáren. Každá transakce je zaznamenána minimálně s přesným časem v milisekundách, cenou aktiva v okamžiku transakce, typem obchodu (nákup nebo prodej) a objemem transakce. V této práci je informace o typu obchodu obsažena v jejím objemu následovně: objem je kladný při provedení nákup a záporný při provedení prodeje.

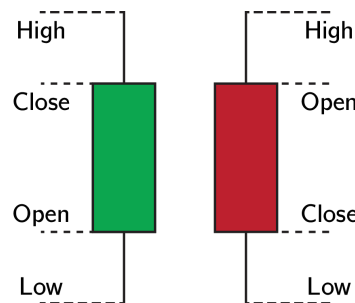
Na obrázku 2.2 jsou tyto transakce označeny jako **Prices** a zobrazené černou čarou. Jednotlivé transakce jsou spojované v pořadí podle času a mají vlastní identifikátor a objem (size).

- **OHLC:** Zkratka pro Open, High, Low, Close. Tyto údaje představují základní informace o cenové aktivitě za určité časové období. Zobrazují se pomocí svíčkového grafu pro zvolený časový rámec. *Open* je cena, za kterou bylo aktivum prvně obchodováno na začátku vybraného časového rámce, *Close* je pak poslední cena. *High* představuje maximální cenu, na kterou se aktivum dostalo během daného časového rámce, a *Low* je minimální cena.

Data ve formátu OHLC se zobrazují pomocí svíčkového grafu, který je možné vidět na obrázku 2.2 červenou a zelenou barvou. Obrázek 2.3 obsahuje vysvětlení zobrazení svíček.



Obrázek 2.2: Ukázka grafu obsahující OHLCV (OHLC + objem) data a jednotlivé transakce. Časový rámec pro OHLC a objemová data je 5 minut. Každá svíčka je vykreslena na začátku časového rámce, pro který je počítána. (Hodnoty pro první svíčku vykreslenou v 00:00 jsou spočítány z transakčních dat z intervalu <00:00 - 00:05>.)



Obrázek 2.3: Ukázka svíček pro OHLC s vysvětlivkami

Pro každou časovou jednotku t je možné získat následující data $open_t$, $high_t$, low_t , $close_t$, tp_t neboli $HLC3$, $HL2$, $OHLC4$, $HLCC4$ a další.

$$HL2_t = \frac{high_t + low_t}{2}$$

$$HLC3_t = TP_t = \frac{high_t + low_t + close_t}{3}$$

$$OHLC4_t = \frac{open_t + high_t + low_t + close_t}{4}$$

$$HLCC4_t = \frac{high_t + low_t + close_t + close_t}{4}$$

Kde TP je typická cena (Typical price).

- **Objem (Volume):** Objem obchodů, udává celkové množství aktiva, které bylo obchodováno v daném časovém rámci. Jedná se o sumu absolutních hodnot všech objemů transakcí v daném časovém rámci. Může se dále dělit na nákupní objem a objem prodeje.

Na obrázku 2.2 je možné vidět objem vyobrazený šedou barvou pod svíčkovým grafem. Protože hodnoty jsou řádově odlišné od ceny aktiva, je nutné vytvořit individuální grafová stopu pro tato data.

- **OHLCV:** Zkratka pro Open, High, Low, Close, Volume. Jedná se o rozšíření OHLC o objem, přesněji o součet objemu nákupu a prodeje.
- **Likvidace:** Data o pozicích, které byly automaticky uzavřeny směnárnou, obvykle kvůli nedostatečnému kapitálu v pákových obchodech. Data obsahují označení finančního nástroje, čas likvidace, cenu, za kterou byla provedena likvidace, a její velikost.
- **Open Interest:** Celková hodnota derivátových kontraktů, které nebyly ještě uzavřeny. Open Interest je měřítkem likvidity a účastníků na trhu vzhledem k danému derivátovému produktu. Růst Open Interest naznačuje zvýšený zájem o trh a může být indikátorem silného tržního trendu, zatímco pokles Open Interest může signalizovat snižující se zájem nebo přechodnou fázi trhu.

Na obrázku 2.4 je ukázka Open Interest dat zobrazena žlutou čarou.

- **Order Book:** Přehled aktuálních nabídek k nákupu a prodeji, včetně cen a množství. Order Book poskytuje hloubkový pohled na trh, zahrnující nabídky nákupu (Buy Orders) (Bids), nabídky prodeje (Sell Orders) (Asks) a to v jednotlivých cenových úrovních. Na každé úrovni je tedy vidět cena a objem. Tento pohled nám také umožňuje vidět spread u daného aktiva.

Na obrázku 2.4 je ukázka Order Book dat, kde na levé straně jsou zeleně zobrazeny nabídky nákupu a na pravé straně červeně nabídky prodeje.

Způsob poskytování dat

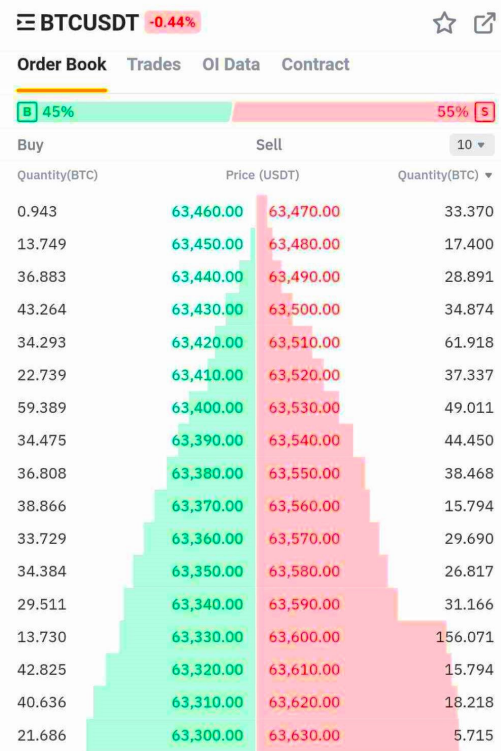
V dnešní době existují tři hlavní způsoby, jak směnárny poskytují data. Aktuální data se nejčastěji poskytují pomocí soketů (Websockets API), kde se, v závislosti na nastavení směnárny, posílají v rámci stanovených intervalů. Například několikrát za vteřinu se mohou posílat informace o jednotlivých transakcích, nebo jednou za vteřinu aktualizovat vteřinové OHLC. Jde tedy o nepřetržitý tok aktuálních dat. Zájemce se k tomuto toku dat může přihlásit a odebírat jej.

Dalším populárním způsobem, jak se předávají data, je prostřednictvím REST API. Zde si klient může specifikovat přesněji, jaká data chce získat. Většinou je možné přistupovat k historickým datům do určité doby. Bývají zde omezení, například data po vteřinách jsou dostupná pouze pár dní zpět, ale data po hodinách mohou být dostupná několik let do historie.

Posledním způsobem, podstatným pro tuto práci, který je obecně používán pro sdílení historických dat, je nějaké úložiště, které může být veřejně dostupné, nebo zpřístupněné po domluvě s poskytovatelem. Většinou neobsahuje nejnovější data, ale jsou do něj postupně ukládány, například s denním zpožděním. Tato data budou pro tuto práci kritická, protože některé směnárny obsahují několikaletou historii dat jednotlivých transakcí.



Obrázek 2.4: Snímek obrazovky mobilní aplikace od směnárny Bybit obsahující ukázkou Open Interest dat



Obrázek 2.5: Snímek obrazovky mobilní aplikace od směnárny Bybit obsahující ukázkou Order Book dat

2.2.4 Typy transakcí

V případě obchodování s nějakým aktivem je možné provést jeho nákup nebo prodej. Díky nákupu obchodník získává přímo aktivum, nebo na něj získává nárok. Následně jej může prodat do výše svého vlastnictví obchodovaného typu aktiva.

U CFD derivátů obchodník může sázet na budoucí růst nebo pokles ceny. V závislosti na typu sázky pak otevírá pozice. V případě, že obchodník očekává růst ceny, vytváří dlouhé (Long) pozice. V opačném případě vytváří krátké (Short) pozice.

2.2.5 Poplatky a další náklady

Poplatky ve směnárnách jsou jedním z hlavních zdrojů příjmů pro tyto finanční instituce a mohou být strukturovány různými způsoby. Pro potřeby této práce je zapotřebí pochopit dva typy poplatku, které jsou na každé směnárně, a to Maker a Taker Fee.

Maker Fee je poplatek účtovaný za transakce, které přidávají objednávky do Order Book, což znamená, že tvoří tržní likviditu. Tyto transakce mají nastavené podmínky, při kterých se vyhodnotí. Nastaví se cena na nějakou hodnotu a musí se počkat, než se vyhodnotí jiná transakce, která bude potřebovat tuto likviditu na nastavené ceně.

Taker Fee je poplatek účtovaný za transakce, které odebírají tržní likviditu. Tyto objednávky jsou obvykle vyhodnoceny okamžitě, protože přijímají podmínky již nabízené na trhu.

Taker Fee je zpravidla větší než Maker Fee, protože se směnárny snaží zajistit co největší tržní likviditu. Na některých směnárnách se může stát, že Maker Fee není ve skutečnosti

poplatek, ale směnárna platí obchodníkovi za vytvoření pozice. V dnešní době se jedná o výjimky, kdy se například nová směnárna tímto způsobem snaží zvýšit svou tržní likviditu.

Kromě poplatků je Spread dalším zásadním prvkem. Spread představuje rozdíl mezi kupní a prodejní cenou aktiva a hraje roli při nákladech na obchodování. Pro dlouhé (Long) pozice bude kupní cena vyšší než středová tržní cena a pro krátké (Short) pozice bude naopak nižší. To znamená, že obchodník vstupuje do pozice s okamžitou ztrátou odvíjející se od velikosti spreadu.

Další aspekty, který je důležité zvážit, jsou Slippage a zpoždění vyhodnocení transakcí. V situacích, kdy trh vykazuje vysokou volatilitu, může dojít k zpoždění mezi zadáním objednávky a jejím vyhodnocením. K zpoždění se také přičítá doba komunikace mezi směnárnou a obchodníkem. Toto zpoždění může způsobit, že skutečná cena se bude lišit od očekávané ceny.

2.3 Typy obchodních strategií

Existují různé druhy obchodních strategií, kde každá strategie vyhovuje různým obchodním stylům a cílům. Čtyři základní typy obchodních strategií jsou swingové obchodování, denní obchodování, skalповé obchodování a arbitráž.

- Swingové obchodování (Swing Trading) je strategie zaměřující se na tvorbu pozic trvajících několik dnů až týdnů.
- Denní obchodování (Day Trading) zahrnuje nákup a prodej v rámci jednoho obchodního dne. Obchodníci, kteří používají tuto strategii, často využívají malých cenových pohybů na finančních nástrojích s vysokou likviditou.
- Skalповé obchodování (Scalp Trading) je obchodní strategie, při které se provádí četné obchody v průběhu dne. Cílem je získat malé zisky z drobných cenových změn nebo vyčkat na ojedinělé příležitosti velkých rychlých cenových pohybů, předvídatelných pomocí různých analýz s nějakou přesností.
- Arbitráž (Arbitrage) spočívá v současné tvorbě různých pozic na různých trzích s cílem profitovat z malých cenových rozdílů mezi trhy. Tato strategie je považována za nízkorizikovou, ale příležitosti jsou vzácné a složitě detekovatelné.

2.4 Analýza trhu

Analýza trhu v kontextu akciového trhu zahrnuje různé přístupy ke studiu a predikci cen akcií. Mezi tyto přístupy patří zejména technická a fundamentální analýza. Celkově analýza trhu poskytuje nezbytný rámec pro pochopení a interpretaci tržních dat, umožňuje identifikovat příležitosti a rizika a podporuje efektivnější a informovanější rozhodnutí. Použití určitého typu analýzy obvykle závisí na cíli obchodníka, obchodní strategii, a také na preferencích a odborných znalostech obchodníka [10, 11].

Technická analýza se soustředí na predikci trhu prostřednictvím analýzy historických tržních cen a technických indikátorů. Využívá grafů a různých technických indikátorů k identifikaci vzorců a trendů v cenách aktiv. Naproti tomu **Fundamentální analýza** se zaměřuje na hodnocení vnitřní hodnoty aktiva, zkoumá ekonomické, finanční a další relevantní faktory, jako jsou ziskovost společnosti, její finanční zdraví, průmyslové podmínky a makroekonomické indikátory [10].

Kromě těchto tradičních přístupů existují i další metody. Jedním příkladem těchto metod může být **Sentimentální analýza**, která posuzuje celkový názor nebo postoj trhu, často prostřednictvím analýzy dat ze sociálních médií a novinových titulků [9, 11].

2.4.1 Technická analýza (TA)

Tato práce se zabývá převážně technickou analýzou. Základem technické analýzy jsou cenové grafy. Tyto grafy mohou být ve formě sloupcových grafů, svíčkových grafů, čárových grafů a dalších. Každý typ grafu poskytuje unikátní vizuální reprezentaci cenového pohybu a může ukazovat různé aspekty vývoje ceny.

Jedním z klíčových konceptů technické analýzy je identifikace trendů. Trh může být ve stoupajícím trendu, klesajícím trendu, nebo se cena může udržovat v pásmu.

Dále jsou pro technickou analýzu důležité úrovně a levely (významné hodnoty), které mohou představovat likviditu v podobě nastavených Take Profitů a Stop Lossů (Resistance a Support).

Nedílnou součástí této analýzy je i hledání opakujících se vzorů. Existuje celá řada vzorů, která se snaží předpovídat budoucí vývoj ceny.

Technická analýza využívá různé indikátory, které pomáhají identifikovat určité tržní podmínky nebo trendy. Mohou být použity samostatně, nebo v kombinaci pro vytváření obchodních strategií [1, 10, 11, 12].

2.4.2 Indikátory pro TA

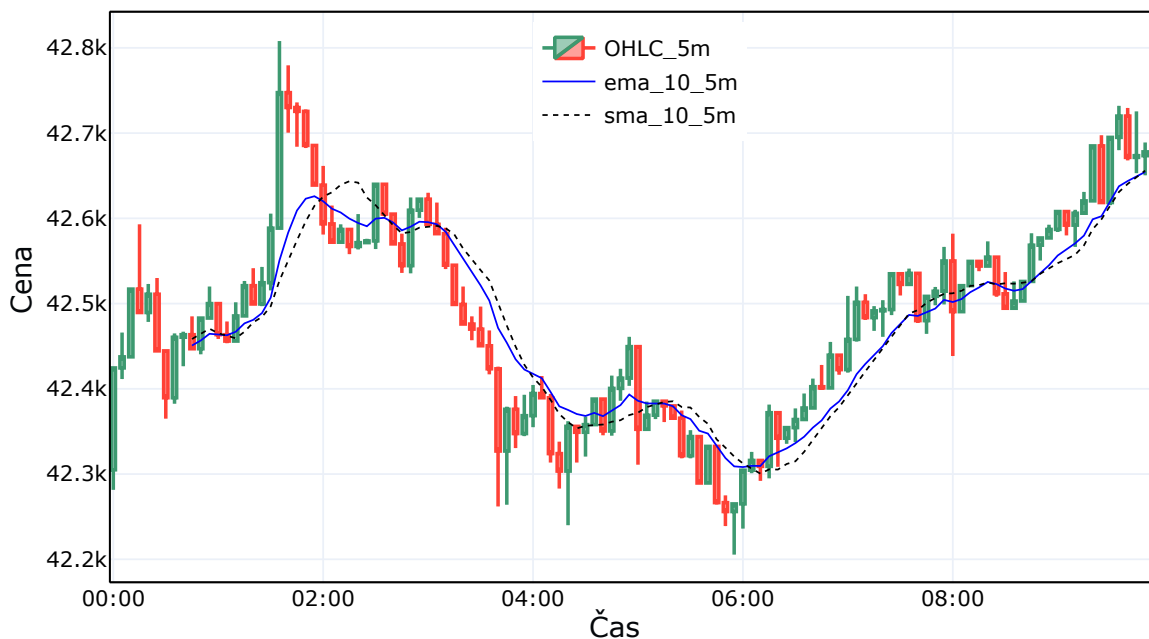
Indikátory v technické analýze jsou matematické výpočty, které se počítají pomocí historických dat obchodování. Existuje velké množství indikátorů. Některé běžně používané jsou klouzavé průměry, RSI, MACD, a mnoho dalších. Každý indikátor může mít různé parametry, které lze upravit pro každou analýzu, jako je délka časového rámce nebo počet uvažovaných rámců (perioda).

Pro představu je v této sekci uveden postup výpočtu hodnot indikátorů: SMA, EMA, Delta a CVD. Postupy pro zbylé indikátory (MACD, RSI a Swingové body), implementované ve frameworku, jsou v příloze A. Indikátory Delta a CVD jsou zde uvedeny pro demonstraci práce s jednotlivými transakcemi a nesou informaci o poměru nákupu a prodeje. Tím se odlišují od běžně používaných indikátorů.

V následujících definicích je použita časová jednotka t , která označuje časový rámec pro indikátor. Může být buď absolutní jako například 10 vteřin (10s), 5 minut (5m), 1 hodina (1h), nebo relativní jako 1 měsíc (1mo) (1M).

Jednoduchý klouzavý průměr (MA) (SMA)

Simple Moving Average (SMA) označovaný také jako Moving Average (MA) počítá průměr určitého počtu datových bodů za definované časové období (periodu). Nazývá se "jednoduchý", protože každému datovému bodu při výpočtu přiřazuje stejnou váhu. SMA se používá k vyhlazení cenových údajů a odfiltrování krátkodobých cenových výkyvů dle zvolené periody. Obvykle se používá kombinace více klouzavých průměrů s různou periodou.



Obrázek 2.6: Ukázka klouzavých průměrů SMA a EMA s periodou 10 a pětiminutovým časovým rámcem

Standardně se počítá z *close* hodnot, ale je možné použít libovolnou ziskatelnou hodnotu z OHLC. SMA se vypočítá následovně:

$$SMA_t(p) = \frac{1}{p} \sum_{i=0}^{p-1} V_{t-i}$$

Kde p je zvolená perioda, t je časová jednotka a V_t je hodnota v čase. Hodnota SMA je validní po $t \geq p$.

Ukázka SMA s periodou 10 a pětiminutovým časovým rámcem je na obrázku 2.6 zobrazena černou tečkovanou čarou.

Exponenciální klouzavý průměr (EMA)

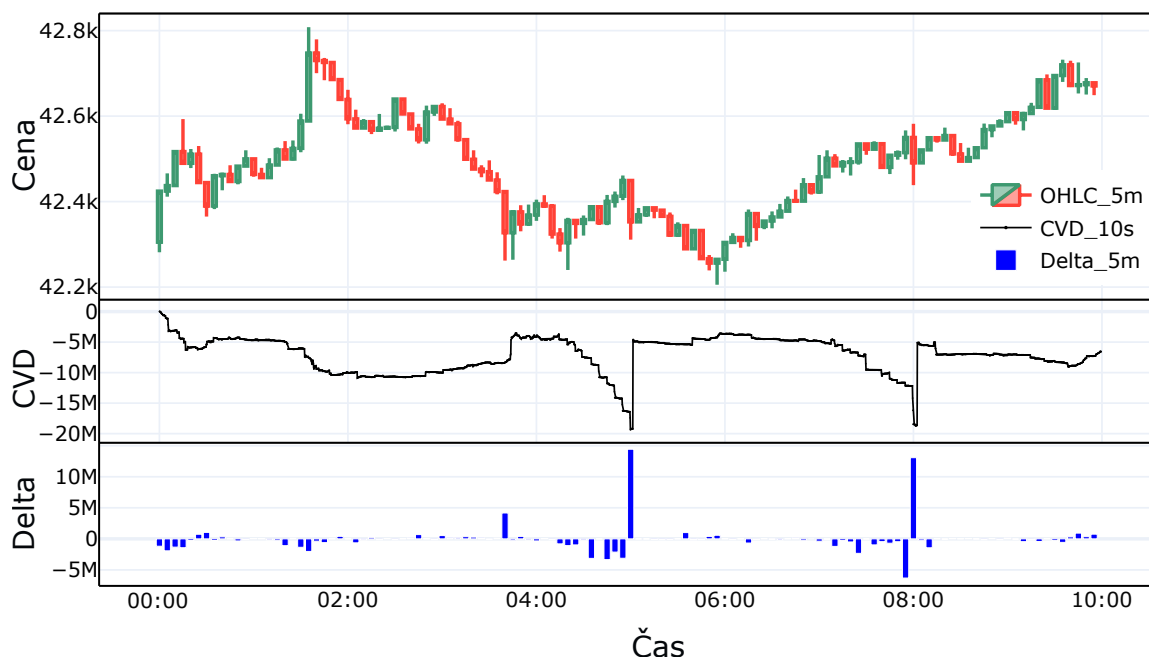
Exponential Moving Average (EMA) je typ klouzavého průměru, který dává větší váhu nedávným cenám, takže ve srovnání s jednoduchým klouzavým průměrem (SMA) lépe reaguje na nedávné změny cen.

Standardně se počítá z *close* hodnot, ale je možné použít libovolnou ziskatelnou hodnotu z OHLC. Následující vzorec pro výpočet EMA zahrnuje násobení konstantou vypočítanou z periody (označovanou jako vyhlazovací faktor):

$$EMA_0(p) = V_0$$

$$EMA_t(p) = \frac{2}{p+1} (V_t - EMA_{t-1}) + EMA_{t-1}$$

Kde p je zvolená perioda, t je časová jednotka a V_t je hodnota v čase. Hodnota EMA je validní po $t \geq p$.



Obrázek 2.7: Ukázka Delty a CVD s pětiminutovým časovým rámcem

EMA má schopnost rychle reagovat na změny cen, díky čemuž je užitečná pro identifikaci krátkodobých trendů a potenciálních vstupních nebo výstupních bodů na trhu. Protože však EMA přikládá větší váhu nedávným cenám, může být také náchylnější k výkyvům nebo falešným signálům na neklidných trzích.

Ukázka EMA s periodou 10 a pětiminutovým časovým rámcem je na obrázku 2.6 zobrazena modrou čarou.

Delta, Kumulativní delta (CVD)

Deltou se v této práci myslí součet všech velikostí nákupů a prodejů. Určuje, jestli byl pro daný časový úsek větší nákup, nebo prodej a o kolik. CVD je pak kumulativní součet všech delt od začátku dat.

Výpočet delty a CVD se dá zapsat jako:

$$M_transakce_t = \{transakce \in vsechny_transakce | (t - 1) \leq cas(transakce) < t\}$$

$$Delta_t = \sum_{i=0}^n velikost(M_transakce_{ti})$$

$$CVD_0 = Delta_0$$

$$CVD_t = CVD_{t-1} + Delta_t$$

Kde $cas()$ je funkce vracející čas transakce, $velikost()$ je funkce vracející velikost transakce, $M_transakce_t$ je množina všech transakcí provedených mezi časovými jednotkami $(t - 1)$ včetně a t , $vsechny_transakce$ jsou všechny transakce.

Na obrázku 2.7 je možné vidět CVD zobrazené uprostřed černou čarou a deltu zobrazenou dole modrou barvou. K výpočtu byl použit pětiminutový časový rámeček.

2.5 Shrnutí v kontextu s cílem práce

Jak je možné odvodit z názvu práce, jedním z hlavních cílů je napsat funkční framework pro backtestování strategií algoritmického obchodování na burze. Cílem je poskytnout robustní a přizpůsobitelný nástroj pro analýzu a simulaci obchodních strategií, který podporuje efektivní rozhodování a optimalizaci v algoritmickém obchodování. Je ovšem potřebné stanovit dílčí cíle této práce a definovat je. Následuje podrobnější specifikace funkcionality frameworku, které vychází z této kapitoly.

2.5.1 Finanční nástroje

Jak bylo popsáno v podsekcí 2.1.4, tato práce se zaměřuje převážně na CFD jako hlavního typu finančních nástrojů pro testování obchodních strategií. CFD (kontrakty na rozdíl) jsou deriváty, které umožňují obchodníkům spekulovat na změny cen finančních nástrojů bez nutnosti skutečného vlastnictví aktiva. Díky této vlastnosti bylo možné v implementaci backtestingu zanedbávat vlastnictví samotných aktiv. CFD umožňuje snadno obchodovat pomocí strategií napříč různými finančními nástroji.

2.5.2 Potřebná data

V analýze bylo zjištěno, jaká data lze získat od směnárů (přesněji v podsekcí 2.2.3). Vzhledem k faktu, že historická data (OHLC, Volume, ...) lze buď dopočítat z jednotlivých transakcí, nebo nejsou dostupná v dostatečné přesnosti (Open Interest, Order Book), jsou jednotlivé transakce považovány za hlavní vstup pro framework. Dalším cílem této práce bylo zajistit získávání historických dat jednotlivých transakcí ze směnárů.

2.5.3 Zpracování dat

Po získání dat je potřeba mít možnost s nimi nějak pracovat. Z toho plyne, že dalším cílem bylo napsat modul, který je schopný genericky zpracovávat jednotlivá data pomocí procesních jednotek, kde každá jednotka řeší zpracování jednoho typu dat (OHLC, CVD, Delta, ...). Uživatel frameworku si může pomocí předdefinovaných metod a funkcionalit jednoduše napsat vlastní procesní jednotky podle potřeby, což mu umožní přidávat zpracování vlastního typu dat. Tato data je možné ukládat na specifikované úložiště.

2.5.4 Vizualizace dat

Data bez možnosti jejich vizualizace se těžko zpracovávají. Jako další cíl bylo stanoveno vytvoření vizualizačního modulu pro zobrazení dat. Modul je schopný zobrazovat jednotlivé typy dat a agregovat je do grafů. Při umožnění uživateli dopsat si vlastní procesní jednotky může nastat, že generické vizualizační metody nebudou mít dostatečnou funkcionality pro zobrazení dat. Proto bylo do tohoto cíle také zahrnuto napsání vizualizačních jednotek, které pracují vždy s jedním typem dat obdobně jako procesní jednotky.

2.5.5 Indikátory

Protože se indikátory vyvíjejí a většina uživatelů by ocenila možnost si jednoduchým způsobem napsat či nadefinovat vlastní indikátory (a to buď úpravou existujících nebo kompletní implementací vlastních), bylo cílem této práce tuto možnost uživatelům poskytnout. Ve frameworku je tedy hierarchie indikátorů s možností jejich dědění, rozšiřování a úpravy.

Všechny indikátory také mohou využívat data ze všech implementovaných procesních jednotek.

2.5.6 Reprezentace strategií

V hlavním cíli jsou zmíněné strategie algoritmického obchodování. Tyto strategie je potřeba nějakým způsobem reprezentovat, a proto bylo potřebné vytvořit způsob, jakým toho dosáhnout. Cílem bylo vytvoření vhodné, čitelné, jednoduše upravovatelné reprezentace jednotlivých strategií.

Tato reprezentace umožňuje užívání indikátorů v libovolné kombinaci a specifikaci jednotlivých parametrů. Také je možné nastavit podmínky pro vzetí pozice (Long, Short). Jednotlivé podmínky mají také možnost kromě stavu indikátorů zahrnout i další parametry jako například omezení na počet aktivních pozic či jejich objem.

Jak bylo zmíněno v podsekcí 2.2.1, trh není konzistentní v průběhu dne a ani v průběhu týdne. Také se mění chování trhu v závislosti na jeho trendu. Z těchto důvodů by bylo vhodné, mít možnost specifikovat stav trhu a doby v týdnu, kdy bude strategie aktivní. Například umožnit aktivovat strategii pro vybranou seanci.

Také je možné reprezentovat kapitál a velikosti pozic pro přesnou simulaci peněženky, pokud se ve strategii uvažovat s podmínkami vázajícími se na stav peněženky. S tím se pojí umožněné nastavení parametrů pro strategii, jako je například výše leverage.

Specifikování Take Profitu spolu se Stop Lossem je zahrnuto v reprezentaci, a navíc je umožněno specifikovat jednotlivé strategie pro posuny těchto hodnot (Trailing Stop Loss).

Kapitola 3

Analýza existujících řešení

Obchodování na burze je populární již velmi dlouhou dobu. Je proto samozřejmé, že lidé si snažili ulehčit práci a zdokonalit strategie pro obchodování na burze pomocí nejrůznějších nástrojů. Tyto nástroje mohou být buď specializované anebo obecné.

Existuje široké spektrum řešení pro backtestování strategií algoritmického obchodování na burze, které umožňují testovat účinnost a robustnost strategií na historických datech. Bohužel každý z těchto nástrojů má svá specifika a nedostatky.

Existující řešení lze rozdělit do několika hlavních kategorií. Tato práce obsahuje analýzu existujících Open-source frameworků, specializovaných knihoven, placených a bezplatných služeb, ať už cloudových nebo desktopových platforem. Zaměření analýzy je na hledání konkurenceschopných řešení, možností inspirace a případné integrace do frameworku.

3.1 Open-source frameworky

Open-source frameworky pro technickou analýzu a backtesting strategií poskytují obchodníkům cenné nástroje pro testování a vývoj obchodních strategií na základě historických dat. Tyto frameworky umožňují simulovat chování strategie na historických datech, což umožňuje optimalizace před nasazením v reálném obchodním prostředí. Cílem této práce byla tvorba nového frameworku s vylepšenými funkcionalitami. Frameworků je celá řada, příkladem můžou být Freqtrade a Backtrader, které jsou v této kapitole více popsány.

3.1.1 Freqtrade

Freqtrade¹ je Open-source obchodní bot pro kryptoměny, napsaný v jazyce Python. Je navržen tak, aby podporoval všechny hlavní kryptoměnové burzy a umožňoval ovládání prostřednictvím Telegramu (aplikace pro posílání zpráv) nebo webového rozhraní. Bot obsahuje nástroje pro backtestování, vizualizaci a správu peněz, stejně jako optimalizaci strategie pomocí strojového učení.

Funkcionalita

Nástroj umožňuje vývoj strategií, které lze psát v Pythonu s knihovnou Pandas. Bot umožňuje stahování historických dat z vybraných burz. Jedná se o data OHLCV. Na těchto datech je poté možné testování strategií. Nástroj umožňuje více módů, a to backtestování

¹<https://www.freqtrade.io/>

na historických datech, či testování na realtime datech, a to buď s virtuální peněženkou, nebo s napojením na reálnou peněženko u dané směnárny.

Tento nástroj používá metody strojového učení pro nalezení nejlepších parametrů strategie. Optimalizace probíhá pomocí hyper-optimalizace parametrů a používá algoritmy z balíčku `scikit-optimize`². Uživatelé mohou vytvářet statické seznamy trhů nebo používat automatické seznamy založené na obchodním objemu nebo cenách. Lze také explicitně vyloučit trhy, na kterých se nemá obchodovat.

Bot umožňuje ovládání pomocí Telegramu nebo webového rozhraní. Lze provádět operace jako spuštění a zastavení bota, zobrazení stavu peněženko, zisku, ztrát, denního souhrnu a výsledku otevřených obchodů.

Freqtrade částečně podporuje řadu burz, včetně Binance, Bitmart, Gate.io, Huobi, Kraken, OKX a dalších.

Nedostatky

Freqtrade, jakožto Open-source obchodní bot pro kryptoměny, nabízí mnoho výhod, avšak pro tuto práci jsou důležité jeho nevýhody a omezení.

Nástroj používá pouze OHLCV, ze kterých nelze zjistit velké množství užitečných informací, jako je například poměr objemů nákupu a prodeje. V případě režimu simulace probíhá vyhodnocování jednou za nastavenou dobu a nevyhodnocují se jednotlivé transakce prováděné na směnárně. Protože používá OHLCV jako vstupní data, je tato doba zdola omezena na jednu vteřinu. Díky limitaci vstupních dat je samotný backtesting nepřesný.

Další nevýhoda je jeho zaměření pouze na burzy s kryptoměnami. Má omezenou funkcionalitu, například vytváření vlastních předzpracovaných dat je nemožné. Vyžaduje hlubší znalost nástroje a pokročilé znalosti programování v Pythonu pro jeho používání.

3.1.2 Backtrader

Backtrader³ je pokročilý Open-source Python framework pro backtesting a obchodování, který se zaměřuje na poskytování robustní platformy pro vývoj, testování a nasazení algoritmických obchodních strategií.

Funkcionalita

Framework umožňuje uživatelům psát a testovat opakovaně použitelné obchodní strategie. Podporuje širokou škálu indikátorů a analyzátorů výkonu, které pomáhají při analýze a optimalizaci strategií. Umožňuje nasazení strategií v reálném čase. Podporuje několik zdrojů dat včetně CSV, databází, Yahoo Finance. Umožňuje generovat grafy pro lepší porozumění výsledkům strategie. Také dovoluje uživateli optimalizovat parametry strategie.

Nedostatky

Hledání nejlepších parametrů probíhá pomocí procházení všech možných kombinací parametrů. Framework pracuje pouze na zpracovaných datech jako je OHLCV. Z toho plynou stejné nedostatky, které již byly zmíněny, a to omezenost získatelných informací, nepřesnost simulace a vyhodnocování nejčastěji po jedné vteřině. Díky tomu dochází ke zkreslení výsledku, které je viditelnější pro pozice trvající krátkou dobu.

²<https://scikit-optimize.github.io/stable/>

³<https://www.backtrader.com/>

3.2 Platformy

Většina směnárny implementuje vlastní platformy. Některé z nich umožňují provádět technickou analýzu a mohou také nabízet funkce pro backtesting, umožňující uživatelům testovat různé technické indikátory a obchodní strategie. Příkladem může být Bybit⁴.

Protože jednotlivé směnárny většinou umožňují uživatelům zobrazit pouze vlastní finanční nástroje, existují firmy vytvářející platformy, které uživatelům umožňují sledovat data z více směnárny současně. Jednou z těchto platform je TradingView⁵. Tyto platformy mohou obsahovat možnosti backtestingu nebo dokonce vyvíjet vlastní programovací jazyky. Například TradingView nabízí jejich Pine Script⁶. Dále umožňují uživatelům také nastavovat různá upozornění na základě jejich preferencí a nastavených indikátorů.

Mimo to existují obchodní platformy, které sice zahrnují integrované nástroje, ale zaměřují se spíše na samotné obchodování: NinjaTrader⁷, Metatrader 4⁸/5⁹, cTrader¹⁰, Atas¹¹, DXtrade¹² a další.

3.3 Knihovny pro technickou analýzu

Existuje řada knihoven, která se zabývá technickou analýzou. Některé jsou specializované na finanční časové řady, jiné jsou spíše obecné. Zde je pár ukázek těchto knihoven.

3.3.1 TA-Lib

TA-Lib¹³ (Technical Analysis Library) je široce používaná Open-source knihovna zaměřená na technickou analýzu finančních časových řad. Používá OHLCV jako vstupní data. TA-Lib poskytuje přes 200 technických indikátorů, jako jsou ADX, MACD, RSI, Stochastic, Bollinger Bands a mnoho dalších. Knihovna také zahrnuje funkce pro rozpoznávání různých vzorů svíčkových grafů.

Zatímco jádro knihovny je napsáno v C/C++, poskytuje také API pro Python, což usnadňuje integraci s Python aplikacemi a skripty. TA-Lib je distribuována pod licencí BSD, což znamená, že ji lze volně integrovat jak do Open-source, tak do komerčních aplikací. Tato knihovna byla vydána v roce 2001 a je stabilní a osvědčená časem.

Jeden z Pythonových wrapperů¹⁴ pro TA-Lib je ta-lib-python¹⁵. Tento wrapper rozšiřuje funkcionalitu původní knihovny o podporu jazyka Python a nabízí několik výhod a funkcí. Je postaven na knihovnách Python Pandas a Numpy. Indikátory mohou být přidány do datového rámce Pandas, což umožňuje integraci s dalšími analytickými procesy. Také umožňuje snadné použití funkcí s výchozími parametry a pro pokročilejší použití existuje

⁴<https://www.bybit.com/>

⁵<https://www.tradingview.com/>

⁶<https://www.tradingview.com/pine-script-docs/en/v5/>

⁷<https://ninjatrade.com>

⁸<https://www.metatrader4.com>

⁹<https://www.metatrader5.com>

¹⁰<https://ctrader.com>

¹¹<https://atas.net>

¹²<https://dx.trade>

¹³<https://ta-lib.org/>

¹⁴Wrapper je program, který obaluje jiný kód, aby zjednodušil jeho použití, rozšířil jeho funkcionalitu, nebo zprostředkoval kompatibilitu.

¹⁵<https://github.com/TA-Lib/ta-lib-python>

tuje Abstract API, které přijímá pojmenované vstupy a pracuje s datovými strukturami, což nabízí větší flexibilitu v manipulaci s daty.

Pandas TA¹⁶ - a Technical Analysis Library in Python 3 je další knihovna používající TA-Lib. Obsahuje více než 130 indikátorů, užitečné funkce a více než 60 vzorů svíčkových grafů. Součástí je mnoho běžně používaných indikátorů. Pro rychlejší zpracování umožňuje Pandas TA využití vícejádrových procesorů, také podporuje integraci s dalšími obchodními knihovnami, jako je Vectorbt¹⁷, YFinance¹⁸.

3.3.2 Technical Analysis Library in Python

Technical Analysis Library in Python¹⁹ je analytická knihovna v Pythonu, která zpracovává OHLCV data. Je postavena na knihovnách Pandas a Numpy a nabízí širokou škálu indikátorů pro technickou analýzu.

3.3.3 Technical Analysis (ta.py)

Knihovna Technical Analysis (ta.py)²⁰ je jednou z dalších Python knihoven, která zpracovává OHLCV data. Umožňuje uživatelům využívat široký rozsah nástrojů pro analýzu trhů. Dále nabízí funkce pro měření volatility, statistické funkce a různé typy grafů.

3.3.4 Shrnutí analýzy knihoven pro TA

Většina knihoven používá OHLCV jako vstupní data, které následně zpracovávají pomocí různých indikátorů a vzorů. Jak již bylo zmíněno, tento přístup má svá omezení, které brání použití zanalyzovaných knihoven ve frameworku pro hlavní výpočty. Je ovšem možné je použít například pro hledání vzorů v grafech po zpracování dat do OHLCV formátu.

3.4 Další užitečné knihovny

Ne všechny knihovny se zabývají technickou analýzou. V tradingu je také potřeba získávat data a případně komunikovat se směnárny.

Příkladem je Cryptocurrency eXchange Trading²¹ (CCXT). CCXT je knihovna s kolekcí tříd pro různé kryptoměnové směnárny.

Všechny třídy směnárny jsou v ní odvozené od základní třídy a sdílejí společnou sadu metod. Každá třída implementuje veřejné a soukromé API pro danou směnárnu. Soukromé API umožňuje správu účtu, vytváření a zrušení pozic, vklady a výběry. Veřejné API umožňuje například načítání trhů, kurzů, Order Book, historických dat. CCXT aktuálně (prosinec 2023) podporuje 93 kryptoměnových směnárny a obchodních API, včetně Binance, Coinbase, Kraken a dalších. Placená verze CCXT Pro je doplněk podporující WebSocket streamy.

Kvůli snaze implementovat jednotné rozhraní pro velké množství směnárny, dochází k vysoké abstrakci, která může zpomalovat jednotlivé úkony a komunikaci se směnárnou. Toto je nejvíce viditelné v případě Skalповého obchodování. Díky častým aktualizacím

¹⁶<https://github.com/twopirllc/pandas-ta>

¹⁷<https://vectorbt.dev/>

¹⁸<https://github.com/ranaroussi/yfinance>

¹⁹<https://github.com/bukosabino/ta>

²⁰<https://github.com/Bitvested/ta.py>

²¹<https://ccxt.com>

a změnám v API směnárny, vyžaduje knihovna CCXT pravidelnou údržbu a aktualizace. Některé pokročilé funkce, jako jsou WebSocket streamy, jsou k dispozici pouze v placené verzi CCXT Pro.

3.5 Shrnutí v kontextu s cílem práce

Bohužel, během analýzy existujících nástrojů a knihoven nebyl nalezen nástroj, který by implementoval práci s jednotlivými transakcemi (používal Order Flow), umožňoval backtesting strategií na transakčních datech a současně umožňoval vylepšovat tyto strategie sofistikovaným způsobem.

3.5.1 Možnosti integrace

Nástroje jako Freqtrade jsou vhodné pro inspiraci během implementace tohoto frameworku a některé knihovny jako TA-Lib by bylo možné integrovat. Kvůli komplexnosti knihoven a nástrojů je jednodušší se jimi inspirovat, implementovat vlastní řešení a případně integrovat vybrané funkcionality jako detekci vzorů v grafech (Jak bylo zmíněno v podsekcí 3.3.4.)

3.5.2 Testování a validace

Po vývoji je také potřebná validace, díky které se ověří správnost fungování frameworku. Cílem bylo procházení jednotlivých výstupů a provádění manuálního ověřování. Existující řešení je možné použít pro ověření a porovnání výsledků ze stejně definovaných, jednoduchých strategií.

3.5.3 Inspirace v backtestingu

Backtesting je poměrně rozsáhlý pojem. Existující řešení většinou nabízejí rychlý backtesting s tím, že využívají již zpracovaná data, což jim umožňuje rychle provádět iterace pro každou strategii.

V této práci má backtesting dvě podoby. První je simulace jednotlivých transakcí za určité období, kde se proudově vyhodnocují jedna transakce po druhé. Druhá podoba je podobná existujícím řešením a má možnost použití zpracovaných dat s možností dodatečného průchodu jednotlivých transakcí v případě potřeby.

Druhá verze slouží k rychlému prototypování. Dává přesnější výsledky než existující řešení, pokud uživatel používá jednotlivé transakce k doladění backtestingu a nepoužívá k vyhodnocování data, která by v době interakce s pozicemi nebyly k dispozici.

Cílem tedy byla implementace obou řešení backtestingu, což s sebou přináší potřebu implementace virtuální směnárny. Hlavní zaměření této práce bylo na průchodovou verzi backtestingu.

3.5.4 Vyhodnocování strategií

Jako u ostatních řešení, framework je schopný pro jednotlivé strategie určit, v jaký okamžik a jak interagovat s pozicemi, a tedy umožní najít a vyhodnotit přesný vstup a výstup pro pozice.

Následně bylo potřeba tyto interakce zpracovat. K tomu slouží virtuální směnárna, která buď v průběhu analýzy řeší kapitál na účtech a případně informuje o likvidaci účtu kvůli nedostatečnému kapitálu. Nebo je v módu, kdy jen vyhodnotí a zaznamená pozice. Následně

generuje report obsahující všechny podstatné informace pro rozhodnutí o kvalitě strategie, jako je například počet transakcí, průměrná doba trvání aktivní pozice, maximální pokles hodnoty peněženky.

3.5.5 Konkurenceschopnost

Z cílů je patrné, že se tento framework odlišuje od ostatních existujících řešení ve více bodech. Jeden z hlavních bodů je, že framework je schopný pracovat s jednotlivými transakcemi a zpracovávat obrovská data pro co nejrealističtější simulaci v backtestingu. Také umožňuje zohledňovat jednotlivé transakce v indikátorech a tímto dosáhnout mnohem přesnějších výsledků pro pozice, které trvají krátkou dobu (vteřiny).

Dalším hlavním přínosem tohoto frameworku byla jeho jednoduchá rozšiřitelnost o vlastní implementace ať už jednotlivých indikátorů nebo procesních a vizualizačních jednotek.

Další hlavní odlišností je umožnění uživateli použít evoluční algoritmy k vylepšování strategie tím, že se postupně hledají lepší a lepší parametry pro strategii. Tato funkcionality je více popsána v kapitole 4.

Jedná se o komplexní návrh s rozsáhlou funkcionalitou. Během studia a zkoumání existujících řešení se nepodařilo najít řešení, které by uživateli umožnilo použít všechny zmíněné funkcionality, nebo alespoň jejich větší část.

Kapitola 4

Evoluční algoritmy (EA)

Tato kapitola poskytuje přehled Evolučních algoritmů (EA), jsou vhodné pro optimalizaci parametrů obchodních strategií z mnoha důvodů. Jedním z hlavních důvodů je jejich schopnost efektivně prozkoumávat velké prostory řešení, na které jsou tradiční optimalizační metody neefektivní nebo nedostačující. Tato schopnost je zásadní pro komplexní obchodní strategie s velkým množstvím kombinací hodnot parametrů.

Evoluční algoritmy se při hledání řešení problémů inspiřují procesy biologické evoluce. Evoluce je proces, který zahrnuje dynamickou změnu populace jedinců v čase pod tlakem selekce neboli přežití nejschopnějších. Tento koncept je základem evolučních algoritmů, ale nejsou jím omezeny. Evoluce v tomto kontextu není zaměřena na přesné modelování existujících evolučních systémů v přírodě, ale spíše na využití jejich adaptačních schopností pro řešení složitých výpočetních problémů. Tato kapitola je inspiřována a čerpá z následujících zdrojů: [3, 4, 5].

4.1 Co jsou EA

Existuje mnoho různých variant evolučních algoritmů, ale společná základní myšlenka všech těchto technik je stejná: Je dána populace jedinců v nějakém prostředí, které má omezené zdroje. Soupeření o tyto zdroje způsobuje přirozený výběr. Tento proces vede ke zlepšení celkové zdatnosti populace.

Pro maximalizaci zdatnosti populace je nejprve nutné vytvořit počáteční populaci například vytvořením náhodných jedinců. Na tato řešení se následně aplikuje funkce kvality jako abstraktní ukazatel zdatnosti, vracející hodnotu zdatnosti (fitness) pro každého jedince. Vybraní kandidáti s lepšími fitness hodnotami jsou poté použiti pro vytvoření nové generace. Toho je dosaženo pomocí křížení nebo mutace, kde procesy generují skupinu nových potomků, jejichž fitness je posouzena se stávajícími členy populace. Následně probíhá soutěž na základě výsledků a případného stáří o místo v příští generaci. Tento cyklus se opakuje, dokud není nalezeno dostatečně kvalitní řešení, nebo dokud není dosažen předem stanovený výpočetní limit. Z popisu můžeme vytvořit obecnou strukturu Evolučního Algoritmu. Algoritmus 1 je pseudokód obecné struktury evolučního algoritmu. Začíná inicializací populace náhodných řešení a pokračuje jejich postupným vývojem prostřednictvím výběru, křížení a mutace, čímž se snaží zlepšovat kvalitu řešení v každé generaci.

Algoritmus 1 Struktura evolučního algoritmu

- 1: Inicializace počáteční populace
 - 2: Vyhodnocení fitness každého jedince
 - 3: **while** není splněna podmínka ukončení **do**
 - 4: Výběr rodičů
 - 5: Křížení rodičů
 - 6: Mutace vzniklých potomků
 - 7: Vyhodnocení fitness nových kandidátů
 - 8: Výběr jedinců pro další generaci
 - 9: **end while**
-

Tento Algoritmus 1 je příliš obecný. Abychom jej mohli přizpůsobit specifickým typům problémů, je nejprve nutné zodpovědět několik klíčových otázek:

- Jak jsou v populaci zastoupeni jedinci?
- Jak se vybírají rodiče?
- Jak velká by měla být populace?
- Jak vznikají potomci z vybraných rodičů?
- Kolik potomků by mělo být vyprodukováno?
- Jak se vybírají přeživší?

4.1.1 Reprezentace jedinců

V evolučních algoritmech se reprezentace jedinců dělí na dva typy: genotypovou a fenotypovou reprezentaci. Genotypové reprezentace napodobují biologický genetický kód, kde jsou řešení zakódována jako řetězce z univerzální abecedy. Fenotypové reprezentace přímo představují řešení problému bez jakéhokoli zprostředkujícího kódování. Prvním krokem při definování evolučního algoritmu je vytvoření spojení mezi reálným problémem a prostorem pro řešení problému, často zahrnující zjednodušení nebo abstrakci. V případě, že jsou jedinci reprezentováni pomocí fenotypové reprezentace, je potřeba vytvořit zakódování na genotypovou reprezentaci.

4.1.2 Velikost populace a počet potomků

V přírodních systémech může velikost populace kolísat v závislosti na faktorech prostředí, ale v evolučních algoritmech se změna velikosti populace v každé generaci neprokázala jako významný výpočetní přínos. Proto většina EA používá model s pevnou velikostí populace, který je jednodušší na implementaci.

Během generace se vyprodukuje určitý počet potomků. Následně mechanismy výběru pro přežití upravují velikost populace na původní velikost. Volba velikosti populace je ovlivněna požadovaným stupněm paralelního vyhledávání v závislosti na požadavcích systému. Počet potomků souvisí se zpožděním zpětné vazby neboli množstvím provedeného průzkumu s aktuální populací před její aktualizací o nové informace.

4.1.3 Selektce a výpočet zdatnosti (fitness)

V evolučních algoritmech jsou dvě základní selektce: při výběru rodičů a při určování přeživších. Pro efektivní selektci se počítá fitness jedinců. Přílišné upřednostňování nejsilnějších jedinců v tomto procesu může vést k rychlé, ale neoptimální konvergenci. Důležité je najít správnou míru selekčního tlaku. Z tohoto důvodu, většina EA uplatňuje zkraslení zdatnosti jedinců. Zkraslení se většinou provádí pouze v jedné fázi výběru. Existují různé výběrové mechanismy, jako je omezený výběr nejvhodnějších jedinců, turnajový výběr, výběr podle pořadí a fitness, které se liší stupněm selekčního tlaku a mohou být implementovány různými způsoby. Při výběru je taktéž nutné zvážít, zda se generace mohou překrývat. Překrývající se generace vedou k rychlejší (možná neoptimální) konvergenci než nepřekrývající se populace.

Výpočet fitness/zdatnosti jedince se provádí pomocí Vyhodnocovací/Fitness funkce, reprezentující požadavky, kterým by se měla populace přizpůsobit. Úkolem Fitness funkce je definování toho, co znamená zlepšení jedince. Často je cílem maximalizace fitness, avšak tento problém lze převést i na minimalizaci. Funkce může být binární, kde hodnota 1 může znamenat jedince splňující všechny požadavky a hodnota 0 vše ostatní. Tento přístup se ale nehodí pro většinu problémů a je potřeba vracet hodnotu zohledňující jednotlivé požadavky. Například je možné jednotlivé požadavky ohodnotit na základě významnosti, a dokonce jim i přidat měřítko, do jaké míry jsou splněné.

4.1.4 Křížení a mutace

V evolučních algoritmech se používají variační operátory pro tvorbu nových jedinců. Tyto operátory se dělí podle arity na dva typy: unární (mutace) a n-ární (křížení). Návrh a použití těchto operátorů jsou zásadní pro zavedení genetické rozmanitosti a usnadnění průzkumu prostoru řešení, což je klíčové pro úspěch evolučního procesu. Mutace a křížení se mohou v EA vyskytovat současně, samostatně, nebo i s různými pravděpodobnostmi jejich provedení.

Mutace modifikuje jeden genotyp a vytváří mírně pozmeněné jedince. Výsledky závisí na náhodných volbách. Je navržen tak, aby vyvolával náhodné, nezkraslené změny. Taktéž je možné se setkat s heuristickými unárními operátory, které se systematicky zaměřují na individuální vylepšení jedinců.

Křížení je n-ární variační operátor, který slučuje znaky z více rodičovských genotypů do jednoho nebo více potomků. Většinou se jedná o binární operátor, který bere znaky ze dvou rodičů. Tento proces, závisí na náhodném výběru rodičovských znaků, které se mají kombinovat. Křížení je založené na principu kombinování různorodých, ale žádoucích vlastností dvou jedinců, což je koncept úspěšně uplatňovaný v přirozeném výběru a tradičním šlechtění.

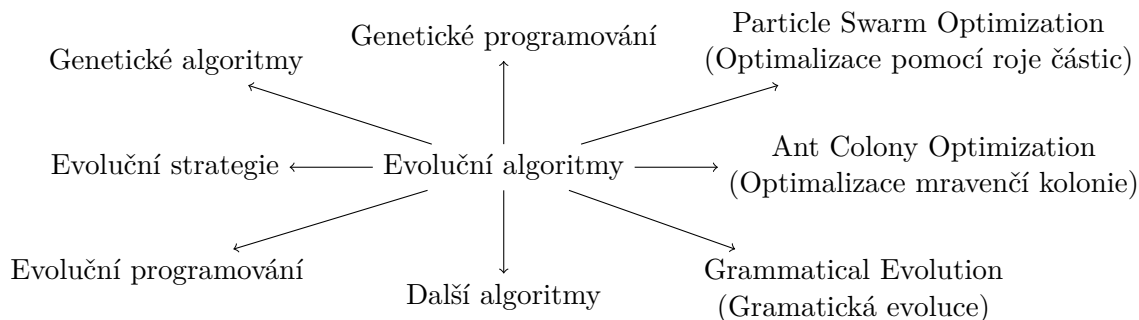
4.1.5 Ukončovací podmínka

V ideálním světě by byla podmínka zastavení evoluce nastavená na objevení řešení. Bohužel kvůli abstrakci a zjednodušování se zavádí šum do výpočtů. Taktéž nemusí existovat záruka nalezení optimálního řešení, či optimální řešení nemusí vůbec existovat. Pokud má problém známou optimální úroveň fitness v rámci dané přesnosti, tak se za podmínku pro ukončení považuje nalezení jedince, který dosahuje potřebné zdatnosti. Tato podmínka nemusí být splněna a algoritmus se nezastaví v přijatelném čase. Proto musíme tuto podmínku rozšířit další, která omezuje zdroje pro běh výpočtu nebo zajišťuje konvergenci k řešení. Může se

skládat například z limitace času procesoru, celkového maximálního počtu vyhodnocení fitness funkce, detekce poklesu zlepšování zdatnosti jedinců po stanovenou dobu nebo poklesu diverzity populace pod mezní hodnotu.

4.2 Typy EA

Existuje celá řada evolučních algoritmů, jejichž částečný přehled je možné vidět na obrázku 4.1. Každý z těchto algoritmů byl navržen tak, aby řešil specifické typy optimalizačních problémů, přičemž využívají různé metody pro prohledávání prostoru možných řešení.



Obrázek 4.1: Přehled některých Evolučních algoritmů [2, 3]

Pro řešení optimalizace v této práci uvažujeme genetické algoritmy (GA) jako dostačující, protože GA poskytují efektivní vyhledávání a schopnost uniknout z lokálních optim. Přesto v některých případech, jako je vyvíjení nových obchodních strategií nebo optimalizace spojitých proměnných, mohou být genetické programování (GP), gramatická evoluce (GE), nebo evoluční strategie (EA) přínosnější.

Genetické algoritmy jsou třídou evolučních algoritmů, které napodobují přirozený evoluční proces jako mechanismus pro nalezení optimálních nebo přibližně optimálních řešení v rozmanitých optimalizačních problémech. GA operují na základě principů genetické selekce, křížení (rekombinace) a mutace. V GA je populace obvykle reprezentována jako sada genomů, kde každý jedinec se dá definovat posloupností hodnot, které kódují parametry řešení.

4.3 Shrnutí v kontextu s cílem práce

Evoluční algoritmy jsou vhodné pro optimalizaci parametrů z mnoha důvodů. Pro tuto práci je podstatná schopnost hledat v rozsáhlých, komplexních prostorech, odolnost vůči uváznutí v lokálních maximech či minimech, paralelizace výpočtů a škálovatelnost. Taktéž je vhodné, že jsou schopny se přizpůsobit měnícím se podmínkám a zachovat různorodost řešení.

4.3.1 Vylepšování parametrů strategií

S pomocí evolučních algoritmů je možné dosáhnout vylepšování parametrů indikátorů ve strategiích. Jeden z cílů této práce bylo specifikování přesných požadavků na evoluční algoritmy a výběr nejvhodnějších algoritmů. Následně implementace algoritmů a jejich úprava, aby byly efektivní pro problematiku této práce.

Vylepšování parametrů strategií je možné dvěma hlavními způsoby. Použitím EA při samotné definici strategií. A při postprocessingu, kde se z nasbíraných dat k jednotlivým pozicím stanovují další parametry, které se mohly vynechat při prvotní definici strategie. Tyto parametry v postprocessingu mohou být pro představu spojené například s různými hodnotami před vzetím pozice, nebo s dobou, kdy byla pozice vytvořena. Vhodným použitím EA v postprocessingu se mnohonásobně urychlí konvergence k optimálnímu řešení, protože postprocessing je mnohonásobně rychlejší než simulace obsažená v backtestingu.

4.3.2 Fitness funkce strategie

Evoluční algoritmy se neobejdou bez metody hodnotící strategii nějakou porovnatelnou hodnotou. Pro jejich správné fungování bylo tedy potřeba navrhnout funkční vyhodnocovací metodu, která hodnotí strategie. Tuto metodu je možné blíže specifikovat v nastavení strategie. Přesněji, uživatel je schopen si nastavit, zda má větší zájem například o co největší profit, nebo o co největší efektivnost strategie.

Kapitola 5

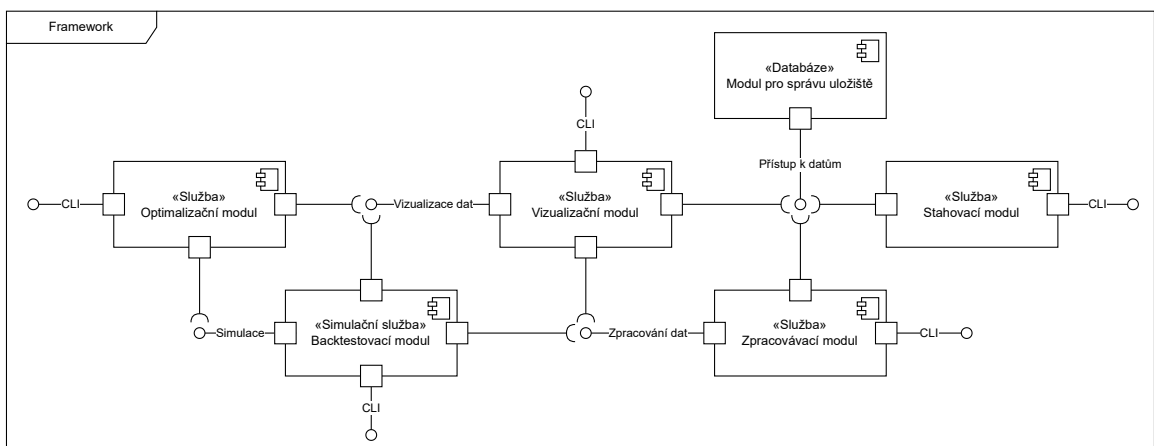
Návrh a architektura frameworku

Tato kapitola se zaměřuje na upřesnění specifikace frameworku. Jsou zde definovány požadavky a způsoby jejich řešení. Celý framework se skládá z několika navzájem provázaných modulů, kde každý modul přidává novou funkcionalitu.

Bylo potřeba navrhnout a implementovat moduly pro následující funkcionality:

- Stahování dat (Stahovací modul)
- Zpracovávání dat (Zpracovávací modul)
- Správu dat (Modul pro správu úložiště)
- Zobrazování dat (Vizualizační modul)
- Backtesting - simulace (Backtestovací modul)
- Vylepšování strategií (Optimalizační modul)

Na obrázku 5.1 je možné vidět jejich vzájemné závislosti. Jediný modul bez možnosti přímého ovládání pomocí rozhraní příkazové řádky (CLI), je Modul pro správu úložiště. Tento modul poskytuje přístup k datům Stahovacímu, Zpracovávacímu a Vizualizačnímu modulu. Zpracovávací modul zpracovává data pro Vizualizační a Backtestovací modul. Vizualizační modul vizualizuje data pro Backtestovací a Optimalizační modul. Backtestovací modul poskytuje simulace pro Optimalizační modul.



Obrázek 5.1: Schéma modulů na nejvyšší úrovni (CLI - rozhraní příkazové řádky)

5.1 Volba programovacího jazyka

Pro tuto práci byl zvolen programovací jazyk Python z několika důvodů. Python má bohatý ekosystém knihoven, který zahrnuje práci s finančními řadami. Jeho syntaxe a jednoduchá čitelnost umožňují rychlý vývoj. Python je platformě nezávislý, což znamená, že framework je možné spustit na různých operačních systémech. Hlavním důvodem tohoto výběru je existence různých balíčků od jednotlivých směnárů usnadňující práci s jejich API.

5.2 Stahování dat

Jak bylo zmíněno v podsekcí 2.5.2, pro tento framework jsou klíčová historická data jednotlivých transakcí. Za účelem stanovení vhodné směnárny pro získávání dat byl uskutečněn rozsáhlý průzkum všech směnárů poskytujících data. Pro získání informací, jako je například objem transakcí prováděných přes směnárnu a dostupné finanční nástroje na směnárně, byly napsány skripty ke scrapování (získávání) informací ze stránek CoinMarketCap¹ a CoinGecko².

Ze získaných dat vznikly po zpracování tabulky v příloze B, ze kterých byly zkopírovány nejzajímavější informace do tabulky 5.1. V tabulce 5.1 je vidět 30 nejlépe hodnocených směnárů. Ke každé směnárně je uveden počet finančních nástrojů, se kterými je možné na této směnárně obchodovat a denní objem všech transakcí. Finanční nástroje jsou v tabulce rozdělené na derivátové a ostatní (spotové). Jedná se jen o orientační hodnoty, které mohou být zkreslené různými faktory. Také se tato data v čase mění. Hodnocení bylo stanoveno na základě hodnoty Alexa Rank³. Jedná se o číslo, které ukazuje, jak dobře se webová stránka umísťuje ve srovnání s ostatními na internetu. Čím nižší číslo, tím lepší pozice.

Následně byl proveden průzkum nejlépe hodnocených směnárů s vysokým objemem transakcí za účelem zjištění stavu dokumentace a dostupnosti historických dat. Tabulka 5.2 obsahuje seznam vybraných směnárů a odkazy na dokumentaci jejich rozhraní. Tabulka 5.3 obsahuje odkazy na úložiště s historickými daty. Byly nalezeny pouze dvě přehledná úložiště, která měla přijatelnou historii a obsahovala jednotlivé transakce pro jednotlivé finanční nástroje.

Na základě průzkumu byly zvoleny směnárny Bybit a Binance jako vyhovující pro získávání dat. Protože u směnárny Bybit se dají data stáhnout jednodušeji, bylo prioritou zprovoznit stahování dat přes tuto směnárnu s možností rozšíření frameworku o další.

Dále bylo nutné stanovit označení finančních nástrojů v celém frameworku. V rámci jedné směnárny mohou mít různé typy finančních nástrojů stejný symbol. A více směnárů může mít stejný typ a symbol pro finanční nástroj. Z toho bylo nutné do označení finančního nástroje zahrnout název směnárny, typ finančního nástroje a symbol například `bybit_inverse_BTCUSD` (kde `inverse` je označení pro některé typy derivátů na Bybit směnárně).

Stahovací modul obsahuje rozhraní příkazové řádky (CLI), které přijímá označení finančního nástroje jako vyžadovaný vstup. Dále je možné specifikovat vybrané období, pro které se mají data stáhnout. Pro využití možnosti stahování více souborů současně, byl implementován přepínač zapínající multiprocessing a parametr pro specifikaci počtu procesů. Celé stahování je také možné spustit ve smyčce s definovatelnou délkou čekání.

¹<https://coinmarketcap.com/>

²<https://www.coingecko.com/>

³https://en.wikipedia.org/wiki/Alexa_Internet#Alexa_Traffic_Rank

Směnárna	Spotové	Derivátové	Celkem	Denní objem [\$]	Alexa Rank
Binance	387	299	686	43,000,000,000	132
Coinbase Exchange	245	153	398	1,000,000,000	730
Bybit	451	349	800	14,000,000,000	1,034
KuCoin	726	229	955	2,800,000,000	1,209
DigiFinex	412	65	477	9,100,000,000	1,922
Gate.io	1,777	319	2,096	2,400,000,000	2,039
MEXC	1,662	349	2,011	5,200,000,000	2,100
OKX	319	429	748	16,000,000,000	2,386
XT.COM	445	348	793	9,200,000,000	3,638
BingX	544	182	726	6,300,000,000	3,932
Bitget	605	224	829	6,100,000,000	4,266
LBank	544	136	680	6,100,000,000	4,403
HTX	603	122	725	3,200,000,000	4,416
WhiteBIT	215	26	241	3,000,000,000	4,538
P2B	146	78	224	750,000,000	4,658
Kraken	240	156	396	650,000,000	4,939
Crypto.com Exchange	292	147	439	1,200,000,000	5,392
Bitrue	521	153	674	10,000,000,000	6,290
Phemex	188	218	406	3,500,000,000	6,399
Upbit Indonesia				9,000,000	6,610
BitMart	727	102	829	11,000,000,000	7,585
YoBit	205	42	247	90,000,000	8,751
Coinsbit	81	54	135	240,000,000	9,227
Bitforex		13	13	3,100,000,000	9,991
CoinEx	739	130	869	240,000,000	10,165
Bitso	57	42	99	36,000,000	10,272
Bitfinex	182	91	273	340,000,000	10,769
Binance US				14,000,000	12,850
Poloniex	410	46	456	200,000,000	13,151
Bithumb	257	0	257	420,000,000	13,467

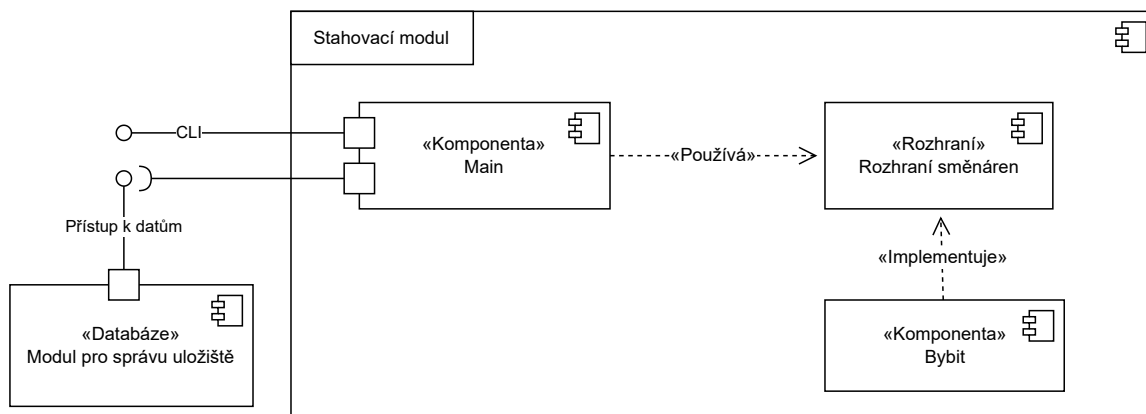
Tabulka 5.1: Informace k jednotlivým směnárnám ze dne 17. 10. 2023

Směnárna	Odkaz na dokumentaci
Bybit	https://bybit-exchange.github.io/docs/v5/intro
Biance	https://binance-docs.github.io/apidocs/spot/en/
Coinbase Exchange	https://docs.cloud.coinbase.com/exchange/docs/welcome
Bitget	https://www.bitget.com/api-doc/
BingX	https://bingx-api.github.io/docs/
Kraken	https://www.kraken.com/features/api-trading
Phemex	https://phemex-docs.github.io/
OKX	https://www.okx.com/docs-v5/en/
DigiFinex	https://docs.digifinex.com/en-ww/spot/v3/rest.html
KuCoin	https://www.kucoin.com/docs/beginners/introduction

Tabulka 5.2: Tabulka obsahující seznam vybraných směnáren a odkazy na dokumentaci k jejich rozhraní

Směnárna	Odkaz na úložiště historických dat
Bybit	https://public.bybit.com/
Biance	https://www.binance.com/en/landing/data

Tabulka 5.3: Tabulka obsahující seznam odkazů na veřejná úložiště směnáren Bybit a Biance



Obrázek 5.2: Diagram komponent Stahovacího modulu

Na obrázku 5.2 je diagram komponent popisující Stahovací modul. Main je spustitelný pomocí CLI a používá Modul pro správu úložiště k ukládání stažených dat. V závislosti na zvolené směnárně se vybere příslušná třída implementující rozhraní pro stahování dat ze směnáren. Tato třída (Bybit) obsahuje metody pro stažení dat a jejich úpravu do potřebného formátu.

5.3 Zpracování dat

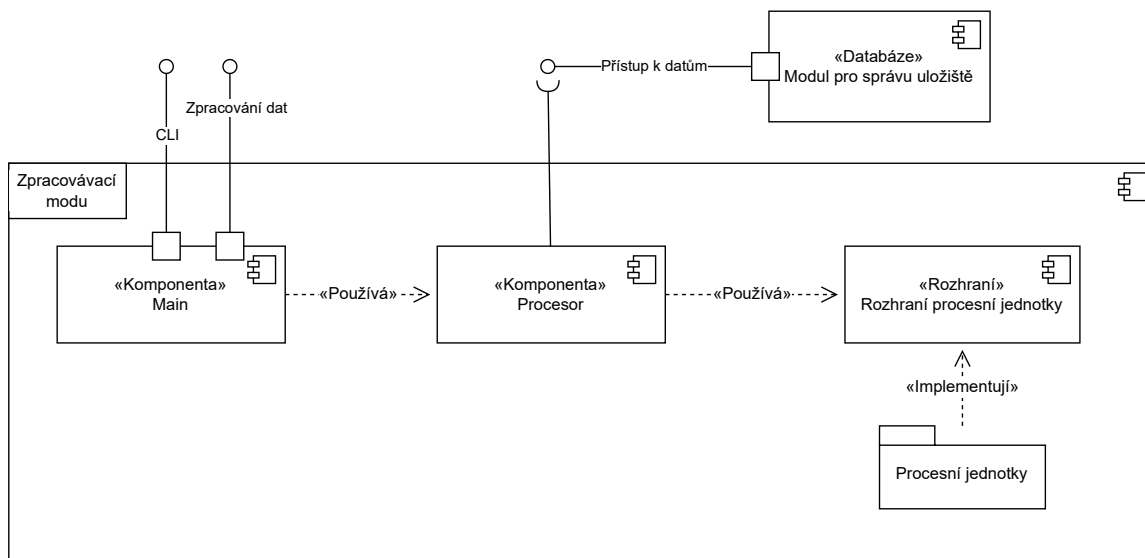
Z jednotlivých transakcí lze dopočítat většinu potřebných a běžně využívaných datových typů (OHLC, objem, ...). Tento framework dokáže tato data dopočítat spolu s dalšími, které si uživatel případně implementuje. Framework tedy poskytuje možnost zpracování dat do libovolného nadefinovaného typu dat pro libovolný časový rámec. Nepoužívá žádné speciální knihovny pro tyto výpočty, vše se zpracovává ze vstupních transakčních dat. Důvodem je kompletní kontrola nad procesem zpracovávání dat.

Většina známých indikátorů jsou zpracovaná data do nějaké podoby, proto se o nich v rámci této práce píše jako o datových typech spolu s dalšími datovými typy, jako jsou například OHLC a transakce.

Pro zachování konzistence dat a umožnění jednoduchého zprovoznění multiprocessingu těchto dat bylo nutné nastavit podmínku vyhodnocování. Uživatel si toto chování může upravit v kódu, ale pro většinu účelů této práce budou využívány časové rámce: měsíc, týden, den a další na ně zarovnatelné. Bylo tedy nutné zajistit pro multiprocessing zpracovávání celého týdne i měsíce a ukládat příslušná data. Proto tento modul zpracovává data zarovnaná pro takové časové rámce.

Zpracovávací modul je opět možné ovládat pomocí CLI, kde si uživatel může specifikovat datové typy pro zpracování; časový úsek, pro který se mají data zpracovat a případně zvolit multiprocessing a zadat počet procesů. Samozřejmě, zpracování probíhá pro zvolený finanční nástroj, který je nutné specifikovat. Modul je schopný zpracovávat data s pomocí událostí z kalendáře událostí popsaného níže, nebo bez něj.

Na obrázku 5.3 je diagram komponent popisující Zpracovávací modul. Main je ovladatelný samostatně pomocí CLI a také nabízí ostatním modulům možnost zpracování dat. Main používá Procesor, který komunikuje s Modulem pro správu úložiště a získává, případně i ukládá, potřebná data. Zpracování pro zvolené datové typy probíhá v jednotlivých Procesních jednotkách, kde každá Procesní jednotka implementuje Rozhraní



Obrázek 5.3: Diagram komponent Zpracovávacího modulu

procesní jednotky. Toto rozhraní obsahuje metody pro získání závislostí k danému datovému typu: inicializační, procesní, evaluační, uklízeční a finalizační metody. Procesní metoda se volá při každé nově přichodící transakci a evaluační při každém události pro vyhodnocení dat. Uklízeční metoda se stará o mazání přebytečných dat. Inicializační je volána úplně na začátku zpracování a finalizační na konci.

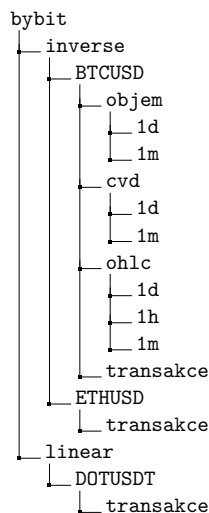
5.4 Ukládání dat

Pro ukládání dat bylo zvažováno použití databáze určené přímo pro časové řady. Pro testování použitelnosti databáze pro framework byla vybrána databáze InfluxDB 3.0⁴, která umožnila ukládání transakcí, jejich zpracování a vizualizaci. Bohužel byla vyhodnocena jako nevyhovující skrze enormní množství dat získávaných ze směnárén a neustálou manipulaci s těmito daty. Databáze sice fungovala rychle, ale nastávaly problémy při manipulaci s daty. Z nutnosti instalace databáze a jejích limitací při snaze zachovat zprovoznění frameworku co nejjednodušší bylo navrženo ukládání dat do souboru, které umožňuje jednoduchou přenositelnost vybraných dat, manipulaci s nimi a snadný přístup. Díky tomuto přístupu má uživatel podrobný přehled nad všemi daty frameworku přímo ve zvoleném adresáři.

Pro ukládání dat do lokálního úložiště bylo potřeba definovat jeho strukturu. Nejvhodnější se ukázala struktura podle označení finančního nástroje, následně datového typu, a nakonec dle zvoleného časového rámce. Na obrázku 5.4 je možné vidět příklad adresářů v úložišti.

Různé datové typy jsou nejčastěji reprezentovány množinou dvojic: jednou číselnou hodnotou pro každý časový rámec. Ale samozřejmě jsou zde i datové typy jako například OHLC a MACD, které mají více hodnot pro každý časový rámec. Také bylo umožněno ukládat datové typy bez určeného časového rámce jako například jednotlivé transakce. Z těchto důvodů bylo nutné pro každý datový typ napsat samostatnou parsovací jednotku, která mapuje uložená data na objekty jednotlivých datových typů a zpět. Parsovací jednotky také umožňují efektivní vyhledávání dat z časových intervalů.

⁴<https://www.influxdata.com/>



Obrázek 5.4: Ukázka adresářů lokálního úložiště, jak Modul pro správu úložiště ukládá data. Na obrázku je vidět stav při stažení transakcí pro tři finanční nástroje a zpracování dat pro jeden z nich.

Při zkoumání možnosti ukládání dat bylo zjištěno, že pro Python existuje více efektivních typů souborů pro ukládání dat. Na základě vlastního testování a různých srovnání na internetu⁵ byly vyhodnoceny formáty Parquet a Feather jako vyhovující. Oba tyto formáty jsou mnohonásobně rychlejší jak pro zápis, tak pro čtení než klasický běžně používaný CSV formát. Sice se jedná o binární nečitelné soubory, ale existuje mnoho způsobů pro zobrazení těchto souborů, jako například pomocí rozšíření `dvirtz.parquet-viewer`⁶ pro Visual Studio Code⁷.

Modul pro správu úložiště tedy umožňuje ukládání různých dat do lokálního úložiště. Také umožňuje načítání těchto dat, a to pro libovolně zvolené časové úseky. Během práce s úložištěm je schopný hodnoty efektivně mapovat do souborů a zpět.

Diagram komponent Modulu pro správu úložiště je možné vidět na obrázku 5.5. Modul obsahuje hlavní komponentu `Main` umožňující ostatním modulům načítat, vyhledávat a ukládat libovolná data. `Main` používá k serializaci a deserializaci dat `Parsovací jednotky`, které jsou unikátní pro každý datový typ a obsahují generické metody. To znamená, že napsání nové `Parsovací jednotky` pro většinu datových typů spočívá pouze v definování struktury dat pro datový typ (float, int, typ třídy) a případném pojmenování hodnot. `Main` také používá `Parsovací jednotky` k efektivnímu vyhledávání potřebných dat. K ukládání a načítání dat jsou implementované metody pro `Lokální úložiště`, které `Main` také používá.

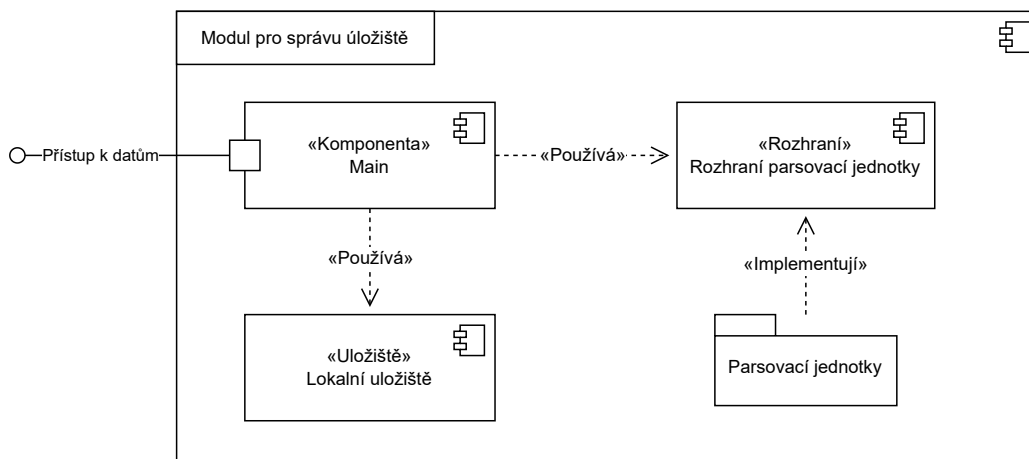
5.5 Zobrazování dat

Pro jednodušší práci s daty, zejména jejich kontrolu, je zapotřebí mít nástroj pro jejich jednoduchou reprezentaci. Pro tento framework je nejvíce vhodná reprezentace ve formě

⁵<https://www.linkedin.com/pulse/comparative-study-among-csv-feather-pickle-parquet-loyola-gonz%C3%A1lez/>

⁶<https://marketplace.visualstudio.com/items?itemName=dvirtz.parquet-viewer>

⁷<https://code.visualstudio.com/>



Obrázek 5.5: Diagram komponent Modulu pro správu úložiště

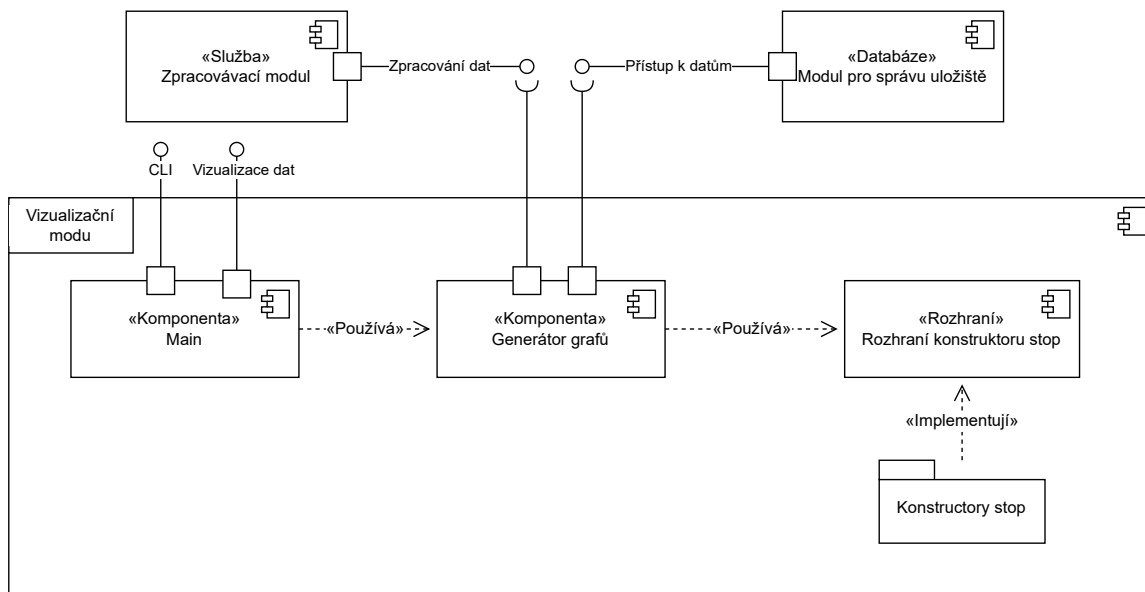
grafu. Je vhodné mít možnost každý zpracovaný datový typ vizualizovat. Protože uživateli bylo umožněno napsat si vlastní procesory pro nové datové typy, tento modul umožňuje uživateli si tyto nové datové typy zobrazit.

Aby byl modul uživatelsky přívětivý, byly implementovány tři možnosti jeho užití:

- První, uživatelsky nejjednodušší, umožňuje zobrazit nezpracovaná data, která se automaticky zpracují z jednotlivých uložených transakcí pomocí **Zpracovávacího modulu**. Stačí specifikovat, jaká data se mají zobrazit, a modul se o veškeré zpracování dat postará, pokud jsou dostupné jednotlivé transakce. Tato možnost zde je, aby uživatel nebyl před každým zobrazením a testováním nových dat nucen si tato data zpracovávat v samostatném modulu.
- Druhá možnost spočívá v definici dat pro zobrazení a jejich následné načtení z úložiště a zobrazení. Tento způsob vizualizace dat je velice efektivní, protože pouze načítá data, která byla dříve předzpracována pomocí **Zpracovávacího modulu** a uložena do lokálního úložiště.
- Třetí možnost, která se například uplatňuje při vizualizaci výsledků ze samotného backtestingu, spočívá v předání dat k zobrazení z jiného modulu. Po specifikaci dat pro zobrazení jiným modulem, je vyžadováno poskytnutí odkazu na instanci **Zpracovávacího modulu** s již zpracovanými daty, ze kterého budou data načtena a následně vizualizována.

Kvůli specifickým požadavkům každého datového typu na zobrazování, bylo nutné napsat jednotlivé třídy vytvářející grafové stopy⁸. Většina stop má obdobné chování, proto bylo napsáno několik generických metod pro tvorbu stop, které je možné využít. Například pro jednoduchou stopu obsahující jednu čáru, není již nutné nic implementovat, pouze je možné nastavit základní barvu, umístění v grafu a případně specifikovat další parametry. Pro složitější datové typy, jako například pro MACD, bylo nutné dopsat část kódu určující, které stopy se mají vytvořit a jakým způsobem. Samozřejmě se jednalo pouze o definování stop použitím generických metod. Tedy například pro MACD se vytvoří tři stopy (MACD čára, signální čára a histogram).

⁸Grafová stopa (trace) označuje jednotlivé komponenty grafu, které reprezentují různé datové řady nebo typy vizualizace.



Obrázek 5.6: Diagram komponent Vizualizačního modulu

Vizualizační modul obsahuje CLI pro ovládání uživatelem. Je jej také možné používat i bez CLI přímo z jiných modulů. Umožňuje načtení zvolených dat z Lokálního úložiště a případně i jejich zpracování stejně jako načítání dat z předaného Zpracovávacího modulu. Tato data následně vykresluje do příslušných grafů a umožňuje uložení grafu do souboru ve formátu `.html`.

Na obrázku 5.6 je zobrazen diagram komponent Vizualizačního modulu. Tento modul obsahuje Main ovladatelný pomocí CLI a umožňující vizualizaci dat ostatním modulům. K vizualizaci používá Generátor grafů. Generátor grafů je na základě požadavků schopen, s možností využití Modulu pro správu úložiště a Zpracovávacího modulu, získat všechna potřebná data. Po získání všech dat k vizualizaci, tento generátor použije Konstruktory stop pro jednotlivé datové typy k tvorbě stop pro výsledný graf. Konstruktory stop implementují Rozhraní konstruktoru stop, ve kterém jsou obsaženy metody pro přípravu dat, generování jednotlivých stop a hlavní metoda vracející seznam stop pro graf.

5.6 Backtesting

Dalším krokem byla implementace Backtestovacího modulu. Bylo potřeba navrhnout vhodnou reprezentaci strategií, které se mají testovat, a celou architekturu pro simulaci. Simulace obsahuje virtuální směnárnu, postupné dávkování dat, zpracovávání těchto dat, vyhodnocování zpracovaných dat na základě indikátorů, evaluaci indikátorů a dalších potřebných informací pro virtuální směnárnu a každou strategii, posílání signálů ze strategií do virtuální směnárny pro tvorbu a manipulaci s pozicemi, vyhodnocování žádostí na straně virtuální směnárny a zpoždování těchto vyhodnocování. Všechny tyto části simulace jsou následně důkladně rozebrány.

Kromě použití Backtestovacího modulu je uživatelům umožněno psaní vlastních simulací s využitím síly frameworku. Uživatel tedy může obejít definování strategií a přímo přistoupit ke zpracovaným datům, která může využít k detekování zajímavých transakcí

a potenciálních vstupů a výstupů z pozic. Samozřejmě, toto umožňuje uživateli obejít postupné zpracování transakcí. Tím se může do simulace vnést velké množství nesrovnalostí, jako například tvorba pozic na základě dat, která v daný okamžik ještě nebyla dostupná. Vlastní backtestování umožňuje rychlé prototypování a testování před napsáním kompletní strategie.

5.6.1 Indikátory

V sekci 5.3 bylo zmíněno, že se v této práci běžně používané indikátory (MACD, SMA, ...) budou nazývat datové typy, protože se nejedná o nic víc než jen o aplikaci známých vzorců. V následujícím textu této práce se pojem indikátor používá pro objekt s vnitřním stavem, který může být buď validní, nebo nevalidní. Jedná se o evaluaci různých dat datových typů na základě vnitřních podmínek indikátoru.

Každý indikátor má identifikátor, typ a data, kde data jsou libovolný počet vstupních parametrů. Uživatel si může napsat vlastní indikátory a využít dědičnost ze základních indikátorů a generických metod. V indikátoru musí specifikovat, jaké datové typy bude indikátor používat a za jakých podmínek se stane validním. Evaluace indikátorů probíhá při každém tiku programu, kde v samotné evaluační metodě je možné specifikovat, že se má indikátor vyhodnocovat pouze za určitých podmínek-například na začátku každého nového časového rámce.

Příkladem indikátoru může být například indikátor pro porovnávání libovolných klouzavých průměrů. Tento indikátor vyžaduje následující vstupní hodnoty: referenční klouzavý průměr k porovnávání, druhou hodnotu k porovnání (může se jednat například o jiný klouzavý průměr, nebo hodnotu z OHLC) a operátor, kde operátor definuje, kdy bude indikátor ve validním stavu. Operátor tedy může být nastavený například pro stejné hodnoty, pro jednu hodnotu větší než druhou, nebo, že jedna hodnota musela v předchozím časovém rámci překročit druhou. Také je možné specifikovat časový rámec, nebo se bere referenční, na který je nastavená celá strategie.

5.6.2 Strategie a jejich reprezentace

Celá strategie se skládá z několika prvků. Je nutné specifikovat, kdy se má na základě strategie vytvořit pozice a o jaký typ pozice se bude jednat. Každé pozici je možné nastavit, kdy se má zrušit (Stop Loss) a kdy se má brát profit (Take Profit). Dále bylo nutné vymyslet, jak tyto pozice následně upravovat-například, kdy a jak posouvat Stop Loss a Take Profit, nebo za jakých podmínek celou pozici zavřít. Samozřejmě je také stěžejní, předem definovat všechny další potřebné parametry.

Parametry strategie

U každé strategie je nutné specifikovat její identifikátor. Vzhledem k tomu, že většina indikátorů vyžaduje tvorbu dat ve stanovených časových rámcích, je možné definovat tento časový rámec pro celou strategii s možností jeho předefinování v dílčích indikátorech. Také se strategii musí nastavit, pro jaké finanční nástroje se má vyhodnocovat. Kvůli omezenosti zdrojů a náročnosti implementace je simulace prováděna nad maximálně jedním finančním nástrojem pro jednu instanci běhu.

Hlavními parametry jsou výše popsané indikátory, kde každý vyžaduje vlastní identifikátor, typ a data. K indikátorům je potřeba specifikovat podmínky pro braní pozic. Po každý typ pozice (Long, Short) je možné specifikovat seznam podmínek. Pro zjednodu-

šení zápisu byla zvolena disjunktivní normální forma (DNF) zápisu těchto podmínek. To znamená, že je možné psát komplexní podmínky, a to pouze ve dvou úrovních zanoření. Označíme-li libovolné dílčí podmínky jednotlivými písmeny, pak zápis komplexní podmínky může vypadat například následovně:

$$((X \wedge Y \wedge Z) \vee (X \wedge Y \wedge U) \vee (U \wedge V))$$

Každá dílčí podmínka má vlastní typ a dále potřebné parametry pro daný typ. Bylo nutné implementovat dva hlavní typy:

- Podmínka pro porovnání hodnot vyhodnocených indikátorů. K tomuto typu je nutné specifikovat identifikátor indikátoru a referenční hodnotu (validní, nebo nevalidní) pro porovnání.
- Ostatní podmínky vyhodnocené na základě jiných informací, než ze zpracovaných dat. Příkladem může být podmínka, že aktuální vyhodnocení probíhá na začátku časového rámce, nebo podmínka na počet aktivních pozic obsahující operátor a referenční hodnotu k porovnání.

Volitelnými parametry jsou dále hodnoty pro brání profitů a Stop Loss, které je možné uvést jako přesnou hodnotu, nebo jako procenta, o kolik se musí cena posunout. Například pro Stop Loss by hodnota 20 % znamenala, že se cena musí posunout o 20 % do ztráty, aby se pozice sama uzavřela na virtuální směnárně.

Kromě indikátorů je možné obdobně definovat i strategie pro manipulaci s vytvořenými pozicemi, které obsahují identifikátor, typ a specifická data pro každou strategii. To přináší nutnost definovat podmínky pro vyhodnocování těchto strategií pro manipulaci s pozicemi. Podmínky se definují stejně, jako pro tvorbu pozic. Díky této funkcionalitě je možné, po vytvoření pozice, posunovat s jednotlivými hodnotami, ať už pro brání profitu, nebo pro Stop Loss. Také je umožněno například okamžité uzavření pozice na základě vyhodnocené podmínky, nebo využít tzv. Trailing Stop Loss.

Každou pozici je možné uzavřít třemi způsoby, a tedy je vhodné alespoň jeden z nich pro každou strategii definovat.

- Pozici je možné uzavřít, nebo částečně uzavřít bráním profitu díky předem nastavené hodnotě ceny aktiva. Těchto hodnot může být libovolné množství s různými váhami a posílají se do virtuální směnárně současně s vytvářením pozice.
- Druhý způsob uzavření pozice je pomocí strategie pro manipulaci s vytvořenými pozicemi. Tento způsob vyžaduje vyhodnocování podmínek na straně strategie pro každou aktivní pozici zvlášť. Pokud jsou podmínky označeny za validní, je poslán požadavek na virtuální směnárně se žádostí o uzavření pozice.
- Poslední způsob uzavření pozice je nežádoucí, ale většinou předejde větší ztrátě kapitálu než té, která je nastavená jako maximální riziko pro vytvořenou pozici. Jedná se o Stop Loss. Jeho cenová hranice je zaslána na virtuální směnárně spolu s hodnotami pro brání profitů.

Dalším volitelným parametrem je plán poplatků, který určí, jaké poplatky ze směnárně se mají aplikovat na jednotlivé transakce.

Reprezentace strategie/í v konfiguračním souboru

Pro reprezentaci strategie bylo zvažováno použití samotného Python kódu, kde by si uživatel nadefinoval každou strategii, jako samostatnou třídu. Vzhledem k tomu, že se jedná pouze o specifikování různých parametrů, byla tato metoda označena za zbytečnou. Druhou zvažovanou reprezentací byl konfigurační soubor. Kvůli rozšířenosti formátu JSON a jeho přehlednosti, byl zvolen jako formát pro konfigurační soubory strategií. Dalšími zvažovanými formáty byly různé značkovací formáty souborů a konfigurační soubory (XML, YAML⁹, TOML¹⁰, ...).

Samotný JSON formát neumožňuje psát komentáře a podporuje pouze základní formáty. Uživatelům bylo umožněno psaní komentářů v souborech při použití prefixu `__` v klíči komentáře: `{"__klic_komentare" : "Toto je komentar"}`. Také bylo umožněno použití funkčních prvků `_UNPACK` a `_PARAM`, mající v prefixu `_`, kde `_PARAM` představuje parametr určený k optimalizaci popsán více v sekci 5.7 a `_UNPACK` slouží k definici hodnot pro dosažení za tento prvek.

Při načítání strategií je nutné prvně odstranit komentáře a následně rozbalit hodnoty definované k rozbalení. Ukázkou definice tohoto prvku může být například stanovení hodnot časového okna strategie: `{"_UNPACK" : ["5m", "10m", "15m"]}`. Pro všechny takové prvky ve strategii se následně vytvoří množina všech možných permutací prvků a vytvoří se příslušný počet strategií s dosazenými hodnotami z této množiny. Bohužel, tento přístup způsobuje enormní nárůst počtu strategií. Například můžeme uvažovat 10 parametrů, kde každý parametr může nabývat 10 hodnot. Výsledný počet roz-generovaných strategií by byl 10^{10} . Bohužel framework není schopný počítat s velkým množstvím strategií v reálném čase. V případě, že uživatel chce optimalizovat dva až tři parametry, či více parametrů s málo hodnotami, je tento přístup pro optimalizaci nevhodnější.

Ve výpise 5.1 je ukázka definice strategie (přeložená do českého jazyka.) Tato ukázka slouží pouze k demonstraci zápisu.

5.6.3 Virtuální směnárna

Funkcionalita pro tvorbu pozic je navržena, ale pro komplexní simulaci je ještě zapotřebí přidat další prvek. Podstatnou součástí simulace je virtuální směnárna, která kontroluje, zda-li je pozici možné vytvořit na základě požadavků ze strategie. Také pozice vytváří a uchovává si o nich informace. Dále na základě požadavků ze strategií s těmito pozicemi manipuluje. Kromě toho může do simulace vnést zpoždění, které by v reálném systému vznikalo (doba příchodu dat do systému + doba zpracování dat + doba odchodu požadavků na směnárnu + doba vyhodnocení na straně směnárny). Další funkcionalitou virtuální směnárny je určování typu a výše poplatků na základě prováděné akce.

Pro uchování informací o pozicích je nutné pro každou strategii vytvořit jednu peněžku, do které se budou ukládat informace o každé jí příslušné pozici. Tyto informace zahrnují sériové číslo pozice, unikátní v rámci celé simulace, typ pozice (Long, Short), čas vzniku pozice, cenu aktiva při vzniku pozice, čas uzavření pozice, cenu aktiva při uzavření pozice, způsob uzavření pozice (normální, Stop Loss), velikost pozice, použitý leverage, odkaz na peněžku, výše poplatku za vytvoření pozice, počáteční a aktuální hodnoty pro braní profitů, počáteční a aktuální Stop Loss a seznam všech částečných uzavření pozice. Každé částečné uzavření pozice obsahuje čas uzavření a cenu aktiva při uzavření pozice,

⁹<https://yaml.org/>

¹⁰<https://toml.io/>

```

1 {
2   "id": "Ukazka_definice_strategie", "jmeno": "Ukazkazkova strategie",
3   "financni_nastroje": [ "bybit_inverse_BTCUSD", "bybit_linear_BTCUSD" ],
4   "casovy_ramec": { "_UNPACK": [ "5m", "15m", "30m" ] },
5   "plan_poplatku": { "_UNPACK": [ "-", "VIP3" ] },
6   "pocatecni_kapital": 1000, "velikost_pozice": 100, "leverage": 5,
7   "stop_loss_v_procentech": 0.2,
8   "indikatory": [
9     {
10      "id": "sma20_nad_ema50_indikator",
11      "typ": "mas_porovnavac",
12      "data": { "ma1": "sma_20", "ma2": "ema_50", "operator": "zkrizene_nahoru" }
13    },
14    {
15      "id": "ema50_nad_sma20_indikator",
16      "typ": "mas_porovnavac",
17      "data": { "ma1": "ema_50", "ma2": "sma_20", "operator": "zkrizene_nahoru" }
18    }
19  ],
20  "strategie_manipulujici_s_pozicemi": [
21    {
22      "id": "strategie_posun_stop_lossu",
23      "type": "strategie_trailing_stop_loss",
24      "priorita": 1, "data": { "hodnota": 0.2 }
25    }
26  ],
27  "podminky_pro_tvorbu_pozic": {
28    "long": [
29      { "typ": "indikator", "indikator": "sma20_nad_ema50_indikator", "hodnota": "pravda" },
30      { "typ": "podminka", "podminka": "pocet_aktivnich_pozic", "operator": "rovnase", "
31      hodnota": 0 },
32      { "typ": "podminka", "podminka": "zacatek_casoveho_ramce" }
33    ]
34  },
35  "short": [ ]
36  },
37  "podminky_pro_strategie_manipulujici_s_pozicemi": {
38    "strategie_zruseni_pozice": [
39      { "typ": "indikator", "indikator": "id_indikatoru", "hodnota": "nepravda"},
40      { "typ": "podminka", "podminka": "zacatek_casoveho_ramce" }
41    ]
42  }
43  ],
44  },
45  "brani_profitu": [
46    { "cil_v_procentech": 0.1, "kvantita": 20 },
47    { "cil": 50, "kvantita": 80 }
48  ]
49 }

```

Výpis 5.1: Ukázka definice strategie. Jedná se o nefunkční strategii sloužící pouze k demonstraci zápisu strategie. Na začátku probíhá definice parametrů strategie. Je možné si všimnout, že tato strategie se pro backtestování roz-generuje na šest různých, protože obsahuje dva funkční prvky `_UNPACK` s postupně třemi a dvěma různými hodnotami pro časový rámec strategie a plán poplatků. Následně se definují jednotlivé indikátory pro strategii a jednotlivé strategie manipulující s pozicemi. S indikátory se, za pomoci jejich identifikátorů, skládají podmínky pro tvorbu pozic a podmínky pro aktivaci strategií manipulujících s pozicemi. Na konci je ukázána definice hodnot pro brání profitu.

Plán poplatků	Spot		Perpetual & Futures		Options	
	Taker Fee	Maker Fee	Taker Fee	Maker Fee	Taker Fee	Maker Fee
VIP 0	0.1000 %	0.1000 %	0.0550 %	0.0200 %	0.0200 %	0.0200 %
VIP 1	0.0800 %	0.0675 %	0.0400 %	0.0180 %	0.0200 %	0.0150 %
VIP 2	0.0775 %	0.0650 %	0.0375 %	0.0160 %	0.0200 %	0.0150 %
VIP 3	0.0750 %	0.0625 %	0.0350 %	0.0140 %	0.0200 %	0.0150 %
VIP 4	0.0600 %	0.0500 %	0.0320 %	0.0120 %	0.0180 %	0.0150 %
VIP 5	0.0500 %	0.0400 %	0.0320 %	0.0100 %	0.0150 %	0.0100 %
Supreme VIP	0.0450 %	0.0300 %	0.0300 %	0.0000 %	0.0150 %	0.0020 %
Pro 1	0.0600 %	0.0500 %	0.0320 %	0.0100 %	0.0180 %	0.0150 %
Pro 2	0.0500 %	0.0400 %	0.0320 %	0.0050 %	0.0150 %	0.0100 %
Pro 3	0.0450 %	0.0300 %	0.0300 %	0.0000 %	0.0150 %	0.0020 %
Pro 4	0.0400 %	0.0300 %	0.0275 %	0.0000 %	0.0100 %	0.0020 %
Pro 5	0.0150 %	0.0050 %	0.0250 %	0.0000 %	0.0100 %	0.0000 %

Tabulka 5.4: Seznam jednotlivých plánů poplatků na směnárně Bybit z dubna 2024

velikost ukončení pozice (váhu) do maximální výše celkové velikosti pozice, typ uzavření pozice (pro určení typu poplatku) a výši poplatku za uzavření.

Z uložených informací ke každé pozici je následně možné dopočítat vše potřebné k ohodnocení strategie. Příkladem může být celkový profit z pozice, dílčí pohyby v peněžence na základě částečných uzavření pozice, celkové trvání pozice, různé informace o poplatcích, počáteční risk kapitálu. Kde to je smysluplné, je možné získat hodnoty jak v absolutní hodnotě, tak i v procentech. Také jsou dostupné hodnoty se započítanými poplatky, stejně jako hodnoty zanedbávající poplatky.

Každá reálná směnárna má nastavené vlastní poplatky a ty se mohou lišit v závislosti na různých faktorech obchodníka, smluvních podmínkách, a především na typu finančního nástroje. Před začátkem simulace je možné nadefinovat tyto poplatky pomocí zvolení plánu poplatků pro simulovanou směnárnu. Každý plán poplatků se skládá ze dvou hodnot, a to Taker Fee (poplatek za transakci odebírající likviditu z trhu) a Maker Fee (za transakci přidávající likviditu do trhu.) Příklad poplatku pro směnárnu Bybit je možný vidět v tabulce 5.4 z dubna 2024¹¹.

Zpoždění událostí jako je tvorba pozice, manipulace s nimi, jejich uzavírání a další, je podstatná část simulace. Kalendář událostí je tedy schopen tyto události zaregistrovat a následně správně vyhodnotit zavoláním příslušných metod.

5.6.4 Data v simulaci

Na reálných směnárnách se vyhodnocují jednotlivé transakce a některé směnárny tyto transakce následně posílají odběratelům, kteří si zažádali o tato data. Samozřejmě dochází ke zpoždění v řádech milisekund. `Backtestovací` modul se v tomto frameworku snaží o simulaci reálné situace, kdy odebírá data s jednotlivými transakcemi ze směnárny a na základě jejich vyhodnocení se rozhoduje o následných krocích. Je tedy v případě backtestování nutné zajistit postupné vyhodnocování těchto transakcí tak, jak by je v reálu bylo možné zpracovávat.

¹¹<https://www.bybit.com/en/help-center/article/Trading-Fee-Structure>

Hlavní jednotkou simulace je generátor postupně dávkující jednotlivé transakce v pořadí za sebou. Další podstatnou jednotkou je kalendář událostí umožňující simulaci reagovat na základě uplynulého času simulace (myšleno času, který by v reálu uplynul, ne doby běhu simulace.) Tento kalendář událostí například umožňuje evaluaci stavu simulace na základě začátku nového časového rámce, nebo opoždění transakcí vyhodnocovaných ve virtuální směnárně.

V simulaci se před evaluací každé transakce ověří, jestli není nějaká událost v kalendáři událostí předcházející příchodu této transakce. Pokud není, tak se provede evaluace transakce. V opačném případě se provede vyhodnocení události.

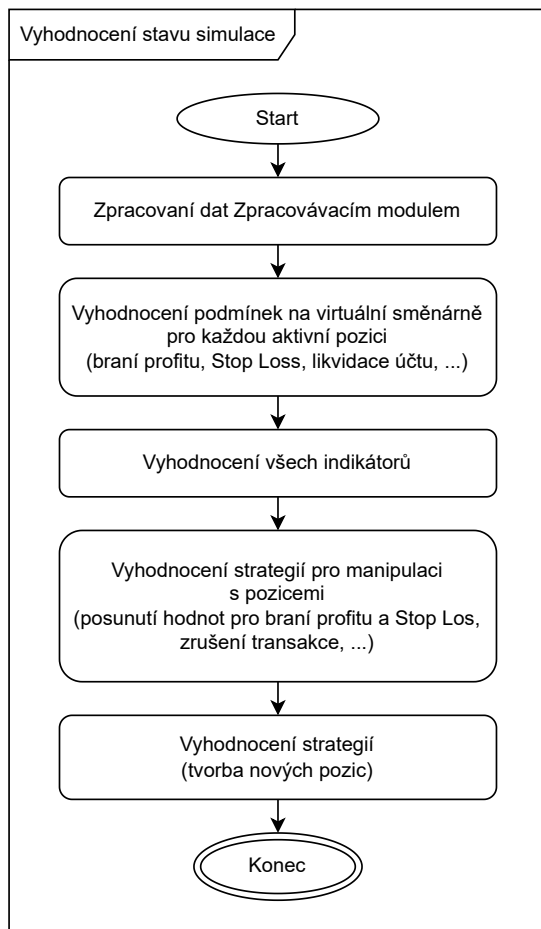
V případě vyhodnocování transakce, nebo události, sloužící k evaluaci na základě začátku nového časového rámce, proběhne vyhodnocení stavu simulace. Tento průběh je zobrazen vývojovým diagramem na obrázku 5.7. Jako první proběhne zpracování dat ve **Zpracovávacím modulu**. Následně se ve virtuální směnárně vyhodnotí podmínky pro uzavření pozice (braní profitu, Stop Loss, případná likvidace účtu) pro každou aktivní pozici a případně se pozice uzavře. Následuje vyhodnocení všech indikátorů v pořadí závislém na jejich závislostech. Po vyhodnocení indikátorů je možné ohodnotit jednotlivé podmínky na straně strategií. Nejprve pro každou aktivní pozici proběhne evaluace podmínek pro strategie manipulující s těmito pozicemi a případně se provedou tyto manipulace. Následně se vyhodnotí podmínky pro tvorbu pozic.

Do kalendáře událostí se ukládají různé události s rozdílnou funkcionalitou. Je nutné tyto události rozlišovat, zdali se má vyhodnotit celý stav simulace, nebo jen zavolat nějaká metoda. Pokud nastala událost například pro simulaci opoždění zpětné vazby ze směnárny, tak probíhá pouze vyhodnocení této události bez vyhodnocování stavu celé simulace. Toto chování je možné vidět na diagramu na obrázku 5.8.

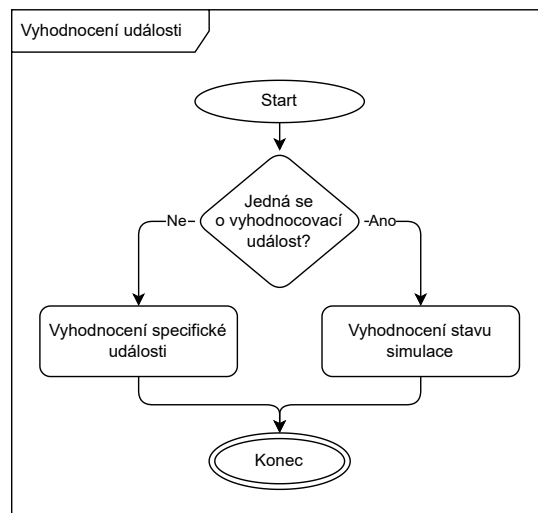
Hlavní prvky simulace a její průběh je možné vidět na vývojovém diagramu na obrázku 5.9. Celá simulace začíná inicializací. Následně probíhá vyhodnocení všech transakcí z datasetu pro backtesting. Před vyhodnocením každé transakce probíhá kontrola existence události předcházející vyhodnocované transakci, která by v reálném čase nastala před příchodem takové transakce. Pokud existuje jedna či více událostí, tak se postupně vyhodnotí, jak bylo popsáno v diagramu na obrázku 5.8. Následně probíhá vyhodnocení stavu celé simulace podle diagramu na obrázku 5.7. Po projití všech transakcí se vyhodnotí všechny zbývající nevyhodnocené události z kalendáře událostí a následuje finalizace celé simulace.

5.6.5 Návrh modulu

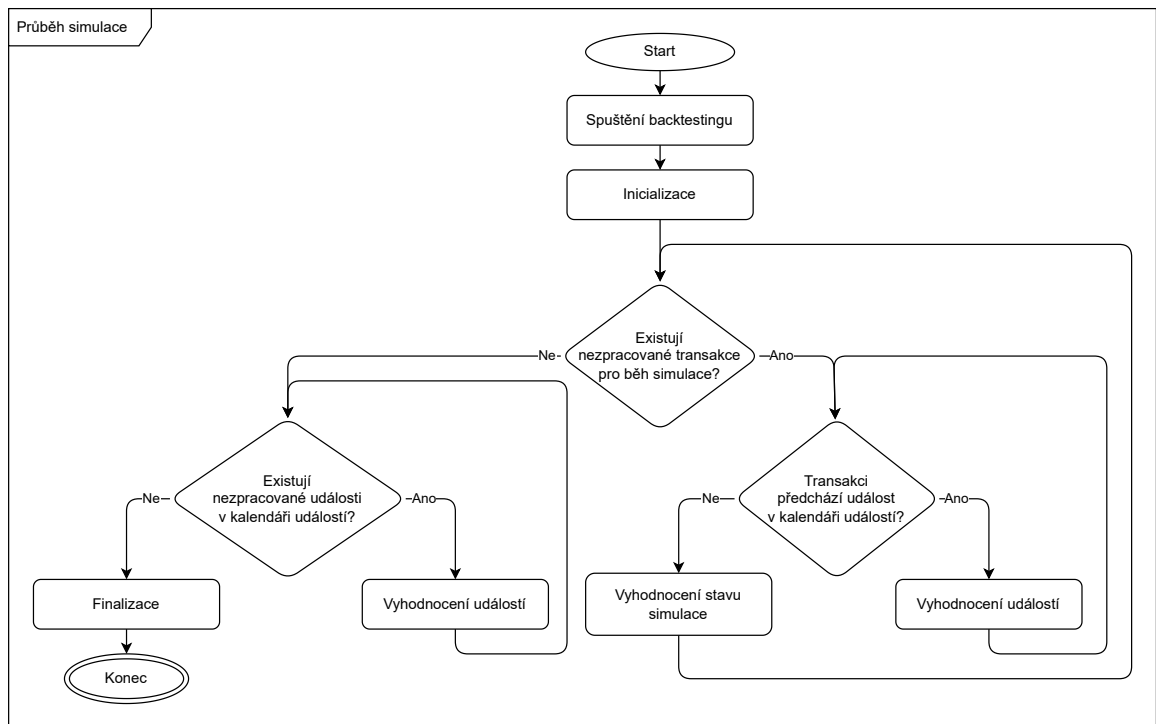
Na obrázku 5.10 je zobrazen diagram komponent **Backtestovacího modulu**. Tento modul obsahuje **Main** ovladatelný pomocí CLI a umožňující provádění simulace strategií ostatním modulům. **Main** používá **Vizualizační modul** k reprezentaci výsledků simulace, které se vyhodnotí ve **Vyhodnocovači**. Simulace probíhá v **Procesoru**, který je ovládán z **Main** komponenty. **Procesor** používá **Kalendář událostí** a získává data v podobě jednotlivých transakcí, které následně zpracovává pomocí **Zpracovávacího modulu**. Během simulace také používá seznam jednotlivých **Strategií** a příslušnou **Virtuální směnárnu**.



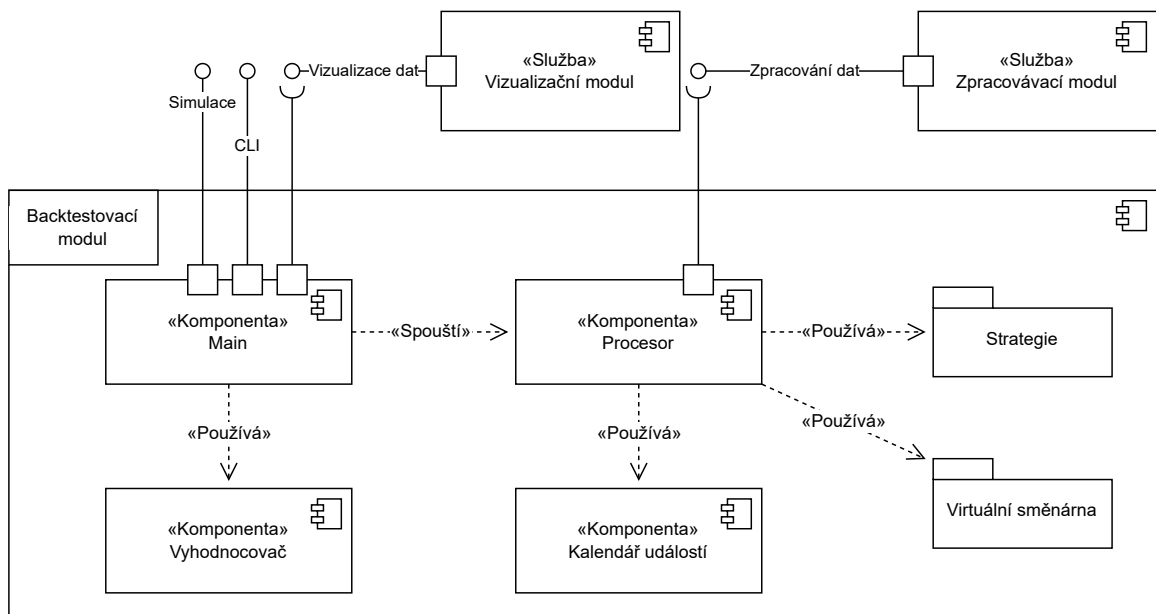
Obrázek 5.7: Vývojový diagram popisující vyhodnocení stavu simulace



Obrázek 5.8: Vývojový diagram popisující vyhodnocení události



Obrázek 5.9: Vývojový diagram popisující průběh simulace. Po spuštění backtestingu a inicializaci se před vyhodnocením každé transakce vyhodnotí události jí předcházející. Ke konci simulace se pak před provedení finalizace se vyhodnotí všechny události, které je zapotřebí zpracovat.



Obrázek 5.10: Diagram komponent Backtestovacího modulu

5.6.6 Vlastní backtesting

Vzhledem k rozsáhlosti frameworku a jeho schopnosti uchovávat zpracovaná data pro libovolné datové typy by byla škoda neumožnit uživateli s těmito daty dále pracovat. Proto je uživateli se znalostmi programování poskytnuto několik užitečných metod pro vyhledávání různých dat v libovolných časových intervalech. Příkladem mohou být metody pracující na libovolných časových intervalech.

- Metoda detekující přesný čas, kdy byla překročena stanovená cena aktiva.
- Metoda získávající přesnou minimální a maximální hodnotu aktiva.
- Metoda počítající přesnou hodnotu objemu transakcí, a to nákupu nebo prodeje.
- Metody pro počítání procent, například pro určení poplatků či ziskovosti.

Tento přístup umožňuje uživateli mít kompletní kontrolu nad tím, co chce testovat. Není nijak omezen, což samozřejmě přináší možnosti zanášení chyb při vyhodnocování. Tento postup je ideální pro navrhování nových datových typů, nových indikátorů a prototypování strategií. Také umožňuje testovat efektivně dlouhodobé strategie u kterých není vyžadována maximální přesnost. Samozřejmě, přesnosti je i v tomto přístupu možné dosáhnout správným počítáním reflektující realitu.

Příklad použití může být například pro dlouhodobé strategie, jako je DCA (dollar-cost averaging) pro pravidelné vkládání kapitálu do aktiv za férovou cenu. Také je možné brát ohled na jakákoliv různá data, která se dají dopočítat, nebo má uživatel k dispozici. Dalším možným použitím může být efektivní nalezení libovolných levelů (významných hodnot) a nalezení jejich proražení, od kterého je možné dopočítat všechna další data, která uživatel může potřebovat.

5.7 Vylepšování parametrů strategií s pomocí evolučních algoritmů

Poslední modul obsažený ve frameworku je **Optimalizační modul**. Pro získání optimální strategie je zapotřebí vykonat velké množství práce. Je potřeba navrhnout strategii její parametry a stanovit, ve kterých podmínkách má fungovat. Bohužel většinou každý finanční nástroj je unikátní a nelze použít strategii optimalizovanou na jeden finanční nástroj na všechny ostatní. Jsou finanční nástroje, mezi kterými je jistá korelace, avšak není nikdy dokonalá. Také trh se nechová pořád stejně. Dají se detekovat různé trendy a většinou každý má své specifické chování a je zapotřebí aplikovat různé strategie. Kvalita strategie se také odvíjí od typu použitých indikátorů a způsobu jejich použití. A samozřejmě každý indikátor může mít spoustu parametrů ovlivňující jeho chování.

Komplexnost celého procesu nalezení optimální strategie je příliš velká pro tento framework. Cílem tohoto modulu není nalezení optimální strategie pro zvolený finanční nástroj. Cílem je nalézt optimální parametry indikátorů předdefinované strategie pro zvolený finanční nástroj na zvoleném časovém intervalu. **Optimalizační modul** je schopný s pomocí evolučních algoritmů efektivně optimalizovat parametry pro strategii za účelem dosažení uživatelského cíle. Cíl může být například maximalizace zisku s předem stanoveným maximálním rizikem, maximalizace profitu za čas použití kapitálu v pozicích, maximalizace objemu transakcí za cílem získání lepších podmínek od směnárny a mnoho dalších.

Ve stručnosti, tento modul umožňuje uživateli nahradit v konfiguraci strategie libovolné hodnoty za parametry a tyto parametry následně blíže omezit. Dále umožňuje nastavení parametrů `Optimalizačního modulu` včetně fitness metody. Výsledkem je strategie s dosaženými hodnotami pro maximalizaci fitness. Efektivita modulu závisí na různých faktorech, jako je nastavení fitness metody, počet iterací, množství poskytnutých dat a parametry učicího algoritmu.

Během učení může nastat problém s případným přetrénováním na poskytnutých datech, proto je klíčové provést i dopředné testování. Z toho vyplývá, že by se měla vynechat část aktuálních dat z datasetu pro optimalizaci a po získání parametrů z `Optimalizačního modulu` provést klasický backtesting na těchto vynechaných datech a tím ověřit funkčnost strategie. Modul také pravidelně po každé iteraci ukládá informace o testovaných strategiích, pozicích vytvořených těmito strategiemi a vyhodnocení jednotlivých strategií obsahující vybrané metriky. Průběžné ukládání je vhodné kvůli možnému přerušení běhu programu a následné ztrátě všech dat.

5.7.1 Parametry

Při analýze existujících řešení se u nástroje Freqtrade ukázalo, že má efektivním způsobem definované parametry¹² pro optimalizaci. Při návrhu došlo k inspiraci těmito definicemi a byly navrženy čtyři základní typy parametrů pro `Optimalizační modul`.

- Celočíslný parametr - definuje celé číslo s horní a dolní hranicí prohledávaného prostoru.
- Desetinný parametr - definuje desetinné číslo s omezeným počtem desetinného rozvoje a s horní a dolní hranicí prohledávaného prostoru.
- Reálný parametr - definuje reálné číslo s horní a dolní hranicí a bez omezení přesnosti.
- Kategorický parametr - definuje parametr s předem určenou množinou hodnot, kterých může nabývat.
 - Logický parametr - jedná se o Kategorický parametr s množinou hodnot obsahující pravdu a nepravdu `{True, False}`.

V závislosti na zvolených typech parametrů a jejich hranicích se dá určit velikost prostoru k prohledání. Je vhodné udržovat prostor co nejmenší. Při použití jednoho a více reálných parametrů se stává tento prostor nekonečným, a proto není doporučeno tento typ parametru používat.

Každý parametr v tomto frameworku může zmutovat, nebo být zkřížen s dalším parametrem stejného typu. Mutace znamená, že je vrácena náhodná hodnota z platného rozsahu parametru, která je odlišná od jeho aktuální hodnoty. Zkřížení dvou parametrů znamená, že je vrácen parametr s hodnotou jednoho z poskytnutých parametrů poskytnutého ke křížení vybraného náhodně.

Ve výpise 5.2 je ukázka části definice strategie s parametry (přeložená do českého jazyka.) Tato ukázka slouží pouze k demonstraci zápisu strategie pro `Optimalizační modul` a obsahuje parametry pro optimalizaci. Je odlišná od normálního zápisu strategie tím, že obsahuje funkční prvek `__SETS`, definující seznamy dosaditelné za výběry hodnot pro

¹²<https://www.freqtrade.io/en/stable/hyperopt/>

kategorické parametry. Dále se odlišuje nahrazením některých hodnot funkčními prvky `__PARAM`, definující prostor určený k prohledání Optimalizačním modulem.

```
1 {
2   "id": "Ukazka_definice_strategie_s_parametry", "jmeno": "Ukazkazkova strategie s parametry",
3   "casovy_ramec": { "_PARAM": {
4     "jmeno": "casovy_ramec", "typ": "kategoricky",
5     "vyber": [ "5m", "10m", "15m" ]
6   }},
7   "stop_loss_v_procentech": { "_PARAM": {
8     "jmeno": "stop_loss_pr", "typ": "desetinny",
9     "dolni_hranice": 0.01, "horni_hranice": 0.5, "precision": 3
10  }},
11  "_SETS" : {
12    "_SET_MA_1": [ "ema_5", "ema_10", "ema_12", "ema_24", "ema_26", "ema_50" ],
13    "_SET_MA_2": [ "ema_24", "ema_26", "ema_50" ]
14  },
15  "indikatory": [
16    {
17      "id": "maA_nad_maB_indikator",
18      "typ": "mas_porovnavac",
19      "data": {
20        "ma1": { "_PARAM": {"jmeno": "ma_A", "typ": "kategoricky", "vyber": "_SET_MA_1"}},
21        "ma2": { "_PARAM": {"jmeno": "ma_B", "typ": "kategoricky", "vyber": "_SET_MA_2"}},
22        "operator": "zkrizene_nahoru"
23      }
24    }
25  ],
26  ...
27 }
```

Výpis 5.2: Ukázka části definice strategie s parametry. Jedná se o nefunkční strategii sloužící pouze k demonstraci zápisu strategie pro Optimalizační modul. Ukázka obsahuje funkční prvek `__SETS`, definující seznamy dosaditelné za výběry hodnot pro kategorické parametry. Také obsahuje, namísto některých hodnot, parametry: `__PARAM`, definující prostor určený k prohledání Optimalizačním modulem.

5.7.2 Genetický algoritmus

Během studia evolučních algoritmů se ukázal Genetický algoritmus jako nejvíce vyhovující pro tento optimalizační problém. Samozřejmě by šly aplikovat i další různé evoluční algoritmy, ale pro tuto práci bylo hlavní správně implementovat genetický algoritmus a umožnit rozšíření frameworku o další algoritmy.

V kontextu genetického algoritmu může být každá instance strategie reprezentována jako jedinec s genomem, který kóduje všechny parametry strategie. Tyto geny mohou nabývat hodnot specifikovaných při definování parametrů strategie. Každá kombinace hodnot těchto genů představuje unikátní konfiguraci strategie, kterou lze pomocí Backtestovacího modulu otestovat a vypočítat fitness jedince s touto konfigurací.

Pro konvergenci k optimálnímu řešení je důležité správně nastavit parametry pro genetický algoritmus. V tomto modulu je možné nastavit velikost populace a velikost počáteční populace, počet přeživších populace, počet iterací (generací), pravděpodobnost mutace potomků, fitness metodu a způsob selekce rodičů. Taktéž je možné nastavit, že pro jedince se stejnou konfigurací bude stačit vyhodnocení pomocí backtestingu pouze jednou, aby se předešlo zbytečnému opakování stejných výpočtů.

5.7.3 Fitness metoda

Pro správnost fungování **Optimalizačního** modulu je potřeba mít funkční fitness metodu ohodnocující zdatnost jedinců v populaci (jednotlivých konfigurací strategie). Při návrhu fitness funkce pro optimalizaci obchodních strategií je důležité zvolit metriky, které správně reflektují cíle strategie. Bohužel neexistuje obecná fitness metoda, která by se dala použít pro všechny typy strategií. Jak již bylo zmíněno, také cíle strategie mohou být různé. Například maximalizace zisku při minimalizaci rizika pro stabilní strategii, maximalizace zisku za dobu použití kapitálu pro možnost kombinování více strategií, maximalizace objemu transakcí za účelem zlepšení podmínek (například snížení poplatků na směnárně.)

Při návrhu fitness metody je možné využít metriky, jako například následující základní jednoduše vypočitatelné hodnoty.

- celkový počet pozic
- míra výhernosti pozic - počet ziskových pozic dělený počtem neziskových a ztrátových pozic
- celkový zisk - součet všech zisků a ztrát ze všech pozic
- průměrný zisk z pozice - celkový zisk dělený celkovým počtem pozic
- maximální zisk z pozice - maximální zisk, v případě pouze ztrátových pozic pak minimální ztráta
- minimální zisk z pozice - maximální ztráta, v případě pouze ziskových pozic pak minimální zisk
- doba trvání pozice s (maximálním / minimálním) ziskem
- součet všech (poplatků / Maker poplatků / Taker poplatků)
- (maximální / minimální) doba trvání jedné pozice
- (celková / průměrná) doba trvání pozic
- maximální počet po sobě jdoucích (výherních / ztrátových) pozic seřazených podle doby (otevření / celkového uzavření) pozice
- maximální pokles hodnoty peněženky
- počáteční (celkové / průměrné / maximální / minimální) riziko - riziko se počítá pro pozici, pokud byl stanovený Stop Loss, jako maximální část kapitálu, kterou je možné ztratit (při zanedbání Slippage)
- možný počáteční (celkový / průměrný / maximální / minimální) zisk - možný zisk se počítá pro pozici, pokud byly stanovené hodnoty pro brání profitu, jako maximální zisk v případě vzetí všech stanovených profitů
- (průměrný / maximální / minimální) poměr zisku a rizika

Všechny metriky je možné použít (v případech, kde to je smysluplné) v absolutní hodnotě, nebo v procentech. Také je možné použít metriky se započítanými poplatky stejně jako

metriky zanedbávající poplatky. Pro demonstraci použití fitness metody stačilo implementovat zmíněné metriky, avšak existuje i celá řada komplexních metrik, počítaných pomocí složitých vzorců.

Během návrhu fitness metody mohou nastat různé problémy. Například, pokud uživatel chce mít co nejlepší poměr výdělečných pozic vůči prodělečným, tak se může stát, že optimalizace najde vysoce riskantní konfiguraci strategie, vytvářející malé množství vysoce výdělečných pozic. Pokud je zase například zdatnost určována převážně maximálním poklesem hodnoty peněženky, tak se sice bude hledat stabilní strategie s nízkým rizikem, ale pravděpodobně nebude příliš profitabilní. V závislosti na cíli může být i neuvážení času, kdy je kapitál v pozici a není dostupný pro jiné účely, celkem problematické.

Jeden z prvních návrhů fitness funkce, snažící komplexně ohodnotit strategii a pokrývající profitabilitu, riziko, stabilitu a dobu pozic, má následující vzorec:

$$\begin{aligned} fitness &= (\text{celkový zisk}) * V_1 \\ &+ (\text{celkový počet pozic}) * V_2 \\ &+ (\text{míra výhernosti pozic v procentech}) * V_3 \\ &- (\text{průměrná doba trvání pozic}) * V_4 \\ &- (\text{maximální pokles hodnoty peněženky v procentech}) * V_5 \end{aligned}$$

Kde V_x jsou váhy. Bohužel, při snaze nalézt funkční váhy se ukázalo, že se jedná o nový optimalizační problém. Po snaze nalézt použitelné váhy následovalo nemalé množství dalších různých návrhů prototypů metody používající vždy vybrané metriky a různé způsoby vyvážení.

Časem se osvědčilo používání různých funkcí pro normalizaci hodnot metrik v procentech. Také zvolení jedné hlavní metriky, jejíž hodnota byla následně dělena upravenými hodnotami dalších důležitých metrik, zjednodušilo výpočet a nastavení fitness funkce. Pořád byl ale problém, jak efektivně hodnoty normalizovat. Proběhlo několik pokusů s různými typy funkcí jako jsou trigonometrické, exponenciální a logaritmické.

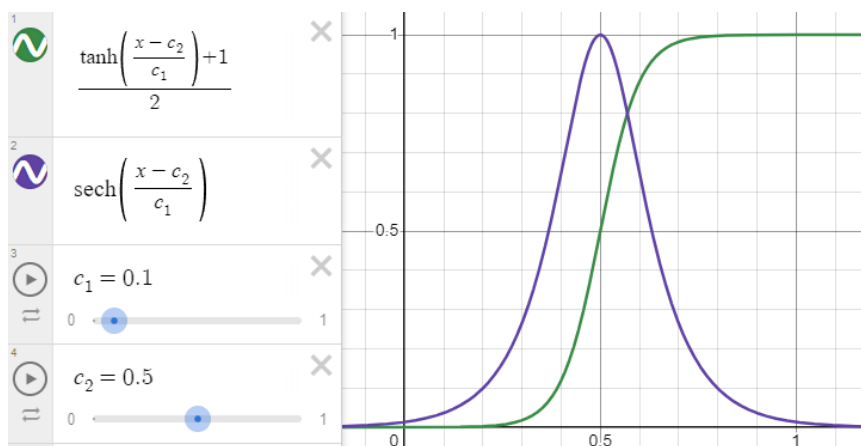
Na konci návrhu byly zmíněné funkce nahrazeny hyperbolickými, které umožňovaly snazší normalizaci a vyhovovaly potřebám pro ohodnocování strategií, přesněji $\tanh()$ (hyperbolický tangens) a $\text{sech}()$ (hyperbolický sekans). Hyperbolický sekans má podobný průběh jako Gaussova funkce, ale Gaussova funkce je strmější, a tak může být její použití v některých případech výhodnější. Pro normalizaci hodnot metrik se osvědčilo používání následujících funkcí:

$$\begin{aligned} f_1 &= \frac{\tanh\left(\frac{x-c_2}{c_1}\right) + 1}{2} \\ f_2 &= \text{sech}\left(\frac{x-c_2}{c_1}\right) \end{aligned}$$

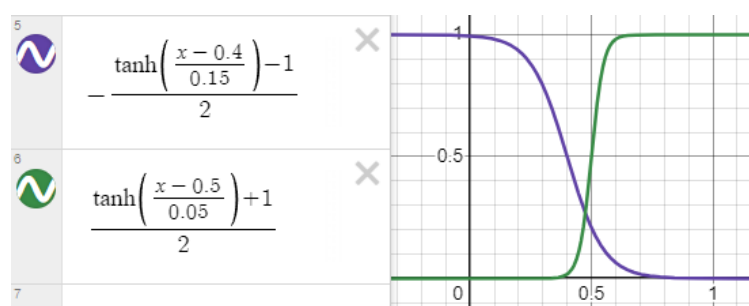
Kde x je hodnota metriky, c_1 je parametr určující sklon funkce a c_2 její posun. Na obrázku 5.11 je zobrazen graf průběhu těchto funkcí (pomocí nástroje [Desmos](https://www.desmos.com/calculator)¹³.)

Pro účely této práce byla navržena následující fitness metoda, kterou je možné použít pro testování a provádění experimentů s optimalizací parametrů. Byly do ní zahrnuty tři metriky, kde celkový zisk v procentech je hlavní metrika a maximální pokles hodnoty peněženky v procentech spolu s mírou výhernosti pozic v procentech jsou vedlejší. Vzorec pro

¹³<https://www.desmos.com/calculator>



Obrázek 5.11: Zobrazení grafu průběhu funkcí vhodných pro aplikaci ve fitness funkci



Obrázek 5.12: Zobrazení grafu průběhu funkcí pro úpravu metrik fitness funkce

výpočet fitness je následující:

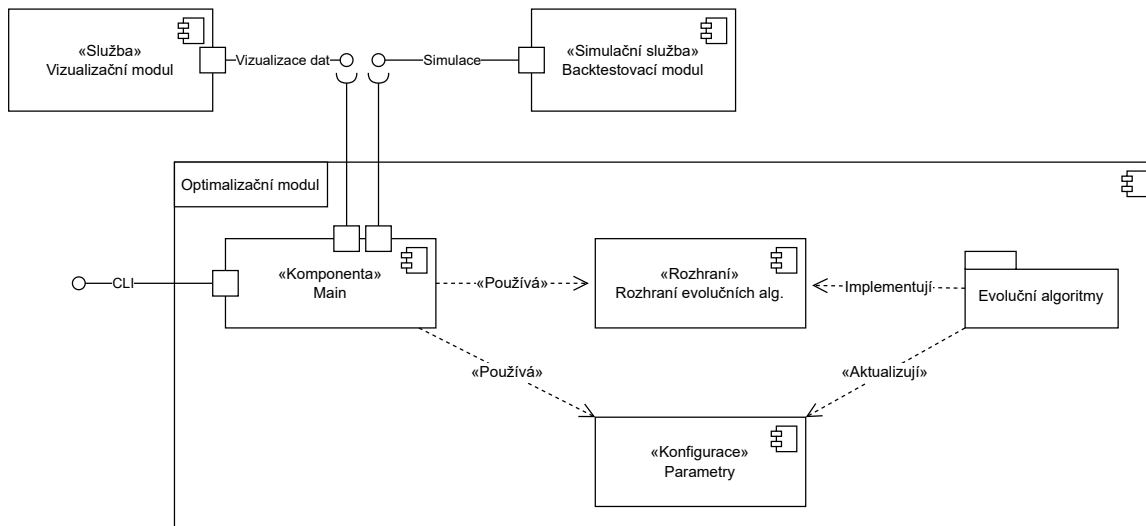
$$m_1 = -\frac{\tanh\left(\frac{mphpp - pmphpp}{0.15}\right) - 1}{2}$$

$$m_2 = \frac{\tanh\left(\frac{mvp - pmvp}{0.05}\right) + 1}{2}$$

$$fitness = czp * ((m_1 * vm_1) + (m_2 * vm_2)) / (vm_1 + vm_2)$$

Kde $mphpp$ je maximální pokles hodnoty peněženky v procentech, $pmphpp$ je povolený maximální pokles hodnoty peněženky v procentech, mvp je míra výhernosti pozic v procentech, $pmvp$ je povolená míra výhernosti pozic v procentech, czp je celkový zisk v procentech, vm_1 a vm_2 jsou váhy pro jednotlivé vedlejší metriky. Výchozí hodnoty jsou $pmphpp = 0.4$, $pmvp = 0.5$, $vm_1 = 1$ a $vm_2 = 1$ a je možné je upravovat dle potřeby. Na obrázku 5.12 je graf zobrazující průběhy funkcí pro úpravu metrik s dosazenými výchozími hodnotami (pomocí nástroje Desmos.)

Optimalizační modul obsahuje CLI pro ovládání uživatelem. Je jej také možné používat i bez CLI přímo z Python kódu pro možnost skriptování úloh. Je schopný načíst jednu strategii, ve které vyhledá parametry a následně pomocí backtestingu testovat konkrétní hodnoty parametrů v zadané strategii. Testování provádí v několika iteracích pomocí genetického algoritmu a na konci optimalizace vybere nejzdatnějšího jedince. Zdatnost se vyhodnocuje na základě poskytnuté fitness funkce. Během analýzy ukládá podstatná data,



Obrázek 5.13: Diagram komponent Optimalizačního modulu

jako je konfigurace všech prošlých strategií, jejich úspěšnost a vytvořené pozice ke každé konfiguraci strategie pro případnou zpětnou analýzu.

Na obrázku 5.13 je zobrazen diagram komponent Optimalizačního modulu. Tento modul obsahuje Main poskytující CLI. K vizualizaci používá Vizualizační modul a k backtestování strategií Backtestovací modul. Main také používá Rozhraní evolučních algoritmů, které implementují jednotlivé Evoluční algoritmy. Modul také obsahuje konfigurace v podobě Parametrů, které se vytváří v Mainu a aktualizují Evolučními algoritmy.

Kapitola 6

Testování, experimenty a validace

Navržené moduly byly implementovány podle návrhu. Implementace tohoto Python frameworku byla provedena v anglickém jazyce. V této kapitole je postupně popsáno testování jednotlivých modulů a validace jejich výstupů. Následně je zde ukázáno možné využití celého frameworku v podobě experimentů.

V této kapitole není popsáno veškeré prováděné testování a experimenty. Cílem této kapitoly je pouze ukázání možností použití frameworku a nastíněný způsob jeho testování a validace.

6.1 Testování stahování dat

První modul, který je při práci s frameworkem nutné použít pro získání dat, je **Stahovací modul**. Pro získání informací o jednotlivých finančních nástrojích dostupných na směnárně, modul obsahuje částečnou integraci API od Bybitu, poskytující přehled o finančních nástrojích, v souboru `src/exchanges/bybit/bybit_api.py`.

Zde je nápověda pro použití CLI k získání dostupných finančních nástrojů na Bybit směnárně:

```
> python src/exchanges/bybit/bybit_api.py -h
usage: bybit_api.py [-h] [-g] [-m MARKET] [-v] [-o OUTPUT_PATH]

options:
  -h, --help            show this help message and exit
  -g, --get             Get available instruments.
  -m MARKET, --market MARKET
                        Market type, one of: [<MARKET_TYPES.LINEAR: 'linear'>,
                        <MARKET_TYPES.INVERSE: 'inverse'>, <MARKET_TYPES.SPOT: 'spot'>,
                        <MARKET_TYPES.OPTION: 'option'>]
  -v, --verbose        Verbose mode.
  -o OUTPUT_PATH, --output_path OUTPUT_PATH
                        Path to output directory; default:
                        C:/_all/Trading/bot/output/instruments_info\2024-04-29_04-07-30
```

CLI nabízí možnost specifikace typu finančních nástrojů pro zobrazení (Bybit nabízí derivátové typy: `linear`, `inverse`, `option`; a typ `spot`.) Získané finanční nástroje přímo odpovídají vstupnímu formátu pro ostatní moduly:

```
> python src/exchanges/bybit/bybit_api.py --get --verbose --market linear
bybit_linear_10000000AIDGUSDT
bybit_linear_10000COQUSDT
bybit_linear_10000LADYSUSDT
bybit_linear_10000NFTUSDT
...
```


Následně je možné použít **Stahovací** modul ke stažení libovolných dostupných transakčních dat. Návod k obsluze CLI je následující:

```
> python src/data_downloading/data_downloading_main.py -h
usage: data_downloading_main.py [-h] -i INSTRUMENTS [-dt DATA_TYPE] [-l] [-d DELAY] [-m]
                               [-th THREADS] [-s START] [-e END]
```

Downloading Module

```
options:
  -h, --help            show this help message and exit
  -i INSTRUMENTS, --instruments INSTRUMENTS
                        Instruments to download, e.g. ["bybit_inverse_BTCUSD',
                        'bybit_linear_BTCUSD']" or bybit_inverse_BTCUSD
  -dt DATA_TYPE, --data_type DATA_TYPE
                        Type of data to download [trades]; default: trades
  -l, --loop            Run in an infinite loop with a delay
  -d DELAY, --delay DELAY
                        Delay between downloads in TimeRange format [3600, 1m, 4h, 12h, 1d,
                        1mo, ...]; default: 4h
  -m, --multithreading Use multithreading
  -th THREADS, --threads THREADS
                        Number of threads to use if multithreading is enabled; default:
                        number of cores - 1
  -s START, --start START
                        Starting time in format: [int, float, YYYY-MM-DD, YYYY-MM-DD HH:MM:SS]
  -e END, --end END    Ending time in format: [int, float, YYYY-MM-DD, YYYY-MM-DD HH:MM:SS]
```

Ke stažení dat několika zajímavých finančních nástrojů byl použit následující příkaz, který pro všechny vypsané finanční nástroje stáhne všechna dostupná transakční data. Stahování dat provádí pro 10 dní (souborů) současně.

```
> python src/data_downloading/data_downloading_main.py --multithreading --threads 10 -i ["\
'bybit_linear_BTCUSD', 'bybit_inverse_BTCUSD', 'bybit_linear_DOGEUSD', 'bybit_linear_DOTUSD', \
'bybit_inverse_DOTUSD', 'bybit_linear_XRPUSD', 'bybit_inverse_XRPUSD', 'bybit_inverse_ETHUSD', \
'bybit_linear_ETHUSD']"
```

CLI Stahovacího modulu také umožňuje specifikaci intervalu, pro který se data mají stáhnout a nebo opakované stahování po specifikované době, jak je vidět v následujících příkazech:

```
> python src/data_downloading/data_downloading_main.py -i "bybit_linear_BTCUSD" \
--start 2024-01-01 --end 2024-03-01
> python src/data_downloading/data_downloading_main.py -i "bybit_linear_BTCUSD" --loop --delay 5h
```

Podařilo se stáhnout nemalé množství jednotlivých transakcí, zobrazené v tabulce 6.1 od počátku historie dat do dubna 2024. Zde je ukázka začátku jednoho souboru s uloženými staženými transakčními daty.

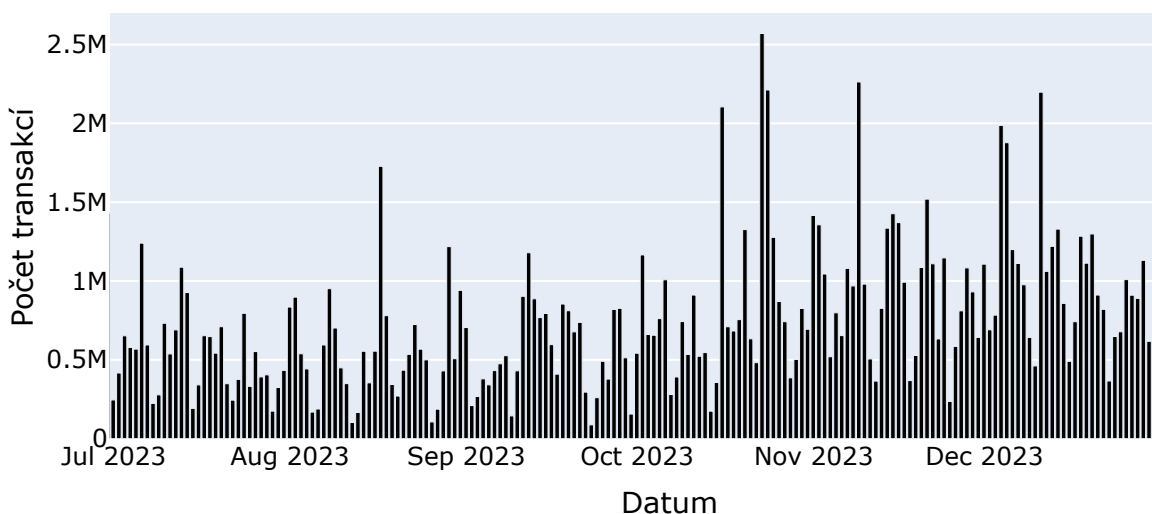
```
{ "id": "0", "ts": 1577836802.438827, "size": 3826, "price": "7168.5" }
{ "id": "1", "ts": 1577836804.846772, "size": 9, "price": "7168.5" }
{ "id": "2", "ts": 1577836805.399127, "size": 1450, "price": "7168.5" }
{ "id": "3", "ts": 1577836805.522619, "size": -500, "price": "7169.0" }
{ "id": "4", "ts": 1577836805.810169, "size": 500, "price": "7168.5" }
...
```

Pro lepší představu rozložení stažených dat v celém období byl napsán skript vizualizující toto rozložení. Na obrázku 6.1 jsou vyobrazeny počty transakčních dat pro každý den. Je možné vidět, že data nejsou rovnoměrně rozložena a dost často se v nich vyskytují nárazově velké výkyvy. Díky této skutečnosti bude doba trvání výpočtu simulace pro každý den odlišná.

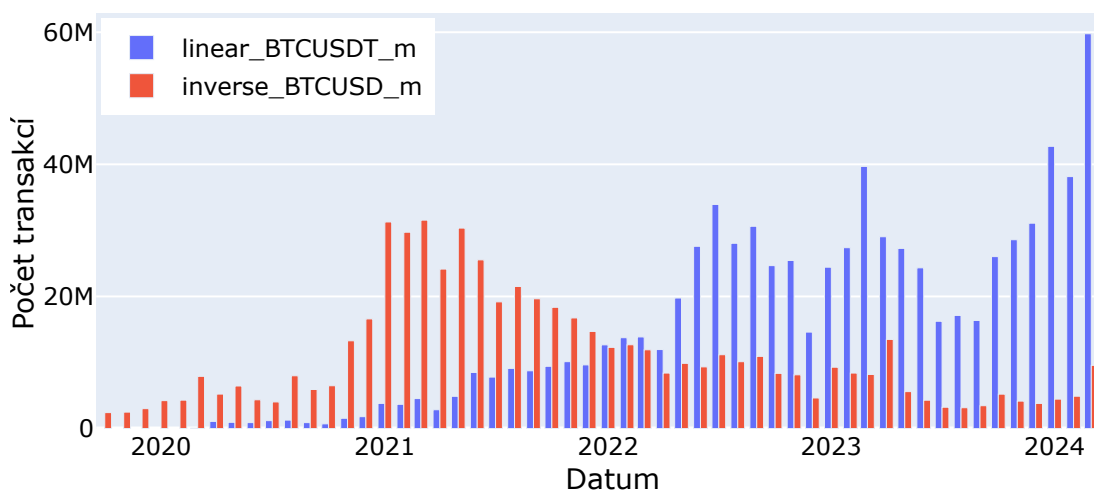
Také pro demonstraci frekvence obchodování různých typů finančních nástrojů vztahujících se k jednomu aktivu byly porovnány `bybit_linear_BTCUSD` s `bybit_inverse_BTCUSD` na obrázku 6.2 a `bybit_linear_ETHUSD` s `bybit_inverse_ETHUSD` na obrázku 6.3. Z tohoto porovnání je jasné, že preference obchodníků se v čase mění.

Finanční nástroj	Počet transakcí
bybit_linear_BTCUSDT	842,573,364
bybit_inverse_BTCUSDT	592,397,546
bybit_linear_ETHUSDT	458,063,513
bybit_inverse_ETHUSDT	249,729,872
bybit_linear_DOGEUSDT	263,658,682
bybit_linear_DOTUSDT	78,786,678
bybit_inverse_DOTUSDT	15,493,558
bybit_inverse_XRPUSD	136,790,225
bybit_linear_XRPUSD	248,999,819

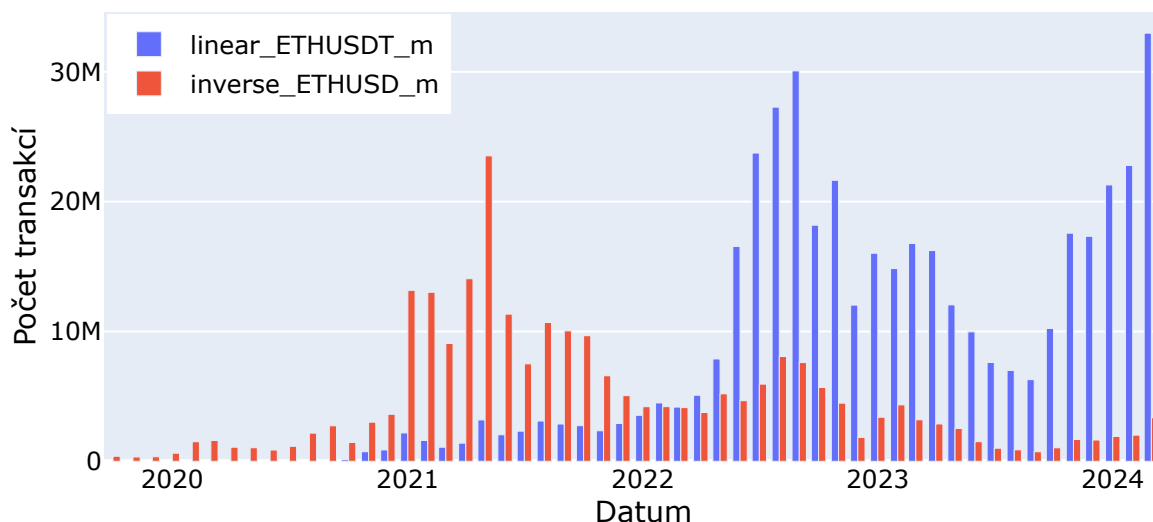
Tabulka 6.1: Počty stažených transakčních dat pro jednotlivé finanční nástroje od počátku historie dat do dubna 2024



Obrázek 6.1: Ukázka počtu stažených transakcí pro jednotlivé dny na finančním nástroji bybit_linear_BTCUSDT ze směnárny Bybit

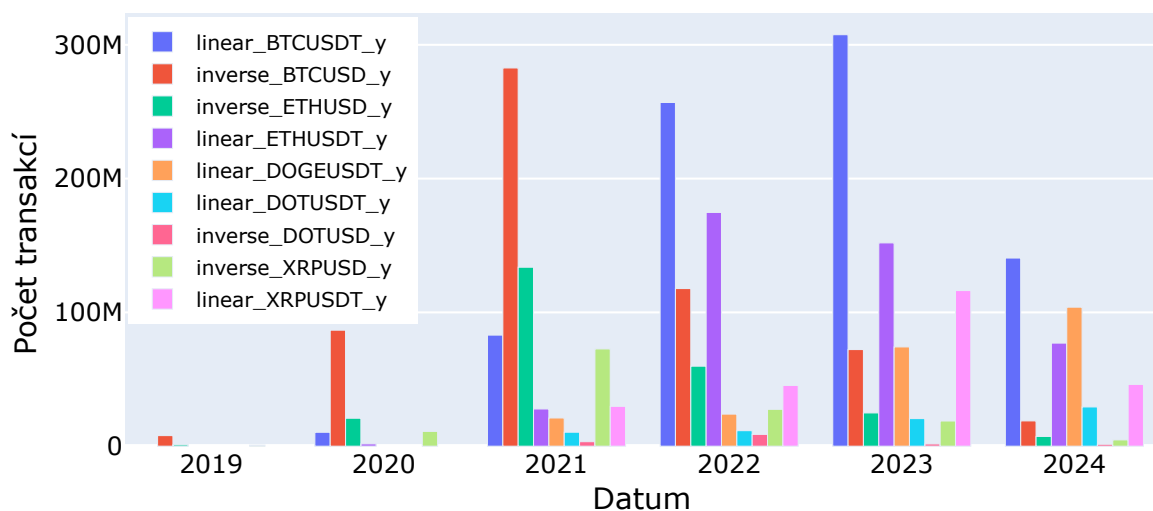


Obrázek 6.2: Porovnání počtu stažených transakcí do dubna 2024 pro jednotlivé měsíce na finančních nástrojích bybit_linear_BTCUSDT a bybit_inverse_BTCUSDT ze směnárny Bybit



Obrázek 6.3: Porovnání počtu stažených transakcí do dubna 2024 pro jednotlivé měsíce na finančních nástrojích bybit_linear_ETHUSD a bybit_inverse_ETHUSD ze směnárny Bybit

Pro komplexnější přehled je na obrázku 6.4 ukázáno pro všechny stažené finanční nástroje, jaký počet transakcí bylo na směnárně Bybit zaznamenáno pro jednotlivé roky.



Obrázek 6.4: Porovnání počtu stažených transakcí do dubna 2024 pro jednotlivé roky na různých finančních nástrojích ze směnárny Bybit

6.2 Testování, zpracování a ukládání dat

Po stažení potřebných dat je možné otestovat jejich zpracování a správné ukládání. CLI Zpracovávacího modulu nabízí následující možnosti:

```
> python src/data_processing/downloaded_data_processing_main.py -h
```

```
usage: downloaded_data_processing_main.py [-h] -i INSTRUMENTS -d DATA [-m] [-t THREADS]
                                           [-s START] [-e END]
```

Processing Module

options:

```
-h, --help            show this help message and exit
-i INSTRUMENTS, --instruments INSTRUMENTS
                        Instruments to process, e.g. ["bybit_inverse_BTCUSD",
                        'bybit_linear_BTCUSDT']" or bybit_inverse_BTCUSD
-d DATA, --data DATA Data to process, e.g. '{"OHLC': ['1h', '4h'], 'CVD': ['1h'],
                        'VOLUME': ['4h']}'" or path to json file.
-m, --multithreading Use multithreading
-t THREADS, --threads THREADS
                        Number of threads to use if multithreading is enabled; default:
                        number of cores - 1
-s START, --start START
                        Starting time in format: [int, float, YYYY-MM-DD, YYYY-MM-DD HH:MM:SS]
-e END, --end END      Ending time in format: [int, float, YYYY-MM-DD, YYYY-MM-DD HH:MM:SS]
```

Pomocí Zpracovávacího modulu bylo provedeno zpracování všech stažených dat pro `bybit_linear_BTCUSDT` a `bybit_inverse_BTCUSD`. Zpracovávaly se základní datové typy na významných časových rámcích pomocí příkazu:

```
> python src/data_processing/downloaded_data_processing_main.py --multithreading --threads 10 \
  -i ["bybit_inverse_BTCUSD", 'bybit_linear_BTCUSDT'] --data data_file.json
```

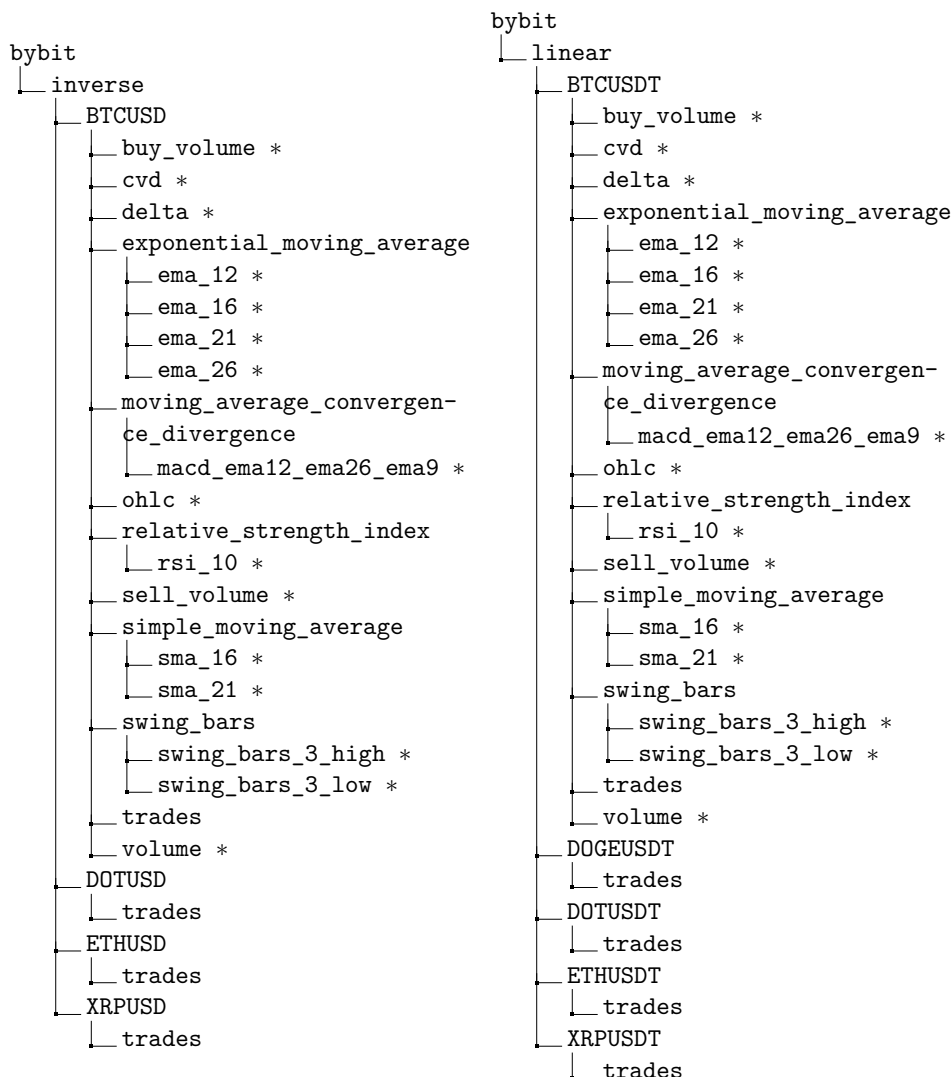
Soubor `data_file.json` obsahuje následující data, reprezentující jednotlivé datové typy a časové rámce určené ke zpracování:

```
{
  "OHLC":           ["m", "5m", "h", "d"],
  "CVD":            ["m", "5m", "h", "d"],
  "delta":         ["m", "5m", "h", "d"],
  "buy_volume":    ["m", "5m", "h", "d"],
  "sell_volume":   ["m", "5m", "h", "d"],
  "volume":        ["m", "5m", "h", "d"],
  "rsi_10":        ["m", "5m", "h", "d"],
  "swing_bars_3_low": ["m", "5m", "h", "d"],
  "swing_bars_3_high": ["m", "5m", "h", "d"],
  "macd_ema12_ema26_ema9": ["m", "5m", "h", "d"],
  "sma_16":         ["m", "5m", "h", "d"],
  "sma_21":         ["m", "5m", "h", "d"],
  "ema_16":         ["m", "5m", "h", "d"],
  "ema_21":         ["m", "5m", "h", "d"]
}
```

Další možný příklady volání CLI Zpracovávacího modulu může být následující příkaz pro zpracování pouze OHLC dat na hodinovém a denním časovém rámci pro konkrétní časový úsek.

```
> python src/data_processing/downloaded_data_processing_main.py -i "bybit_inverse_BTCUSD" \
  --data '{"OHLC': ['1h', '1d']}' --start 2024-01-01 --end 2024-03-01
```

Byl zvolen formát `parquet` pro efektivní ukládání a načítání dat. Zpracování dat trvalo 65 minut pro `bybit_inverse_BTCUSD` (přibližně 600M transakcí) a 105 minut pro `bybit_linear_BTCUSDT` (přibližně 850M transakcí) při zpracovávání v deseti vláknech současně. V lokálním úložišti přibylo několik nových adresářů se zpracovanými daty, jak je možné vidět na obrázku 6.5. Je možné si všimnout, že byla zpracována data pro datové typy EMA(12) a EMA(26), které nebyly určeny pro zpracování. Je tomu z důvodu, že se tyto datové typy zpracovávaly pro výpočet MACD, a tak je modul také uložil.



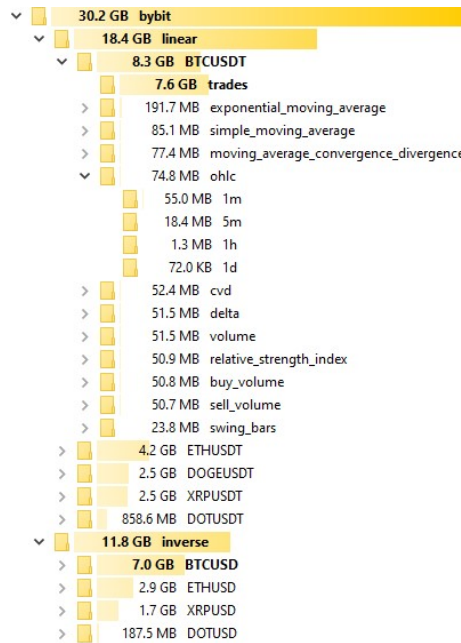
Obrázek 6.5: Ukázka adresářů lokálního úložiště po zpracování některých dat. Označení * znamená, že adresář obsahuje další adresáře pro data rozdělená podle jednotlivých časových rámců. V tomto případě konkrétně adresáře: 1d, 1h, 5m a 1m.

Nejvíce místa zabírají v lokálním úložišti transakční data. Celkem stažená a zpracovaná data zabírají okolo 30 GB paměti. Na obrázku 6.6 je vidět stav paměti lokálního úložiště s testovanými daty. (Jedná se o snímek obrazovky z nástroje [TreeSize](https://www.jam-software.com/treesize)¹.)

Zde je ukázka uložených, zpracovaných dat pro datový typ OHLC s nastaveným hodinovým časovým rámcem. Jedná se o ukázkou začátku souboru.

```
{ "ts": "1577836800", "o": 7169, "h": 7169, "l": 7154.5, "c": 7154.5 }
{ "ts": "1577840400", "o": 7154.5, "h": 7207, "l": 7154.5, "c": 7193 }
{ "ts": "1577844000", "o": 7193, "h": 7222, "l": 7193, "c": 7221.5 }
{ "ts": "1577847600", "o": 7221.5, "h": 7222, "l": 7198, "c": 7201 }
{ "ts": "1577851200", "o": 7201, "h": 7201.5, "l": 7193.5, "c": 7195.5 }
...
```

¹<https://www.jam-software.com/treesize>



Obrázek 6.6: Stav paměti lokálního úložiště s testovanými daty

6.3 Validace dat pomocí vizualizace

K validaci dat je možné použít Vizualizační modul. Pro validaci každého datového typu lze provést několik náhodných výběrů zpracovaných dat, vizualizovat je a ověřit manuálními výpočty. Také je možné u některých typů dat provést porovnání s daty z nějaké platformy.

Nápověda CLI Vizualizačního modulu je následující:

```
> python src/visualizer/data_previewer.py -h
usage: data_previewer.py [-h] -i INSTRUMENT [-pr] [-sh] [-sv] [-f FILE] [-pa PATH]
                        [-t TITLE] [-d DATA] [-s START] [-e END] [-tr TRADES] [-l]

Visualization Module

options:
  -h, --help                show this help message and exit
  -i INSTRUMENT, --instrument INSTRUMENT
                            Instrument to visualize, e.g bybit_inverse_BTCUSD
  -pr, --process            Process data if true, otherwise only load pre-processed data.
  -sh, --show              Show data after processing.
  -sv, --save              Save data after processing.
  -f FILE, --file FILE    File name to save data; default: export_2024-04-29_04-27-17.html
  -pa PATH, --path PATH   Path to save data; default: C:/_all/Trading/bot/output
  -t TITLE, --title TITLE
                            Title for the graph.
  -d DATA, --data DATA  Data to load, e.g. '{"OHLC': ['1h', '4h'], 'CVD': ['1h'], 'VOLUME':
                            ['4h']}' or path to json file.
  -s START, --start START
                            Starting time in format: [int, float, YYYY-MM-DD, YYYY-MM-DD
                            HH:MM:SS]
  -e END, --end END       Ending time in format: [int, float, YYYY-MM-DD, YYYY-MM-DD
                            HH:MM:SS]
  -tr TRADES, --trades TRADES
                            Trades intervals in list: "[['value','value'],['value','value']]"
                            where value in format: [int, float, YYYY-MM-DD, YYYY-MM-DD
                            HH:MM:SS]
  -l, --light              Light mode for the graph.
```



Obrázek 6.7: Porovnání OHLC dat s denním časovým rámcem pro `bybit_inverse_BTCUSD` z frameworku (světlejší svíčkový graf v popředí) a pro `BYBIT BTCUSD.P` z platformy TradingView (tmavší svíčkový graf v pozadí)

Příkladem validace dat může být náhodné vybrání a zobrazení dat datového typu jako třeba OHLC s denním časovým rámcem pro `bybit_inverse_BTCUSD` ve frameworku a na platformě TradingView². Oba výstupy je možné ověřit kontrolou libovolných hodnot. Na obrázku 6.7 jsou hodnoty porovnány pro 4. 12. 2021. Nebo ověřit vizuálně, jak je ukázáno na obrázku 6.7 překrytím svíčkových grafů. Těmito způsoby, spolu s manuálními výpočty hodnot, byly úspěšně porovnány a validovány náhodné výstupy Zpracovávacího modulu.

Následující příkaz byl použit pro vizualizaci svíčkového grafu pro obrázek 6.7 Vizualizačním modulem, který načtl data z paměti a zobrazil je na světlém pozadí pro zvolený čas.

```
> python src/visualizer/data_previewer.py -i bybit_inverse_BTCUSD --light --show \
  --start "2021-10-01 00:00:00" --end "2022-01-01 0:00:00" --data "{OHLC:['1d']}"
```

Následuje příkaz, který vykreslí a uloží grafy na obrázku 6.8 s tmavým pozadím. V příkazu je specifikováno, že se mají všechna data dopočítat z transakčních dat pro stanovený časový interval. Je možné si všimnout, jak framework rozděluje jednotlivé grafové stopy do příslušných grafů na obrázku 6.8 na základě definic v jednotlivých Generátorech stop.

```
> python src/visualizer/data_previewer.py -i bybit_inverse_BTCUSD --process \
  --title "Ukazka vizualizace" --show --save --file "ukazka_vizualizace.html" \
  --data "{OHLC:['1m','5m','15m'],'CVD':['5m'],'delta':['5m'],'sma_16':['5m'],'ema_21':['5m'], \
    'swing_bars_3_high':['15m'],'swing_bars_3_low':['15m'],'buy_volume':['15m'], \
    'sell_volume':['15m'],'rsi_10':['15m'],'macd_ema12_ema26_ema9':['15m']}" \
  --start "2021-01-01 00:00:00" --end "2021-01-01 12:00:00"
```

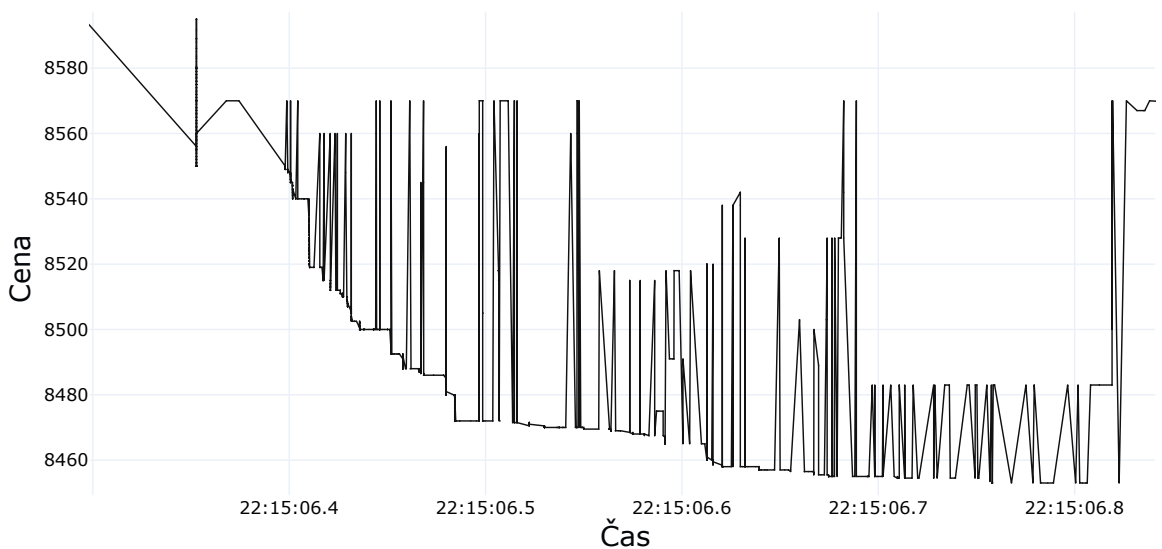
Během zkoumání dat se podařilo nalézt místa se zvýšenou likviditou. Příklad Supportu může být vidět na obrázku 6.9, na kterém je zobrazeno zhruba půl vteřiny transakčních dat. Je hezky viditelné, jak se cena finančního nástroje `bybit_inverse_BTCUSD` snaží prorazit Support. Pro demonstraci, jak vykreslit tento graf pomoci Vizualizačního modulu, je zde příkaz pro vizualizaci jednotlivých transakčních dat pro obrázek 6.9:

```
> python src/visualizer/data_previewer.py -i bybit_inverse_BTCUSD --light --show \
  --trades "[['2020-01-19 22:15:06','2020-01-19 22:15:07']]"
```

²<https://www.tradingview.com/>



Obrázek 6.8: Ukázka výstupu Vizualizačního modulu s rozsáhlou vizualizací dat. Je možné si na pravé straně v legendě všimnout dvou prvních, méně výrazných popisků pro OHLC. Tyto dvě stopy jsou na grafu skryté. (Je možné zobrazit, nebo skrýt jakoukoli stopu kliknutím na text v legendě.) Grafy postupně od shora obsahují v 1. grafu: OHLC, Swingové body, EMA, SMA; 2.: objem nákupu a prodeje; 3.: CVD; 4.: Deltu; 5.: MACD; 6.: RSI.



Obrázek 6.9: Ukázka půl vteřiny transakčních dat, kde je snaha o proražení Supportu u finančního nástroje bybit_inverse_BTCUSD

Samozřejmě je možné kombinovat vizualizaci transakčních dat spolu s dalšími datovými typy. V kapitole 2 byl obrázek 2.2 získán následujícím příkazem:

```
> python src/visualizer/data_previewer.py -i bybit_inverse_BTCUSD --light --show \  
--process --data '{"OHLC':['5m'],'volume':['5m']}' \  
--start "2024-01-01 00:00:00" --end "2024-01-01 01:00:00" \  
--trades "[['2024-01-01 00:00:00', '2024-01-01 01:00:00']]"
```

6.4 Testování funkcionalit Backtestovacího modulu

Uživatel si může napsat vlastní strategie, které chce otestovat. Pro otestování frameworku byly dotázány dostupné generativní modely s žádostí o vytvoření funkční strategie pro algoritmické obchodování na burze. Odpovědi jednotlivých modelů jsou přiložené v příloze C. Všechny strategie z odpovědí byly složeny z indikátorů používající následující datové typy: EMA, SMA, MACD, RSI. Pro účely testování funkcionality modulu byla vybrána odpověď od modelu GPT-4 C.4. Poskytnutý algoritmus 4 (v příloze C), obsahující pouze datové typy SMA(20) a EMA(50), byl přepsán na funkční konfiguraci strategie viditelnou ve výpise 6.1.

Možnosti použití CLI Backtestovacího modulu:

```
> python src/backtesting/backtesting_main.py -h  
usage: backtesting_main.py [-h] -i INSTRUMENT [-sp STRATEGIES_PATH] [-o OUTPUT_PATH]  
[-s START] [-e END] [-sh]
```

Backtesting Module

options:

```
-h, --help          show this help message and exit  
-i INSTRUMENT, --instrument INSTRUMENT  
                    Instrument to download, e.g. bybit_inverse_BTCUSD  
-sp STRATEGIES_PATH, --strategies_path STRATEGIES_PATH  
                    Path to strategies directory; default:  
                    C:/_all/Trading/bot/strategies  
-o OUTPUT_PATH, --output_path OUTPUT_PATH  
                    Path to output directory; default:  
                    C:/_all/Trading/bot/output/backtesting\2024-04-29_04-29-01  
-s START, --start START  
                    Starting time in format: [int, float, YYYY-MM-DD, YYYY-MM-DD  
                    HH:MM:SS]  
-e END, --end END  Ending time in format: [int, float, YYYY-MM-DD, YYYY-MM-DD  
                    HH:MM:SS]  
-sh, --show        Show graphs after processing
```

Na obrázku 6.10 je možné vidět, jak bylo manipulováno s jednotlivými pozicemi v čase. Kvůli jednoduchosti strategie jsou zde pouze na sebe navazující pozice s pouze jedním kompletním uzavřením každé pozice. V případě více nastavených Take Profitů a manipulací se strategií, by byl graf více větvený.

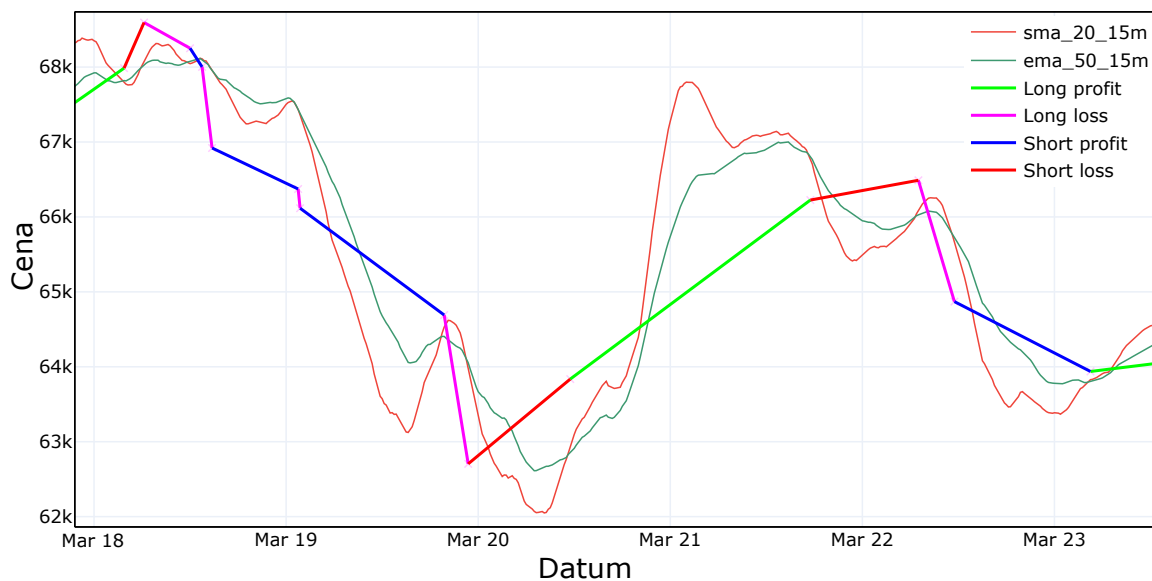
Strategie byla backtestována na historických datech finančního nástroje `bybit_inverse_BTCUSD` a na obrázku 6.11 je možné vidět vývoj stavu peněženky během simulace. Ve statistikách backtestingu (za 6 testovaných měsíců) je, že bylo vytvořeno 393 pozic, kde 29.5 % pozic bylo výdělečných, a celková ztráta je 3 %.

```

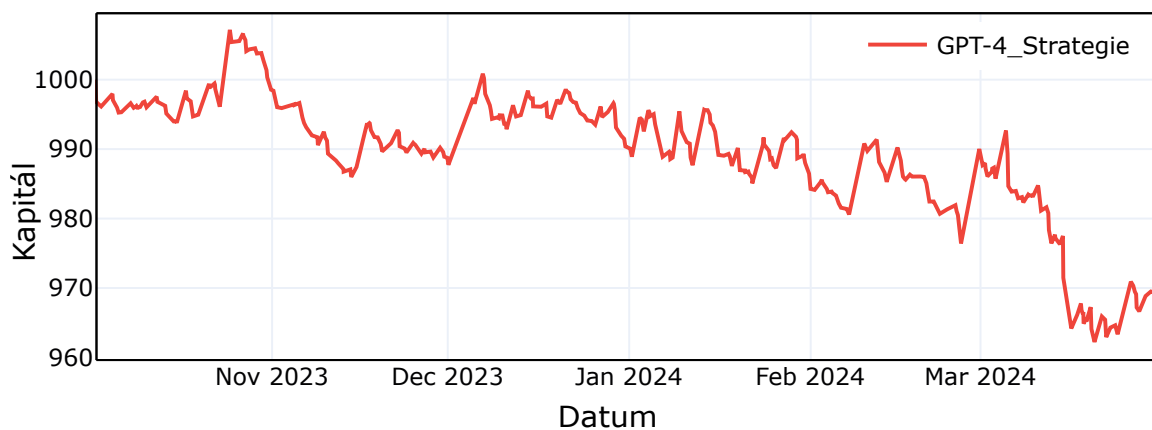
1 {
2   "id": "GPT-4_Strategie",
3   "name": "GPT-4 Strategie",
4   "instruments": [
5     "bybit_inverse_BTCUSD"
6   ],
7   "exchange_fee_plan": "VIP3",
8   "time_range": "15m",
9   "stop_loss_percent": 0.2,
10  "indicators": [
11    {
12      "id": "short_sma_crosses_above_long_ema_indicator",
13      "type": "mas_comparison",
14      "data": {
15        "ma1": "sma_20",
16        "ma2": "ema_50",
17        "operator": "crossed_above"
18      }
19    },
20    {
21      "id": "short_sma_crosses_below_long_ema_indicator",
22      "type": "mas_comparison",
23      "data": {
24        "ma1": "sma_20",
25        "ma2": "ema_50",
26        "operator": "crossed_below"
27      }
28    }
29  ],
30  "take_conditions": {
31    "long": [
32      [
33        { "type": "indicator", "indicator": "short_sma_crosses_above_long_ema_indicator", "
34        value": "valid" },
35        { "type": "field", "field": "active_trades_cnt", "operator": "equal", "value": 0 },
36        { "type": "field", "field": "start_of_time_range" }
37      ],
38      "short": [
39        [
40          { "type": "indicator", "indicator": "short_sma_crosses_below_long_ema_indicator", "
41          value": "valid" },
42          { "type": "field", "field": "active_trades_cnt", "operator": "equal", "value": 0 },
43          { "type": "field", "field": "start_of_time_range" }
44        ]
45      ],
46      "position_manipulation_strategies_conditions": {
47        "cancel_position_strategy": [
48          [
49            { "type": "indicator", "indicator": "short_sma_crosses_above_long_ema_indicator", "
50            value": "valid" },
51            { "type": "field", "field": "start_of_time_range" }
52          ],
53          [
54            { "type": "indicator", "indicator": "short_sma_crosses_below_long_ema_indicator", "
55            value": "valid" },
56            { "type": "field", "field": "start_of_time_range" }
57          ]
58        ]
59      ]
60    }
61  }

```

Výpis 6.1: Konfigurace strategie vytvořená na základě odpovědi od modelu GPT-4. Strategie používá indikátory detekující křížení klouzavých průměrů SMA(20) a EMA(50).



Obrázek 6.10: Ukázka manipulací s pozicemi pro GPT-4 Strategii. Strategie reaguje na překřížení SMA a EMA. Na obrázku jsou čtyřmi různými barvami odlišeny typy pozic (Long, Short) a jejich profitabilita (zisková/profit, ztrátová/loss). Body čar, reprezentující pozice, jsou na úrovni ceny, která byla v době manipulace s pozicí.



Obrázek 6.11: Vývoj kapitálu v USD virtuální peněženky v čase pro GPT-4 Strategii



Obrázek 6.12: Ukázka DCA strategie na bybit_inverse_BTCUSD datech. Každý den bylo investováno 100 USD. V prvním grafu je zobrazena výše investice a hodnota vlastněného aktiva / portfolia. V druhém grafu pak celkový zisk ke každému dni.

6.5 Testování vlastního backtestingu

K otestování složitosti používání vlastního backtestingu byla naprogramovaná DCA (dollar-cost averaging) strategie. Jedná se o jednoduchý algoritmus, který každý den investuje stejné množství prostředků do vybraného aktiva. Na obrázku 6.12 je výstup této strategie pro kompletní dostupná historická data finančního nástroje bybit_inverse_BTCUSD. Celkový zisk je 228 %.

Ve zdrojovém kódu stačilo importovat knihovny, definovat konstanty a vytvořit instance Modulu pro správu úložiště a Vizualizačního modulu. Získat denní OHLC data a výši poplatků z Bybit směnárně. Provést cyklus počítání kapitálu a poplatků pro každý den, specifický pro DCA strategii. Získaná data šlo poté jednoduše zobrazit pomocí Vizualizačního modulu.

Pro porovnání výsledků byla tato DCA strategie otestována na internetové stránce dcaBTC³, která zanedbává poplatky směnárně. Její výsledek je možné vidět na obrázku 6.13. Výsledky jsou téměř shodné, drobné odchylky v hodnotách jsou způsobené uvažováním poplatků ve frameworku.

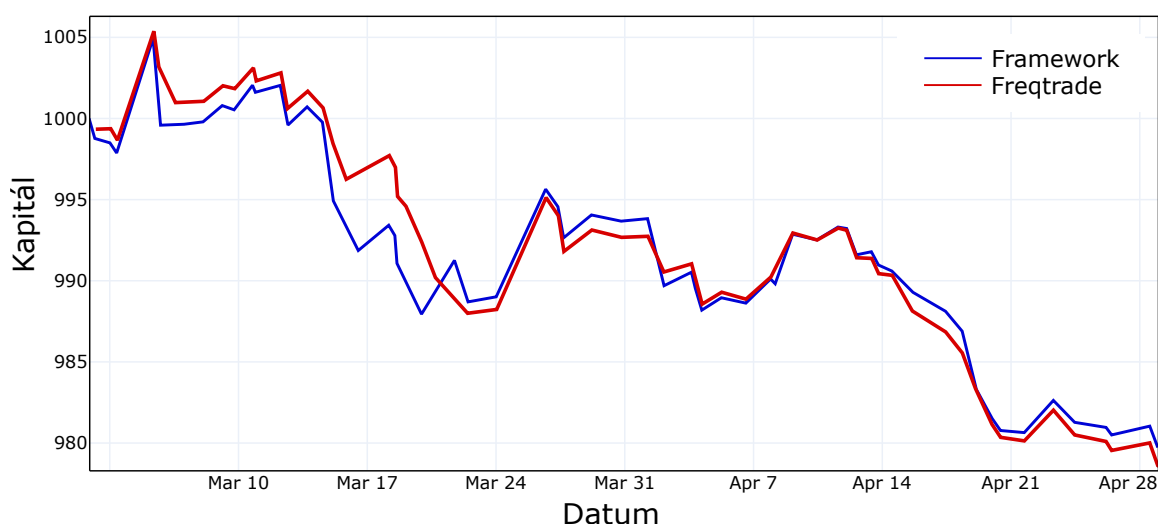
6.6 Porovnání výsledku s Freqtrade frameworkem

Pro ověření správnosti výsledků byl proveden backtesting pomocí Freqtrade frameworku za účelem porovnání výsledků. Data pro tento backtesting byly z Bybit směnárně pro BTC/USDT, co odpovídá bybit_linear_BTCUSDT. Přesnější popis strategie napsané pro Freqtrade framework, její definice a výstup backtestingu je v příloze D. Také byl napsán

³https://dcabtc.com?sd=2019-10-01&sda=custom&f=daily&d=5_years&ac=10000&c=false



Obrázek 6.13: Vývoj hodnoty portfolia pro DCA strategie z internetové stránky dcaBTC. Denní vklad byl nastaven na 100 USD. Na ose x je kapitál v USD a na ose y čas.



Obrázek 6.14: Ukázka stavu peněženek v USD pro backtesting jednoduché strategie v toto frameworku a ve Freqtrade frameworku

ekvivalent strategie pro tento framework. Vzhledem k absenci vizualizace stavu peněženky ve Freqtrade frameworku (nebo jejímu nenalezení), bylo doprogramováno vykreslování stavu peněženky a výsledky obou backtestingů je možné vidět na obrázku 6.14.

Strategie měly stejné parametry, ale kvůli odlišnostem jako je například počítání poplatků, vyhodnocování indikátorů a kvůli odlišnostem ve vstupních datech, je možné vidět drobné odchylky.

6.7 Experimenty s optimalizací parametrů strategií

Pro experimenty s optimalizací parametrů strategií, byla zvolena GPT-4 Strategie z výpisu 6.1, u které byl nahrazen časový rámec spolu se všemi klouzavými průměry za kategorie parametry a Stop Loss hodnota za desetinný parametr. Úprava konfigurace strategie je ve výpisu 6.2.

```

1 {
2   "time_range": { "_PARAM": {"name": "time_range", "type": "categorical", "choices": [ "5m", "10m",
3     "15m" ]}},
4   "stop_loss_percent": { "_PARAM": {"name": "stop_loss", "type": "decimal", "lower_bound": 0.01, "
5     upper_bound": 0.5, "precision": 3}},
6   "_SETS" : {
7     "_SET_MA_1": [
8       "ema_2", "ema_3", "ema_4", "ema_5", "ema_6", "ema_7", "ema_8", "ema_9", "ema_10", "ema_11",
9       "ema_12", "ema_13", "ema_14", "ema_15", "ema_16", "ema_17", "ema_18", "ema_19", "ema_20",
10      "ema_21", "ema_22", "ema_23", "ema_24", "ema_25", "ema_26", "ema_27", "ema_28", "ema_29",
11      "ema_30", "ema_35", "ema_40", "ema_45", "ema_50",
12      "sma_2", "sma_3", "sma_4", "sma_5", "sma_6", "sma_7", "sma_8", "sma_9", "sma_10", "sma_11",
13      "sma_12", "sma_13", "sma_14", "sma_15", "sma_16", "sma_17", "sma_18", "sma_19", "sma_20",
14      "sma_21", "sma_22", "sma_23", "sma_24", "sma_25", "sma_26", "sma_27", "sma_28", "sma_29",
15      "sma_30", "sma_35", "sma_40", "sma_45", "sma_50"
16    ]
17  },
18  "indicators": [
19    {
20      "id": "short_sma_crosses_above_long_ema_indicator",
21      "type": "mas_comparison",
22      "data": {
23        "ma1": { "_PARAM": {"name": "ema_A", "type": "categorical", "choices": "_SET_MA_1"}},
24        "ma2": { "_PARAM": {"name": "ema_A", "type": "categorical", "choices": "_SET_MA_1"}},
25        "operator": "crossed_above"
26      }
27    }
28  ]
29 }

```

Výpis 6.2: Úpravy konfigurace GPT-4 strategie pro Optimalizační modul. Nahrazení hodnot za parametry.

Následuje ukázka CLI Optimalizačního modulu. Další parametry pro evoluční algoritmus se nastavují přímo v Python kódu.

```

> python src/evo/evo_main.py -h
usage: evo_main.py [-h] -i INSTRUMENT -sp STRATEGY_PATH [-o OUTPUT_PATH] [-s START]
                  [-e END] [-sh] [-ic ITERATION_COUNT] [-pm POPULATION_MULTIPLIER]

```

Optimization Module

options:

```

-h, --help          show this help message and exit
-i INSTRUMENT, --instrument INSTRUMENT
                    Instrument to download, e.g. bybit_inverse_BTCUSD
-sp STRATEGY_PATH, --strategy_path STRATEGY_PATH
-o OUTPUT_PATH, --output_path OUTPUT_PATH
                    Path to output directory; default:
                    C:/_all/Trading/bot/output\evo\2024-04-29_04-30-05
-s START, --start START
                    Starting time in format: [int, float, YYYY-MM-DD, YYYY-MM-DD
                    HH:MM:SS]
-e END, --end END   Ending time in format: [int, float, YYYY-MM-DD, YYYY-MM-DD
                    HH:MM:SS]
-sh, --show         Show graphs after processing
-ic ITERATION_COUNT, --iteration_count ITERATION_COUNT
                    Number of iterations
-pm POPULATION_MULTIPLIER, --population_multiplier POPULATION_MULTIPLIER
                    Population multiplier

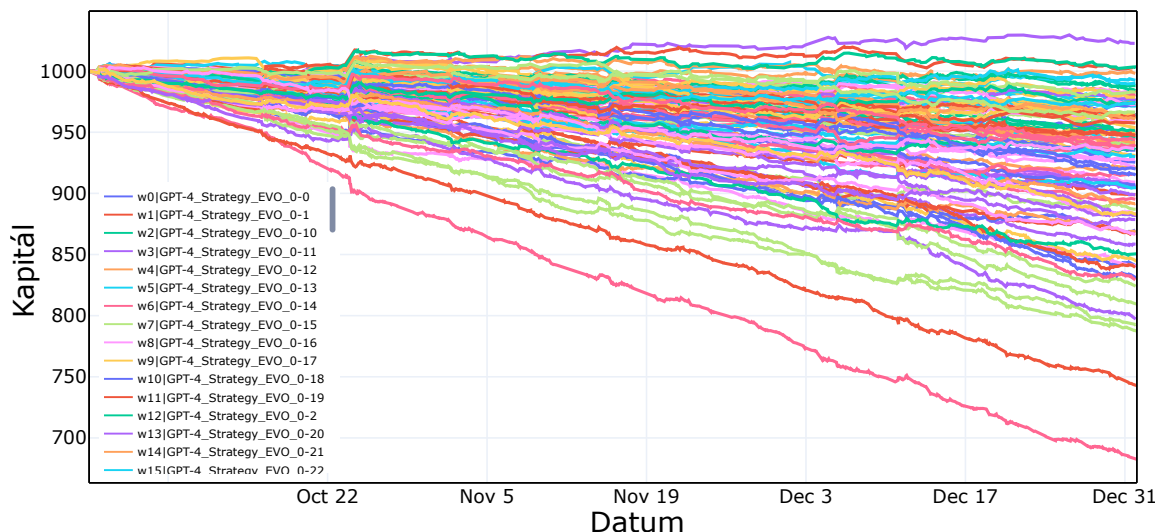
```

První optimalizace byla spuštěna následujícím příkazem pro otestování správnosti generování konfigurací strategií a validaci výstupů z Optimalizačního modulu.

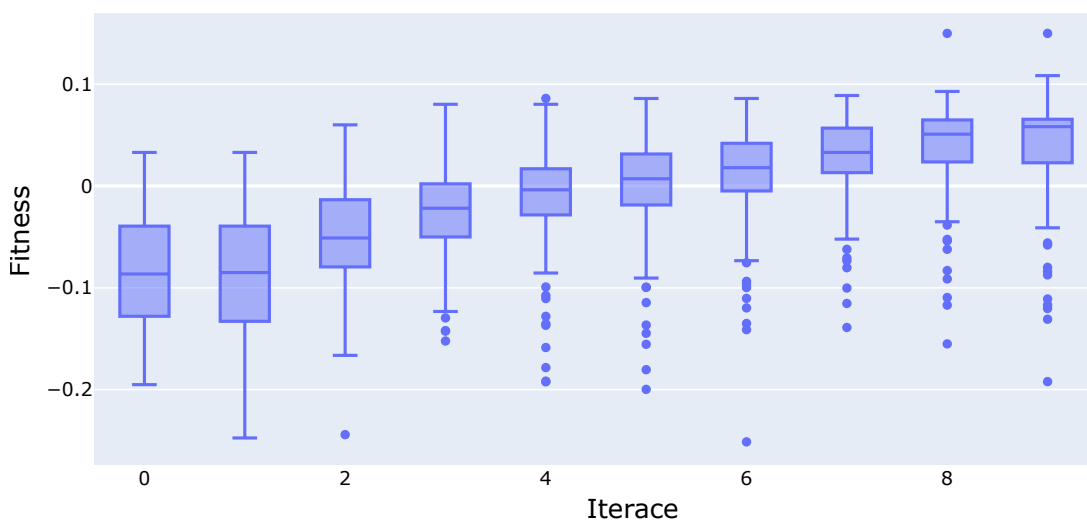
```

> python src/evo/evo_main.py -i bybit_inverse_BTCUSD --iteration_count 3 --population_multiplier 5 \
  --strategy_path "path/evo_strategies/GPT4_Strategy_EVO.json" \
  --start "2023-10-01" --end "2024-01-01"

```



Obrázek 6.15: Ukázka stavu peněženek během jedné iterace optimalizace



Obrázek 6.16: Krabicový graf reprezentující zdatnost populace v každé iteraci

Velikost populace byla nastavena na $5 * 10$ jedinců. Backtesting trval 2.3 hodiny. Na obrázku 6.15 je ukázka stavu peněženek během první iterace optimalizace.

Po otestování funkcionalit backtestingu proběhlo otestování schopností **Optimalizačního** modulu vylepšit parametry. Následujícím příkazem bylo provedeno 10 iterací genetickým algoritmem s populací o velikosti 100 jedinců. Backtesting trval 11.5 hodiny. Výsledkem bylo velké množství dat. Pro jednoduchou vizualizaci je na obrázku 6.16 krabicový graf reprezentující zdatnost populace v každé iteraci.

```
> python src/evo/evo_main.py -i bybit_inverse_BTCUSD \
  --iteration_count 10 --population_multiplier 10 \
  --strategy_path "path/evo_strategies/GPT4_Strategy_EVO.json" \
  --start "2023-10-01" --end "2024-01-01" > evo_out_1.tmp
```



Obrázek 6.17: Vývoj kapitálu virtuálních peněženek v USD v čase pro GPT-4 Strategie i a její optimalizovanou verzi

Získané data ukázaly konfigurace strategií, které byly výdělečné na datech, na kterých se učili. Předposlední experiment, uvedený v této práci, spočíval v dopředném ověření získaných výdělečných strategií na datech, které Optimalizační modul neměl k dispozici. Výsledky byly překvapující, protože se podařilo optimalizovat triviální strategii vygenerovanou GPT modelem na opravdu výdělečnou strategii. Výsledky jedné z optimalizovaných strategií GPT-4_Strategie_EVO je, s porovnáním vůči původní vygenerované verzi, vidět na obrázku 6.17. Optimalizace byla provedena na třech měsících dat od 1. 10. 2023 do 1. 1. 2024 a dopředném ověření na sedmi měsících dat od 1. 10. 2023 do 1. 5. 2024. Parametry optimalizované strategie s původní je možné vidět v tabulce 6.2.

Metrika	GPT-4_Strategie	GPT-4_Strategie_EVO
Celkový zisk [%]	-3.097	9.933
Průměrný zisk na pozici [%]	-0.008	0.017
Maximální pokles hodnoty peněženky [%]	-4.46	-1.17
Počet pozic	393	579
Výhernost pozic [%]	29.517	56.304
Průměrná doba trvání pozice [ms]	39879.39	27181.87

Tabulka 6.2: Tabulka obsahující jednotlivé metriky strategií

Experimenty s testováním byly prováděny na stolním počítači s OS: Windows 10 Pro, Verze: 22H2, CPU: 13th Gen Intel(R) Core(TM) i5-13500 2.50 GHz, RAM: 48.0 GB, SSD. Ověření funkčnosti v prostředí Linuxu proběhlo otestováním modulů bez komplikací na notebooku s OS: Ubuntu, Verze: 22.04.4 LTS, CPU: Intel® Core™ i7-8850H CPU @ 2.60GHz, RAM 16 GB, SSD.



Obrázek 6.18: Vývoj kapitálu virtuálních peněženek v USD v čase pro GPT-4 Strategii, GPT-4_Strategie_EVO a jejich zpožděné verze o 10 vteřin

6.8 Experiment se zpožděním reakce směnární

Testování a experimentování se zpožděním vyhodnocení manipulací s pozicemi na straně směnární bylo prováděno rozšířením konfigurace strategií o *exchange_delay* parametr následovně:

```
"exchange_delay": { "_UNPACK": [ 0.1, 0.5, 1, 5, 10 ] }
```

Přidáním parametru pro backtestování byly získány výsledky, které byly málo odlišné od originálních. Bylo to způsobené dobou trvání pozic, která byla v řádech hodin. Na obrázku 6.18 jsou opět stavy peněženek pro GPT-4_Strategie, GPT-4_Strategie_EVO a stavy peněženek při zpoždění vyhodnocování na straně směnární těchto strategií o 10 vteřin.

Kapitola 7

Závěr

Cílem této práce bylo prozkoumat téma algoritmického obchodování na burze s důrazem na backtesting, zahrnující získávání a zpracování dat. Součástí průzkumu byly také možnosti aplikace evolučních algoritmů k optimalizaci strategií pro automatické obchodování. Hlavním cílem bylo vytvořit komplexní framework skládající se z několika modulů. Výsledkem měl být framework pracující s historickými daty ve formě jednotlivých transakcí, který by byl schopen získat, zpracovat, vizualizovat tato data a provádět nad nimi simulace. Měl být kladen důraz na tvorbu simulace odpovídající co nejvíce reálnému obchodování. Framework měl být schopen backtestovat jednotlivé konfigurace strategií a také provádět optimalizace jejich parametrů.

Hlavním důvodem, proč tato práce vznikla, byla neexistence veřejně dostupného komplexního přístupu k backtestování algoritmických obchodních strategií za použití nezkreslených dat. Byl proveden rozsáhlý průzkum existujících řešení, který ukázal, že všechny dostupné nástroje, platformy a knihovny pracují s již zpracovanými daty nejčastěji ve formátu OHLCV. Tento přístup představuje omezení v podobě limitace přesnosti simulace na kratším časovém rámci, než je přesnost zpracovávaných dat. Také ochuzuje analýzu o cenné informace ztrátovou kompresí cenných transakčních dat.

Podařilo se zmapovat a analyzovat danou problematiku a navrhnout jednotlivé moduly. Tyto moduly následně implementovat podle návrhu s kompletní funkcionalitou. Jednotlivé moduly, ovladatelné pomocí příkazové řádky, jsou schopné provést všechny potřebné úkony pro uživatele, který si chce otestovat vlastní strategii a případně ji i vylepšit. Framework aktuálně obsahuje podporu pro získávání dat ze směnárny Bybit, zpracování základních datových typů a jejich vizualizaci. Dále umožňuje, když si uživatel nadefinuje vlastní strategie, provádět backtestování strategií a optimalizace parametrů pomocí genetického algoritmu. Implementace byla provedena s efektivním používáním abstrakce, dědičnosti a polymorfismu, aby uživatel mohl libovolně a s co nejmenší námahou rozšiřovat tento komplexní framework.

V rámci experimentů bylo navrženo pár strategií, které byly následně backtestovány a optimalizovány. Během provádění simulací byla jejich správnost ověřována pomocí kontrolních výstupů v podobě zaznamenávání změn indikátorů a informací o manipulacích s pozicemi. Získané výstupy byly ověřovány. Po úspěšném vylepšení strategie, kterou navrhovaly různé generativní modely, byly splněny všechny cíle této práce. Samozřejmě, efektivnost optimalizace závisela na účinnosti poskytnuté strategie a mnoha dalších faktorech.

Tato práce byla příspěvkem na konferenci Excel@FIT 2024. Za účelem nominace na tuto konferenci byl navržen plakát přiložený v příloze E spolu s logem a abstraktem práce. Také bylo natočeno krátké stručné video ukazující práci s frameworkem.

7.1 Návrhy na vylepšení frameworku

Během návrhu, implementace a testování frameworku postupně vzniklo velké množství návrhů, které by mohly vylepšit dílčí funkcionality jednotlivých modulů, stejně jako celého frameworku. Podstatné návrhy jsou zde uvedeny. Předmětem pokračování vývoje a testování tohoto frameworku by mohlo být:

- Provedení rozsáhlých experimentů ve snaze najít optimální profitabilní strategii, která by fungovala v jakémkoli trendu pro zvolený finanční nástroj.
- Implementace více datových typů a indikátorů umožňující tvorbu komplexnějších strategií.
- Rozšíření podpory pro směnárnu Binance a případně další.
- Rozšíření simulace o možnost vlastnění aktiva pro umožnění testování nákupu a prodeje aktiv v `Backtestovacím` modulu.
- Napsání post-simulačních optimalizačních strategie. Například optimalizace na základě omezení času, pro který má být strategie aktivní. Mohlo by se provést vyhodnocení pro jednotlivé seance a další časová rozmezí.
- Napsání detektoru vzorů svíčkových grafů, nebo integrace knihovny pro rozpoznávání těchto vzorů.
- Implementace grafického rozhraní. Například vytvoření webového rozhraní, kde si uživatel bude moci navolit vše, co by jinak musel vypisovat do příkazů do CLI jednotlivých modulů.
- Paralelizace simulací a provádění více iterací současně v `Optimalizačním` modulu.
- Rozšíření simulace aby uvažovala spread pro reálnější výsledky.
- Vylepšení fitness metody a implementace dalších metrik pro vyhodnocování efektivnosti strategií.
- Úprava práce s poplatky, aby bylo možné uvažovat Funding Fee (poplatek pomáhající vyrovnávat ceny derivátových finančních nástrojů s aktuálními tržními cenami aktiv.)
- Přidání podpory pro skoky cen finančních nástrojů, které vznikají mezi jednotlivými obchodními obdobími na burze.
- Upravení frameworku pro práci s daty ze směnárny v reálném čase. Umožnění komunikace s reálnou směnárnou pro vytvoření kompletního samostatného řešení obchodujícího s reálným kapitálem na reálné směnárně. Vzhledem ke snaze napsat framework co nejvíce reflektující realitu by toto rozšíření nemělo být příliš náročné. Mohlo by se jednat o implementaci `Real time` modulu pracujícího s aktuálními daty v reálném čase.

- Přehlednější monitorování stavu simulací a optimalizací. V případě spouštění nástroje na serveru by mohlo být vhodné monitorovat stav a vytíženost serveru (například použitím nástrojů Grafana¹ a Telegraf²) pro zaručení efektivního využívání prostředků.
- Automatické generování strategií za použití všech dostupných indikátorů a následné optimalizace parametrů.
- Optimalizace nástroje z pohledu časové náročnosti, která je poměrně vysoká kvůli zvolenému prototypovacímu jazyku Python a architektuře frameworku. Jedno z možných vylepšení je přepsání celého frameworku do nízkourovňového programovacího jazyka s důrazem na zefektivnění výpočtů pomocí vektorizace a paralelizace. Další možností je upravení Python kódu a celé architektury frameworku do podoby umožňující použití vektorizace.

¹<https://grafana.com/>

²<https://www.influxdata.com/time-series-platform/telegraf/>

Literatura

- [1] ACHELIS, S. B. *Technical Analysis from A to Z*. McGraw Hill New York, 2001.
- [2] BRABAZON, A., O'NEILL, M. a MCGARRAGHY, S. *Natural computing algorithms*. Berlin, Heidelberg: Springer Berlin, Heidelberg, 2015. Dostupné z: <https://doi.org/10.1007/978-3-662-43631-8>.
- [3] BRABAZON, A. a O'NEILL, M. *Biologically inspired algorithms for financial modelling*. Berlin, Heidelberg: Springer Berlin, Heidelberg, 2006. Dostupné z: <https://doi.org/10.1007/3-540-31307-9>.
- [4] DE JONG, K. Generalized Evolutionary Algorithms. In: ROZENBERG, G., BÄCK, T. a KOK, J. N., ed. *Handbook of Natural Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, s. 625–635. DOI: 10.1007/978-3-540-92910-9_20. ISBN 978-3-540-92910-9. Dostupné z: https://doi.org/10.1007/978-3-540-92910-9_20.
- [5] EIBEN, A. E. a SMITH, J. E. *Introduction to evolutionary computing*. Berlin, Heidelberg: Springer Berlin, Heidelberg, 2015. Dostupné z: <https://doi.org/10.1007/978-3-662-44874-8>.
- [6] FOREX.COM TRADING ACADEMY. Accessed: 2023-11-21. Dostupné z: <https://www.forex.com/ie/trading-academy>.
- [7] FTMO ACADEMY. Accessed: 2023-12-10. Dostupné z: <https://academy.ftmo.com>.
- [8] HARRIS, L. *Trading and Exchanges: Market Microstructure for Practitioners*. New York, NY: Oxford University Press, 2003. Financial Management Association Survey and Synthesis Series. ISBN 0-19-514470-8.
- [9] MEHTA, P., PANDYA, S. a KOTECHA, K. Harvesting social media sentiment analysis to enhance stock market prediction using deep learning. *PeerJ Computer Science*. PeerJ Inc. 2021, sv. 7, s. e476.
- [10] NTI, I. K., ADEKOYA, A. F. a WEYORI, B. A. A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*. Springer. 2020, sv. 53, č. 4, s. 3007–3057.
- [11] PICASSO, A., MERELLO, S., MA, Y., ONETO, L. a CAMBRIA, E. Technical analysis and sentiment embeddings for market trend prediction. *Expert Systems with Applications*. Elsevier. 2019, sv. 135, s. 60–70.
- [12] SCHWAGER, J. D. *Getting Started in Technical Analysis*. New York, NY: John Wiley & Sons, Inc., 1999. Getting Started In. ISBN 0-471-29542-6.

Příloha A

Další indikátory pro TA

Vzhledem k tomu, že se tato práce nezaměřuje na implementaci jednotlivých indikátorů, jejich vlastností nebo srovnání, jsou zbývající implementované ukazatele uvedeny v této příloze.

V následujících definicích bude použita časová jednotka t , která označuje časový rámec pro indikátor. Může být buď absolutní jako například 10 vteřin (10s), 5 minut (5m), 1 hodina (1h), nebo relativní jako 1 měsíc (1mo) (1M).

A.1 Klouzavý průměr konvergence divergence (MACD)

Moving Average Convergence Divergence (MACD) je populární indikátor technické analýzy, který se používá k identifikaci trendů a potenciálních změn ve směru hybnosti trhu.

Počítá se jako rozdíl dvou klouzavých průměrů, nejčastěji EMA(12) a EMA(26), a tím sleduje sbíhavost a rozbíhavost klouzavých průměrů. Výsledkem je čára MACD, která se doplňuje o signální čáru. Ta se počítá jako klouzavý průměr z MACD čáry, nejčastěji EMA(9). Následně je možné vypočítat histogram jako rozdíl MACD čáry a signální čáry.

$$\begin{aligned}MACD_L_t &= MA_t(p_1) - MA_t(p_2) \\MACD_S_t &= MA(p_3) \text{ pro hodnoty z } MACD_L \\MACD_H_t &= MACD_L_t - MACD_S_t\end{aligned}$$

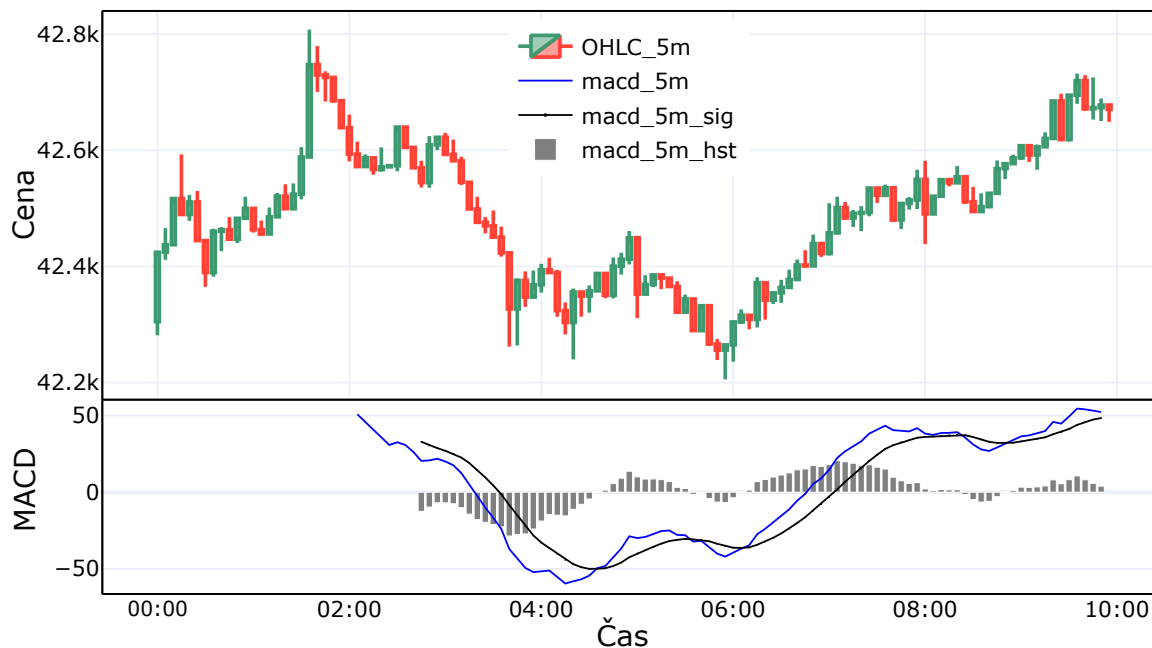
Kde t je časová jednotka, MA je zvolený klouzavý průměr (SMA, nebo EMA), p_1 je perioda prvního klouzavé průměru, p_2 je perioda druhého klouzavé průměru, p_3 je perioda klouzavé průměru pro výpočet signální čáry, $MACD_L$ je MACD čára, $MACD_S$ je signální čára a $MACD_H$ je histogram. Platí $p_2 > p_1$. Hodnota $MACD_L$ je validní po $t \geq p_2$, hodnoty $MACD_S$ a $MACD_H$ pak pro $t \geq (p_2 + p_3)$.

Nejčastější využití indikátoru je hledání křížení MACD čáry a signální čáry jako změny trendu. Také poskytuje další náhled na sílu a hybnost trendu.

Na obrázku [A.1](#) je ukázka MACD s pětiminutovým časovým rámcem a klouzavými průměry EMA(12) a EMA(26). Signální čára je počítána jako EMA(9). Modrou barvou je zobrazena MACD čára, černou signální čára a šedou barvou histogram.

A.2 Index relativní síly (RSI)

Relative Strength Index (RSI) je oscilátor hybnosti měřící rychlost a změnu cenových pohybů. RSI osciluje mezi 0 a 100, trh je považován za překoupený, pokud je hodnota indexu



Obrázek A.1: Ukázka MACD s pětiminutovým časovým rámcem a klouzavými průměry EMA(12) a EMA(26). Signální čára je počítána jako EMA(9).

větší než x a přeprodáný pokud je menší než y , kde x a y jsou zvolené konstanty, nejčastěji $x = 70$ a $y = 30$.

Standardně se počítá z *close* hodnot, ale je možné použít libovolnou ziskatelnou hodnotu z OHLC. RSI se vypočítá následovně:

$$\begin{aligned}
 G_0 &= 0; G_t = \max(0, V_t - V_{t-1}) \\
 L_0 &= 0; L_t = (-1)\min(0, V_t - V_{t-1}) \\
 AG_t(p) &= \frac{1}{p} \sum_{i=0}^{p-1} G_{t-i} \\
 AL_t(p) &= \frac{1}{p} \sum_{i=0}^{p-1} L_{t-i} \\
 RS_t(p) &= \frac{AG_t(p)}{AL_t(p)} \\
 RSI_t(p) &= 100 - \frac{100}{1 + RS_t(p)}
 \end{aligned}$$

Kde p je zvolená perioda, t je časová jednotka, RS je relativní síla, AG je průměrný zisk neboli průměrná velikost pohybu nahoru pro posledních p hodnot, AL je průměrná ztráta neboli průměrná velikost pohybu dolů pro posledních p hodnot, G je zisk, L je ztráta, Hodnoty RSI , RS , AG_t , AL_t jsou validní po $t \geq p$. Pro $AG_t(p) = 0$ je hodnota $RS = inf$.

Ukázka RSI s periodou 10 a pětiminutovým časovým rámcem je na obrázku A.2 zobrazena černou čarou. Dále jsou v grafu zobrazeny červenou barvou signální čáry na hodnotách 30 a 70.



Obrázek A.2: Ukázka RSI s periodou 10 a pětiminutovým časovým rámcem

A.3 Swingové body

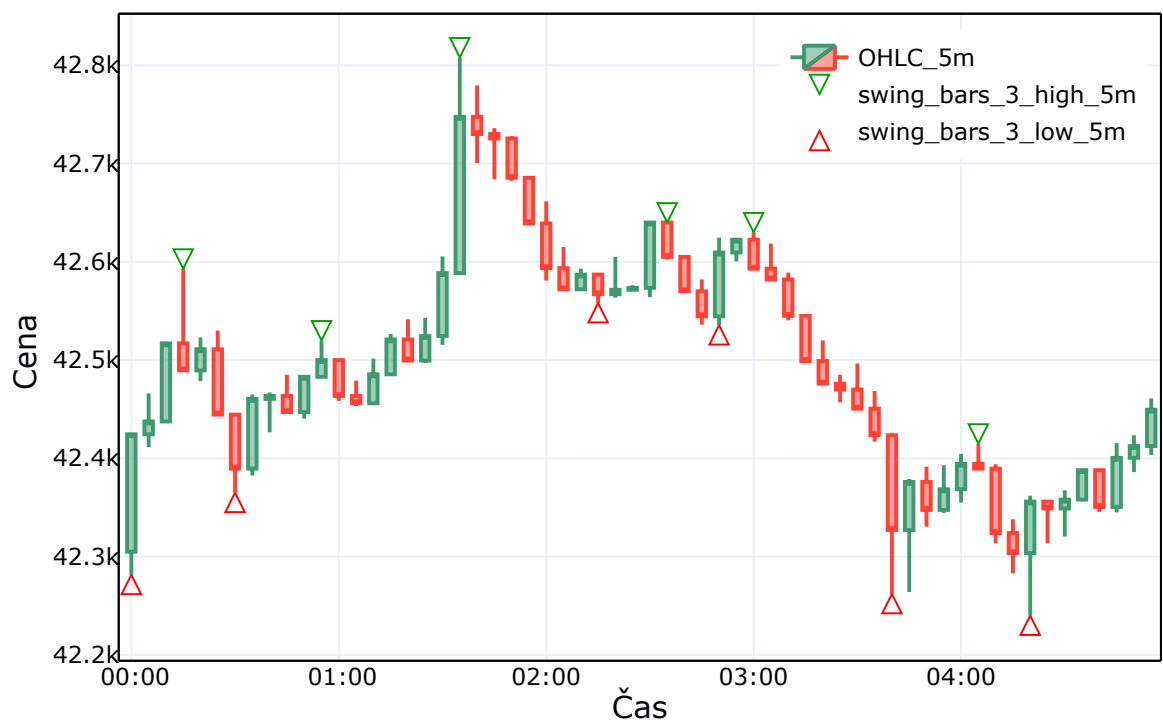
V této práci budeme uvažovat swingová maxima (Swing High) a swingová minima (Swing Low) jako swingové body. Jedná se o lokální extrémy ve svíčkových grafech v určeném okolí. Okolí je nejčastěji zvolené jako 2 nebo 3 svíčky.

Body můžeme definovat následovně:

$$\begin{aligned}
 o(t) &= \{ohlc_i | i \in \mathbb{N}_0 : ((t - n * r) \leq i < t \vee t < i \leq (t + n * r))\} \\
 oh(t) &= \{high(ohlc) | \forall ohlc \in o(t)\} \\
 ol(t) &= \{low(ohlc) | \forall ohlc \in o(t)\} \\
 SM &= \{(t, high_t) | \forall t \in \mathbb{N}_0 : high_t > max(oh(t))\} \\
 SL &= \{(t, low_t) | \forall t \in \mathbb{N}_0 : low_t < min(ol(t))\}
 \end{aligned}$$

Kde t je časová jednotka, r je časový rámeček, n je přirozené číslo označující velikost okolí, SL je množina všech swingových minim, SM je množina všech swingových maxim, $ol()$ je funkce vracející všechny low hodnoty v okolí, $oh()$ je funkce vracející všechny $high$ hodnoty v okolí, $o()$ je funkce vracející všechny $ohlc$ hodnoty v okolí, $low()$ je funkce vracející low hodnotu z $ohlc$, $high()$ je funkce vracející $high$ hodnotu z $ohlc$.

Na obrázku A.3 je možné vidět vyznačené swingové body na svíčkovém grafu pro pětiminutový časový rámeček s okolím nastaveným na 3 svíčky.



Obrázek A.3: Ukázka swingových bodů na svíčkovém grafu s pětiminutovým časovým rámcem

Příloha B

Tabulka směnáren

Za účelem stanovení vhodné směnárny pro získávání dat byl uskutečněn rozsáhlý průzkum všech směnáren poskytujících data. Pro získání informací byly napsány skripty ke scrapování (získávání) informací ze stránek CoinMarketCap¹ a CoinGecko².

Ze získaných dat vznikly po zpracování tabulky **B.1**, **B.2** a **B.3**, ve kterých je vidět 30 nejlépe hodnocených směnáren. Jedná se jen o orientační hodnoty, které mohou být zkreslené různými faktory. Také se tato data v čase mění. Hodnocení bylo stanoveno na základě hodnoty Alexa Rank³. Jedná se o číslo, které ukazuje, jak dobře se webová stránka umísťuje ve srovnání s ostatními na internetu. Čím nižší číslo, tím lepší pozice.

Data obsahují vždy název směnárny. V tabulce **B.1** dále informaci, jestli je směnárna centralizovaná, počty finančních nástrojů pro deriváty: perpetuální, futures; spotové, jejich součty: derivátové, celkem; a zemi, pod kterou je směnárna vedena.

V tabulce **B.2** pak obsahuje spotový, derivátový a celkový denní objem v USD, denní derivátový Open Interest, seznam hlavních podporovaných fiat měny a jejich celkový počet.

Tabulka **B.3** pak obsahuje datum spuštění směnárny, Alexa Rank, skóre důvěry⁴, CM skóre⁵, počet denních návštěv, počet sledujících na sociální síti Twitter a zdroje ze kterých bylo čerpáno, kde CM reprezentuje CoinMarketCap a GC CoinGecko.

¹<https://coinmarketcap.com/>

²<https://www.coingecko.com/>

³https://en.wikipedia.org/wiki/Alexa_Internet#Alexa_Traffic_Rank

⁴Skóre důvěry (Trust Score) hodnota z algoritmu vyvinutého společností CoinGecko, který hodnotí legitimitu objemu obchodování na burze.

⁵CoinMarketCap skóre řadí a hodnotí burzy na základě více faktorů, kterým jsou přiřazeny různé váhy.

Směnárna	Centralizace	Spotové f. n.	Perpetuální f. n.	Futures f. n.	Derivátové f. n.	Celkem f. n.	Země
Binance	centralized	387	266	33	299	686	Cayman Islands
Coinbase Exchange	centralized	245			153	398	United States
Bybit	centralized	451	287	62	349	800	British Virgin Islands
KuCoin	centralized	726	227	2	229	955	Seychelles
DigiFinex	centralized	412			65	477	Seychelles
Gate.io	centralized	1,777	319	0	319	2,096	Cayman Islands
MEXC	centralized	1,662	349	0	349	2,011	Seychelles
OKX	centralized	319	175	254	429	748	Seychelles
XT.COM	centralized	445	348	0	348	793	Seychelles
BingX	centralized	544	182	0	182	726	British Virgin Islands
Bitget	centralized	605	224	0	224	829	Seychelles
LBank	centralized	544			136	680	British Virgin Islands
HTX	centralized	603	114	8	122	725	Seychelles
WhiteBIT	centralized	215	26	0	26	241	Lithuania
P2B	centralized	146			78	224	Lithuania
Kraken	centralized	240	119	37	156	396	United States
Crypto.com Exchange	centralized	292	133	14	147	439	Malta
Bitrue	centralized	521	153	0	153	674	Singapore
Phemex	centralized	188	218	0	218	406	Singapore
Upbit Indonesia	centralized						Indonesia
BitMart	centralized	727			102	829	Cayman Islands
YoBit	centralized	205			42	247	Russia
Coinsbit	centralized	81			54	135	Estonia
Bitforex	centralized		13	0	13	13	Seychelles
CoinEx	centralized	739	130	0	130	869	Samoa
Bitso	centralized	57			42	99	Gibraltar
Bitfinex	centralized	182	91	0	91	273	British Virgin Islands
Binance US	centralized						United States
Poloniex	centralized	410	46	0	46	456	Seychelles
Bithumb	centralized	257	0	0	0	257	South Korea

Tabulka B.1: Informace získané k jednotlivým směnárnám ze dne 17. 10. 2023 - 1. část

Směnárna	Spotový denní objem [€]	Derivátový denní objem [€]	Celkový denní objem [€]	D. derivátový Open interest [€]	Podporované fiat měny	P. fiat měn
Binance	7,200,000,000	36,000,000,000	43,000,000,000	7,600,000,000	"EUR, GBP, BRL, ... "	11
Coinbase Exchange	1,000,000,000		1,000,000,000		"USD, EUR, GBP "	3
Bybit	1,600,000,000	12,000,000,000	14,000,000,000	4,600,000,000	"USD, EUR, GBP, ... "	6
KuCoin	980,000,000	1,800,000,000	2,800,000,000	1,200,000,000	"USD, AED, ARS, ... "	48
DigiFinex	910,000,000	8,200,000,000	9,100,000,000	2,700,000,000	"USD, EUR, AED, ... "	28
Gate.io	780,000,000	1,600,000,000	2,400,000,000	1,100,000,000	"EUR, USD, GBP, ... "	22
MEXC	940,000,000	4,300,000,000	5,200,000,000	1,100,000,000		
OKX	1,300,000,000	15,000,000,000	16,000,000,000	3,200,000,000	"AED, ARS, AUD, ... "	46
XT.COM	840,000,000	8,400,000,000	9,200,000,000	150,000,000	"USD, KRW, NGN, ... "	12
BingX	1,100,000,000	5,200,000,000	6,300,000,000	1,600,000,000	"USD, THB, MYR, ... "	7
Bitget	480,000,000	5,600,000,000	6,100,000,000	2,200,000,000	"USD, EUR, GBP, ... "	15
LBank	660,000,000	5,400,000,000	6,100,000,000	9,400,000	"EUR, USD, CAD, ... "	10
HTX	1,400,000,000	1,800,000,000	3,200,000,000	180,000,000	"ALL, AUD, BRL, ... "	50
WhiteBIT	760,000,000	2,200,000,000	3,000,000,000	560,000,000	"USD, EUR, UAH, ... "	10
P2B	750,000,000		750,000,000			
Kraken	640,000,000	9,000,000	650,000,000	140,000,000	"USD, EUR, GBP, ... "	7
Crypto.com Exchange	510,000,000	680,000,000	1,200,000,000	15,000,000	EUR	1
Bittrue	1,200,000,000	8,900,000,000	10,000,000,000	460,000,000	"USD, EUR, GBP, ... "	10
Phemex	64,000,000	3,400,000,000	3,500,000,000	990,000,000		
Upbit Indonesia	9,000,000		9,000,000			
BitMart	1,800,000,000	9,100,000,000	11,000,000,000	860,000,000	"USD, EUR "	2
YoBit	90,000,000		90,000,000			
Coinsbit	240,000,000		240,000,000		"USD, EUR "	2
Bitforex	1,500,000,000	1,600,000,000	3,100,000,000	110,000,000		
CoinEx	25,000,000	220,000,000	240,000,000	26,000,000		
Bitso	36,000,000		36,000,000		"MXN, ARS, BRL, ... "	4
Bitfinex	140,000,000	200,000,000	340,000,000	440,000,000	"USD, EUR, GBP, ... "	4
Binance US	14,000,000		14,000,000			
Poloniex	190,000,000	14,000,000	200,000,000	4,800,000	"USD, CAD, JPY, ... "	36
Bithumb	420,000,000	0	420,000,000		KRW	1

Tabulka B.2: Informace získané k jednotlivým směnárnám ze dne 17. 10. 2023 - 2. část

Směnárna	Datum spuštění	Alexa Rank	Skóre důvěry	CM skóre	Denní návštěvy	Počet sledujících na Twitteru	Zdroje
Binance	Jul-17	132	10	9.9	1,400,000	10,825,944	CM, GC
Coinbase Exchange		730	10	8.5	440,000	1,212,856	CM, GC
Bybit	Mar-18	1,034	10	7.3	480,000	2,682,123	CM, GC
KuCoin	Aug-17	1,209	9	7.3	160,000	2,594,115	CM, GC
DigiFinex	Apr-18	1,922	8	5.7	200,000	170,629	CM, GC
Gate.io	Jan-13	2,039	9	6.9	400,000	1,382,153	CM, GC
MEXC	Apr-18	2,100	10	6.4	210,000	987	CM, GC
OKX	Jan-17	2,386	10	7.2	740,000	3,470,172	CM, GC
XT.COM	Oct-18	3,638	8	5.8	320,000	192,581	CM, GC
BingX	May-18	3,932	8	5.4	230,000	189,155	CM, GC
Bitget	Apr-18	4,266	9	6.4	240,000	1,173,795	CM, GC
LBank	Oct-16	4,403	7	6.4	320,000	445,705	CM, GC
HTX	Sep-13	4,416	10	6.7	330,000	59,836	CM, GC
WhiteBIT	Dec-18	4,538	8	5.2	440,000	125,126	CM, GC
P2B		4,658	8	5.7	160,000	77,835	CM, GC
Kraken	Jul-11	4,939	10	8.3	150,000	1,510,942	CM, GC
Crypto.com Exchange	Nov-19	5,392	10	6.4	90,000	2,693,462	CM, GC
Bittrue	Jul-18	6,290	7	6	150,000	459,443	CM, GC
Phemex	Nov-19	6,399	8	5.3	42,000	502,337	CM, GC
Upbit Indonesia		6,610			2,000	8,281	GC
BitMart	Mar-18	7,585	8	5.8	420,000	873,510	CM, GC
YoBit		8,751	4	2.8	36,000	165,679	CM, GC
Coinsbit		9,227	8	5.4	310,000	81,147	CM, GC
Bitforex		9,991	5		180,000	159,170	GC
CoinEx	Dec-17	10,165	7	3.2	38,000	296,873	CM, GC
Bitso		10,272	8	5.2	24,000	115,735	CM, GC
Bitfinex	Oct-12	10,769	10	7	22,000	933,325	CM, GC
Binance US		12,850	10		29,000	435,559	GC
Poloniex	Jan-14	13,151	8	5.6	26,000	556,821	CM, GC
Bithumb		13,467	7	6.6	52,000	251,473	CM, GC

Tabulka B.3: Informace získané k jednotlivým směnárnám ze dne 17. 10. 2023 - 3. část

Příloha C

Strategie vygenerované pomocí generativních modelů

V této příloze jsou celé odpovědi jednotlivých generativních modelů na žádost o vygenerování strategie. Odpovědi byly bohužel podobné a všechny modely kombinovaly indikátory MACD, RSI, SMA a EMA.

Žádost zněla: Vytvoř jednu funkční strategii pro algoritmické obchodování. Použij některé známé indikátory. Napiš ji v jednoduše čitelném formátu ve pseudokódu se všemi parametry nastavenými tak, aby maximalizovala zisk pro BTSUSD pár.

Výsledky jsou z března 2024. Kvůli přesnějším odpovědím byl dotaz pro modely v anglickém jazyce. Pro lepší čitelnost byly odpovědi formátovány pro lepší zobrazení v tomto dokumentu se zachováním obsahu.

C.1 Instrukce pro modely:

Create one functional strategy for algorithmic trading. Use some well-known indicators for it. Write it in a simple readable format in pseudocode with all parameters set up to maximize profit for BTSUSD instrument.

C.2 Odpověď od modelu GPT-3.5

Sure, here's a simple pseudocode for an algorithmic trading strategy using the Moving Average Convergence Divergence (MACD) indicator along with the Relative Strength Index (RSI) indicator for trading the BTSUSD instrument: [Algorithm 2](#)

This pseudocode defines a simple trading strategy using the MACD and RSI indicators. It buys when the MACD histogram crosses below zero and the RSI is below the oversold threshold, and sells when the MACD histogram crosses above zero and the RSI is above the overbought threshold. Adjusting the parameters and adding more conditions can further refine the strategy. Remember, this is a basic framework and should be thoroughly backtested before use in live trading.

Algorithmus 2 Algorithm from GPT-3.5

```
1: capital ← 1000                                ▷ Initial capital
2: position_size ← 0.2                            ▷ Percentage of capital to allocate per trade
3: short_ma_period ← 12                          ▷ Short moving average period for MACD
4: long_ma_period ← 26                           ▷ Long moving average period for MACD
5: signal_period ← 9                             ▷ Signal line period for MACD
6: rsi_period ← 14                               ▷ RSI period
7: rsi_oversold_threshold ← 30                   ▷ RSI oversold threshold
8: rsi_overbought_threshold ← 70                 ▷ RSI overbought threshold
9: btcusd_data ← get historical data("BTSUSD")    ▷ Get historical price data
10: for each new_data_point in btcusd_data do
11:   short_ema ← calculate EMA(btcusd_data, short_ma_period)
12:   long_ema ← calculate EMA(btcusd_data, long_ma_period)
13:   macd_line ← short_ema − long_ema
14:   signal_line ← calculate EMA(macd_line, signal_period)
15:   macd_histogram ← macd_line − signal_line
16:   close_prices ← btcusd_data[: current_index]
17:   rsi ← calculate RSI(close_prices, rsi_period)
18:   if macd_histogram > 0 and rsi > rsi_overbought_threshold then
19:     sell_size ← position_size × capital
20:     execute sell order("BTSUSD", sell_size)
21:     capital ← capital − sell_size
22:   else if macd_histogram < 0 and rsi < rsi_oversold_threshold then
23:     buy_size ← position_size × capital
24:     execute buy order("BTSUSD", buy_size)
25:     capital ← capital − buy_size
26:   end if
27: end for
```

C.3 Odpověď od modelu claude-3-opus-20240229

Here's a simple algorithmic trading strategy using the Moving Average Convergence Divergence (MACD) and Relative Strength Index (RSI) indicators for the BTSUSD instrument, written in pseudocode: Algorithm 3

Explanation: The strategy uses the MACD and RSI indicators to generate buy and sell signals. The MACD is calculated using the calculateMACD function with the specified fastPeriod, slowPeriod, and signalPeriod. The RSI is calculated using the calculateRSI function with the specified rsiPeriod.

The trading conditions are as follows: **Buy signal:** If the MACD crosses above zero (turns positive) and the previous MACD value was less than or equal to zero, and the RSI is below the rsiUpperThreshold, and there is no existing position, a buy order is placed. **Sell signal:** If the MACD crosses below zero (turns negative) and the previous MACD value was greater than or equal to zero, and the RSI is above the rsiLowerThreshold, and there is an existing position, a sell order is placed.

The tradingAmount parameter determines the amount of BTC to be traded in each order. The previous MACD value is updated after each iteration to keep track of the cross-

Algoritmus 3 Algorithm from claude-3-opus-20240229

```
1: fastPeriod ← 12
2: slowPeriod ← 26
3: signalPeriod ← 9
4: rsiPeriod ← 14
5: rsiUpperThreshold ← 70
6: rsiLowerThreshold ← 30
7: tradingAmount ← 0.01                                ▷ BTC amount per trade
8: position ← 0
9: macdPrevious ← 0
10: while true do
11:   price ← fetchLatestPrice("BTSUSD")
12:   macd ← calculateMACD(price, fastPeriod, slowPeriod, signalPeriod)
13:   rsi ← calculateRSI(price, rsiPeriod)
14:   if macd > 0 and macdPrevious ≤ 0 and rsi < rsiUpperThreshold and
     position == 0 then
15:     buyOrder ← placeBuyOrder("BTSUSD", tradingAmount, price)
16:     position ← 1
17:   else if macd < 0 and macdPrevious ≥ 0 and rsi > rsiLowerThreshold and
     position == 1 then
18:     sellOrder ← placeSellOrder("BTSUSD", tradingAmount, price)
19:     position ← 0
20:   end if
21:   macdPrevious ← macd
22:   wait(60)                                             ▷ Wait for 1 minute before the next iteration
23: end while
```

sover. The strategy waits for a specified interval (e.g., 1 minute) before the next iteration to avoid excessive trading.

Note: This is a simplified example, and you would need to implement the `fetchLatestPrice`, `calculateMACD`, `calculateRSI`, `placeBuyOrder`, and `placeSellOrder` functions based on your specific trading platform or API. Additionally, you may need to adjust the parameters and trading conditions based on your own analysis and risk tolerance to maximize profit for the BTSUSD instrument.

C.4 Odpověď od modelu GPT-4

Strategy: Moving Average Crossover

Indicators: Simple Moving Average (SMA), Exponential Moving Average (EMA)

Pseudocode: Algorithm 4

This strategy uses a short period SMA and a long period EMA. When the SMA crosses above the EMA, it indicates a possible upward trend, so the strategy buys the instrument. When the SMA crosses below the EMA, it indicates a possible downward trend, so the strategy sells the instrument.

C.5 Odpověď od modelu Gemini Pro

Algorithmic Trading Strategy (Pseudocode) - BTSUSD

Algorithm 5

This strategy combines momentum (RSI) with trend direction and potential reversals (MACD). Entry and exit signals involve both RSI and MACD for confirmation. Parameters can be adjusted based on risk tolerance and market conditions. Backtesting is crucial before deploying with real capital. This is a simplified version; real trading requires additional logic for order management and risk control.

Disclaimer: Educational purposes only, not financial advice. Algorithmic trading involves inherent risks, and past performance does not guarantee future results.

C.6 Odpověď od modelu Gemini 1.5 Pro

Algorithmic Trading Strategy for BTSUSD: Moving Average Crossover with RSI Filter This strategy utilizes two well-known indicators: Moving Averages and the Relative Strength Index (RSI), aiming to capitalize on trends while filtering out potentially false signals.

Indicators: Short-term Moving Average (SMA_short): 5 periods Long-term Moving Average (SMA_long): 20 periods Relative Strength Index (RSI): 14 periods, overbought threshold 70, oversold threshold 30

Pseudocode: Algorithm 6

Use code with caution.

Explanation: The strategy enters a long position when the short-term SMA crosses above the long-term SMA, indicating an uptrend, and the RSI is below the overbought threshold (70) to confirm momentum. Conversely, it enters a short position when the short-term SMA crosses below the long-term SMA and the RSI is above the oversold threshold (30). The strategy exits positions when the SMA crossover reverses, aiming to capture profits or limit losses.

Notes: This is a basic example and requires further optimization and risk management strategies. Backtesting and paper trading are crucial before deploying real capital. Consider incorporating additional indicators or conditions for more robust signals. Adjust the SMA periods and RSI thresholds based on your risk tolerance and market conditions.

Disclaimer: This is not financial advice, and any trading decisions should be made based on your own research and risk tolerance.

Algorithmus 4 Algorithm from GPT-4

```
1: Parameters:
2: short_sma_period = 20
3: long_ema_period = 50
4: instrument = "BTSUSD"
5: amount_to_trade = 100
6:
7: while market is open do
8:   short_sma ← calculate_SMA(instrument, short_sma_period)
9:   long_ema ← calculate_EMA(instrument, long_ema_period)
10:
11:   if short_sma crosses above long_ema then
12:     if not in position then
13:       buy(instrument, amount_to_trade)
14:     else
15:       do nothing
16:     end if
17:   else if short_sma crosses below long_ema then
18:     if in position then
19:       sell(instrument, amount_to_trade)
20:     else
21:       do nothing
22:     end if
23:   end if
24:
25:   wait for next market tick
26: end while
27:
28: Functions:
29:
30: function CALCULATE_SMA(instrument, period)
31:   get close prices for instrument for the last 'period' time
32:   return average of these prices
33: end function
34:
35: function CALCULATE_EMA(instrument, period)
36:   get close prices for instrument for the last 'period' time
37:   calculate EMA using these prices
38:   return EMA
39: end function
40:
41: function BUY(instrument, amount)
42:   submit a buy order for 'amount' of 'instrument'
43: end function
44:
45: function SELL(instrument, amount)
46:   submit a sell order for 'amount' of 'instrument'
47: end function
```

Algorithmus 5 Algorithm from Gemini Pro

Input: Entry_Long_Threshold, Exit_Long_Threshold, Entry_Short_Threshold, Exit_Short_Threshold, Minimum_Trade_Amount, Order_Validity

Output: Buy/Sell orders

- 1: **Indicators:**
 - 2: - Relative Strength Index (RSI): Measures momentum and identifies overbought/oversold conditions.
 - 3: - Moving Average Convergence Divergence (MACD): Identifies trend direction and potential reversals.
 - 4:
 - 5: **Logic:**
 - 6: - Continuously monitor BTSUSD market price.
 - 7: - Calculate the Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD).
 - 8: - **IF** RSI is greater than $30 + \text{Entry_Long_Threshold}$ **AND** MACD crosses above its signal line **AND** current position is not long:
 - 9: Place a buy order for Minimum_Trade_Amount.
 - 10: Set order validity to Order_Validity.
 - 11: - **IF** RSI is less than $70 - \text{Exit_Long_Threshold}$ **OR** MACD crosses below its signal line (bearish crossover) **AND** current position is long:
 - 12: Place a sell order to close the long position.
 - 13: - **IF** RSI is less than $70 - \text{Entry_Short_Threshold}$ **AND** MACD crosses below its signal line (bearish crossover) **AND** current position is not short:
 - 14: Place a sell short order for Minimum_Trade_Amount.
 - 15: Set order validity to Order_Validity.
 - 16: - **IF** RSI is greater than $30 + \text{Exit_Short_Threshold}$ **OR** MACD crosses above its signal line (bullish crossover) **AND** current position is short:
 - 17: Place a buy order to close the short position.
-

Algorithmus 6 Algorithm from Gemini 1.5 Pro

```
1: // Initialization
2:  $SMA_{short} = 0$ 
3:  $SMA_{long} = 0$ 
4:  $RSI = 0$ 
5:  $position = 0$  ▷ 0 = no position, 1 = long, -1 = short
6:
7: function UPDATE_INDICATORS(close_price)
8:    $SMA_{short} = \text{calculate\_SMA}(\text{close\_price}, 5)$ 
9:    $SMA_{long} = \text{calculate\_SMA}(\text{close\_price}, 20)$ 
10:   $RSI = \text{calculate\_RSI}(\text{close\_price}, 14)$ 
11: end function
12:
13: while true do
14:   // Get current price
15:    $current\_price = \text{get\_current\_price}("BTSUSD")$ 
16:
17:   // Update indicators
18:   UPDATE_INDICATORS( $current\_price$ )
19:
20:   // Entry/Exit conditions
21:   if  $position == 0$  then ▷ No current position
22:     if  $SMA_{short} > SMA_{long}$  and  $RSI < 70$  then ▷ Enter long position
23:       BUY("BTSUSD")
24:        $position = 1$ 
25:     else if  $SMA_{short} < SMA_{long}$  and  $RSI > 30$  then ▷ Enter short position
26:       SELL("BTSUSD")
27:        $position = -1$ 
28:     end if
29:   else if  $position == 1$  then ▷ Long position
30:     if  $SMA_{short} < SMA_{long}$  then ▷ Exit long position
31:       SELL("BTSUSD")
32:        $position = 0$ 
33:     end if
34:   else if  $position == -1$  then ▷ Short position
35:     if  $SMA_{short} > SMA_{long}$  then ▷ Exit short position
36:       BUY_TO_COVER("BTSUSD")
37:        $position = 0$ 
38:     end if
39:   end if
40:
41:   // Wait for next time interval (e.g., 1 minute)
42:   sleep(60)
43: end while
```

Příloha D

Ukázka Freqtrade backtestingu

V rámci ověřování správnosti výsledků a zkoumání existujících řešení, byl proveden backtesting pomocí Freqtrade frameworku za účelem porovnání výsledků. Data pro tento backtesting byly z Bybit směnárny pro BTC/USDT. Zde je podstatná část definice strategie:

```
class MyStrategy(IStrategy):
    stoploss = -0.02
    timeframe = '15m'
    ...

    def populate_indicators(self, dataframe: DataFrame, metadata: dict) -> DataFrame:
        dataframe['sma_20'] = ta.SMA(dataframe, timeperiod=20)
        dataframe['ema_50'] = ta.EMA(dataframe, timeperiod=50)
        return dataframe

    def populate_entry_trend(self, dataframe: DataFrame, metadata: dict) -> DataFrame:
        dataframe.loc[ (
            (dataframe['sma_20'].crossed_above(dataframe['ema_50'])) # Long entry
        ), 'enter_long'] = 1
        return dataframe

    def populate_exit_trend(self, dataframe: DataFrame, metadata: dict) -> DataFrame:
        dataframe.loc[ (
            (dataframe['sma_20'].crossed_below(dataframe['ema_50'])) # Long exit
        ), 'exit_long'] = 1
        return dataframe
```

Dále v konfiguračním souboru byl nastaven počáteční kapitál na 1000 USDT, výše pozice na 100 USDT a poplatky ekvivalentní základním poplatkům pro Bybit směnárnu.

Pro stažení dat byl použit příkaz:

```
freqtrade download-data --exchange bybit --pairs BTC/USDT -t 15m --days 400
```

Pro zahájení backtestingu se strategií MyStrategy pro dva měsíce dat začínající 1. 3. 2024 byl použit příkaz:

```
freqtrade backtesting --strategy MyStrategy --timerange 20240301-20240501
```

Backtesting vrátil následující výsledky a uložil informace o jednotlivých pozicích do souboru.

```
Result for strategy MyStrategy
===== BACKTESTING REPORT =====
| Pair | Entries | Avg Profit % | Tot Profit USDT | Tot Profit % | Avg Duration | Win | Draw | Loss | Win% |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| BTC/USDT | 61 | -0.37 | -22.303 | -2.23 | 11:33:00 | 19 | 0 | 42 | 31.1 |
| TOTAL | 61 | -0.37 | -22.303 | -2.23 | 11:33:00 | 19 | 0 | 42 | 31.1 |

===== LEFT OPEN TRADES REPORT =====
| Pair | Entries | Avg Profit % | Tot Profit USDT | Tot Profit % | Avg Duration | Win | Draw | Loss | Win% |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TOTAL | 0 | 0.00 | 0.000 | 0.00 | 0:00 | 0 | 0 | 0 | 0 |
```

```

===== ENTER TAG STATS =====
| TAG | Entries | Avg Profit % | Tot Profit USDT | Tot Profit % | Avg Duration | Win Draw Loss Win% |
|-----|-----|-----|-----|-----|-----|-----|-----|
| OTHER | 61 | -0.37 | -22.303 | -2.23 | 11:33:00 | 19 0 42 31.1 |
| TOTAL | 61 | -0.37 | -22.303 | -2.23 | 11:33:00 | 19 0 42 31.1 |

```

```

===== EXIT REASON STATS =====
| Exit Reason | Exits | Avg Profit % | Tot Profit USDT | Tot Profit % | Avg Duration | Win Draw Loss Win% |
|-----|-----|-----|-----|-----|-----|-----|-----|
| exit_signal | 48 | 0.13 | 6.260 | 0.63 | 13:18:00 | 19 0 29 39.6 |
| stop_loss | 13 | -2.20 | -28.564 | -2.86 | 5:09:00 | 0 0 13 0 |
| TOTAL | 61 | -0.37 | -22.303 | -2.23 | 11:33:00 | 19 0 42 31.1 |

```

```

===== SUMMARY METRICS =====
| Metric | Value |
|-----|-----|
| Backtesting from | 2024-03-01 00:00:00 |
| Backtesting to | 2024-05-01 00:00:00 |
| Max open trades | 1 |
| Total/Daily Avg Trades | 61 / 1.0 |
| Starting balance | 1000 USDT |
| Final balance | 977.697 USDT |
| Absolute profit | -22.303 USDT |
| Total profit % | -2.23% |
| CAGR % | -12.63% |
| Sortino | -9.21 |
| Sharpe | -3.93 |
| Calmar | -25.41 |
| Profit factor | 0.57 |
| Expectancy (Ratio) | -0.37 (-0.30) |
| Avg. daily profit % | -0.04% |
| Avg. stake amount | 99.965 USDT |
| Total trade volume | 6097.881 USDT |
| Best Pair | BTC/USDT -2.23% |
| Worst Pair | BTC/USDT -2.23% |
| Best trade | BTC/USDT 6.88% |
| Worst trade | BTC/USDT -2.20% |
| Best day | 6.887 USDT |
| Worst day | -4.395 USDT |
| Days win/draw/lose | 14 / 16 / 30 |
| Avg. Duration Winners | 22:17:00 |
| Avg. Duration Loser | 6:42:00 |
| Max Consecutive Wins / Loss | 2 / 12 |
| Rejected Entry signals | 0 |
| Entry/Exit Timeouts | 0 / 0 |
| Min balance | 977.697 USDT |
| Max balance | 1005.339 USDT |
| Max % of account underwater | 2.75% |
| Absolute Drawdown (Account) | 2.75% |
| Absolute Drawdown | 27.642 USDT |
| Drawdown high | 5.339 USDT |
| Drawdown low | -22.303 USDT |
| Drawdown Start | 2024-03-05 08:30:00 |
| Drawdown End | 2024-04-30 08:30:00 |
| Market change | -0.99% |

```

Backtested 2024-03-01 00:00:00 -> 2024-05-01 00:00:00 | Max open trades : 1

```

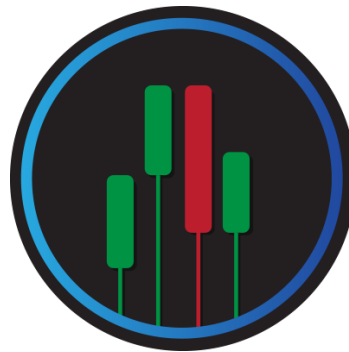
===== STRATEGY SUMMARY =====
| Strategy | Entries | Avg Profit % | Tot Profit USDT | Tot Profit % | Avg Duration | Win Draw Loss Win% | |
|---|---|---|---|---|---|---|---|
| MyStrategy | 61 | -0.37 | -22.303 | -2.23 | 11:33:00 | 19 0 42 31.1 | 27.642 USDT |
| Drawdown | 2.75% |

```

Příloha E

Podklady na konferenci Excel@FIT 2024

Tato práce se účastnila konference Excel@FIT 2024¹. Na stránkách konference je možné tuto práci dohledat pod číslem 39. Pro nominaci musel být vytvořen plakát, logo a článek. Logo je možné vidět na obrázku E.1. Plakát² zaměřený na výsledky práce a článek³ v podobě upraveného rozšířeného abstraktu jsou na následujících stránkách.



Obrázek E.1: Logo frameworku vytvořené pro konferenci Excel@FIT 2024

¹<https://excel.fit.vutbr.cz/>

²https://excel.fit.vutbr.cz/submissions/2024/039/39_poster.pdf

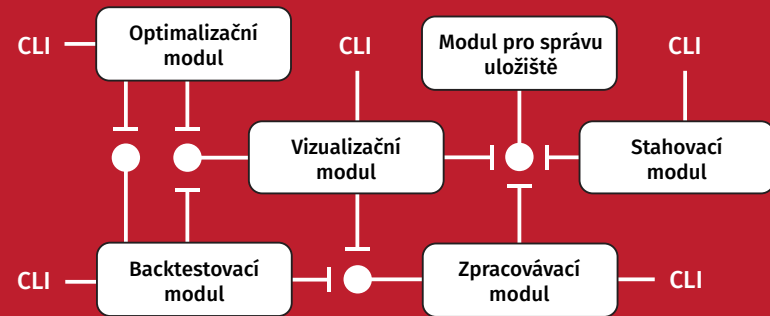
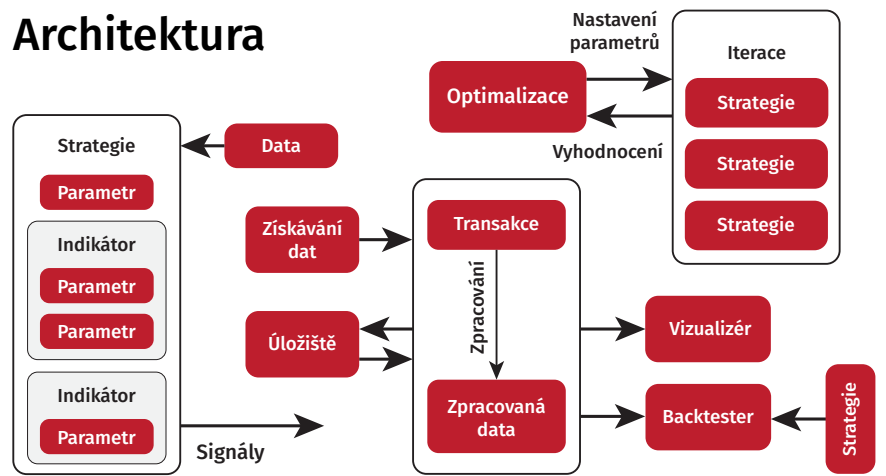
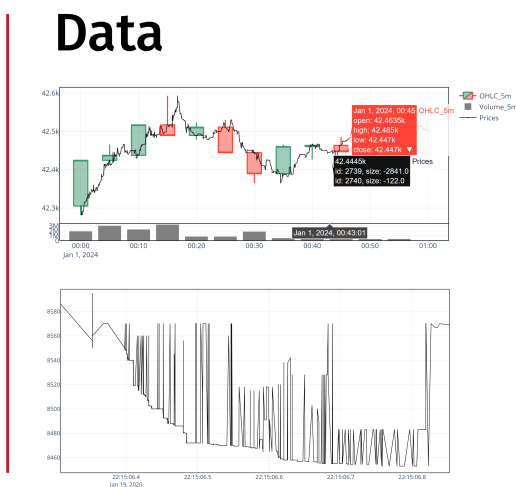
³<https://excel.fit.vutbr.cz/submissions/2024/039/39.pdf>

Trading Framework

Bc. Martin Kmenta | Ing. Igor Szóke, Ph.D.

Excel@FIT 2024

VYSOKÉ UČENÍ FAKULTA TECHNICKÉ INFORMATIČNÍCH V BRNĚ TECHNOLOGIÍ



Kompletní framework:

- Práce s tokovými daty na trhu (jednotlivými transakcemi)
- Backtestování strategií
- Optimalizace strategií s pomocí evolučních algoritmů

Strategie

Id: GPT-4 Strategie
časový_rámec: 15 min
stop_loss: 20 %
plán_poplatků: VIP3

Indikátory

sma20_nad_ema50_indikátor
typ: mas_porovnávač

ma1: sma_20, ma2: ema_50
operátor: zkřížené_nahorů

sma20_pod_ema50_indikátor
typ: mas_porovnávač

ma1: sma_20, ma2: ema_50
operátor: zkřížené_dolů

Braní pozic

Long

sma20_nad_ema50_indikátor
začátek_časového_rámce
žádná_aktivní_pozice

Short

sma20_pod_ema50_indikátor
začátek_časového_rámce
žádná_aktivní_pozice

Manipulace s pozicemi

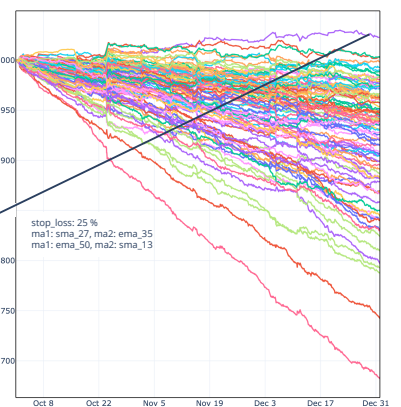
Rušení pozic

sma20_nad_ema50_indikátor
začátek_časového_rámce

sma20_pod_ema50_indikátor
začátek_časového_rámce



Optimalizace



Framework pro backtestování strategií algoritmického obchodování na burze včetně podpory pro vylepšování strategií s pomocí evolučních algoritmů.

Martin Kmenta*

Abstrakt

Tato diplomová práce se soustředí na **vývoj** pokročilého **frameworku pro backtestování algoritmických obchodních strategií**, přičemž klade důraz na **optimalizaci strategií pomocí evolučních algoritmů**. Zabývá se analýzou a aplikací technické analýzy v kontextu obchodování na burze. Dále se zaměřuje na návrh a vývoj modulů pro efektivní získávání, zpracování, vizualizaci a analýzu různých typů tržních dat, což umožňuje uživatelům **vytvářet a backtestovat** své **vlastní indikátory a obchodní strategie** s využitím robustního frameworku.

*xkment06@vut.cz, Faculty of Information Technology, Brno University of Technology

1. Úvod

Tato diplomová práce se soustředí na vývoj pokročilého frameworku pro backtestování algoritmických obchodních strategií a jejich optimalizace. Zabývá se analýzou a aplikací technické analýzy v kontextu obchodování na burze, přičemž se snaží detekovat co nejvíce faktorů ovlivňující výkonnost strategií. Práce obsahuje rozsáhlý průzkum existujících řešení za účelem zjištění konkurenceschopnosti frameworku, možné inspirace a případné integrace těchto řešení. Taktéž se snaží o efektivní optimalizace obchodních strategií převážně s pomocí evolučních algoritmů.

Dále se práce zaměřuje na samotný návrh a implementaci jednotlivých modulů pro efektivní získávání, zpracování, vizualizaci a analýzu různých typů tržních dat stejně jako o implementaci optimalizačního modulu se snahou vytvořit jednoduchý a snadno rozšiřitelný framework.

2. Algoritmické obchodování

Algoritmické obchodování, často označované jako automatizované obchodování, je charakterizováno využitím algoritmů a matematických modelů pro vytváření a realizaci obchodních příkazů na základě obchodních strategií. Tento přístup umožňuje rychlou a efektivní reakci na tržní změny, minimalizaci chyb

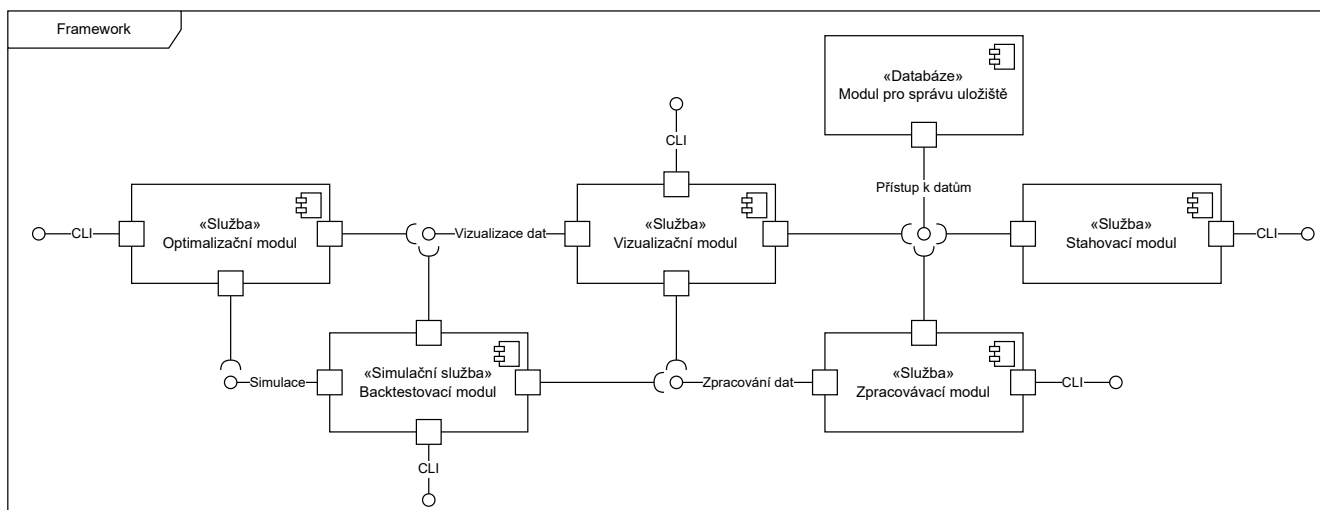
způsobených lidským faktorem a využití složitých strategií, které by pro člověka byly obtížně realizovatelné.

Backtestování je klíčový proces, při kterém se používají historická data k ověření efektivity a výkonnosti obchodních strategií. Je to zásadní součást algoritmického obchodování, protože umožňuje investorům simulovat jejich strategie a poskytuje důležité informace pro jejich optimalizaci a vylepšení. Tímto způsobem mohou investoři identifikovat potenciální slabiny ve svých obchodních strategiích a přizpůsobit je, aby více vyhovovaly jejich preferencím.

3. Motivace

Hlavním důvodem vzniku této práce je **neexistence** veřejně dostupného komplexního **přístupu k backtestování** algoritmických obchodních strategií **za použití nezakreslených dat**. Všechny prozkoumané dostupné nástroje zkrslují výsledky samotného backtestingu tím, že používají nejčastěji vstupní data ve formátu OHLCV¹. Tento přístup neumožňuje pracovat s jednotlivými transakcemi a ochuzuje analýzu o cenné informace, jako je například poměr objemu nákupu a prodeje, tím, že seskupuje jednotlivé transakce do

¹OHLCV (zkratka pro Open, High, Low, Close, Volume) představuje základní informace o cenové aktivitě za určité časové období.



Obrázek 1. Schéma modulů na nejvyšší úrovni

jedné informace. Sice umožňuje pracovat s informacemi seskupenými po jednotlivých vteřinách avšak například pro Scalp trading², kde transakce mohou trvat jednotky vteřin, je tento přístup příliš omezující.

Dalším důvodem je **optimalizace strategií**. Dostupné frameworky a platformy sice nabízí různé optimalizace parametrů ať už jednoduchým prohledáváním všech permutací, nebo sofistikovanějšími metodami strojového učení a biologicky inspirovanými algoritmy, ale tyto řešení mají vždy nemalé limitace kvůli omezením samotných nástrojů.

Cílem diplomové práce je vyvinutí pokročilého frameworku, který eliminuje zmíněné omezení a bude konkurenceschopný s existujícími řešeními ať už z pohledu funkcionality, uživatelské přívětivosti nebo rozšiřitelnosti.

4. Dosažené výsledky

Podářilo se implementovat framework, který splňuje veškeré stanovené požadavky. Přesněji bylo implementováno šest spolu spolupracujících modulů, kde každý se stará o jednu z těchto funkcionalit: **získávání dat**, **zpracování dat**, **správu úložiště**, **vizualizaci**, **backtestování strategií** a **optimalizaci parametrů strategie**. Moduly a jejich vzájemnou komunikaci je možné vidět na obrázku 1. Celý framework je možné ovládat pomocí příkazové řádky nebo jej přímo integrovat do python projektu.

Backtestování probíhá formou simulace reálného prostředí, do kterého je zahrnuta směnárna a obchodníci.

²Scalp trading je obchodní strategie, která využívá krátkodobého vstupu a výstupu z pozic k zisku z malých cenových pohybů na trhu.

Během simulace jsou vyhodnocovány historické transakce jedna po druhé, což mnohonásobně zvyšuje přesnost backtestingu vůči všem veřejně dostupným nástrojům. Každý obchodník žádá směnárnu o vytvoření pozic³ na základě vyhodnocení dostupných dat. Také žádá směnárnu o další manipulace s vytvořenými pozicemi, jako je například jejich předčasné zrušení. Virtuální směnárna přijímá požadavky od obchodníků a vytváří, nebo manipuluje s pozicemi, které v průběhu času také uzavírá podle jejich parametrů.

Kvůli časové náročnosti simulace a nemožnosti projít množinu všech možných permutací parametrů komplexnějších strategií v reálném čase, byl vytvořen modul pro učení parametrů s pomocí genetického algoritmu. Framework byl testován a ověřován. Při experimentování se podařilo optimalizovat parametry několika strategií. Efektivnost optimalizace závisela na účinnosti poskytnuté strategie, rozsahu dat, počtu iterací a nastavení algoritmu. Také byly navrženy další zajímavé možnosti pro následný vývoj frameworku.

Poděkování

Rád bych poděkoval panu Ing. Igorovi Szóke, Ph.D. za vedení a odbornou pomoc v průběhu řešení této diplomové práce.

³Pozic je myšleno spekulace na pohyb cen aktiva, bez nutnosti jeho vlastnictví. V obchodování CFD (smlouvy na rozdíl) je tento koncept validně realizovatelný.