



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA STROJNÍHO INŽENÝRSTVÍ**

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV AUTOMATIZACE A INFORMATIKY**

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**Využití umělé inteligence pro automatizaci  
obchodování na burze**

THE USE OF ARTIFICIAL INTELLIGENCE TO AUTOMATE TRADING ON THE EXCHANGE

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. František Čermák**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**prof. Ing. Radomil Matoušek, Ph.D.**

**BRNO 2024**



# Zadání diplomové práce

Ústav:	Ústav automatizace a informatiky
Student:	<b>Bc. František Čermák</b>
Studijní program:	Aplikovaná informatika a řízení
Studijní obor:	bez specializace
Vedoucí práce:	<b>prof. Ing. Radomil Matoušek, Ph.D.</b>
Akademický rok:	2023/24

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

## Využití umělé inteligence pro automatizaci obchodování na burze

### Stručná charakteristika problematiky úkolu:

Cílem bude prozkoumat dostupné možnosti automatizovaných obchodovacích strategií a následně navrhnout, implementovat, otestovat a porovnat vlastní řešení.

Bude využita informace dat pohybu cen mezi jednotlivými aktivy a indikátory trhu. Bude prozkoumáno možné využití strategií: Deep learning, Adaptive reinforcement learning aj. Programovací jazyk: Python. Prezentace se předpokládá na burzovních demo účtech, ke kterým se bot připojí pomocí API klíče.

### Cíle diplomové práce:

Rešerše možností a dostupných existujících řešení.  
Návrh vlastního automatizovaného obchodovacího systému.  
Návrh vlastní obchodovací strategie s využitím umělé inteligence.  
Implementace navrženého řešení.  
Otestování systému a vyhodnocení výsledků.

### Seznam doporučené literatury:

HUANG, Boming, et al. "Automated trading systems statistical and machine learning methods and hardware implementation: a survey." Enterprise Information Systems 13.1 (2019): 132-144.

BIGIOTTI, Alessandro, and Alfredo NAVARRA. "Optimizing automated trading systems." Digital Science. Springer International Publishing, 2019.

DEMPSTER, Michael AH, and Vasco LEEMANS. "An automated FX trading system using adaptive reinforcement learning." Expert systems with applications 30.3 (2006): 543-552.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2023/24

V Brně, dne

L. S.

---

doc. Ing. Zdeněk Hadaš, Ph.D.  
ředitel ústavu

---

doc. Ing. Jiří Hlinka, Ph.D.  
děkan fakulty

## **ABSTRAKT**

Tahle diplomová práce se zabývá využitím umělé inteligence pro automatizaci obchodování na burze. Hlavním cílem bylo prozkoumat současné technologie aplikované v algoritmickém obchodování a následně navrhnout a vyvinout automatizovaný obchodní systém využívající umělou inteligenci. Práce se zaměřuje na různé aspekty algoritmického obchodování, včetně vysokofrekvenčního obchodování, cloudových řešení, strojového učení, blockchainu a smart contracts. Dále zkoumá aplikace umělé inteligence v obchodování, jako je prediktivní analytika a zpracování přirozeného jazyka, a diskutuje etické a regulační výzvy spojené s touto technologií. Návrh a vývoj automatizovaného obchodního systému je popsán detailně, včetně architektury systému, volby programovacích jazyků a nástrojů, a implementace obchodních algoritmů. Výsledky ukazují, že využití umělé inteligence může výrazně zvýšit efektivitu a přesnost obchodování na burze, avšak je třeba vzít v úvahu technologická a etická rizika. Tato práce přináší významný příspěvek k výzkumu v oblasti algoritmického obchodování a poskytuje základy pro další výzkum v optimalizaci obchodních algoritmů a integraci nových technologií.

## **ABSTRACT**

This thesis deals with the use of artificial intelligence for automating stock trading. The main objective was to investigate current technologies applied in algorithmic trading and then to design and develop an automated trading system using artificial intelligence. The work focuses on various aspects of algorithmic trading, including high frequency trading, cloud solutions, machine learning, blockchain and smart contracts. It also explores the applications of AI in trading, such as predictive analytics and natural language processing, and discusses the ethical and regulatory challenges associated with this technology. The design and development of an automated trading system is described in detail, including system architecture, choice of programming languages and tools, and implementation of trading algorithms. The results show that the use of artificial intelligence can significantly increase the efficiency and accuracy of stock trading, but technological and ethical risks must be considered. This thesis makes a significant contribution to research in the field of algorithmic trading and provides a foundation for further research in optimizing trading algorithms and integrating new technologies.

## **KLÍČOVÁ SLOVA**

algoritmické obchodování, umělá inteligence, automatizované obchodní systémy, prediktivní analytika, vysokofrekvenční obchodování, strojové učení, blockchain, cloudová řešení, zpracování přirozeného jazyka, obchodní strategie

## **KEYWORDS**

algorithmic trading, artificial intelligence, automated trading systems, predictive analytics, high-frequency trading, machine learning, blockchain, cloud solutions, natural language processing, trading strategies



2024

## **BIBLIOGRAFICKÁ CITACE**

ČERMÁK, František. *Využití umělé inteligence pro automatizaci obchodování na burze*. Brno, 2024. Dostupné také z: <https://www.vut.cz/studenti/zav-prace/detail/161714>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky. Vedoucí práce Radomil Matoušek.



## **PODĚKOVÁNÍ**

Přeji si v první řadě poděkovat svému otci, který mi byl po celou dobu psaní téhle práce na blízku a podporoval mě. Dále patří poděkování mým nejbližším přátelům, Ondrovi, Patrikovi, Lukášovi, Kubovi, Markovi a zejména Dominikovi, kteří tam byli vždy, i v ty nejkritičtější okamžiky. Na závěr si přeji poděkovat svému vedoucímu práce, který vždy povzbudil a usměrnil v práci.



## ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že, že tato práce je mým původním dílem, vypracoval jsem ji samostatně pod vedením vedoucího práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury.

Jako autor uvedené práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následku porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestně právních důsledků.

V Brně dne 24. 5. 2024

.....

František Čermák



# OBSAH

<b>1</b>	<b>ÚVOD.....</b>	<b>15</b>
<b>2</b>	<b>SOUČASNÉ TECHNOLOGIE PRO ALGORITMICKÉ OBCHODOVÁNÍ NA BURZE.....</b>	<b>17</b>
2.1	Historie a vývoj algoritmického obchodování.....	17
2.2	Technologie a infrastruktura v algoritmickém obchodování.....	18
2.3	Strategie a modely Algoritmického obchodování .....	19
2.4	Výzvy a rizika algoritmického obchodování.....	19
	Technologická rizika.....	20
2.5	Budoucí trendy a příležitosti v algoritmickém obchodování.....	20
	Umělá inteligence a strojové učení .....	21
<b>3</b>	<b>APLIKACE UMĚLÉ INTELIGENCE PŘI OBCHODOVÁNÍ NA BURZE</b>	<b>22</b>
3.1	Úvod do umělé inteligence v obchodování na burze.....	22
3.2	Prediktivní analytika v obchodování na burze .....	22
3.3	Automatizované obchodní systémy .....	23
3.4	Zpracování přirozeného jazyka .....	24
3.5	Zlepšená správa portfolia a snížení rizik .....	24
3.6	Etické a regulační výzvy.....	25
3.7	Budoucí trendy .....	25
<b>4</b>	<b>NÁVRH A VÝVOJ AUTOMATIZOVANÉHO OBCHODNÍHO SYSTÉMU</b>	<b>27</b>
4.1	Architektura systému .....	27
4.1.1	API Binance.....	27
4.1.2	Python jako primární programovací jazyk pro vývoj algoritmu. ....	28
	• Python v Algoritmickém Obchodování:.....	28
	• Python a Umělá Inteligence:.....	28
4.1.3	AI a fundamentální analýza kryptotrhu .....	29
4.1.4	AI a technická analýza trhu .....	30
4.1.5	Shrnutí architektury automatizovaného obchodního systému.....	31
4.2	Praktický vývoj automatizovaného systému .....	31
4.2.1	Vyhodnocování fundamentálních zpráv .....	31
4.2.2	Výsledky vyhodnocování fundamentálních zpráv .....	57
4.2.3	Automatizované obchodování na burze na základě technické analýzy.....	59
4.2.4	Výsledky Automatizovaného obchodování na burze na základě technické analýzy	74
<b>5</b>	<b>ZÁVĚR .....</b>	<b>78</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>79</b>



# 1 ÚVOD

Algoritmické obchodování, které kombinuje finanční trhy a pokročilé technologie, se stalo nedílnou součástí moderního finančního světa. S rychlým vývojem výpočetní techniky a umělé inteligence (AI) získávají algoritmické obchodní systémy na popularitě díky své schopnosti analyzovat velké objemy dat a provádět obchodní rozhodnutí během zlomků sekundy. Tato práce se zaměřuje na využití AI pro automatizaci obchodování na burze, konkrétně na vývoj automatizovaného obchodního systému využívajícího pokročilé techniky umělé inteligence.

Cílem této diplomové práce je prozkoumat aktuální technologie a strategie používané v algoritmickém obchodování a následně navrhnout a implementovat obchodní systém, který využívá AI k optimalizaci obchodních rozhodnutí a zvýšení ziskovosti. Práce je rozdělena do několika částí, které postupně analyzují současné technologie, aplikace AI v obchodování a detaily návrhu a vývoje automatizovaného systému.

První kapitola poskytuje úvod do algoritmického obchodování, popisuje jeho historii a vývoj, a analyzuje klíčové technologie a infrastrukturu, jako je vysokofrekvenční obchodování (HFT), cloudová řešení, strojové učení, blockchain a smart contracts. Dále jsou diskutovány různé obchodní strategie a modely, stejně jako výzvy a rizika spojená s algoritmickým obchodováním.

Druhá kapitola se zabývá aplikacemi umělé inteligence v obchodování na burze. Zaměřuje se na prediktivní analytiku, automatizované obchodní systémy a zpracování přirozeného jazyka. Tato část také diskutuje etické a regulační výzvy spojené s využitím AI v obchodování.

Třetí kapitola se soustředí na návrh a vývoj automatizovaného obchodního systému. Popisuje architekturu systému, volbu technologií a programovacích nástrojů, a detaily implementace algoritmů. Důležitou součástí je také testování a hodnocení výkonnosti systému.

Výsledky této práce ukazují, že AI může výrazně zlepšit efektivitu a přesnost obchodování na burze. Automatizovaný systém vyvinutý v rámci této práce dosáhl pozitivních výsledků ve zhodnocení portfolia, což potvrzuje potenciál AI v této oblasti. Přesto je důležité zvážit i technologická a etická rizika spojená s nasazením těchto technologií.

Tahle práce přináší významný příspěvek k výzkumu a vývoji v oblasti algoritmického obchodování a poskytuje základy pro další výzkum, zaměřený na optimalizaci obchodních algoritmů a integraci nových technologií, jako je kvantové počítání.



## 2 SOUČASNÉ TECHNOLOGIE PRO ALGORITMICKÉ OBCHODOVÁNÍ NA BURZE

Současné technologie pro algoritmické obchodování se vyvíjejí ve velmi dynamickém prostředí, kde klíčovou roli hrají inovace a digitalizace. Významným trendem v tomto sektoru je stále rostoucí adopce cloudových řešení [1], která umožňují obchodníkům automatizovat obchodní procesy, udržovat obchodní data efektivně a optimalizovat náklady. Cloudové platformy nabízejí lepší škálovatelnost a snadnější správu dat. To je zvláště užitečné pro testování nových obchodních strategií a provádění časových analýz a zpětných testů.

Kromě toho, vývoj v algoritmickém obchodování značně zvyšuje účinnost transakcí na trzích, jako jsou akcie, devizy, dluhopisy a kryptoměny. Například na akciových trzích, které představují jednu z největších oblastí pro algoritmické obchodní strategie, se využívá různorodých přístupů k maximalizaci zisku a řízení rizik. [2]

Jedním z hlavních hráčů na trhu algoritmického obchodování jsou institucionální investoři, jako jsou banky, pojišťovny a investiční fondy, kteří používají sofistikované algoritmické strategie pro zlepšení efektivity, optimalizaci provádění transakcí a řízení rizik [2]. Tyto strategie často zahrnují vysokofrekvenční obchodování (HFT), které vyžaduje extrémně rychlé a efektivní algoritmické systémy schopné reagovat na tržní data rychleji než lidé. [1]

Globálně je severoamerický trh s algoritmickým obchodováním jedním z největších a nejdynamičtějších [1]. To je dáno přítomností mnoha klíčových hráčů v tomto regionu a rozsáhlými investicemi do obchodních technologií.

Z těchto důvodů je zřejmé, že trh s algoritmickým obchodováním se bude i nadále dynamicky rozvíjet, a to zejména díky pokračující digitalizaci, inovacím v oblasti umělé inteligence a strojového učení, a rostoucímu využívání cloudových řešení, které nabízejí nové možnosti pro obchodování a řízení dat. [4]

### 2.1 Historie a vývoj algoritmického obchodování

Algoritmické obchodování má hluboké kořeny v technologických inovacích a regulatorních změnách, které otevřely dveře efektivnějším a rychlejším tržním transakcím. Jeho historie sahá až do 70. let 20. století, kdy se začaly používat počítače k výpočtu objednávek na burze. V 80. letech s elektronizací NASDAQ a zavedením digitálního obchodního systému začalo dávat smysl, že technologie může přinést výrazný posun v obchodování. To umožnilo rozšíření složitých obchodních algoritmů.

V 90. letech byl růst algoritmického obchodování výrazně podpořen klesajícími náklady na výpočetní techniku a stále větší dostupností internetu [5]. Díky internetu se obchodování stalo rychlejším a dostupnějším. Na začátku 21. století se algoritmické obchodování dále transformovalo s nástupem vysokofrekvenčního obchodování (HFT), a to přineslo ještě větší objem obchodů a efektivnější cenové rozpětí. [3]

S přijetím amerického zákona Sarbanes-Oxley Act v roce 2002 se zvýšila transparentnost finančních informací, tím byly posíleny algoritmické strategie. V posledním desetiletí technologie jako strojové učení a umělá inteligence přinesly novou úroveň sofistikace v obchodování, tím byla zapříčiněna rychlejší analýza tržních dat a komplexnější modelování. [2]

Cloud computing a využívání distribuovaných sítí umožnily firmám dosáhnout ještě vyšší efektivity a snížit náklady na datové transakce [1]. Tento trend byl obzvláště patrný během pandemie COVID-19, kdy digitální transformace a stoupající počet online obchodů umožnily obchodníkům provádět operace přes digitální platformy. [3]

## 2.2 Technologie a infrastruktura v algoritmickém obchodování

### Vysokofrekvenční obchodování (HFT)

Vysokofrekvenční obchodování zahrnuje použití algoritmů, které provádějí obchody v řádu milisekund. Důležitou součástí infrastruktury pro HFT jsou vysoce výkonné počítačové systémy a připojení s extrémně nízkou latencí, čímž je možno rychle reagovat na tržní změny. Přístup k nejnovějším tržním datům a schopnost rychle reagovat na tyto informace jsou zásadní pro úspěch v HFT. [5] [6]

### Cloudové řešení

Cloudové platformy hrají stále větší roli v algoritmickém obchodování díky své schopnosti poskytovat škálovatelnost, snížení nákladů a vysokou dostupnost dat a výpočetního výkonu [7]. Cloudové služby umožňují obchodníkům flexibilně testovat a provádět obchodní strategie bez potřeby velkých počátečních investic do hardwaru a infrastruktury. [1]

### Strojové učení a umělá inteligence

AI a strojové učení se stávají stěžejními technologiemi v algoritmickém obchodování pro rozvoj prediktivních modelů a automatizaci rozhodovacích procesů [4]. Tyto technologie umožňují analyzovat obrovské objemy dat v reálném čase a přizpůsobovat obchodní strategie aktuálním tržním podmínkám. Investice do AI a strojového učení nabízejí obchodníkům příležitost zvýšit přesnost a efektivitu jejich algoritmických strategií. [7]

### Blockchain a smart contracts

Blockchainové technologie a chytré kontrakty přinášejí do oblasti algoritmického obchodování nové možnosti pro zvýšení transparentnosti a bezpečnosti. Tyto technologie nabízejí decentralizované řešení, které může redukovat potřebu tradičních finančních prostředníků a zároveň zlepšit efektivitu a snížit náklady na transakce. [7]

### Automatizované obchodní systémy (ATS)

ATS umožňují implementaci složitých obchodních strategií s minimálním lidským zásahem. Platformy jako MetaTrader a další poskytují nástroje pro vývoj, testování a provádění algoritmů, které mohou operovat na různých finančních trzích a v různých obchodních podmínkách. [5]

## 2.3 Strategie a modely Algoritmického obchodování

V algoritmickém obchodování existuje široká škála strategií, které obchodníci používají pro maximalizaci zisku a minimalizaci rizik. Zde jsou některé z nejčastějších a nejúspěšnějších modelů a strategií:

### Hedging a Diverzifikace

- **Hedging:** Technika snižování rizika, při které se obchodník snaží chránit před možnými ztrátami vytvořením kompenzační pozice. Často se k tomu využívají opce nebo futures kontrakty pro zajištění proti kolísání cen. [8]
- **Diverzifikace:** Strategie, která spočívá v rozložení investic do různých sektorů, aby se snížila zranitelnost vůči jednotlivým trhům nebo strategiím.

### Jednoduché Strategie

- **Klouzavý průměr:** Sleduje průměrnou cenu v určitých obdobích a používá se k určení trendů a jejich směru. Klasickým příkladem je křížení 50denního a 200denního klouzavého průměru, které dává signály pro nákup či prodej. [9]
- **Vážený průměr:** Strategie, která sleduje časově nebo objemově vážený průměr a obchoduje kolem těchto průměrů. Pomáhá vyhnout se extrémním cenám v nestálých trzích. [10]

### Pokročilé Strategie

- **Statistická arbitráž:** Tato metoda využívá cenových nesrovnalostí mezi různými podobnými aktivy. Algoritmy identifikují akcie nebo skupiny akcií s vysokou korelací a profitují z rozdílů v jejich cenách. [10]
- **Momentum trading:** Zaměřuje se na trend a sílu cenových pohybů. Algoritmy hledají aktiva s významnými pohyby jedním směrem a podle toho se rozhodují pro dlouhé nebo krátké pozice. [9]

### Zajištění Rizik

- **Stop-loss:** Nastavení limitů pro ukončení obchodů, aby se předešlo nadměrným ztrátám v případě nečekaného pohybu na trhu.
- **Optimalizace signálů:** Neustálé vyhodnocování a vylepšování obchodních signálů může vést k přesnějším a ziskovějším obchodům. [11]

Každý obchodník může najít nejlepší strategii pro své potřeby testováním, úpravou signálů a neustálým monitorováním tržních podmínek. Strategie by měla odpovídat typu aktiv a časovému horizontu, který si obchodník zvolí. [12]

## 2.4 Výzvy a rizika algoritmického obchodování

Algoritmické obchodování přináší významné výhody, ale je také spojeno s řadou rizik a výzev, které musí obchodníci a investoři zohlednit.

### **Technologická rizika**

Algoritmické obchodování závisí na složitých technologických infrastrukturách, včetně obchodních platforem a připojení k datům. Selhání hardwaru nebo softwaru, problémy s připojením nebo chyby v datech mohou vést k významným finančním ztrátám [13]. Je nezbytné mít spolehlivé záložní systémy a nepřetržitý monitoring, aby se minimalizovala tato rizika.

### **Rizika spojená s exekucí**

I přes pokročilé algoritmy může dojít k odchýlkám od zamýšlených obchodních cen, zejména v důsledku nestability trhu nebo změn v likviditě. Tyto faktory mohou způsobit skluz (slippage), při kterém jsou obchody realizovány za odlišné ceny, než bylo zamýšleno. [13]

### **Rizika tržních podmínek**

Finanční trhy jsou vysoce nestálé a nepředvídatelné. To může vést k rychlým změnám, které znehodnotí dříve ziskové strategie. Algoritmy musí být dostatečně adaptabilní, aby se mohly rychle přizpůsobit měnícím se podmínkám.

### **Rizika modelů**

Výkonnost algoritmů závisí na jejich podkladových modelech a předpokladech. Chyby v modelech, chyby ve vyhodnocení parametrů nebo zkreslení mohou vést k nesprávným obchodním rozhodnutím. Je nezbytné provádět důkladné zpětné testování a průběžné hodnocení výkonnosti, aby se identifikovala a řešila rizika modelů.

### **Regulační rizika**

Algoritmické obchodování je předmětem regulačního dohledu a musí splňovat požadavky na správné řízení rizik a transparentnost. Nedodržení regulatorních požadavků může mít za následek právní důsledky a poškození reputace. [13]

### **Etické výzvy**

Využívání algoritmického obchodování přináší otázky týkající se férovosti, transparentnosti a soukromí dat. Je klíčové, aby byly algoritmy navrženy a provozovány způsobem, který podporuje integritu trhu a chrání osobní údaje uživatelů. [4]

Tyto výzvy vyžadují neustálou pozornost a proaktivní přístup k řízení rizik, aby se maximalizovaly přínosy algoritmického obchodování a minimalizovaly potenciální negativní dopady.

## **2.5 Budoucí trendy a příležitosti v algoritmickém obchodování**

Algoritmické obchodování čeká zajímavá budoucnost plná výzev i příležitostí, které jsou definovány pokrokem v oblastech umělé inteligence (AI), automatizace, velkých dat a regulací.

### **Umělá inteligence a strojové učení**

AI a strojové učení se stále více stávají základem pro vývoj pokročilých obchodních algoritmů. Tyto technologie umožňují analýzu rozsáhlých datových sad a identifikaci obchodních vzorů a trendů [14], což přináší zvýšenou efektivitu a ziskovost algoritmického obchodování. Využití prediktivní analytiky pomáhá rozhodování a plánování strategií na základě předpovědí tržních trendů. [15]

### **Velké množství Dat a analytika**

Exponenciální růst dat ve finančních trzích znamená, že algoritmy, které efektivně zpracovávají a analyzují velké objemy dat, mohou odhalit nové příležitosti pro obchodování. Tato schopnost je klíčová pro identifikaci trendů a sentimentu na trhu. [15]

### **Blockchain a decentralizované financování (DeFi)**

Technologie blockchain a aplikace v DeFi nabízejí nové možnosti pro zabezpečení, transparentnost a efektivitu obchodních operací [15]. AI hraje klíčovou roli ve vývoji chytrých kontraktů a decentralizovaných obchodních platform. Tím se stávají finanční služby stále demokratičtějšími. [16]

### **Automatizace a integrace**

Automatizace rutinních úkolů a integrace různých obchodních systémů a nástrojů zvyšují efektivitu a snižují náklady. Robotic Process Automation (RPA) se stává běžnou součástí algoritmických obchodních operací. Použití RPA umožňuje obchodníkům soustředit se na strategičtější aspekty obchodování. [17]

### **Regulace a etické otázky**

Rozvíjející se regulativní rámec klade důraz na transparentnost a prevenci tržního zneužívání. Algoritmické obchodování musí respektovat stále se měnící regulace a zároveň čelit etickým výzvám, jako je zajištění spravedlivého přístupu a zabránění manipulativním praktikám. [15] [17]

### **Globální expanze a kryptoobchodní platformy**

Algoritmické obchodování se rozšiřuje na globální úrovni, kde jsou zahrnuty různorodé trhy a regulativní prostředí. V oblasti kryptoměn je očekáván pokračující růst a inovace, které přinášejí nové strategie a způsoby řízení obchodních rizik. [18]

## 3 APLIKACE UMĚLÉ INTELIGENCE PŘI OBCHODOVÁNÍ NA BURZE

### 3.1 Úvod do umělé inteligence v obchodování na burze

Umělá inteligence (AI) má významný dopad na obchodování na burze tím, že automatizuje a vylepšuje tradiční obchodní strategie pomocí prediktivní analytiky, zpracování dat a sofistikovaných algoritmů. Umožňuje rychlejší a přesnější obchodní rozhodnutí, poskytuje hluboký pohled na tržní trendy a predikce a optimalizuje správu portfolia.

- **Prediktivní Analytika:** AI modely využívají historická data a analyzují vzory k přesnější predikci pohybů na trhu. Tímto způsobem poskytují obchodníkům konkurenční výhodu, která byla dříve nedostupná. Algoritmy také zvládají analyzovat a zpracovávat rozsáhlé datové sady, což je klíčové při určování tržního sentimentu a odhalování příležitostí.
- **Automatizované obchodní systémy:** Obchodní algoritmy poháněné AI dovedou automatizovat obchodní rozhodnutí, což umožňuje vysokou frekvenci a přesnost transakcí. High-frequency trading (HFT) je příkladem, kdy systémy analyzují trh a provádějí obchody v mikrosekundách.
- **Zlepšená správa portfolia:** AI modely pomáhají při řízení rizik a optimalizaci portfolia identifikací příležitostí na základě individuálního rizikového profilu a investičních cílů. Díky strojovému učení se strategie neustále upravují na základě změn trhu. [19]

Pokrok v AI tak umožňuje obchodníkům získávat přehled o tržních trendech a poskytuje vyšší úroveň efektivity a flexibility při provádění obchodů.

### 3.2 Prediktivní analytika v obchodování na burze

#### Úvod do prediktivní analytiky s AI

Prediktivní analytika využívá umělou inteligenci (AI) k analýze velkého množství dat, odhalování skrytých vzorců a předpovídání tržních trendů s vysokou přesností. Tato technologie přeměňuje tradiční obchodní strategie tím, že nabízí hlubší vhled do tržních pohybů a umožňuje obchodníkům reagovat dynamicky na měnící se podmínky.

#### Techniky a modely

V prediktivní analýze se široce využívají různé AI modely, jako jsou neuronové sítě a strojové učení, které umožňují nejen identifikaci a využití tržních vzorců, ale také kvantifikaci a řízení rizik [20]. Modely jsou trénovány na historických datech a optimalizovány pro maximální výkon při predikci budoucích cen akcií a tržních pohybů. [21]

### **Aplikace v reálném světě**

AI je také nasazována pro sentimentální analýzu, což zahrnuje sběr a analýzu dat z novinových článků a sociálních médií k určení tržního sentimentu. Tato technika umožňuje obchodníkům pochopit, jak veřejné vnímání ovlivňuje tržní ceny, a přizpůsobit své obchodní strategie těmto vnímáním. [22]

### **Výzvy a omezení**

Přestože prediktivní analytika nabízí mnoho výhod, stále čelí výzvám jako je přesnost modelů, potenciál pro nadměrné přizpůsobení (overfitting) a závislost na kvalitě a dostupnosti dat [21]. Modely vyžadují neustálé aktualizace a úpravy, aby reflektovaly nejnovější tržní podmínky a zůstaly relevantní.

### **Závěr**

Prediktivní analytika s použitím AI mění obchodování na burze tím, že nabízí sofistikované nástroje pro hlubší pochopení trhu a efektivnější obchodní rozhodování. S rostoucí integrací AI se očekává, že její role v obchodování bude nadále růst, což umožní obchodníkům lépe navigovat v komplexním a dynamickém tržním prostředí.

## **3.3 Automatizované obchodní systémy**

### **Analýza dat a identifikace obchodních příležitostí s pomocí AI**

Automatizované obchodní systémy s AI využívají sofistikované algoritmy k analýze velkého množství finančních dat a identifikaci obchodních příležitostí. Tyto systémy nejen že zkoumají historická data, ale také provádějí sentimentální analýzu zpráv a sociálních médií, aby předpovídaly tržní pohyby [22]. Použitím metod jako je zpracování přirozeného jazyka, systémy efektivně skenují a interpretují subjektivní informace, poskytující uživatelům cenné náhledy, které mohou ovlivnit tržní rozhodnutí. [23]

### **Schopnost provádět obchody s vysokou rychlostí a přesností**

AI v obchodních systémech umožňuje provádění obchodů s bleskovou rychlostí a vysokou úrovní přesnosti, často v rámci mikrosekund, což je zásadní v prostředích jako je high-frequency trading (HFT). Tyto systémy neustále monitorují trh, aby optimalizovaly obchodní strategie a adaptovaly se na měnící se tržní podmínky nebo neefektivitu trhu. Díky schopnosti rychle reagovat na tržní signály a automaticky provádět obchody, AI obchodní systémy minimalizují riziko lidských chyb a zvyšují šance na ziskové obchody. [24]

Tahle kombinace pokročilé analýzy a rychlého provedení obchodu činí AI neocenitelným nástrojem pro moderní obchodníky a investory, kteří se snaží využít každou tržní příležitost.

### 3.4 Zpracování přirozeného jazyka

#### **Analýza zpravodajství a sociálních médií**

NLP technologie hraje klíčovou roli v analýze zpravodajství a sociálních médií, kde umožňuje identifikaci sentimentu trhu. Tato technologie sbírá a analyzuje textová data, jako jsou novinové články a příspěvky na sociálních sítích, aby odhalila, jaké emoce a nálady panují mezi veřejností vzhledem k určitým tématům nebo akciím [25]. Pomocí NLP lze sledovat, jak se veřejné vnímání mění v reálném čase, což poskytuje obchodníkům cenné náhledy do pohybu trhu. [26]

#### **Vliv sentimentu na obchodní rozhodování**

Schopnost rychle analyzovat sentiment z médií umožňuje obchodníkům adaptovat své strategie podle aktuálního vnímání trhu. Například, pokud NLP nástroje identifikují pozitivní sentiment ve zprávách týkajících se určité akcie, mohou obchodníci tohle vnímání využít k rychlým nákupům této akcie před tím, než cena vzroste [27]. Naopak, negativní sentiment může obchodníky varovat, aby se dané akcie vyvarovali nebo ji prodali, čímž minimalizují ztráty. [21]

Tato dynamika ukazuje, jak může moderní technologie NLP převádět rozsáhlé a různorodé datové sady na stručné, akční informace, které pomáhají obchodníkům udržet krok s rychle se měnícím trhem.

### 3.5 Zlepšená správa portfolia a snížení rizik

#### **Optimalizace Správy Portfolia pomocí AI**

Umělá inteligence (AI) hraje stále významnější roli v optimalizaci správy investičních portfolií. S využitím pokročilých algoritmů a strojového učení, AI dokáže analyzovat velké množství finančních dat, identifikovat optimální kombinaci investičních aktiv a přizpůsobovat investiční strategie individuálním cílům a toleranci k riziku investora. Tato technologie umožňuje investorům dosáhnout vyšší míry diverzifikace a efektivity při minimálním zásahu člověka a to vede k lepšímu výkonu portfolia. [28] [29]

#### **Vylepšení Analýzy Rizik**

AI rovněž revolucionalizuje způsoby, jakými jsou rizika hodnocena a řízena. S pomocí algoritmů může AI provádět složité analýzy rizik v reálném čase, simulovat extrémně nepříznivé tržní podmínky a predikovat potenciální dopady na portfolia. Tato schopnost umožňuje rychle reagovat na tržní změny a přizpůsobovat strategie tak, aby byla rizika efektivně minimalizována [28]. Díky AI mohou být portfolio manažeři proaktivní místo reaktivní, mohou lépe chránit investice v nestabilních tržních podmínkách.

AI a její integrace do správy portfolia přináší revoluční změny ve způsobu, jakým jsou investice řízeny. Díky přesnějším predikcím a pokročilému řízení rizik nabízí investorům

možnost dosahovat lepších výsledků, zatímco zároveň snižuje riziko a zvyšuje transparentnost a kontrolu nad jejich investičními rozhodnutími.

### 3.6 Etické a regulační výzvy

#### **Etické Výzvy: Ochrana Osobních Údajů a Prevence Tržní Manipulace**

V oblasti algoritmického obchodování přináší umělá inteligence (AI) nejen efektivitu, ale také etické výzvy, zejména ve směru ochrany osobních údajů a prevence tržní manipulace. Využívání AI pro analýzu rozsáhlých datových sad může vést k otázkám o spravedlivosti a transparentnosti výsledků [30], tak se děje že vznikají obavy z potenciálního zneužití těchto technologií pro manipulativní účely. Zajištění, že systémy AI respektují soukromí [31] a nejsou využívány k narušování tržní rovnováhy, je klíčové pro udržení důvěry ve finanční trhy.

#### **Regulační Výzvy a Dodržování Pravidel**

AI také nabízí nástroje, které mohou pomoci finančním institucím lépe dodržovat regulace a předcházet podvodům. Systémy mohou automatizovaně monitorovat obchodní aktivity a identifikovat potenciální rizika nebo nekalé praktiky, což pomáhá ve snaze minimalizovat finanční podvody a zneužití [32]. Využití AI pro stresové testy a simulace tržních scénářů umožňuje institucím lepší předpovědi a řízení rizik, což zvyšuje celkovou stabilitu finančního systému. [33]

Výzvy spojené s integrací AI do finančních procesů vyžadují neustálé hodnocení a aktualizaci regulačních rámců, aby se zajistilo, že technologický pokrok nepředbíhá etické normy a zákonné požadavky. Tato snaha o rovnováhu mezi inovacemi a regulačními požadavky je klíčová pro budoucí vývoj etického a udržitelného využívání AI ve finančním sektoru.

### 3.7 Budoucí trendy

#### **Vyhlídky pro AI v obchodování na burze**

V obchodování na burze se očekává, že umělá inteligence (AI) bude hrát stále důležitější roli, když se stane integrální součástí všech aspektů finančních trhů, od obchodních platforem po investiční poradenské služby. AI bude pokračovat ve zdokonalování svých prediktivních schopností [34]. Tím budou umožněny přesnější předpovědi tržních trendů a cen akcií. Tato schopnost přinese obchodníkům a finančním institucím významnou konkurenční výhodu, protože budou moci efektivněji reagovat na dynamicky se měnící tržní podmínky.

#### **Očekávané inovace a trendy**

Mezi očekávané inovace v AI obchodování patří rozvoj pokročilých algoritmických obchodních systémů, které provádějí obchody s vysokou frekvencí a přesností. Tyto systémy budou schopné analyzovat rozsáhlá množství dat v reálném čase, a tak se budou

zlepšovat schopnosti predikovat tržní pohyby s větší přesností [35]. Dále se předpokládá, že se budou rozvíjet nástroje pro personalizovanou obchodní analýzu, které budou poskytovat obchodníkům na míru šité doporučení založené na jejich specifických potřebách a preferencích. [36]

Vývoj v oblasti AI a její aplikace v obchodování na burze naznačuje, že finanční trhy vstupují do nové éry digitalizace a automatizace, což může zásadně změnit způsob, jakým jsou obchody prováděny a řízeny.

## 4 NÁVRH A VÝVOJ AUTOMATIZOVANÉHO OBCHODNÍHO SYSTÉMU

### 4.1 Architektura systému

V této kapitole se zaměříme na návrh a vývoj automatizovaného obchodního systému pro kryptoburzu Binance, který využívá umělou inteligenci pro analýzu a rozhodování. Cílem je vytvořit robustní platformu, která efektivně zpracovává obchodní příkazy a vyhodnocuje tržní data s vysokou přesností a rychlostí. Systém bude integrovat několik klíčových technologií a metod:

1. **API Binance** pro přístup k tržním datům a provedení obchodů.
2. **Python** jako primární programovací jazyk pro vývoj algoritmu.
3. **AI** a Fundamentální analýza kryptotrhu
4. **AI** a Technická analýza kryptotrhu

V následujících sekcích podrobně prozkoumáme každou z těchto komponent, vysvětlíme si, jak jsou vzájemně propojené a jaké technologie a bezpečnostní opatření jsou implementovány pro zajištění spolehlivého a efektivního obchodování.

#### 4.1.1 API Binance

Binance API [38] je nástroj, který umožňuje vývojářům a obchodníkům automatizovaně interagovat s kryptoburzou Binance pomocí programovacích jazyků, jako je Python. API poskytuje rozhraní pro získávání informací o tržních datech, provádění obchodů, správu účtů a sledování objednávek v reálném čase. Umožňuje uživatelům vytvářet personalizované obchodní aplikace, které mohou reagovat na tržní změny rychleji, než je to možné manuálně.

Jak funguje Binance API:

1. **Získání API klíče:** Nejprve je nutné zaregistrovat se u Binance a získat API klíč a tajný klíč, které legitimizují a zabezpečují přístup API. Tyto klíče jsou zásadní pro autentizaci a komunikaci s Binance servery přes programové rozhraní.
2. **Vytváření a správa objednávek:** Binance API umožňuje uživatelům vytvářet, sledovat a spravovat obchodní objednávky. Uživatelé mohou nastavit objednávky na nákup nebo prodej kryptoměn za specifické ceny a podmínky. API poskytuje také přístup k historickým obchodním datům, která mohou být využita pro zpětné testování a optimalizaci obchodních strategií.
3. **Bezpečnost a regulace:** Bezpečnostní opatření, jako je šifrování komunikace a omezení přístupu pomocí IP adres, jsou klíčová pro ochranu obchodních a osobních údajů. Binance API také vyžaduje, aby všechny požadavky byly podepsány a ověřeny, což pomáhá chránit uživatele před neoprávněným přístupem a manipulací.

Tato flexibilita a široká škála funkcí činí Binance API populární volbou pro vývojáře algoritmických obchodních systémů, kteří hledají efektivní způsoby, jak maximalizovat své obchodní operace na jedné z největších kryptoměnových burz světa.

Pro Python byla v práci využita knihovna `binance-futures-connector`. Zamýšlím při obchodování využívat páku, z tohoto důvodu jsem vybral knihovnu umožňující přístup k futures kontraktům [37]. Dále je použita knihovna `python-binance` pro získávání historických dat z burzy.

#### 4.1.2 Python jako primární programovací jazyk pro vývoj algoritmu.

Python se stal jedním z nejpobulárnějších programovacích jazyků v oblasti finanční technologie a obchodování, a to díky své flexibilitě, široké škále knihoven a snadné čitelnosti. V kontextu vývoje algoritmů pro automatizované obchodování je Python obzvláště ceněn pro jeho schopnost rychle implementovat komplexní matematické modely a zpracovávat obrovské množství dat, což je klíčové pro real-time obchodní aplikace.

- **Python v Algoritmickém Obchodování:**

**Snadná integrace a široká podpora:** Jedním z hlavních důvodů, proč je Python preferovaný pro vývoj obchodních algoritmů, je jeho schopnost snadno se integrovat s různými datovými zdroji a obchodními platformami [1]. Python podporuje řadu knihoven, jako jsou `Pandas` pro manipulaci s daty, `NumPy` pro výpočty a `Matplotlib` pro vizualizace, které jsou nezbytné pro analýzu finančních dat. Navíc, Pythonové frameworky jako je `backtrader` nebo `pyalgotrade` umožňují testování strategií a simulaci obchodů, což výrazně urychluje vývoj a testování nových obchodních strategií. [11]

**Komunita a dostupné zdroje:** Python těží z velmi aktivní komunity vývojářů a široké škály otevřeně dostupných projektů a tutoriálů, které mohou vývojáři využít pro rychlé řešení problémů nebo pro inspiraci při vývoji vlastních algoritmických obchodních systémů [20]. Tato podpora je zásadní pro rychlé prototypování a inovace v oblasti algoritmického obchodování.

- **Python a Umělá Inteligence:**

**AI a strojové učení:** Python je také lídrem v oblasti umělé inteligence a strojového učení, s knihovnami jako `TensorFlow`, `Keras` a `Scikit-learn`, které umožňují vývojářům vytvářet sofistikované prediktivní modely [20]. Tyto nástroje jsou klíčové pro vývoj algoritmů, které mohou naučit se rozpoznávat vzory v historických cenových datech a predikovat budoucí cenové pohyby, což je neocenitelné pro automatizované obchodní strategie.

Python se stal základním stavebním kamenem pro mnohé, kdo se snaží prorazit ve světě finančních technologií, díky jeho schopnosti snadno zpracovávat a analyzovat data, jeho integrace s dalšími technologiemi a podpoře širší komunity. Jeho pokračující evoluce a adaptace na nové výzvy v oblasti datové analýzy a umělé inteligence slibuje udržet jeho postavení jako primárního nástroje pro vývojáře finančních aplikací i do budoucna. A tedy proto jsem se rozhodl využít Python pro svou práci i já.

### 4.1.3 AI a fundamentální analýza kryptotruhu

Je důležité vyjasnit si, jak a které fundamentální zprávy z trhu ovlivňují hodnotu určitého krypta. Existují zprávy, které jsou pro krypto bullish, bearish a neutrální. Cílem téhle kapitoly práce bude zisk zpráv z médií a vyhodnotit pomocí umělé inteligence jejich vliv na pohyby cen kryptoměn. Pojdme se napřed seznámit s vyskytujícími se typy zpráv:

#### **Bullish fundamentální zprávy:**

Zprávy a události, které obvykle působí na trhu s kryptoměnami bullish signály, zahrnují různé faktory, jako jsou institucionální investice, schválení regulačních rámců, technologické inovace a makroekonomické změny.

1. **Institucionální investice:** Když velké finanční instituce investují do kryptoměn, může to zvýšit důvěru v trh a vést k růstu cen. Například, zprávy o investicích společností jako MicroStrategy nebo Tesla měly v minulosti významný pozitivní vliv na ceny Bitcoinu.
2. **Schválení ETF:** Diskuse a potenciální schválení burzovně obchodovaných fondů (ETF) pro Bitcoin by mohlo přilákat další institucionální peníze a retailové investory, což by mohlo vést k růstu cen. [39]
3. **Technologické inovace a upgrady:** Velké technologické aktualizace, jako jsou například upgrade Ethereum na Proof of Stake (PoS), mohou také působit býčí signály, jelikož naznačují pokrok a zlepšení v síti. [40]
4. **Regulační pokrok:** Když vlády a regulační orgány přijímají kryptoměny prostřednictvím pozitivních regulačních rozhodnutí, může to zvýšit investorův zájem a působit na trhy býčí signály. Přijetí progresivních regulačních politik ve velkých trzích, jako je USA, má obvykle pozitivní dopad na trhy. [39]
5. **Makroekonomické faktory:** Rozhodnutí centrálních bank, jako je snížení úrokových sazeb nebo kvantitativní uvolňování, může vést k hledání alternativních investic, jako jsou kryptoměny, což může způsobit růst cen. [40]

#### **Neutrální fundamentální zprávy:**

Obvykle jde o zprávy, které poskytují běžné informace o trhu, jako jsou celkové tržní kapitalizace nebo objemy obchodování, bez specifických událostí nebo změn, mohou být považovány za neutrální, protože neuvádějí žádné nové nebo neočekávané informace, které by mohly trh výrazně ovlivnit. [41]

#### **Bearish fundamentální zprávy:**

Zprávy, které obvykle signalizují medvědí (bearish) trh pro kryptoměny, často zahrnují negativní faktory jako jsou regulace, negativní zprávy, tržní volatilita a obavy o celkové zdraví globální ekonomiky.

1. **Regulační zásahy:** Když vlády nebo regulační orgány zavedou přísnější pravidla pro kryptoměny, může to vést k nejistotě a poklesu cen [42]. Například zákazy těžby nebo obchodování s kryptoměnami v některých zemích mají často negativní dopad na trh.
2. **Technologické a institucionální selhání:** Kolaps velkých krypto institucí nebo technologické selhání může vést k rychlému poklesu důvěry investorů a cen

kryptoměn. Příklady zahrnují bankroty velkých burz nebo selhání významných krypto projektů.

3. **Negativní makroekonomické faktory:** Obavy z globálního ekonomického zpomalení nebo nejistota ohledně měnové politiky, jako jsou zvyšování úrokových sazeb, mohou mít negativní dopad na kryptoměnové trhy [43]. Tyto faktory mohou ovlivnit investiční strategie a vést k prodeji kryptoměn.
4. **Negativní zprávy a veřejné vnímání:** Negativní mediální pokrytí, jako jsou zprávy o podvodech, hackech nebo kritice ze strany vládních činitelů [44], může negativně ovlivnit nálady na trhu a vést k poklesu cen.

Tím by byly vysvětleny obecné základní typy zpráv a jejich vliv na cenu kryptoměn. Tyhle zprávy jsou zásadní pro celé krypto. U menších kryptoprojektů existují další specifičtější typy zpráv, které mají vliv na budoucí růst hodnoty takových kryptoměn.

### **Specifické zajímavé typy zpráv:**

Zprávy, které jsou důležité pro malé kryptoprojekty, se obvykle ještě týkají několika klíčových oblastí, včetně vývoje projektu, partnerství, regulace a přijetí trhu. Zde je několik konkrétních příkladů důležitých zpráv:

1. **Vývoj projektu a technologické inovace:** Zprávy o vývoji projektu, jako jsou aktualizace, vylepšení protokolu nebo nové verze, jsou klíčové. Příklady zahrnují zavedení nových funkcí nebo integraci s dalšími technologiemi a platformami. [45]
2. **Partnerství a kolaborace:** Nová partnerství s jinými technologickými nebo finančními firmami mohou znamenat klíčové kroky pro adopci a růst malých kryptoprojektů [46]. Například, spojení s většími platformami nebo získání podpory od významných investorů může výrazně zvýšit viditelnost a důvěryhodnost projektu.
3. **Regulativní zprávy:** Zprávy týkající se regulačních změn nebo stanovisek mohou mít výrazný dopad na malé kryptoprojekty [42]. Pozitivní regulační vyjádření může podpořit adopci, zatímco negativní může vést k nejistotě a sníženému zájmu investorů.
4. **Tržní přijetí a komunitní podpora:** Úspěch malých kryptoprojektů často závisí na podpoře a zapojení komunity [42]. Zprávy o rostoucích uživatelských bazích nebo komunitních iniciativách, které podporují projekt, mohou být pozitivním signálem pro potenciální investory a uživatele.

#### 4.1.4 AI a technická analýza trhu

V kryptoměnovém obchodování přináší umělá inteligence (AI) značné výhody, zvláště v oblasti technické analýzy. S využitím AI mohou obchodníci identifikovat tržní trendy a významné obchodní signály s přesností a rychlostí, které tradiční metody nemohou dosáhnout. Cílem této části práce bude využít umělé inteligence k rozhodování o obchodech na základě technické analýzy trhu. Následující sekce popisuje praktické postupy, které budou využity pro technickou analýzu kryptotrhu.

### **Prediktivní Modelování:**

AI algoritmy, jako jsou neuronové sítě a hluboké učení, jsou trénovány na rozsáhlých datech historických cen, objemu obchodů a dalších relevantních indikátorů, aby předpovídaly budoucí cenové pohyby. Tyto modely dokážou identifikovat složité vzorce, které jsou často neviditelné pro lidské analytiky. Například, modely mohou detekovat skryté vzorce v pohybech cen Bitcoinu během určitých tržních podmínek, což umožňuje obchodníkům lepší rozhodování o koupi nebo prodeji. [34]

### **Automatizované Obchodní Strategie:**

S pomocí AI mohou být obchodní strategie automatizovány tak, aby využívaly technické indikátory, jako jsou klouzavé průměry, RSI (Relative Strength Index), a MACD (Moving Average Convergence Divergence) [34]. AI nejenže interpretuje tyto indikátory v reálném čase, ale může také zpracovávat zprávy a sociální signály, aby upravila obchodní taktiky podle aktuálního sentimentu na trhu. [35]

#### **4.1.5 Shrnutí architektury automatizovaného obchodního systému**

Pro obchodování bude využita kryptoburza Binance, jako programovací jazyk bude použit Python. Vzhledem k umělé inteligenci jsou v tuhle chvíli dvě oblasti práce, dva úkoly:

- a) Vyhodnocování fundamentálních zpráv
- b) Vyhodnocování technických dat

Následující kapitola vysvětluje praktické řešení a postup práce na každém úkolu zvlášť.

## **4.2 Praktický vývoj automatizovaného systému**

### **4.2.1 Vyhodnocování fundamentálních zpráv**

Je důležité určit si odkud čerpat info. Pro sběr informací ze světa je zvolena platforma X (dříve Twitter). Známé osobnosti se tam často vyjadřují o důležitých událostech ze světa, které mají vliv na cenu Bitcoinu a ostatních kryptoměn. Na twitteru také začínající projekty dávají zprávy o navazování nových spoluprací a vývoji jejich projektu. Je důležité rozlišit zajímavé zprávy od těch méně zajímavých. Na X je mnoho zajímavých účtů, které je vhodné sledovat pro získávání zpráv o aktuálním dění kolem kryptoměn. Některé z nich mohou být následující:

#### **Elon Musk [47]**

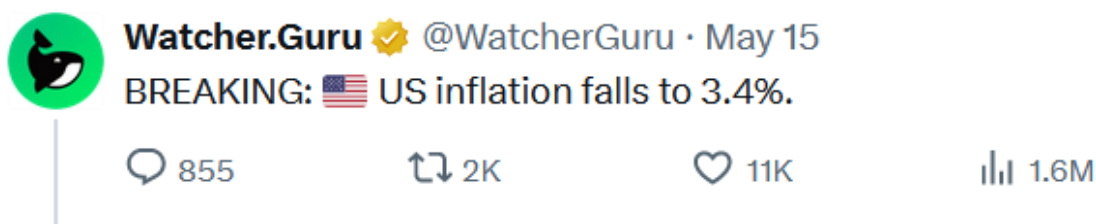
Elon Musk je známý vzhledem ke kryptoměnám mimo jiné kvůli jeho firmě Tesla a jeho podpoře Dogecoinu. V minulosti se cena Bitcoinu podezřele rychle pohybovala ihned po Elonových tweetech. Např. když tweetnul, že Tesla začne přijímat Bitcoin jako platidlo, cena Bitcoinu povyroستla během následujících několik hodin o XY %.

Podobný případ nastal u Dogecoinu. Když Elon tweetoval o tomhle coinu, jeho cena na Elonovy pozitivní tweety rychle reagovala. Jenže Elon tweetuje i o jiných věcech než o Tesle nebo Dogecoinu, většina jeho tweetů bývá pro kryptotrž neutralní.

Cílem je rozlišit zajímavé tweety od Elona hned jak vzniknou a na ty zajímavé upozornit.

### Watcher Guru [48]

Tenhle účet, podobně jako Elon Musk, občas tweetne více „zajímavé“ tweety, občas méně „zajímavé“ tweety. Příklad zajímavého tweetu:



Obr. 1: Zajímavý tweet ohledně inflace v US

Tweet zobrazený na obrázku informuje o poklesu inflace v US na 3,4 % (z 3,5 %). To je pro trh pozitivní zpráva, Podívejme se, jak zareagovala na tuhle informaci cena Bitcoinu:



Obr. 2: Reakce BTC na změnu inflace v US

Z obrázku 2 je patrné že od chvíle oznámení americké vlády o snížení inflace cena Bitcoinu během následujících pěti hodin vyrostla přibližně o 5,5 %.

Ale i tenhle účet poměrně často vysílá do světa méně zajímavé zprávy, například:



Obr. 3: Nezajímavá informativní zpráva ohledně ceny BTC

Tweet na obrázku 3 oznamuje aktuální hodnotu Bitcoinu. Je to pěkné, ale prakticky k ničemu.

U všech pozoruhodných účtů na twitteru je to podobně. Je potřeba tu značně převažující většinu nezajímavých zpráv odfiltrovat, vytvořit „síto“, které propustí všechn písek, avšak zlatá zrnka ponechá. Účtů ke sledování je mnoho, mohu zmínit například ještě Bitcoin Magazine [49], představitele burzy Binance CZ [50] a další ...

### Krok 1: Získání tweetů pro natrénování neuronové sítě k detekci zajímavých zpráv

Nejpřímější cestou k získání tweetů by bylo použití API klíčů twitteru a stáhnout tweety přímo z něj. Jenže tady mám smůlu, nedávno Twitter velmi omezil a zpoplatnil využívání svých API klíčů.

	Free	Basic	Pro	Enterprise
Getting access	Get Started	Get Started	Get Started	Get Started
Price	Free	\$100/month	\$5000/month	
Access to X API v2	✓ (Only Post creation)	✓	✓	
Access to standard v1.1	✓ (Only Media Upload, Help, Rate Limit, and Login with X)	✓ (Only Media Upload, Help, Rate Limit, and Login with X)	✓ (Only Media Upload, Help, Rate Limit, and Login with X)	
Project limits	1 Project	1 Project	1 Project	
App limits	1 App per Project	2 Apps per Project	3 Apps per Project	
Post caps - Post	1,500	3,000	300,000	
Post caps - Pull	✗	10,000	1,000,000	
Filtered stream API	✗	✗	✓	
Access to full-archive search	✗	✗	✓	
Access to Ads API	✓	✓	✓	

Obr. 4: Nedostatky oficiálních API klíčů Twitteru [51]

Z informací na obrázku vyplývá, že stažení 10 000 tweetů by jednoho stálo 100 USD. V tenhle okamžik volím pátrat po tweetech ještě někde jinde. Pokud má být moje práce efektivní, potřebuji se u jednotlivých účtů dostat co nejdále do historie. Samotný účet Elona Muska vydá za tisíce tweetů. Pro mou práci by jich bylo potřeba mnohem více.

## Pokus s Nitterem

Byl realizován pokus využít Nitter, což je alternativní rozhraní Twitteru, které je zaměřeno na ochranu soukromí uživatelů. Pomocí různých návodů na Youtube a knihoven pro Python jsem se mnoha různými způsoby pokoušel tweety získat. Výsledky však byly neuspokojivé, získat tweety touhle cestou je možné, podařilo se získat několik desítek tweetů, ovšem někdy pokus selhal ihned, někdy po pěti tweetech ... Jeden se zkrátka stále nemohl dostat hluboko do historie jednotlivých účtů na Twitteru.

## Zakoupení dat s tweety

Další variantou se nabízí objemy tweetů zkrátka zakoupit ještě někde jinde. Na stránkách WebAutomation [52] je možné si tweety koupit. Respektive bylo:



### X (Twitter) Profile Scraper

Our pre-built Twitter Scraper tool will scrape all tweets, retweets, replies starting from a Twitter profile page. Wit...

Webautomation 687

Down for maintenance

Obr. 5: Stažení tweetů aktuálně nedostupné [52]

Elonovy změny kolem přístupu k tweetům postihly, zdá se, všechny. Je to škoda, zde by se totiž dalo získat 4 800 000 tweetů v ceně 74 USD. To už by na kvalitní dataset stačit mohlo a současně bylo za cenu přijatelné.

### Project (Annual)



\$74/mo

\$890 Annually

Ideal for Small and medium scale projects where continuous data is required

Buy now

4.8M Row Credits annually

1 Free Extractor Build

Obr. 6: Přijatelná cena za tweety [52]

## Vytvoření tweetů

Vzhledem k okolnostem okolo Twitteru jsem se rozhodl tweety pro akademické účely vytvořit. Později, až se situace s Twitterem urovná, bude využito dat skutečných. Nyní však budiž postačujícím využití tweetů mnou samotným vytvořených.

Sice si lečjaké tweety pamatuji, ale dát dohromady stovky tweetů je relativně náročný úkol jak mentálně, tak i časově. Zde byl využit Chat-GPT4, kde byl použit předdefinovaný chatbot na vytváření virálních tweetů. Chatbotovi byl zadán úkol vytvořit stovky tweetů s následujícími tématy:

### Pozitivní témata

- Vývoj projektu a technické inovace
- Kolaborace
- Pozitivní regulativní zprávy
- Tržní přijetí a komunitní podpora
- Institucionální investice
- Makroekonomické faktory
- Listing nového krypta na burzu

### Negativní témata

- Negativní regulativní zprávy
- Regulační zásahy
- Technologické a institucionální selhání
- Negativní makroekonomické faktory
- Negativní zprávy a veřejné vnímání

### Neutrální témata

- Neutrální, oznamovací zprávy

Ukázka skutečného tweetu ohledně listingu nového krypta na burzu:



Obr. 7: Pozitivní zpráva z Twitteru

Ukázka jednoho z tweetů chatbotem vytvořeným ohledně listingu nového krypta na burzu:

🌟 New listing alert! [Crypto Project] will soon be on [Exchange Name]! Excitement is in the air!  
🌟 #CryptoAlert

Obr. 8: Vygenerovaná pozitivní zpráva

## Závěr

Tweety nejsou stejné, ale podobnost tam je. Všechny tweety jsou uloženy v souboru `lists_of_tweets.py`. Tímhle způsobem by tedy byly zajištěny aspoň provizorní tweety, než se situace s přístupem ke skutečným tweetům urovná.

### Krok 2: Vytvoření datasetu

V tuhle chvíli jsou tedy zajištěny tweety. Nyní od sebe potřebuji rozlišit pozitivní, negativní a neutrální zprávy. Pozitivním tweetům dávám ohodnocení „1“, negativním dávám „2“ a neutrálním dávám „0“. Vytvořený dataset se všemi tweety má název `tweets_all_final.csv`.

### Krok 3: Natrénování neuronové sítě

Kód s trénováním neuronové sítě má název `neuron_training.py`. Rozhodl jsem se použít knihovnu TensorFlow [53]. TensorFlow je výkonná platforma pro strojové učení, která umožňuje efektivní trénování neuronových sítí pro různé úkoly, včetně rozpoznávání textu. Pomocí TensorFlow lze vytvořit a trénovat modely hlubokého učení, které dokážou analyzovat textová data a rozpoznávat vzory a významy v textu.

Tento proces zahrnuje přípravu a načítání dat, konstrukci modelu s využitím vrstev neuronové sítě, trénování modelu na rozsáhlých textových datových sadách a jeho následnou optimalizaci. TensorFlow poskytuje nástroje a knihovny, jako je Keras, které zjednodušují tento vývojový cyklus.

Následuje popis kódu:

#### 1. Import knihoven.

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import tensorflow as tf
5 import tensorflow_hub as hub
```

Obr. 9: Python knihovny pro trénování neuronové sítě

NumPy je knihovna pro vědecké výpočty v Pythonu. Poskytuje podporu pro velké, vícerozměrné pole a matice, spolu s množstvím matematických funkcí pro práci s těmito poli. V tomto kódu se používá pro rozdělení dat na trénovací, validační a testovací sady.

Pandas je knihovna pro manipulaci a analýzu dat. Nabízí datové struktury a operace pro manipulaci s číselnými tabulkovými daty. V tomto kódu se používá k načítání CSV souboru do DataFrame a manipulaci s daty.

Matplotlib je knihovna pro tvorbu statických, animovaných a interaktivních vizualizací v Pythonu. Ve vytvořeném se používá pro vizualizaci histogramu hodnocení. Tohle pro trénování jako takové není nutné, slouží pouze pro kontrolu rozdělení kategorií v datasetu.

TensorFlow je open-source platforma pro strojové učení. Poskytuje komplexní, flexibilní ekosystém nástrojů, knihoven a komunitních zdrojů, které umožňují výzkumníkům posunout stav umělé inteligence a vývojářům snadno vytvářet a nasazovat aplikace založené na strojovém učení.

TensorFlow Hub je knihovna pro publikování, vyhledávání a používání předtrénovaných modelů. Tyto modely mohou být znovu použity v nových úlohách, což může výrazně zkrátit čas a výpočetní prostředky potřebné pro trénování modelů od začátku. V mém kódu se používá k načtení předtrénované vrstvy vektorové reprezentace textu.

## 2. Načtení csv souboru s trénovacím datasetem.

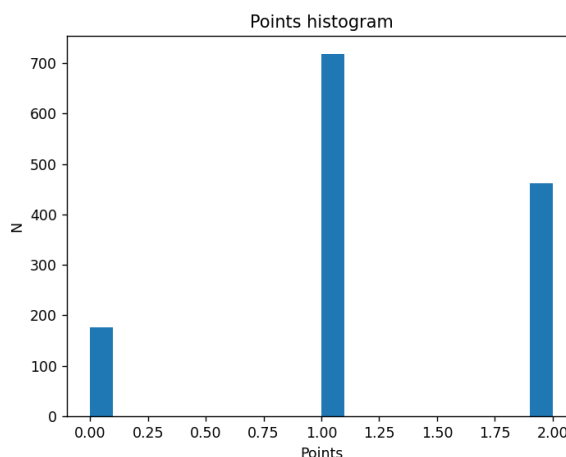
```
8 # Načtení csv souboru
9 df = pd.read_csv("tweets_all_final.csv", sep=';', usecols=['tweet', 'rating'])
10
11 # Odstranění řádků, kde nejsou žádné hodnoty
12 df = df.dropna(subset=["tweet", "rating"])
13 print(df.head())
```

Obr. 10: Načtení trénovacího datasetu

Následuje kontrola rovnoměrnosti rozdělení mých trénovacích dat do kategorií.

```
15 # obrázek vizualizace mých dat a jejich ohodnocení.
16 plt.hist(df.rating, bins=20)
17 plt.title("Points histogram")
18 plt.ylabel("N")
19 plt.xlabel("Points")
20 plt.show()
```

Obr. 11: Řádky kódu pro kontrolu datasetu

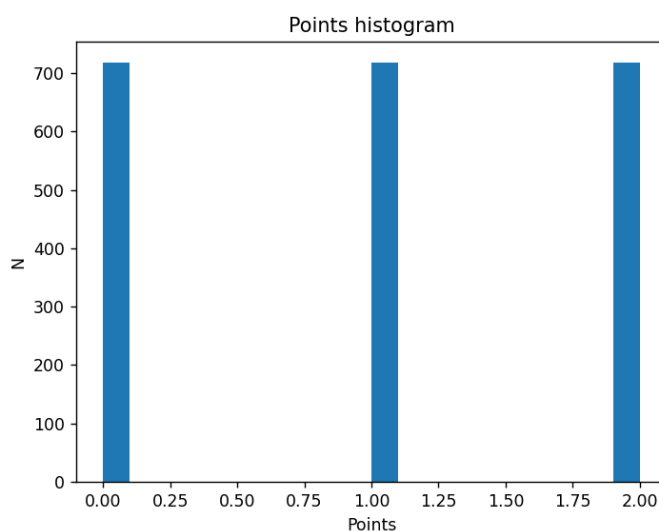


Obr. 12: Nerovnoměrné rozdělení ohodnocení dat

Při prvotním pokusu bylo vidno, že máme hodně dat s kategorií ratingu 1 nebo 2, ale velmi málo dat s přiřazením ratingu 0. Je záhodno, aby byly třídy v datasetu rozděleny co nejrovnoměrněji. Vyrovnané rozdělení tříd v datasetu zamezuje zkreslení modelu, zlepšuje jeho přesnost a zajišťuje férovost a robustnost. Díky vyrovnaným třídám jsou metriky výkonnosti přesnější a model má lepší schopnost generalizace na nové a různorodé datové sady. Nerovnoměrné rozdělení tříd v datasetu lze vyřešit několika způsoby. Zde jsou dva běžné přístupy:

- A) **Oversampling (překročování):** Zvětší se počet řádků v datasetu s méně zastoupenými třídami tak, že se tyto řádky replikují.
- B) **Undersampling (podvzorkování):** Zmenší se počet řádků v datasetu s více zastoupenými třídami tak, že se některé z nich odstraní.

Byla zvolena varianta řešení A). Byl tedy vytvořený krátký kód s názvem *class\_balance.py*. Tenhle kód zkrátka vezme původní soubor *tweets\_all\_final.csv* a vytvoří nový soubor *tweets\_all\_final\_balanced.csv*, kde je již rozdělení tříd rovnoměrné.



Obr. 13: Upravený dataset, rovnoměrné rozdělení tříd

### 3. Rozdělení dat na trénovací, validační a testovací.

```
22 # Rozdělení dat na trénovací, validační a testovací
23 train, val, test = np.split(df.sample(frac=1), [int(0.8*len(df)), int(0.9*len(df))])
```

Obr. 14: Rozdělení dat na trénovací, validační a testovací

- **`df.sample(frac=1)`** zamíchá řádky v dataframe `df` náhodným způsobem. Argument `frac=1` znamená, že se použije 100 % řádků, ale v náhodném pořadí. Tímto krokem zajišťujeme, že rozdělení dat je náhodné a nezávislé na původním pořadí řádků v datasetu.
- Funkce **`np.split`** rozděluje data na tři části podle indexů uvedených ve druhém argumentu.
- **`[int(0.8*len(df)), int(0.9*len(df))]`** jsou indexy, které určují hranice mezi těmito částmi:
  - **`int(0.8*len(df))`**: Tento index určuje hranici mezi trénovací a validační sadou. Znamená to, že prvních 80 % řádků (po zamíchání) bude použito jako trénovací data.

- `int(0.9*len(df))`: Tento index určuje hranici mezi validační a testovací sadou. Znamená to, že následujících 10 % řádků bude použito jako validační data.
- Posledních 10 % řádků bude použito jako testovací data.
- Výsledek funkce `np.split` je trojice dataframe objektů, které jsou přiřazeny do proměnných `train`, `val` a `test`.
  - `train`: Trénovací data obsahující prvních 80 % zamíchaných řádků.
  - `val`: Validační data obsahující následujících 10 % zamíchaných řádků.
  - `test`: Testovací data obsahující posledních 10 % zamíchaných řádků.

Tohle rozdělení je důležité pro trénování a hodnocení modelu, protože zajišťuje, že model je trénován, validován a testován na různých částech dat, což pomáhá předcházet overfittingu a poskytuje přesnější odhady výkonnosti modelu na nových datech.

#### 4. Vytvoření funkce pro převod našeho dataframe na TensorFlow dataset.

```

25 # Převod Pandas dataframe na Tensorflow dataset
26 def df_to_dataset(dataframe, shuffle=True, batch_size=512):
27     df = dataframe.copy()
28     labels = df.pop('rating')
29     df = df["tweet"]
30     ds = tf.data.Dataset.from_tensor_slices((df, labels))
31     if shuffle:
32         ds = ds.shuffle(buffer_size=len(dataframe))
33     ds = ds.batch(batch_size)
34     ds = ds.prefetch(tf.data.AUTOTUNE)
35     return ds

```

Obr. 15: Převod dataframe na Tensorflow dataset

- **`df = dataframe.copy()`**
  - Tenhle řádek vytvoří kopii vstupního dataframe, aby se původní data nezměnila.
- **`labels = df.pop('rating')`**
  - Metoda `pop` odstraní sloupec `rating` z kopie dataframe a uloží jej do proměnné `labels`.
  - Tímto způsobem oddělíme cílové hodnoty (ratingy) od vstupních hodnot (tweetů).
- **`df = df["tweet"]`**
  - Zkopírovaný dataframe `df` nyní obsahuje pouze sloupec `tweet`, protože sloupec `rating` byl odstraněn v předchozím kroku.
  - Tato řádka extrahuje sloupec `tweet` a přiřadí jej zpět do `df`.
- **`ds = tf.data.Dataset.from_tensor_slices((df, labels))`**
  - Funkce `from_tensor_slices` vytvoří TensorFlow Dataset z dvojice (`df`, `labels`). Každý prvek datasetu bude dvojice (tweet, rating).
  - Tento dataset je základním datovým formátem pro trénování v TensorFlow.
- **`if shuffle`**
  - Pokud je parametr `shuffle` nastaven na `True`, následující kód zamíchá dataset.
- **`ds = ds.shuffle(buffer_size=len(dataframe))`**

- Metoda shuffle zamíchá data v datasetu. `buffer_size=len(dataframe)` znamená, že se použije buffer velikosti celého dataframe, což zajistí efektivní zamíchání dat.
- **`ds = ds.batch(batch_size)`**
  - Metoda batch rozdělí data do menších skupin (batchí) o velikosti určené parametrem `batch_size` (výchozí hodnota je 512).
  - Batching je klíčový pro efektivní trénování modelu, protože zpracování dat po batchích je rychlejší a efektivnější než zpracování jednoho vzorku najednou.
- **`ds = ds.prefetch(tf.data.AUTOTUNE)`**
  - Metoda prefetch umožňuje, aby se data načítala a připravovala v předstihu během trénování, čímž se minimalizují prodlevy způsobené I/O operacemi.
  - `tf.data.AUTOTUNE` automaticky nastaví optimální počet prvků, které se mají předem načítat.
- **`return ds`**
  - Funkce vrátí vytvořený TensorFlow Dataset, který je připravený pro použití v trénovacím procesu modelu.

Cílem funkce `df_to_dataset` je převést Pandas DataFrame na TensorFlow Dataset, který odděluje vstupní data (tweet) od cílových hodnot (rating). Funkce navíc zajišťuje, že data jsou volitelně zamíchána a rozdělena do batchí pro efektivní trénování. Přidáním přednačítání dat zvyšuje výkon při trénování modelu.

#### 5. Rozdělení dat na trénovací, validační a testovací pomocí předešlé funkce.

```
37 # Rozdělení dat na trénovací, validační a testovací
38 train_data = df_to_dataset(train)
39 valid_data = df_to_dataset(val)
40 test_data = df_to_dataset(test)
```

Obr. 16: Rozdělení dat pomocí funkce

#### 6. Použití předtrénovaného modelu "Neural Network Language Model" (NNLM) s 50-dimenzionálními vektory.

```
42 # ## Vložení a model
43 # Jde o to, že využiji předtrénovanou vrstvu vektorové
44 # reprezentace ("neural network language model" (NNLM))
45 embedding = "https://tfhub.dev/google/nnlm-en-dim50/2"
```

Obr. 17: Použití předtrénovaného modelu

Embedding (česky "zapouzdření" nebo "vložená reprezentace") je způsob, jak převést kategorické nebo textové údaje do numerických vektorů, které mohou být použity jako vstupy do strojového učení nebo neuronových sítí. V kontextu zpracování přirozeného jazyka (NLP) se embeddingy používají k reprezentaci slov nebo frází v kontinuálním vektorovém prostoru.

V tenhle okamžik jsem měl dvě možnosti. Buďto bych si vytvořil vlastní embedding, to by znamenalo že bych vytvořil model, který bych naučil převádět text na jeho číselnou reprezentaci. Nebo je zde druhá varianta, že bych použil již existující model pro převod

slov na číselnou reprezentaci. Nakonec jsem zvolil použít předtrénovaný model NNLM. Podívejme se na možné výhody použití již předtrénovaného modelu:

- **Rychlejší trénování**

Předtrénovaný model již prošel rozsáhlým tréninkem na velkém množství dat, což znamená, že jeden již nemusí trávit čas trénováním modelu od začátku. To umožňuje rychlejší vývoj a nasazení aplikací.

- **Lepší výkon**

Předtrénované modely, jako NNLM, byly trénovány na velkých a různorodých textových korpusech, což znamená, že jsou schopny lépe zachytit složité jazykové struktury a významy než modely trénované na menších datových sadách. To vede k lepšímu výkonu při zpracování přirozeného jazyka (NLP) v různých úlohách.

- **Generalizace na nové úlohy**

Předtrénované modely mají schopnost dobře generalizovat na různé úlohy zpracování přirozeného jazyka. To znamená, že mohou být snadno přizpůsobeny specifickým úlohám, jako je sentimentová analýza, klasifikace textu, pojmenované entity atd., aniž by bylo potřeba mnoho dalších dat pro doladění.

- **Redukce dimenzionality**

50-dimenzionální vektory poskytují kompaktní reprezentaci textu, snižuje se tím složitost modelu a potřeba paměti, zatímco stále zachovávají dostatečně bohatou informaci pro mnoho aplikací. To je užitečné zejména v prostředích s omezenými výpočetními zdroji.

- **Snadná integrace:**

TensorFlow Hub poskytuje jednoduchý způsob, jak integrovat předtrénované modely do vašich aplikací. To umožňuje vývojářům snadno využívat pokročilé techniky strojového učení bez potřeby hlubokých znalostí v oblasti trénování modelů.

Použití předtrénovaného NNLM s 50-dimenzionálními vektory tedy může značně zjednodušit a urychlit proces vývoje modelů pro zpracování přirozeného jazyka, zatímco poskytuje vysokou úroveň výkonu a efektivity. Vraťme se nyní k popisu mého kódu:

- **embedding = ...**

Proměnná embedding je nastavena na URL adresu předtrénované embedding vrstvy, kterou poskytuje TensorFlow Hub.

- **"<https://tfhub.dev/google/nnlm-en-dim50/2>"**

Tohle je URL adresa specifického modelu vektorové reprezentace textu, nazývaného "Neural Network Language Model" (NNLM) s 50-dimenzionálními vektory.

- Tento model byl předtrénován na velkém množství textových dat a dokáže převádět text (věty, slova) do vektorového formátu, který zachycuje sémantický obsah textu.

#### 7. Vytvoření modelu neuronové sítě.

```
47 # Vytvoření modelu neuronové sítě
48 model = tf.keras.Sequential([
49     hub.KerasLayer(embedding, dtype=tf.string, trainable=True),
50     tf.keras.layers.Dense(4, activation='relu'),
51     tf.keras.layers.Dropout(0.4),
52     tf.keras.layers.Dense(128, activation='tanh'),
53     tf.keras.layers.Dropout(0.4),
54     tf.keras.layers.Dense(3, activation='softmax')
55 ])
```

Obr. 18: Model neuronové sítě pro klasifikaci textu

Tato část kódu vytváří model neuronové sítě pomocí Keras API v TensorFlow. Model je sekvenční, což znamená, že vrstvy jsou přidávány jedna za druhou. Pojďme si podrobně vysvětlit jednotlivé vrstvy a jejich funkce:

- **hub.KerasLayer(embedding, dtype=tf.string, trainable=True)**
  - Účel: Tato vrstva je předtrénovaná embedding vrstva z TensorFlow Hub, která převádí text na vektorovou reprezentaci, jak jsme si vysvětlili v předchozím bodu.
  - Parametry:
    - embedding: Odkaz na URL předtrénovaného embeddingu.
    - dtype=tf.string: Vstupní data jsou řetězce (text).
    - trainable=True: Umožňuje trénování této vrstvy, to znamená, že se její váhy mohou během trénování upravovat.
- **tf.keras.layers.Dense(4, activation='relu')**
  - Účel: Hustá (plně propojená) vrstva se 4 neurony.
  - Parametry:
    - 4: Počet neuronů ve vrstvě.
    - activation='relu': ReLU (Rectified Linear Unit) aktivační funkce, která pomáhá modelu učit se nelineární vztahy.
- **tf.keras.layers.Dropout(0.4)**
  - Účel: Dropout vrstva, která náhodně vypíná 40 % (0.4) neuronů během trénování.
  - Parametry:
    - 0.4: Pravděpodobnost vypnutí každého neuronu. Dropout pomáhá předcházet overfittingu.
- **tf.keras.layers.Dense(128, activation='tanh')**
  - Účel: Hustá vrstva se 128 neurony.
  - Parametry:
    - 128: Počet neuronů ve vrstvě.

- `activation='tanh'`: Tanh (hyperbolická tangenta) aktivační funkce, která pomáhá modelu učit se složité vztahy.
- **`tf.keras.layers.Dense(3, activation='softmax')`**
- Účel: Výstupní vrstva s 3 neurony, odpovídající třem třídám v klasifikační úloze.
- Parametry:
  - 3: Počet neuronů, odpovídající počtu tříd v klasifikaci.
  - `activation='softmax'`: Softmax aktivační funkce, která převádí výstupy na pravděpodobnosti, které sumují na 1. Tato funkce je vhodná pro více třídovou klasifikaci.

Tahle část kódu popisuje můj návrh modelu Neuronové sítě pro klasifikaci textu do tří kategorií. Počet vrstev a parametry mnou zvolené jsou výsledkem experimentování s nimi. K experimentování se více vyjádřím ještě později.

## 8. Definování ztrátové funkce

```
65 # Aktualizace ztrátové funkce (určena pro použití s více třídami)
66 loss_function = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False)
```

Obr. 19: Ztrátová funkce

- **Definice ztrátové funkce**
  - Tento řádek kódu definuje ztrátovou funkci pomocí `tf.keras.losses.SparseCategoricalCrossentropy`. Ztrátová funkce měří, jak dobře nebo špatně model funguje během trénování, a poskytuje základ pro úpravu vah modelu.
- **Sparse Categorical Crossentropy**
  - `SparseCategoricalCrossentropy` je speciální typ ztrátové funkce používaný pro klasifikaci, kde cílové hodnoty jsou ve formě celých čísel, která reprezentují třídy (např. 0, 1, 2).
  - Tato ztrátová funkce je vhodná pro více třídovou klasifikaci, kde jsou cílové hodnoty diskrétní (např. klasifikace do tří tříd).
- **from\_logits=False**
  - Parametr `from_logits` specifikuje, zda výstupy modelu jsou logits nebo pravděpodobnosti.
  - Logits jsou surové výstupy neuronové sítě před aplikací aktivační funkce `softmax`.
  - Pravděpodobnosti jsou výstupy po aplikaci `softmax`, která převádí logits na pravděpodobnosti, které sumují na 1.
  - V tomto případě je `from_logits=False`, což znamená, že výstupy modelu jsou pravděpodobnosti (tj. `softmax` byla aplikována na výstupy poslední vrstvy modelu).

Tento řádek kódu nastavuje ztrátovou funkci `SparseCategoricalCrossentropy` pro klasifikační úlohu, kde model předpovídá pravděpodobnosti tříd (po aplikaci

softmax) a cílové hodnoty jsou ve formě celých čísel. Tato ztrátová funkce pomáhá modelu učit se minimalizovat chyby v klasifikaci během trénování.

#### 9. Nastavení způsobu učení modelu.

```
68 # Nastavení jak se model bude učit
69 model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
70               loss=loss_function,
71               metrics=['accuracy'])
```

Obr. 20: Kompilace modelu NN

Tento blok kódu nastavuje parametry pro proces učení modelu pomocí metody `compile`.

- **`model.compile`**
  - Tato metoda připravuje (kompiluje) model pro trénování. Specifikuje, jakým způsobem se model bude učit a jak se bude hodnotit jeho výkon během trénování.
- **`optimizer=tf.keras.optimizers.Adam(learning_rate=0.001)`**
  - **Optimizer:** Optimalizátor je algoritmus, který upravuje váhy modelu na základě gradientu ztrátové funkce.
  - **Adam:** Adam (Adaptive Moment Estimation) je pokročilý optimalizační algoritmus, který kombinuje výhody dvou jiných rozšířených optimalizátorů: AdaGrad a RMSProp. Je známý svou efektivitou a stabilitou při trénování hlubokých neuronových sítí.
  - **learning\_rate=0.001:** Parametr `learning_rate` určuje, jak velký krok bude optimalizátor dělat při každé aktualizaci vah. Hodnota 0.001 je běžná volba, která často dobře funguje v praxi.
- **`loss=loss_function`**
  - **Loss:** Ztrátová funkce měří, jak dobře model funguje při předpovídání cílových hodnot. Definovaná ztrátová funkce `loss_function` je `SparseCategoricalCrossentropy`, což je vhodné pro více třídovou klasifikaci s diskrétními cílovými hodnotami (třídami), jak jsme si uvedli dříve.
- **`metrics=['accuracy']`**
  - **Metrics:** Metriky slouží k hodnocení výkonu modelu. Metriky nejsou použity při trénování modelu (neovlivňují aktualizaci vah), ale poskytují užitečné informace o tom, jak dobře model funguje.
  - **Accuracy:** Metrika `accuracy` měří přesnost modelu, což je poměr správných předpovědí k celkovému počtu předpovědí. Je to běžná metrika pro klasifikační úlohy.

Celkově tento krok připravuje model pro proces učení, specifikuje, jakým způsobem se budou aktualizovat váhy a jak se bude měřit výkon modelu během trénování.

## 10. Trénování modelu.

```
73 # trénování modelu
74 history = model.fit(train_data, epochs=100, validation_data=valid_data)
```

Obr. 21: Trénování modelu

Tento řádek kódu provádí trénování modelu a ukládá výsledky trénování do proměnné `history`.

- **model.fit**
  - Metoda `fit` trénuje model na poskytnutých trénovacích datech. Během trénování model optimalizuje své váhy, aby minimalizoval ztrátovou funkci a zlepšil výkon podle specifikovaných metrik (např. přesnost).
- **train\_data**
  - `train_data` je dataset použitý pro trénování modelu. Obsahuje vstupní data (`tweet`) a odpovídající cílové hodnoty (`rating`).
- **epochs=100**
  - Parametr `epochs` určuje počet průchodů celým trénovacím datasetem. V tomto případě je model trénován na celých trénovacích datech 100krát. Více epoch obvykle znamená, že model má více příležitostí se učit, ale může také zvýšit riziko overfittingu.
- **validation\_data=valid\_data**
  - Parametr `validation_data` specifikuje validační dataset, který se používá k vyhodnocení výkonu modelu po každé epoše trénování. Validační data nejsou použita pro úpravu vah modelu, ale slouží k monitorování toho, jak dobře model generalizuje na neviděná data.
- **history**
  - `history` je objekt typu `History`, který obsahuje informace o průběhu trénování. Tento objekt ukládá hodnoty ztráty a metrik pro každou epochu trénování a validace.
  - Je možné přistupovat k těmto informacím pomocí `history.history`, což je slovník s klíči jako `'loss'`, `'accuracy'`, `'val_loss'` a `'val_accuracy'`, které odpovídají ztrátě a metrikám na trénovacích a validačních datech.

## 11. Vyhodnocení výkonu natrénovaného modelu na trénovacích, validačních a testovacích datech.

```
76 # Vyhodnocení výkonu natrénovaného modelu na zadaných trénovacích datech
77 train_eval = model.evaluate(train_data)
78 print(f"Training Data - Loss: {train_eval[0]}, Accuracy: {train_eval[1]}")
79
80 # Vyhodnocení výkonu natrénovaného modelu na zadaných validačních datech
81 valid_eval = model.evaluate(valid_data)
82 print(f"Validation Data - Loss: {valid_eval[0]}, Accuracy: {valid_eval[1]}")
83
84 # Vyhodnocení výkonu natrénovaného modelu na zadaných testovacích datech
85 test_eval = model.evaluate(test_data)
86 print(f"Test Data - Loss: {test_eval[0]}, Accuracy: {test_eval[1]}")
87
88 # Uložení modelu do keras souboru
89 model.save('neuron_model_03.keras')
90 print('Model has been saved')
```

Obr. 22: Vyhodnocení a uložení modelu

Tyto řádky kódu vyhodnocují výkonnost natrénovaného modelu na trénovacích, validačních a testovacích datech. Výsledky těchto vyhodnocení poskytují informace o tom, jak dobře model funguje na trénovacích, validačních a testovacích datech, což pomáhá posoudit schopnost modelu generalizovat na nová data.

## 12. Grafické zobrazení výsledků trénování

```
92 # Grafické zobrazení výsledků trénování
93 plt.figure(figsize=(12, 6))
94
95 # Graf ztráty
96 plt.subplot(1, 2, 1)
97 plt.plot(history.history['loss'], label='Training Loss')
98 plt.plot(history.history['val_loss'], label='Validation Loss')
99 plt.xlabel('Epochs')
100 plt.ylabel('Loss')
101 plt.title('Training and Validation Loss')
102 plt.legend()
103 plt.savefig('training_validation_loss.png')
104
105 # Graf přesnosti
106 plt.subplot(1, 2, 2)
107 plt.plot(history.history['accuracy'], label='Training Accuracy')
108 plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
109 plt.xlabel('Epochs')
110 plt.ylabel('Accuracy')
111 plt.title('Training and Validation Accuracy')
112 plt.legend()
113 plt.savefig('training_validation_accuracy.png')
114
115 plt.show()
```

Obr. 23: Grafické zobrazení výsledků trénování

Tahle část kódu vytváří dva grafy, které zobrazují průběh trénování modelu neuronové sítě. První graf zobrazuje ztrátu (loss) a druhý graf zobrazuje přesnost (accuracy) modelu během trénovacích epoch pro trénovací i validační data.

## a) Graf ztráty

Na vodorovné ose je počet epoch trénování. Na svislé ose je potom hodnota ztrátové funkce, která měří chybu modelu. Křivka Training Loss reprezentuje ztrátu na trénovacích datech v průběhu epoch. Validation Loss reprezentuje ztrátu na validačních datech v průběhu epoch.

Tento graf umožňuje sledovat, jak se hodnota ztrátové funkce mění v průběhu trénování. Ideálně by měla trénovací ztráta klesat a validační ztráta by měla také klesat, ale s určitou stabilitou, bez výrazného nárůstu, což by mohlo indikovat overfitting.

## b) Graf přesnosti

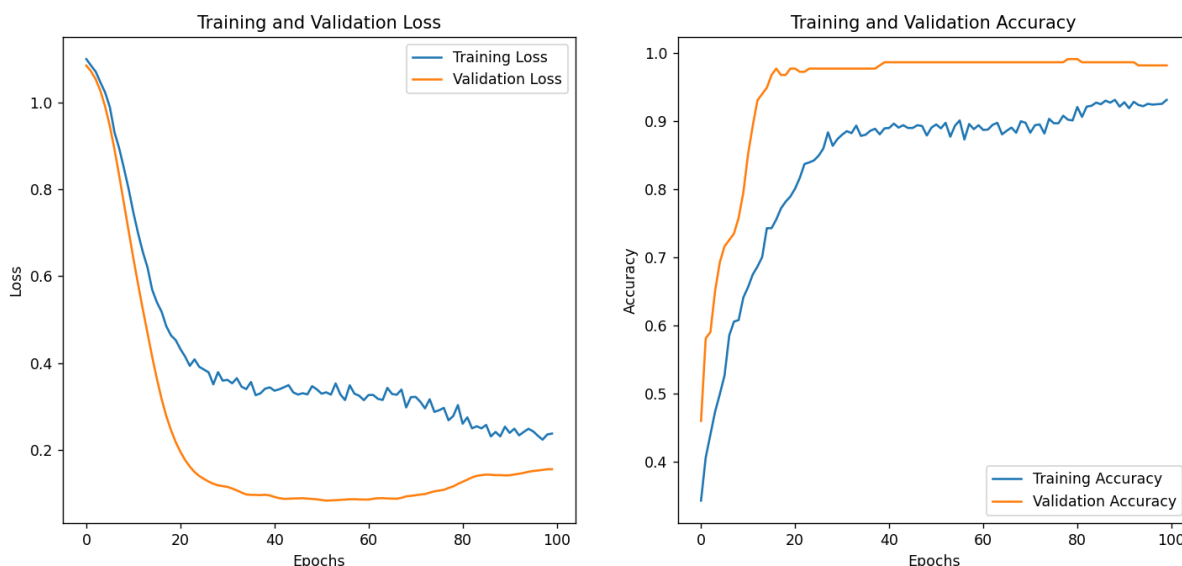
Na vodorovné ose je počet epoch trénování. Na svislé ose je hodnota přesnosti modelu. Křivka Training Accuracy reprezentuje přesnost na trénovacích datech v průběhu epoch. Validation Accuracy reprezentuje přesnost na validačních datech v průběhu epoch.

Tento graf umožňuje sledovat, jak se hodnota přesnosti mění v průběhu trénování. Ideálně by měla trénovací přesnost stoupat a validační přesnost by měla také stoupat nebo se stabilizovat, což by indikovalo, že model se učí správně a generalizuje dobře na neviděná data.

Tyto dva grafy dohromady poskytují vizuální přehled o průběhu trénování modelu. Graf ztráty ukazuje, jak dobře se model učí minimalizovat chybu, zatímco graf přesnosti ukazuje, jak dobře model klasifikuje data.

Krok 4: Experimentování a výsledky trénování neuronové sítě

Na následujícím obrázku jsou grafy výsledků trénování neuronové sítě s parametry uvedenými v předešlém textu:



Obr. 24: Grafy výsledků prvního trénování

Z obrázku je patrné že křivky Training Loss, Training Accuracy a Validation Accuracy jsou v pořádku. Křivka Validation Loss však v pořádku není. Problém je že ke konci

epoch začíná mírně narůstat. Když validační ztráta po určitém počtu epoch začne mírně narůstat, zatímco trénovací ztráta stále klesá nebo se stabilizuje, obvykle to znamená, že model začíná **overfitovat** na trénovací data.

Overfitting nastává, když se model příliš přizpůsobí specifikům trénovacích dat a ztrácí schopnost generalizace na neviděná data (validační nebo testovací data). Takle záležitost lze napravit vícero způsoby. Lze například:

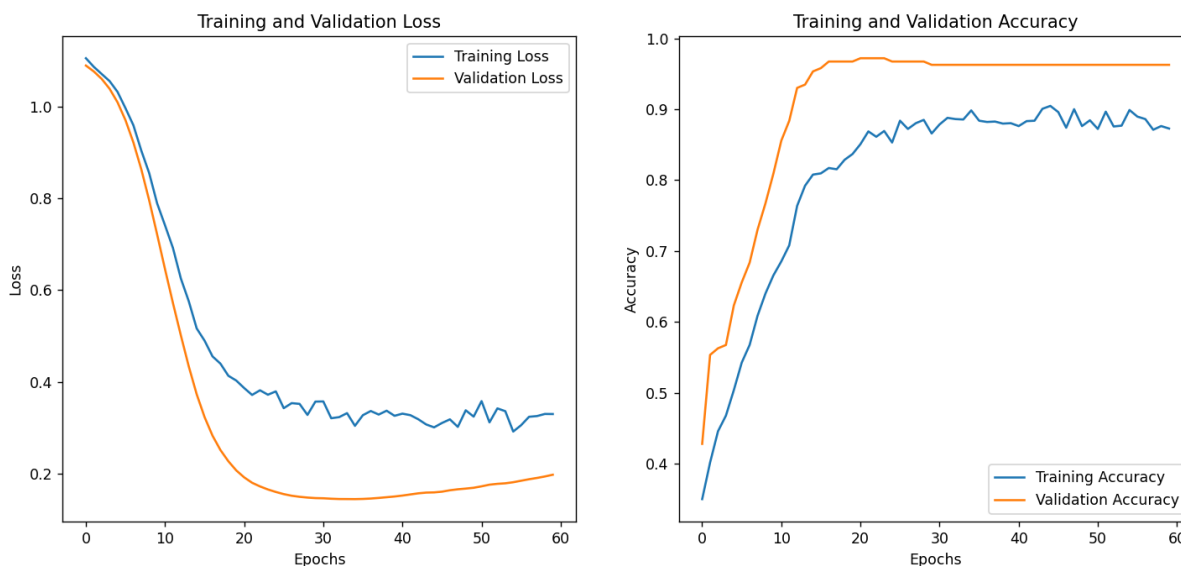
- **Použití Early Stopping**
  - Přestat trénovat model, jakmile validační ztráta přestane klesat. Toho lze dosáhnout pomocí callbacku `EarlyStopping` v Keras.
- **Snížení počtu epoch**
  - Zkrátit počet epoch, například na 60, kde je přibližně bod, kde se validační ztráta přestane zlepšovat.
- **Zvýšení regularizace**
  - Zvýšit dropout rate nebo přidat L2 regularizaci k vahám modelu.
- **Použití menšího modelu**
  - Snížit počet vrstev nebo neuronů ve vrstvách, což sníží kapacitu modelu přizpůsobit se trénovacím datům.

Při pohledu na výsledky testování neuronové sítě na trénovacích, validačních a testovacích datech se zdají být výsledky relativně v pořádku. Přesnost modelu nad 90 % je relativně dobrý výsledek, cílem je nyní dostat ztrátu pod 10 %.

```
Training Data - Loss: 0.0407567135989666, Accuracy: 1.0
1/1 [=====] - 0s 178ms/step - loss: 0.1551 - accuracy: 0.9814
Validation Data - Loss: 0.1551082581281662, Accuracy: 0.9813953638076782
1/1 [=====] - 0s 276ms/step - loss: 0.1289 - accuracy: 0.9815
Test Data - Loss: 0.128932923078537, Accuracy: 0.9814814925193787
```

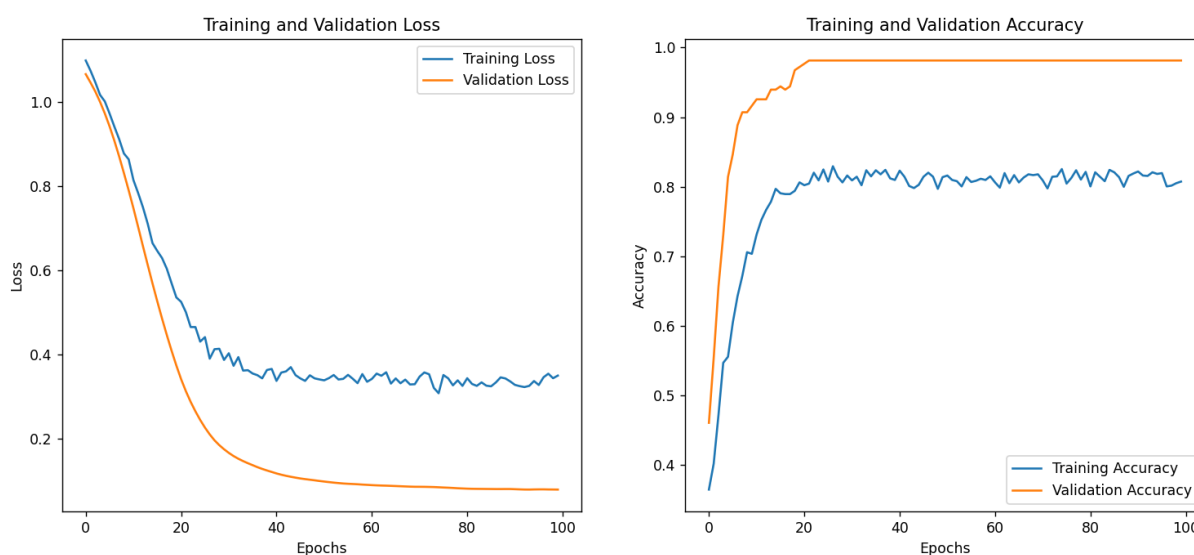
Obr. 25: Přesnost a ztráta prvního trénování

Pro svůj problém nyní volím snížení počtu epoch na 60. Na následujícím obrázku jsou výsledky s upraveným počtem epoch:



Obr. 26: Výsledky druhého pokusu trénování NN

Úprava počtu epoch zjevně nepomohla. Po několika dalších pokusech bylo dosaženo následujících výsledků:



Obr. 27: Nejlepší výsledky trénování NN

Nakonec byl počet epoch ponechán na hodnotě 100. Úprava spočívala ve změně počtu neuronů vrstvy s aktivační funkcí „tanh“ snížením na 32. Všechny ostatní parametry zůstaly ponechány bezezměny jak byly uvedeny dříve. Dosažená přesnost modelu je 98 % se ztrátou 7,8 %. Tenhle výsledek považuji za uspokojivý.

```
Training Data - Loss: 0.006722636055201292, Accuracy: 1.0
1/1 [=====] - 0s 166ms/step - loss: 0.0786 - accuracy: 0.9814
Validation Data - Loss: 0.07856275886297226, Accuracy: 0.9813953638076782
1/1 [=====] - 0s 189ms/step - loss: 0.0774 - accuracy: 0.9815
Test Data - Loss: 0.07739969342947006, Accuracy: 0.9814814925193787
```

Obr. 28: Číselné výsledky přesnosti a ztráty

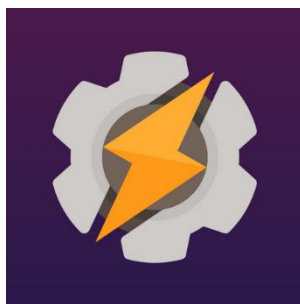
V průběhu pokusů bylo zajímavé například zjištění, že s přibývajícím počtem vrstev modelu klesá jeho přesnost a zvyšuje se ztráta. Důležité je také použití aktivační funkce „tanh“. Bez použití téhle funkce model za použití pouze aktivační funkce „relu“, „sigmoid“ nebo kombinace těchto dvou byly získávány značně méně uspokojivé výsledky.

### Krok 5: Zisk zpráv z Twitteru v reálném čase

Neuronová síť je připravena, stačí již pouze zajistit, aby četla text tweetu ve chvíli, kdy vznikne a ihned na tweet reagovala. Jak již víme, API twitteru v tenhle okamžik není schůdná varianta. Rozhodl jsem se podniknout následující tah, vytvořit systém, pomocí kterého se nakonec podařilo tweety získávat ve chvíli jejich vzniku. Zmíněný systém získávání zpráv se skládá z více částí, ke každé se nyní v rámci informovaného vysvětlení vyjádřím:

#### ❖ Mobilní aplikace Tasker

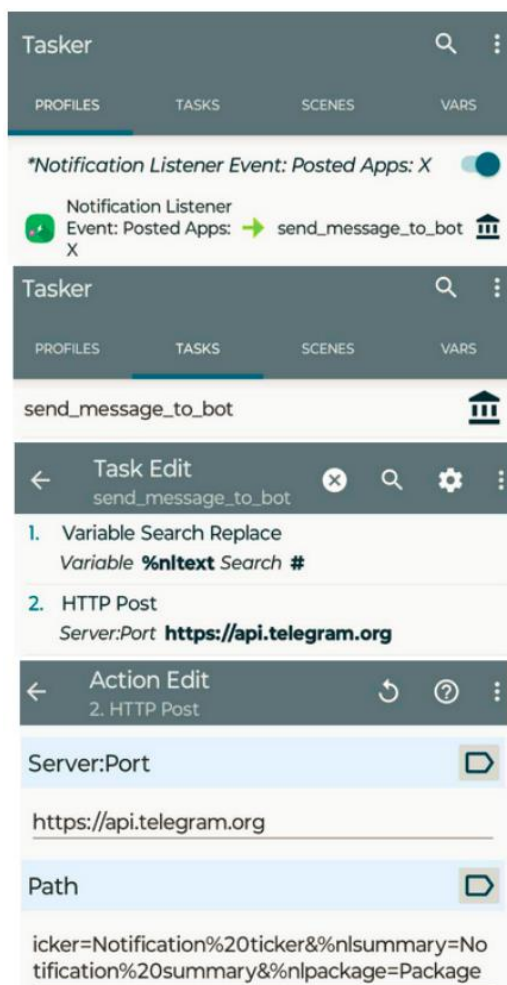
Tasker je pokročilá mobilní aplikace pro Android, která umožňuje automatizovat širokou škálu úkolů na mobilním zařízení. Uživatelé mohou nastavit různé podmínky, při jejichž splnění se spustí předem definované akce [55]. Tyto podmínky mohou být založeny na čase, poloze, stavu zařízení, událostech nebo použití konkrétních aplikací. [54]



Obr. 29: Logo aplikace Tasker [56]

Tahle aplikace byla použita pro automatické přeposílání oznámení od aplikace twitter. Tedy ve chvíli, kdy na twitteru vznikne tweet, mně do telefonu přijde oznámení s daným tweetem, které se ihned přepoše dále.

Stručně řečeno byl v aplikaci vytvořen profil jedním úkolem, který obsahuje dvě funkce. Jedna funkce zajišťuje přeposlání zprávy pomocí url odkazu skrze bota na Telegramu, druhá funkce zprávu upravuje do přeposlatelného tvaru. Nicméně cílem téhle práce není rozbrat práci s aplikací Tasker. Nastavení a mechanika přeposílání zpráv by vystačila minimálně na jednu kapitolu navíc, na což tu nyní není prostor. Následující obrázek ukazuje ukázkou z prostředí Tasker a nastavení přeposílání zpráv.



Obr. 30: Ukázka nastavení systému v aplikaci Tasker

#### ❖ Telegram: vlastní bot Tanya

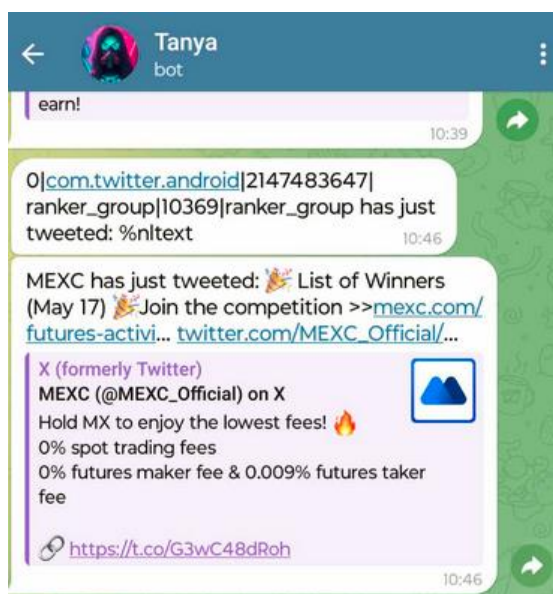
Aby mi mohly na Telegram přicházet zprávy skrze Tasker, potřebuji vlastního bota, skrze kterého by to bylo realizovatelné. Vytváření botů na Telegramu je relativně intuitivní záležitost. Stačí napsat zprávu účtu BotFather, který s tvorbou bota pomůže.



Obr. 31: Vytvoření Telegram bota

Bot byl pojmenován „Tanya“. Současně s vytvořením bota uživatel získá i jeho tzv. TOKEN, pomocí kterého může daného bota ovládat například skrze Python, pomocí kterého uživatel může bota dále konfigurovat.

Takový TOKEN vypadá vzhledově podobně jak API klíč a v podstatě tak i funguje. A právě tohoto tokenu bylo využito v Aplikaci Tasker, kde byl vložen do cesty pro přeposílání zprávy. Takže v tuhle chvíli oznámení z Twitteru přicházející na telefon vypisuje bot do chatu na Telegramu.



Obr. 32: Ukázka zpráv z twitteru přeposílaných skrze bota

#### ❖ Telegram: Vestavěný bot na přeposílání zpráv

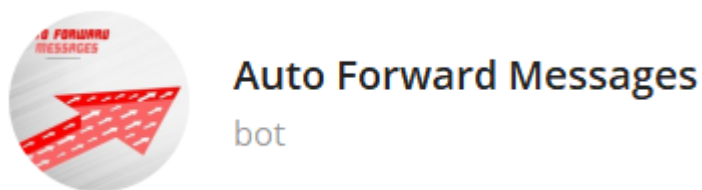
V tuhle chvíli by bylo ideální, kdyby bot mohl „načítat“ vlastní zprávy, které by dále zpracovával. Tohle však telegramový bot nezvládne, nemá na to zkrátka funkce. Další variantou se nabízí přeposlat zprávu od tohoto bota přímo k jinému botovi, jenže tohle také nelze. Telegram zamítl botům vzájemné čtení zpráv, aby nevznikaly nekonečné smyčky. [57]

#### **Why doesn't my bot see messages from other bots?**

Bots talking to each other could potentially get stuck in unwelcome loops. To avoid this, we decided that bots will not be able to see messages from other bots regardless of mode.

Obr. 33: Vysvětlení proč mezi sebou boti nemohou na telegramu interagovat

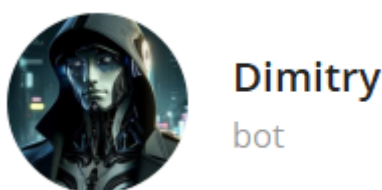
Jenže, existuje vestavěný bot na Telegramu, který má funkci přeposílání zpráv skrze uživatelův osobní profil. Jmenuje se „Auto Forward Messages“. Tohoto bota lze nakonfigurovat, aby přijaté zprávy od uživatelem určených účtů přeposílal prostřednictvím uživatele účtu do chatů jiných. Takže tenhle bot byl využit jako prostředník mezi botem Tanya a dalším botem, který má název „Dimitry“ a který již bude moci přijatou zprávu „uchopit“ a dále s ní pracovat.



Obr. 34: Interní bot na Telegramu

#### ❖ Telegram: Vlastní bot Dimitry

Jak již bylo uvedeno v předešlém odstavci, náš kanál přeposílání zpráv dovedl zprávu k botu Dimitry. Tenhle bot je již nakonfigurovaný pomocí Pythonu, přijímá zprávu, upravuje ji do správného tvaru a nechává neuronovou síť určit kategorii přijaté zprávy.



Obr. 35: Dimitry\_bot

#### Krok 6: Zpracování zpráv z Twitteru v reálném čase

Nyní je tedy zpráva u Dimitryho. V následujících odstavcích popisují skript, na základě kterého Dimitry funguje. Hlavní soubor má název *Dimitry\_bot.py*, soubor s funkcemi pro neuronovou síť se jmenuje *Dimitry\_neuron\_classification.py*.

Na začátku skriptu jsou importovány potřebné knihovny a moduly, včetně telegram a telegram.ext pro práci s Telegram Bot API, a dva vlastní moduly *Dimitry\_neuron\_classification* a *Dimitry\_token*.

```

1 from typing import Final
2 from telegram import Update
3 from telegram.ext import Application, CommandHandler, MessageHandler, filters, ContextTypes, CallbackContext
4
5 import Dimitry_neuron_classification
6 import Dimitry_token

```

Obr. 36: Import potřebných knihoven

Konstanty `TOKEN` a `BOT_USERNAME` jsou načteny z modulu *Dimitry\_token*, který obsahuje klíč bota a jeho jméno.

```

8 # Dimitry
9 TOKEN: Final = Dimitry_token.dimitry_key
10 BOT_USERNAME: Final = Dimitry_token.dimitry_name

```

Obr. 37: Dimitryho klíče

Bot má definované dva příkazy: `custom_command` a `start_command`. Oba příkazy posílají jednoduchou textovou odpověď. Jde o příkazy, které se objeví při otevření chatu s botem v telegramu. V tuhle chvíli jsou zde pouze pro otestování funkčnosti bota, aby se jednoduše zjistilo, jestli funguje. Používám zde asynchronní funkce.

Asynchronní funkce (také známé jako async funkce) umožňují, aby program pokračoval v běhu i během čekání na dokončení některých operací, což zlepšuje efektivitu a výkon aplikace. Jsou zvláště užitečné při provádění úloh, které by mohly trvat delší dobu, aniž by blokovaly hlavní běh programu. Asynchronní funkce jsou definovány pomocí klíčového slova `async` a jejich volání je obvykle prováděno pomocí klíčového slova `await` [58]. Když se taková funkce setká s `await`, řízení se může předat zpět do hlavní smyčky událostí (event loop), což umožňuje provádění jiných úloh během čekání na dokončení operace.

```

12 # Commands
13 async def custom_command(update: Update, context: ContextTypes.DEFAULT_TYPE):
14     await update.message.reply_text('Light talk.')
15
16 async def start_command(update: Update, context: CallbackContext) -> None:
17     await update.message.reply_text('Hey.')
```

Obr. 38: Funkce pro otestování funkčnosti Dimitry bota

Dále zde je nachystaná funkce `handle_response`. Tahle funkce zatím není využita pro funkci bota. Pouze je zde zatím ponechána z principu připravena, kdyby se někdy hodila. Tahle funkce by obstarávala důmyslnou odpověď bota na nějaký vstup. Nicméně v tuhle chvíli něco takového nebude potřeba, pouze bude stačit, aby bot přiřadil přijatému textu číslo.

```

20 def handle_response(text: str) -> str: #Když přijme string, vrátí string
21     processed: str = text#.lower() #nebude rozdíl mezi velkým a malým písmenem
22
23     return
```

Obr. 39: Funkce připravená pro práci budoucí

Důležitá je následující funkce `handle_message`. Zde se provádí přiřazení třídy textu. Dovolím si nyní několik řádků skriptu přeskočit, prakticky se tam pouze provádí některé operace pro správnou přípravu textu pro následné předání neuronové síti. Podívejme se na tu zajímavou část kódu, kde je řešena klasifikace textu:

```

48 # Zpracování zprávy neuronovou sítí
49 class_prediction = Dimitry_neuron_classification.message_classification(text)
50
51 if class_prediction == 1:
52     print('Zpráva je bullish')
53     target_chat_id = Dimitry_token.Bullish_Messages # Bullish Messages
54     await context.bot.send_message(chat_id=target_chat_id, text= text)
55
56 elif class_prediction == 2:
57     print('Zpráva je bearish')
58     target_chat_id = Dimitry_token.Bearish_Messages # Bearish Messages
59     await context.bot.send_message(chat_id=target_chat_id, text= text)
60
61 elif class_prediction == 0:
62     print('Zpráva je Neutrální')
63     target_chat_id = Dimitry_token.Neutral_Messages # Neutral Messages
64     await context.bot.send_message(chat_id=target_chat_id, text= text)
```

Obr. 40: Část kódu s určením ohodnocení a tříděním zpráv

Nejprve je zavolán soubor `Dimitry_neuron_classification` a funkce v něm `message_classification`, do které je vložen přijatý text. Tahle funkce zařídí použití natrénované neuronové sítě a vyhodnotí text. Vrábí nám hodnotu:

- 1 ... pokud je zpráva pozitivní
- 2 ... pokud je zpráva negativní
- 0 ... pokud je zpráva neutrální

#### ❖ Telegram: Chaty pro dané typy zpráv

Po vyhodnocení zprávy, jestli je neutrální, pozitivní nebo negativní, pošle Dimitry zprávu do jednoho ze tří chatů na Telegramu, určeného pro daný typ zprávy. Byly vytvořeny tři chaty:

- Bullish Messages pro pozitivní zprávy
- Bearish Messages pro negativní zprávy
- Neutral Messages pro neutrální zprávy



Obr. 41: Loga chatů pro Bullish, Neutrální a Bearish zprávy

Na konci skriptu je potom hlavní funkce, která zajišťuje že je Dimitry bot stále aktivní. Na začátku tohoto úseku se inicializuje aplikace, registrují příkazy handlerů a handler na zpracování přijatých zpráv, dále handler chyb a úplně na konci se spouští nekonečná smyčka, která každé 3 sekundy kontroluje nové zprávy a události na Telegramu.

```

78 if __name__ == '__main__':
79     print('Starting bot ...')
80     app = Application.builder().token(TOKEN).build()
81
82     # Commands
83     app.add_handler(CommandHandler('custom', custom_command))
84     app.add_handler(CommandHandler('start', start_command))
85
86     # Messages
87     app.add_handler(MessageHandler(filters.CHAT, handle_message)) # tohle se postará o zprávy
88
89     # Errors
90     app.add_error_handler(error)
91
92     # Stále kontrolovat updaty na nové zprávy
93     print('polling') # ujistíme se, že bot běží
94     app.run_polling(poll_interval=3) # kontroluje chat každých x sekund (poll_interval=3)
95

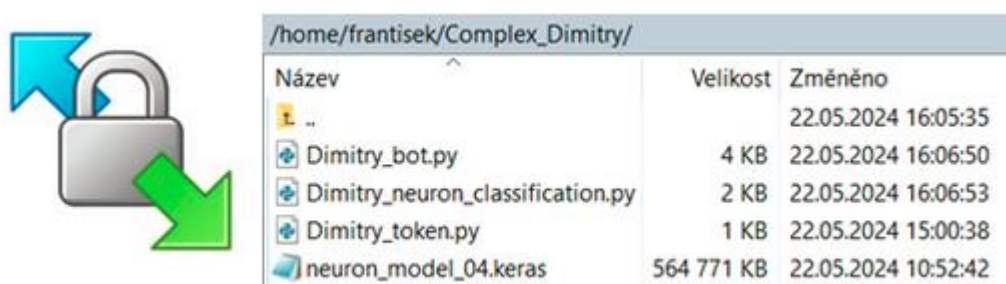
```

Obr. 42: Zajištění vytrvalého běhu bota

### Krok 7: Zajištění, aby Dimitry a celý systém fungoval nepřetržitě

Pro efektivní sběr a vyhodnocování přijatých zpráv je důležité, aby Dimitry pracoval neustále, a proto byl umístěn na server, čímž je taková činnost umožněna. Zde náleží díky mému otci, který mi pro tenhle i ostatní úkoly poskytl jeho domácí servery.

Nejprve je potřeba všechny potřebné soubory nahrát na server, k tomu byl použit program WinSCP. WinSCP je open-source program pro Windows, který umožňuje bezpečné kopírování souborů mezi lokálním počítačem a vzdálenými servery pomocí různých protokolů, jako jsou SFTP, SCP, FTP, FTPS, WebDAV a Amazon S3. [59]



Obr. 43: Logo software WinSCP a uložení souborů na server

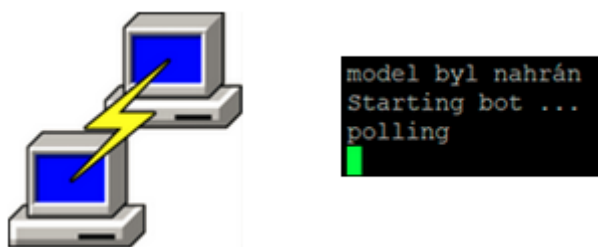
Pro připojení k serveru byl poté použit software PuTTY. PuTTY je volně dostupný a open-source software, který slouží jako terminálový emulátor, sériová konzole a aplikace pro přenos souborů po síti. Původně byl vyvinut pro platformu Windows, ale díky své popularitě byl přizpůsoben i pro další operační systémy, jako jsou macOS a Linux [60].

Protože otcovy servery využívají Linux, je nutné tomu pythonové soubory přizpůsobit. To se zařídí tak, že se u každého souboru nastaví jeho formát na unixový, pomocí příkazu „set ff=unix“. Díky tomu bude zajištěno správné řádkování a kód tak bude moci v linuxovém prostředí správně fungovat.

Dále je důležité na začátek každého skriptu připsat následující řádek, který zajistí, že systém bude používat správnou verzi Pythonu:

```
#!/usr/bin/env python3
```

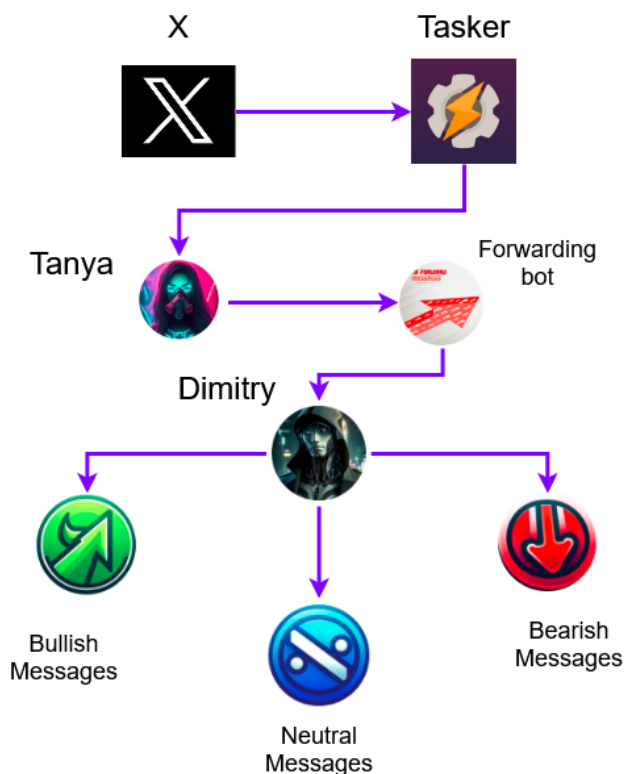
Zbývá tedy bota Dimitry spustit a sledovat výsledky.



Obr. 44: Logo PuTTY a spuštění bota an serveru

### Krok 8: Shrnutí systému pro sledování a analyzování fundamentálních zpráv

Je tedy vytvořen systém pro sledování, ohodnocování a třídění zpráv. Následující schéma popisuje celý proces od vzniku zprávy na Twitteru až po její zařazení do příslušného chatu:



Obr. 45: Cesta tweetu od jeho vzniku

#### 4.2.2 Výsledky vyhodnocování fundamentálních zpráv

Podívejme se nyní, jak si celý systém pro vyhodnocování Fundamentálních zpráv vedl. Chatovací skupiny fungují, zprávy od bota Dimitry vypadají následovně:

##### ❖ Bullish Chat



Obr. 46: Ukázka zpráv přiřazených do Bullish chatu

### Komentář:

První tweet uvedený na obrázku hovoří o technologickém updatu ohledně projektu Hedera. To je jednoznačně pozitivní zpráva, a tedy právem náleží do Bullish Messages chatu.

Druhá zpráva hovoří o tom, že by návrh nového zákona FIT 21, který má stanovit regulační rámec pro krypto, mohl mít negativní dopad na „běžné investory, kteří se snaží vybudovat bohatství.“ Z obsahu zprávy nelze jednoznačně určit, jestli je vážné nebo ironické povahy.

Každopádně vzhledem k okolnostem se spíše domnívám že jde o ironický obsah, vzhledem k tomu že poslední dny se kryptu spíše daří, stále další státy přijímají BTC a ETH ETF apod. V tomhle případě by zpráva mohla být považována za pozitivní, protože se regulace okolo kryptoměn uvolňuje.

Třetí zpráva informuje o nárůstu ceny Bitcoinu za posledních 7 dní. Je to pěkné, je to relativně hodně procent za krátký čas, nicméně tuhle zprávu považuji spíše za neutrální.

### ❖ Neutral Chat



Obr. 47: Ukázka zpráv přiřazených do Neutrálního chatu

### Komentář:

První zpráva je dle mého na pomezí pozitivní/neutrální. Grayscale je na trhu kryptoměn i obecně na světě velký hráč. Na jednu stranu je pozitivní, že se jeho vztah k ETF Etherea nějak vyvíjí, na druhou stranu je pravda že jde spíše o zprávu informativního charakteru.

Druhá zpráva jednoznačně informuje o průběhu nějakého podcastu, správně je tedy v chatu s neutrálními zprávami.

Třetí zpráva informuje že transakce protokolu Hedera probíhají tak jak mají, to je samozřejmě pozitivní, nicméně v našem kontextu jde o zprávu neutrální.

❖ **Bearish chat**

Obr. 48: Ukázka zpráv přiřazených do Bearish chatu

Komentář:

První zpráva není negativní, spíše bych ji označil za neutrální.

Druhá zpráva je jednak komická, protože vysvětluje, že pokud má člověk modré vlasy a není si jist svým genderem, tak krypto \$MAGA nejspíše nebude pro něj, a jednak tedy snad oprávněně v chatu s negativními zprávami, lidé s modrými vlasy si totiž na \$MAGA zkratka nejspíše nesáhnou ...

Třetí zpráva je sice negativní, protože se tam píše o tom, že Joe Biden bude označen za podvodníka, pokud začne přijímat krypto příspěvky. Takle zpráva obecně je negativní, nicméně ne pro kryptotrž. V našem kontextu bych ji spíše označil za neutrální.

Shrnutí:

Na základě zpráv rozdělených do příslušných chatů lze konstatovat, že trend rozhodování neuronové sítě je přibližně správným směrem. Někdy se stane, jednoznačně určí zprávu špatně, nicméně ve většině případech lze rozdělení považovat za adekvátní.

Vzhledem k povaze trénovacího datasetu považuji dosavadní výsledky za úspěch. Každopádně lze vidět nedostatky systému. Pokud má být skutečně kvalitní a robustní, potřebuje být neuronová síť natrénována na více kvalitnějších datech. Kromě toho zůstává jako úkol do budoucna vytvořit novou kategorii, kam by neuronová síť posílala skutečné „perly“, které v krátkém čase způsobí prudký nárůst nebo pokles ceny daného krypta.

4.2.3 **Automatizované obchodování na burze na základě technické analýzy**

Tahle kapitola se zabývá implementací umělé inteligence při obchodování na burze na základě technických parametrů. Strategií na obchodování je nespočet, stejně tak indikátorů, podle kterých lze rozhodovat. Nakonec bylo započato s něčím relativně primitivním, leč zajímavým na implementaci.

Zvolená strategie je následující: Natrénovat neuronovou síť, naučit ji, jak vypadají svíčky poměrově vzhledem k sobě vzájemně v určitém časovém úseku a s přihlédnutím k aktuální hodnotě RSI v časovém horizontu pěti timeframů odhadnout změnu trendu a na základě tohoto odhadu vytvořit obchodní pozice.

Cílem je zajistit co nejstabilnější růst portfolia. Pojdme si nejprve vysvětlit, co navržená strategie obsahuje.

### Vysvětlení technikálií:

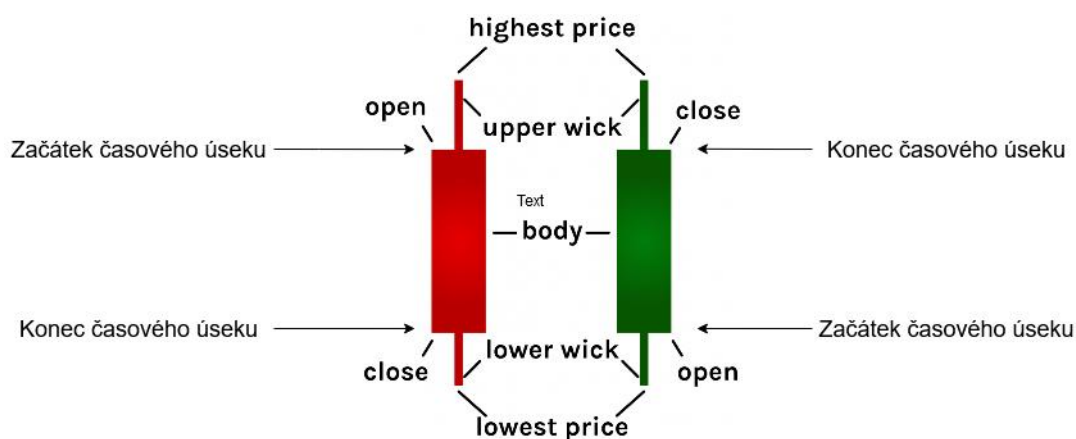
#### a) Svíčky

Svíčky (nebo také svícný, anglicky "candlesticks") jsou vizuální nástroje používané v technické analýze k zobrazení cenového pohybu aktiv, jako jsou akcie, komodity nebo měny, v určitém časovém období.

Každá svíčka reprezentuje pohyb cen během jednoho časového úseku a skládá se ze čtyř základních hodnot: otevírací cena (open), nejvyšší cena (high), nejnižší cena (low) a zavírací cena (close). Svíčky poskytují detailní informace o cenovém chování a psychologii trhu v daném časovém úseku. Různé formace svíček a jejich vzory mohou obchodníkům pomoci identifikovat možné budoucí pohyby cen. [61]

### Struktura svíčky:

1. **Tělo svíčky (Body):** Tělo svíčky je část mezi otevírací a zavírací cenou. Pokud je zavírací cena vyšší než otevírací, tělo svíčky je obvykle zelené (nebo bílé), což signalizuje růst ceny. Pokud je zavírací cena nižší než otevírací, tělo svíčky je červené (nebo černé), což signalizuje pokles ceny.
2. **Knoty (Wicks) nebo stíny (Shadows):** Vertikální čáry nad a pod tělem svíčky představují nejvyšší a nejnižší cenu v daném časovém období. Horní knot ukazuje nejvyšší dosaženou cenu, zatímco spodní knot ukazuje nejnižší cenu.
3. **Barva:** Barva těla svíčky pomáhá rychle identifikovat, zda cena rostla (zelená) nebo klesala (červená) během daného období.



Obr. 49: Vysvětlení svíček v kontextu pohybů ceny aktiva [62]



Obr. 50: Pohyb ceny aktiva v podobě svíček [63]

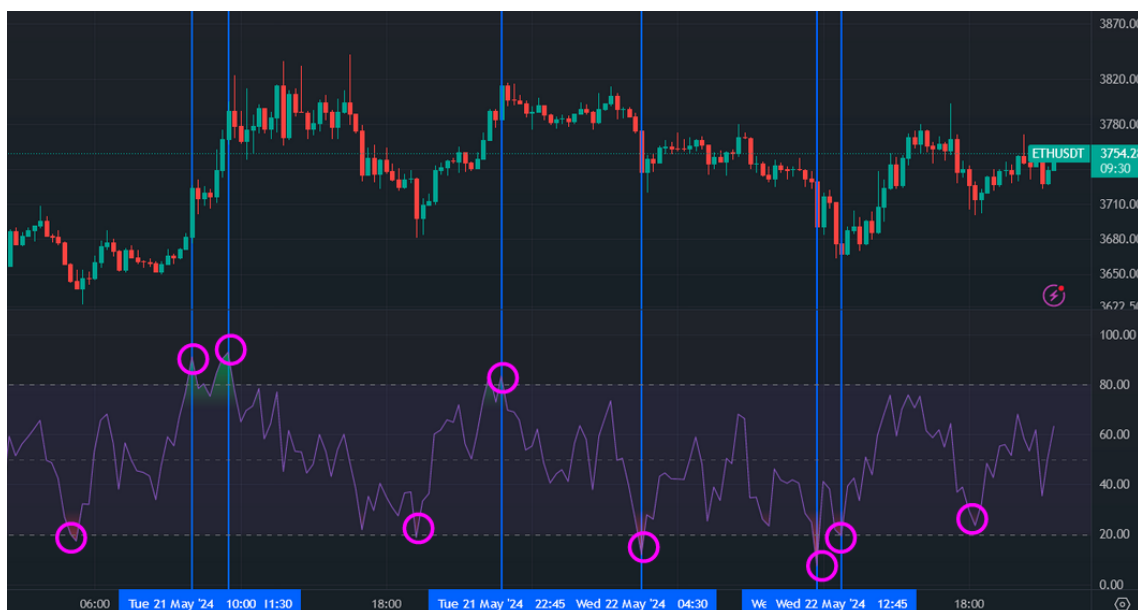
#### b) Indikátor RSI

Indikátor RSI (Relative Strength Index) je momentum oscilátor vyvinutý J. Wellesem Wilderem, který měří rychlost a změnu cenových pohybů. RSI se pohybuje mezi hodnotami 0 a 100 a jeho hlavním cílem je identifikovat podmínky překoupenosti a přeprodanosti na trhu.

- Hodnoty nad 70 jsou obvykle považovány za překoupené, což může signalizovat možný obrat směrem dolů.
- Hodnoty pod 30 jsou považovány za přeprodané, což může signalizovat možný obrat směrem nahoru. [64]

Ve zvolené strategii je zamýšleno využít RSI s délkou 5 časových horizontů. Když je délka RSI nastavena na 5, znamená to, že RSI je vypočítáváno na základě průměrného zisku a ztráty za posledních 5 časových období.

V následujícím obrázku vidíme v horní části časový průběh ceny Ethera, kde jedna svíčka představuje časový horizont 15 minut. V dolní části vidíme časový průběh RSI Ethera s délkou 5 horizontů. Všiměme si lokálních maxim a minim RSI a jak se chovala cena Ethera v těchto oblastech. Je vidno, že po dosažení maxima překoupení nastává obrat trendu, tedy cena jde pak trochu dolů. U relativního podkoupení je tomu naopak, tam nastává nárůst ceny.



Obr. 51: Ukázka vztahu indikátoru RSI a pohybu ceny ETH [63]

Tím by byly vysvětleny technické indikátory, se kterými budu dále pracovat.

### **Realizace obchodního systému:**

#### **Krok 1: Sběr dat a příprava datasetu**

Pro sběr dat a přípravu datasetu k trénování byl vytvořen skript s názvem `dataset_preparation_v3.py`. V tomhle kroku je cílem získat historické hodnoty open, high, low a close vždy tří posledních svíček relativně vzhledem k aktuální svíčce. Následující obrázek lépe popisuje tuhle myšlenku:



Obr. 52: Vysvětlení relativního ohodnocení významných údajů svíček

High aktuální svíčky má hodnotu „1“. Low aktuální svíčky má hodnotu „0“. Hodnoty open a close jsou někde mezi tím. Hodnoty low, high, open a close předešlých svíček jsou vypočítány poměrově vzhledem k první svíčce. Všiměme si, že high svíčky úplně vlevo má hodnotu 2,30. Když se podíváme, jak široké rozpětí je od hodnoty „0“ k hodnotě „1“, vidíme že ohodnocení 2,30 je přibližně adekvátní. Vzdálenost od „0“ k „2,30“ je minimálně dvakrát tak velká jako vzdálenost od „0“ k „1“.

Tímhle způsobem lze číselně vyjádřit vzhled svíček. Výhoda tohoto označení je univerzálnost použití. Ať už bude systém vyhodnocovat jakýkoliv graf kteréhokoliv aktiva, svíčky tam budou velikostně poměrově mezi sebou vždy podobné, nehelede na skutečnost, jestli se hodnota aktiva pohybuje v rámci stovek nebo desetitisíců.

Jako datové aktivum pro sběr údajů volím pár BTC/USDT, protože právě Bitcoin je nejstarší kryptoměna, bude mít nejvíce dat k dispozici. Nejlépe by bylo sehnat data úplně až od vzniku Bitcoinu, jenomže data burzy Binance sahají pouze do roku 2019. To je však postačující, důležité je mít v rámci dat obsažen aspoň jeden bullrun, kde se vyskytuje jak uptrend, tak i downtrend. Časové rozpětí jedné svíčky volím 15 minut.

Dále si přeji zjistit hodnotu RSI každého timeframu. Nyní stanovím, za jakých podmínek bych si přál, aby můj budoucí bot nakoupil, otevřel pozici long a za jakých podmínek si přeji, aby otevíral pozici short, tedy vydělal na tom, že cena půjde dolů. Podmínky pro pozici long jsou následující:

1. Aktuální hodnota RSI je menší jak 30.
2. Hodnota aktiva v následujících třech časových úsecích vzrostla aspoň o 0,3 %.
3. Žádné low následujících tří svíček není nižší než aktuální low.

... Je tedy nutné zjistit informace o low hodnotách následujících tří svíček a jak vypadá cena aktiva a jak to vypadá s cenou aktiva od nynějšího timeframu vzhledem ke třetímu následujícímu.

Podmínky pro pozici short pozici jsou následující:

1. Aktuální hodnota RSI je větší jak 70.
2. Hodnota aktiva v následujících třech intervalech klesla aspoň o 0,3 %.
3. Žádné high následujících tří svíček není vyšší než aktuální high.

Těmito podmínkami bych mohl odhadnout skutečnou změnu trendu a natrénovat neuronovou síť podle vzhledu svíček, které takovým změnám trendu předcházejí. Vzniklý soubor csv ze skriptu *dataset\_preparation\_v3.py* tedy obsahuje několik sloupců, jde o sloupce následující:

- ❖ Datum a čas
- ❖ Aktuální hodnota RSI
- ❖ Relativní hodnoty nynější svíčky a tři předchozích svíček
- ❖ Absolutní hodnoty high a low tří následujících svíček

- ❖ Procentuální změna aktiva za následující tři timeframy
- ❖ Vyhodnocení akce, co udělat. K tomu je vyhrazen sloupec WhatToDo, v tomhle sloupci mohou vystupovat hodnoty 0, 1 a 2:
  - Hodnota 0 představuje nečinnost. Není ani vhodná doba na nákup ani na prodej
  - Hodnota 1 představuje nákup, otevření pozice long
  - Hodnota 2 představuje prodej, otevření pozice short

#### Stručný popis kódu `dataset_preparation_v3.py`:

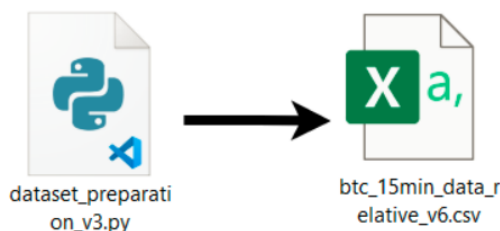
Nejprve jsou načteny potřebné knihovny a inicializovány API klíče pro přístup k Binance. Kód vytváří relaci s mechanismem opakování, aby zajistil spolehlivé načítání dat i při dočasných výpadech nebo chybách. Následně je inicializován klient pro komunikaci s Binance API a je mu přiřazena vlastní relace pro robustní požadavky na server.

Funkce `get_historical_klines` je zodpovědná za načtení historických dat pro zadaný symbol a interval z Binance. Data jsou načítána po blocích (každý až 500 řádků) a zpracována do pandas DataFrame. Časové značky jsou převedeny na datetime formát a nastaveny jako index.

Následně se vypočítává RSI (Relative Strength Index) pro každou svíčku pomocí knihovny `ta`, a relativní hodnoty (otevření, uzavření, maxima a minima) se normalizují vzhledem k aktuální svíčce. Dále se vypočítávají relativní hodnoty pro tři předchozí svíčky vůči aktuální svíčce.

Poté jsou vypočítány absolutní hodnoty pro následující tři svíčky a procentuální změna ceny mezi aktuální a třetí následující svíčkou. Na základě těchto výpočtů jsou definovány podmínky pro nákup a prodej: pokud RSI je menší než 30 a očekává se růst ceny, nebo pokud RSI je větší než 70 a očekává se pokles ceny, pak se v příslušných řádcích DataFrame nastaví hodnota sloupce WhatToDo na 1 (nákup) nebo 2 (prodej).

Na závěr jsou vyfiltrovány požadované sloupce, odstraněny řádky s chybějícími hodnotami a výsledný DataFrame je uložen do CSV souboru. Náš soubor má název `btc_15min_data_relative_v6.csv` a budeme s ním pracovat dále.



Obr. 53: Uložení historických, relativních dat svíček Bitcoinu

1	datetime	RSI	rel_open	rel_high	rel_low	rel_close
2	08.09.2019 19:00	100.0	0.0	1.0	0.0	1.0
3	08.09.2019 19:15	100.0	0.5105946684894493	1.0	0.0	0.8010936431989302
4	08.09.2019 19:30	96.25806538609308	0.862951807228938	1.0	0.0	0.4171686746987681
5	08.09.2019 19:45	93.49050526448492	0.8114663726570388	1.0	0.0	0.19184123484027646
6	08.09.2019 20:00	94.1904403248094	0.01480111008332637	1.0	0.0	0.8862164662349281
7	08.09.2019 20:15	89.63124157525189	0.6439333862013169	1.0	0.0	0.07850911974621222
8	08.09.2019 20:30	90.1866897502054	0.22283865002311085	1.0	0.0	0.5312066574202592
9	08.09.2019 20:45	84.20180713091263	0.9254538431826831	1.0	0.0	0.6519891850135024
10	08.09.2019 21:00	67.1837531499256	0.9100529100529311	1.0	0.0	0.14744268077596778
11	08.09.2019 21:15	71.67237181336148	0.2563499529632574	1.0	0.0	0.8744120413922839
12	08.09.2019 21:30	75.63536828728171	0.18698710433766733	1.0	0.0	0.9431418522860859
13	08.09.2019 21:45	83.57224565953904	0.02676133260512263	1.0	0.0	1.0
14	08.09.2019 22:00	85.94141343343868	0.3053359683794643	1.0	0.0	1.0
15	08.09.2019 22:15	72.35788254878192	0.7567567567567612	1.0	0.0	0.03783783783784625
16	08.09.2019 22:30	70.3930810179551	0.3285625388922795	1.0	0.0	0.0267579340385998
17	08.09.2019 22:45	58.09025073423428	1.0	1.0	0.0	0.0

Obr. 54: Ukázka tabulky získaných hodnot RSI a relativních údajů

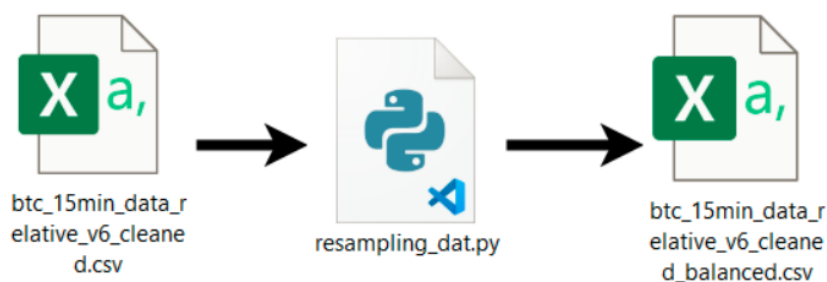
### Krok 2: Očištění souboru csv od nyní již nepotřebných informací a resampling dat

Nedává smysl neuronovou síť učit na informace budoucí. Ty nemůže v daný okamžik odniknut vyčíst, a tedy jsou zbytečné. Je tedy záhodno vzniklý csv soubor od informací ohledně budoucích hodnot high, low a procentuální změny aktiva očistit.

Dále je také potřeba odstranit datum. K tomuto účelu byl vytvořen skript `data_to_test.py`, který uvedené sloupce z původního souboru odstraní a získáváme nový, upravený soubor s názvem `btc_15min_data_relative_v6_cleaned.csv`. Můžeme si v něm všimnout, že většina řádků má ve sloupci WhatToDo ohodnocení „0“. Ohodnocení „1“ nebo „2“ se objevuje pouze zřídka. Již nyní je jasné, že je soubor nevyvážený co do počtu přiřazených kategorií.

Aby se neuronová síť správně natrénovala, je potřeba zařídit vyrovnanou četnost ohodnocení dat. Jak jsme si již uvedli v předchozí kapitole u kategorizování textu a neuronové sítě, i zde nyní použijí oversampling. K tomuto účelu byl vytvořen soubor `resampling_dat.py`. Tento kód se tedy snaží vyřešit problém s nevyváženými daty tím, že zvyšuje počet vzorků v menšinových třídách na úroveň majoritní třídy.

Výsledkem je nový csv soubor (`btc_15min_data_relative_v6_cleaned_balanced.csv`), kde jsou třídy již vyrovnané.



Obr. 55: Vyvážení dat s kategoriemi obsaženými v datasetu

### Krok 3: Natrénování neuronové sítě

Skript s trénováním neuronové sítě má název *neuron\_training\_a\_vahy.py*. Zde opět proběhne načtení csv souboru, data se rozdělí na trénovací, validační a testovací sady, Pandas dataframe se převede na Tensorflow dataset. Důležitý je potom následující úsek:

```
31 # Výpočet vah tříd
32 class_weights = {0: 1., 1: len(y) / (2 * y.value_counts()[1]), 2: len(y) / (2 * y.value_counts()[2])}
```

Obr. 56: Výpočet vah tříd

Tenhle řádek slouží k vyvážení dat v průběhu trénování modelu neuronové sítě. Vzhledem k tomu, že třídy v datasetu mohou být nerovnoměrně zastoupeny, váhy tříd umožňují modelu věnovat větší pozornost menšinovým třídám, což zvyšuje pravděpodobnost, že se model naučí rozpoznávat všechny třídy rovnoměrně. Přestože máme již více-méně vyvážený dataset, rozhodl jsem si vyváženost tříd ještě pojistit tímto způsobem.

- **class\_weights** je slovník, který přiřazuje váhy jednotlivým třídám.
- **0: 1.** znamená, že třída s hodnotou 0 má váhu 1. Tím se udržuje její význam v původní formě, protože je předpokládána jako většinová třída.
- **1: len(y) / (2 \* y.value\_counts()[1])** a **2: len(y) / (2 \* y.value\_counts()[2])** počítají váhy pro třídy 1 a 2. Tyto váhy se počítají tak, že celkový počet vzorků (len(y)) se dělí dvojnásobkem počtu vzorků dané třídy. Tímto způsobem se váhy pro třídy 1 a 2 zvyšují, aby se vyrovnal jejich menší počet ve srovnání s třídou 0.

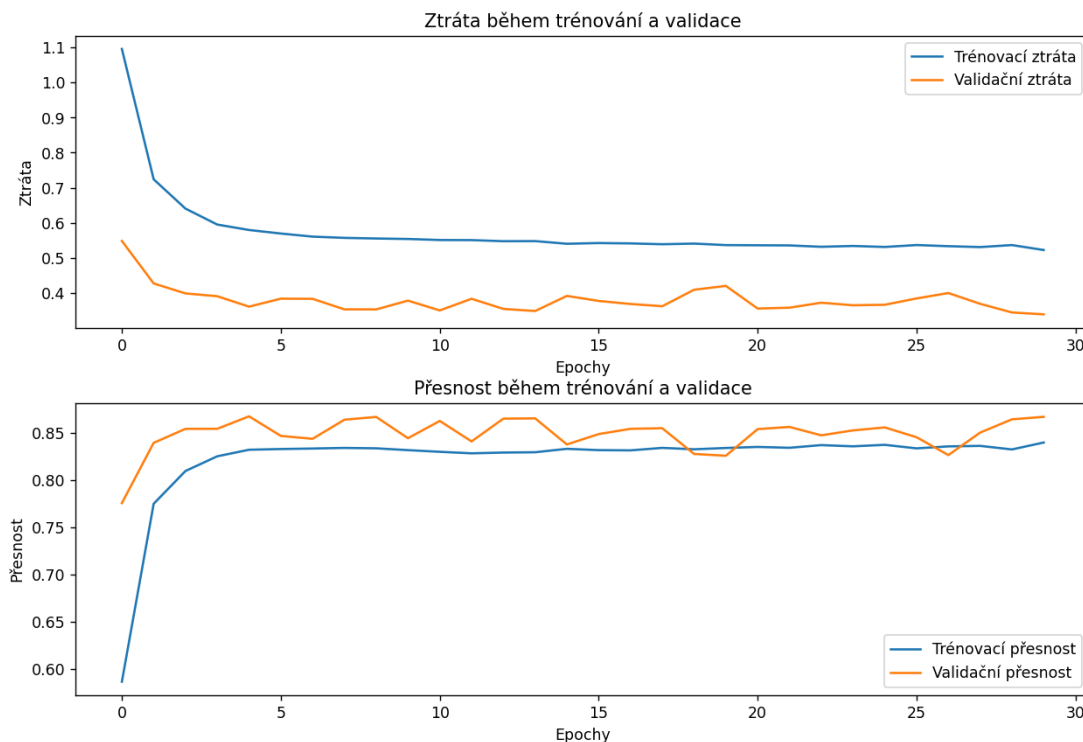
Dále pak proběhne definice modelu a trénování, jak to již známe z předchozího obsahu. Bylo odzkoušeno několik různých kombinací parametrů `batch_size`, počet epoch, vrstev neuronové sítě apod, nicméně nejlepších dosažených výsledků bylo dosaženo následující kombinací trénovacích parametrů:

- `Batch_size = 128`
- `Epochs = 30`
- A následujícím modelem sítě:

```
34 # Definice modelu
35 model = tf.keras.Sequential([
36     tf.keras.layers.Dense(64, activation='relu'),
37     tf.keras.layers.Dropout(0.4),
38     tf.keras.layers.Dense(32, activation='relu'),
39     tf.keras.layers.Dropout(0.4),
40     tf.keras.layers.Dense(16, activation='relu'),
41     tf.keras.layers.Dropout(0.4),
42     tf.keras.layers.Dense(8, activation='tanh'),
43     tf.keras.layers.Dropout(0.4),
44     tf.keras.layers.Dense(3, activation='softmax')
45 ])
```

Obr. 57: Struktura neuronové sítě pro technickou analýzu

Výsledky trénování jsou následující:



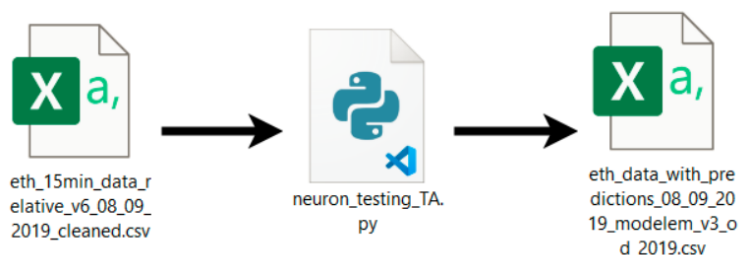
Obr. 58: Výsledky trénování NN pro technickou analýzu

Přesnost 87 % a ztráta 35 %. Lepších výsledků se doposud nepodařilo docílit. Záležitost dosažení lepší přesnosti a nižší ztráty pro nyní zůstává úkolem budoucím. Při zkoušení různých kombinací parametrů pro trénování bylo zajímavé například zjištění, že se zvýšením proměnné `batch_size` klesá přesnost a zvyšuje se ztráta.

#### Krok 4: Testování neuronové sítě na historických datech a Vyhodnocení

Nyní se pojděme podívat, jak by si naše natrénovaná neuronová síť vedla, kdybychom ji spustili v minulosti a nechali ji obchodovat od roku 2019. Neuronová síť byla trénována na datech Bitcoinu, bylo by zajímavé otestovat ji na jiném obchodním páru, zkusme tedy ETH/USDT. K tomu bylo zapotřebí získat soubor csv s historickými údaji o svíčkách páru ETH/USDT a hodnotami RSI.

K získání těchto údajů bylo využito předešlých souborů, kde byl tedy soubor očištěn nyní navíc i o sloupec „WhatToDo“. Za účelem získání predikcí neuronové sítě u ETH/USDT byl vytvořen soubor `neuron_testing_TA.py`. Tenhle soubor načte historická data ETH/USDT, použije neuronovou síť a vytvoří nový csv soubor, který již obsahuje rozdělení jednotlivých časových úseků do tříd.



Obr. 59: Příprava dat Ethera k prověření vytvořené neuronové sítě

Nyní tedy máme soubor, který je rozdělen do tříd. Stojíme před otázkou, za jakých podmínek bychom si přáli vybrat profit a za jakých okolností bychom přiznali ztrátu. Dostáváme se tedy k risk managementu. Otázkou je, jak by měl být takový risk management nastaven, aby byl náš budoucí bot co nejvíce profitový?

Za účelem vyhodnocení nejlepšího risk managementu byl vytvořen kód s názvem *vypocet\_uspesnosti.py*. Podstatou tohoto skriptu je vyzkoušet několik různých kombinací parametrů pro výběr profitu nebo přiznání ztráty. Podívejme se na příklad možné kombinace takovýchto parametrů:

- take\_profit = 0,006
- stop\_loss = 0,002

V takovém případě se bude dít následující: Pokud se cena od zadání obchodní pozice dříve posune ve směru predikce o 0,6 %, taková pozice se uzavře a připíše se zisk 0,6 %. Ovšem pokud se cena dříve posune o 0,2 % proti směru naší predikce, pozice se uzavře a nám se připíše ztráta 0,2 %. Následující dva řádky specifikují, jaké všechny hodnoty nám vstupují do kombinace:

```

140 # Definování rozsahů pro stop-loss a take-profit
141 take_profit_values = [0.002, 0.003, 0.004, 0.005, 0.006, 0.007]
142 stop_loss_values = [0.001, 0.002, 0.003, 0.004, 0.005]
  
```

Obr. 60: Kombinace parametrů take\_profit a stop\_loss

Kód každou uvedenou kombinaci vezme, projde s ní všechny predikce vytvořené neuronovou sítí, vykreslí graf vývoje portfolia a vytvoří csv soubor s výsledky obchodů.

V téhle části práce bylo obzvláště zapeklité správné uchopení času vzniku pozice a času realizace zisku nebo ztráty. Pro skutečně správné výsledky bylo nutné získat co nejjemnější časové intervaly historie vývoje ceny ETH. Jedině tímto způsobem je možné co nejpřesněji určit úspěšnost obchodování. Pro zisk jednodominutových časových intervalů byl vytvořen skript s názvem *data\_1min\_save.py*. tenhle skript načítá a ukládá všechny dostupné hodnoty svíček s jedno-minutovým časovým intervalem až do datumu 11.05.2020, kam tahle historie ETH sahá.

Soubor *vypocet\_uspesnosti.py* funguje tak, že se postupně zjišťuje zadané obchodní pozice a u každé z nich potom sleduje jemné jednodominutové časové horizonty

a zjišťuje, jestli bylo dříve dosaženo hodnoty také `take_profit` nebo `stop_loss`. Následující tabulka ukazuje výsledky vyhodnocování optimálního risk managementu:

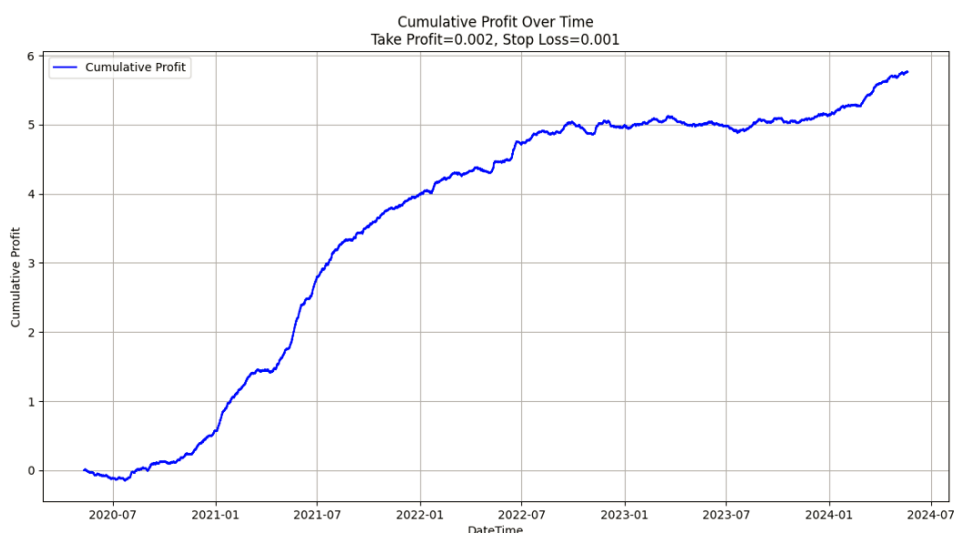
1	Column1	Column2	Column3	Column4
2	take_profit	stop_loss	success_rate	total_profit
3	0.002	0.001	0.38029477183291377	5.764
4	0.002	0.002	0.5296116148901328	4.846
5	0.002	0.003	0.6228338181018258	4.671
6	0.002	0.004	0.6865543959132794	4.882
7	0.002	0.005	0.7338987607850805	5.617000000000001
8	0.003	0.001	0.2802043360301127	4.9430000000000005
9	0.003	0.002	0.42201745166572974	4.504
10	0.003	0.003	0.5163395497763547	4.011
11	0.003	0.004	0.5860484442597708	4.1869999999999985
12	0.003	0.005	0.6399921785251631	4.9069999999999999
13	0.004	0.001	0.21861022168992741	3.8069999999999999
14	0.004	0.002	0.3458558404419133	3.0740000000000003
15	0.004	0.003	0.43619387480751837	2.1830000000000002
16	0.004	0.004	0.5058050008554739	1.9000000000000004
17	0.004	0.005	0.5623395986605725	2.4979999999999993
18	0.005	0.001	0.17847627893334636	2.8989999999999996
19	0.005	0.002	0.29286534842226186	2.0480000000000005
20	0.005	0.003	0.37728839244249995	0.7490000000000017
21	0.005	0.004	0.44462640236599615	0.06700000000000064
22	0.005	0.005	0.500721042211522	0.2950000000000004
23	0.006	0.001	0.1516877276171388	2.5289999999999995
24	0.006	0.002	0.2551267323344658	1.6780000000000002
25	0.006	0.003	0.33361034390047173	0.10200000000000223
26	0.006	0.004	0.3970131742966783	-1.2219999999999998
27	0.006	0.005	0.45125021386845254	-1.4780000000000004
28	0.007	0.001	0.13113191406154523	2.0069999999999983
29	0.007	0.002	0.22584508591401267	1.334
30	0.007	0.003	0.29824261237259553	-0.7190000000000003
31	0.007	0.004	0.35834575807200647	-2.3810000000000007
32	0.007	0.005	0.4114584606359837	-2.5520000000000014

Obr. 61: Výsledky úspěšnosti kombinací parametrů `take_profit` a `stop_loss`

Z hodnot z tabulky vyplývá, že největšího profitu by člověk dosáhl při následujícím risk managementu (RM):

- `take_profit` = 0,002
- `stop_loss` = 0,001

Přestože úspěšnost takového RM je pouze 38 %, zisk je z celé tabulky nejvyšší, tedy 576 %. Podívejme se nyní na průběh takovýchto obchodů:



Obr. 62: Graf vývoje portfolia při nejprofitovějším RM

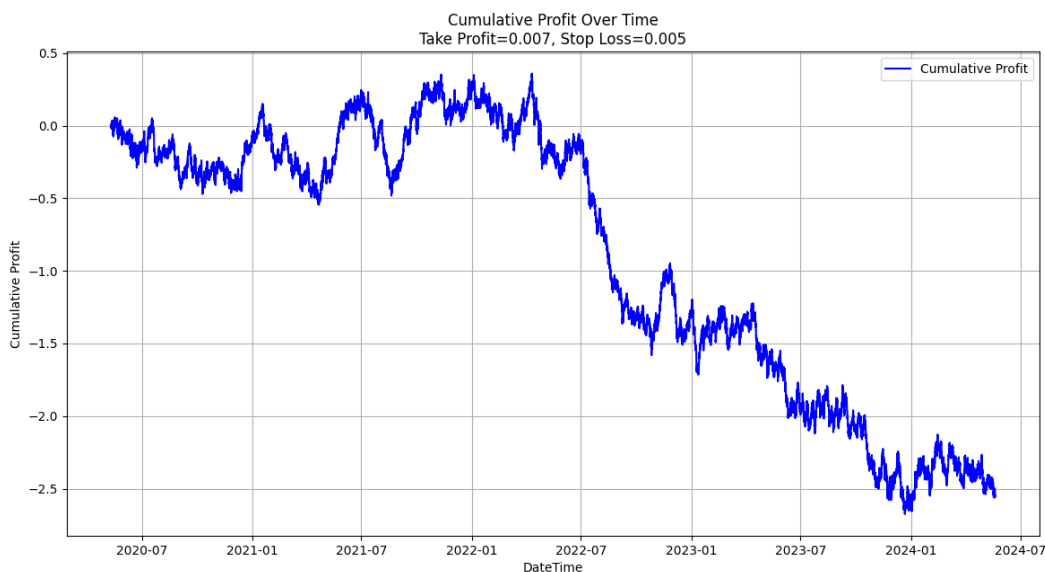
Z grafu je patrné, že volatilita celkové hodnoty portfolia je relativně nízká. Na začátku klesla hodnota portfolia až k 85 % jeho původní hodnoty, do konce roku 2020 se však hodnota vyšplhala o 50 % vzhůru a od té doby až do současného okamžiku by se původní hodnota portfolia znásobila 5,8násobně. Při takových číslech je nutné výsledky zkontrolovat „na vlastní oči“.

Je tedy k dispozici soubor `eth_results_tp_0.002_sl_0.001.csv`, kde jsou vypsané výsledky této obchodní strategie. Pomocí Tradingview bylo zkontrolováno posledních 20 obchodů, jestli by se skutečně realizovaly podle vyhodnocení skriptu. Bylo zjištěno že obchody skutečně sedí. Příkazy `take_profit` a `stop_loss` by se skutečně realizovaly podle tabulky.



Obr. 63: Prověření výsledků nejprofitovějšího RM

Podívejme se nyní na úplně opačný protipól úspěšnosti, jak by situace vypadala s nejztrátovějším risk managementem, kde `take_profit = 0,007` a `stop_loss = 0,005`:



Obr. 64: Graf vývoje portfolia při nejztrátovějším RM

Každá mince má dvě strany. Při nesprávném RM bychom ztratili relativně mnoho peněz.

### Krok 5: Propojení neuronové sítě z burzou a deploy bota

Pro vytvoření a nasazení obchodního bota na burze bylo vytvořeno a umístěno několik skriptů na server. Tyto skripty zahrnují následující soubory: "`NancyTA_bot.py`", "`binance_futures.py`", "`keys.py`" a "`neuron_model_balanced_v3_od_08_09_2019_v3.h5`".

Hlavním skriptem, který koordinuje činnost všech ostatních, je skript "`NancyTA_bot.py`". Následující odstavec stručně vysvětluje obsah tohoto textu.

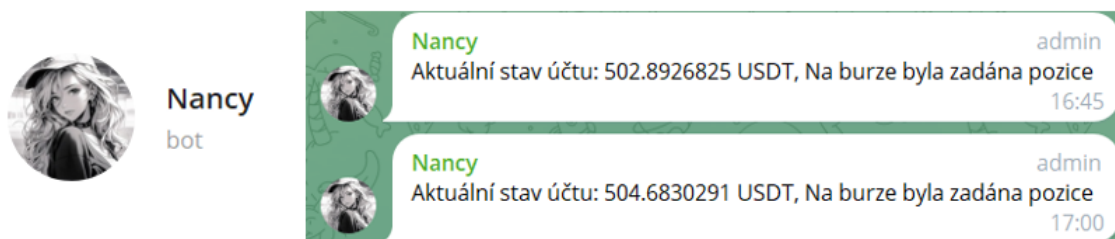
### **Vysvětlení skriptu `NancyTA_bot.py`:**

Tento skript představuje komplexní systém pro automatické obchodování na burze pomocí umělé inteligence a neuronové sítě. Hlavní funkce a části skriptu jsou následující:

1. **Načítání a inicializace modelu neuronové sítě:**
  - Model neuronové sítě je načítán ze souboru `neuron_model_balanced_v3_od_08_09_2019_v3.h5`.
2. **API klíče a relace:**
  - Binance API klíče jsou načítány z externího souboru `keys`.
  - Vytváří se relace s mechanismem opakování pro stabilnější připojení k Binance API.
3. **Funkce pro načítání historických dat:**
  - Funkce `get_historical_klines` načítá historická data obchodování a převádí je do formátu `DataFrame` pro další zpracování.
4. **Výpočet technických indikátorů:**

- Funkce `calculate_rsi` vypočítává Relative Strength Index (RSI).
  - Funkce `calculate_relative_values` a `calculate_past_relative_values` vypočítávají relativní hodnoty svíček.
5. **Funkce pro získání aktuálních dat a jejich zpracování:**
- Funkce `get_current_data` získává aktuální data, vypočítává RSI a relativní hodnoty pro aktuální i předchozí svíčky.
6. **Predikce a rozhodování:**
- Funkce `make_decision` získává aktuální data, připravuje je pro model neuronové sítě a provádí predikci, zda koupit nebo prodat.
7. **Automatické obchodování:**
- Funkce `check_candles` a `check_orders` pravidelně kontrolují svíčky a zadávají obchodní příkazy na základě predikcí modelu.
  - Funkce `set_orders` zajišťuje komunikaci s Binance API a zadávání obchodních příkazů.
8. **Telegram bot:**
- Implementace Telegram bota, který umožňuje vzdálené sledování stavu účtu a obchodních příkazů.

Celkově tento skript automatizuje proces obchodování na burze pomocí pokročilých metod umělé inteligence a technické analýzy, zajišťuje robustní komunikaci s Binance API a poskytuje uživateli možnost monitorovat obchodní procesy prostřednictvím Telegramu. V současném okamžiku je skript nastaven tak, aby skrze zprávy na Telegramu informoval o činnosti obchodování a aktuálním zůstatku na účtu.



Obr. 65: NancyTA\_bot a ukázka zpráv ohledně obchodů na burze

### Obsah skriptu *binance\_futures.py*

Skript "*binance\_futures.py*" slouží k automatizaci obchodování na platformě Binance Futures prostřednictvím rozhraní API. Skript využívá knihovnu `binance_um_futures` a implementuje několik klíčových funkcí, které umožňují plně automatizované obchodování. Hlavní funkce zahrnují:

- **Inicializace Binance Futures klienta:** Skript načítá API klíče z externího souboru `keys` a inicializuje klienta pro komunikaci s Binance Futures API.
- **Získání zůstatku v USDT:** Funkce `get_balance_usdt` získává aktuální zůstatek účtu v USDT.
- **Získání seznamu tickerů:** Funkce `get_tickers_usdt` vrací seznam všech obchodovatelných párů v USDT.

- **Nastavení páky a režimu marže:** Funkce `set_leverage` a `set_mode` nastavují páku a typ marže (Cross nebo Isolated) pro daný obchodní pár.
- **Získání přesnosti cen a množství:** Funkce `get_price_precision` a `get_qty_precision` vrací přesnost cen a množství pro daný symbol.
- **Otevření obchodního příkazu:** Funkce `open_order` umožňuje vytvoření nového nákupního nebo prodejního příkazu. Zahrnuje nastavení páky, výpočet množství a ceny, a zadání stop-loss a take-profit příkazů.
- **Kontrola otevřených pozic a objednávek:** Funkce `check_positions` vrací seznam aktuálně otevřených pozic, zatímco `close_open_orders` uzavírá všechny otevřené příkazy pro daný symbol. Funkce `check_orders_for_symbol` kontroluje, zda existují specifické typy objednávek (take profit nebo stop market) pro daný symbol.
- **Pomocné funkce:** Například `get_duration_time` nastavuje dobu trvání objednávky na 14 minut od aktuálního času.

Tento skript je navržen tak, aby poskytoval kompletní sadu nástrojů pro efektivní a bezpečné obchodování na Binance Futures, čímž umožňuje automatizované strategie obchodování s minimálním zásahem uživatele.

Na následujícím obrázku je ukázka části kódu, kde se zadává pozice pro nákup. Přestože jde o relativní proces, zadávání pozice se nakonec ukázalo být docela výzvou vzhledem k okolnostem, se kterými bylo potřeba se průběžně vypořádávat. Například nelze pouze sdělit „Chci nakoupit za 2 USDT“, ale je potřeba vypočítat množství Etherea, které skutečně má být s pákou nakoupeno, aby se pozice realizovala.

```
# BUY pozice
if side == 'Buy':
    try:
        # První request, zadání nového příkazu
        resp1 = um_futures_client.new_order(
            symbol=symbol,
            side='BUY',
            type='Limit',
            quantity=qty01, # Množství, co jsem vypočítal dřív
            timeInForce='GTD', # timer do vykonání příkazu "GTC" je good till cancelled
            goodTillDate= new_timestamp_ms,
            price = first_entry_price # jde o limitní cenu, pokud jdu market, tak tohle nemusí být
        )
        # type 'LIMIT' je že zadám nějaký limit kde otevřít pozici
        # type 'MARKET' nakoupí nyní za aktuální cenu
```

Obr. 66: Ukázka vystavení nákupní pozice (long) na burzu

Dále také bylo nutné zajistit, aby otevřené objednávky na `take_profit` nebo `stop_loss` neexistovaly donekonečna. Pokud se například realizovala objednávka `take_profit`, tak zmizela. Bylo nutné zařídit, aby současně s ní zmizela i nerealizovaná objednávka `stop_loss`. Bez ošetření této záležitosti by systém nemohl fungovat.

Pro uvedení strategie do provozu bylo nutné spuštění bota také správně načasovat, aby se svíčky kontrolovaly synchronně se svíčkami na burze, ideálně sekundu nebo dvě před uzávěrem intervalu a systém mohl pracovat co nejpřesněji.

#### 4.2.4 Výsledky Automatizovaného obchodování na burze na základě technické analýzy

Podarilo se natrénovat neuronovou síť, která nyní úspěšně vyhodnocuje situace na trhu. Vzhledem k tomu, že je neuronová síť v provozu na serveru pouze několik dní, není zatím vhodné detailně analyzovat několik málo obchodů, které byly realizovány. Nicméně, lze konstatovat, že optimální úspěšnost obchodů (winrate 38 %) při parametrech  $take\_profit = 0,002$  a  $stop\_loss = 0,001$  byla dosud udržena.

B ETHUSD Perp Cross			Closed
Closing PNL			Closed Vol.
<b>-0.51</b>			<b>0.136 ETH</b>
Entry Price	Avg. Close Price	Max Open Interest	
3672.57	3668.80	0.136 ETH	
Opened	2024-05-23 22:15:03		
Closed	2024-05-23 22:15:05		

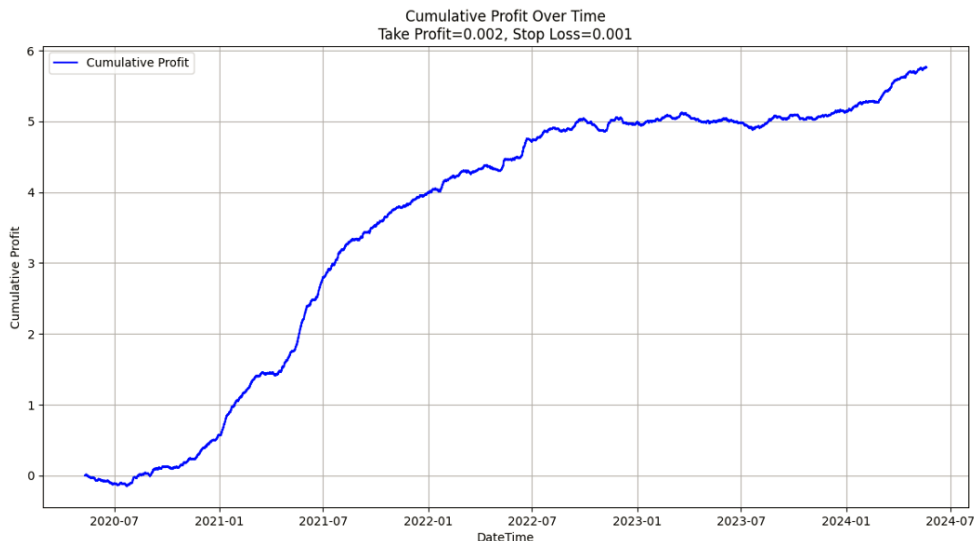
S ETHUSD Perp Cross			Closed
Closing PNL			Closed Vol.
<b>1.00</b>			<b>0.131 ETH</b>
Entry Price	Avg. Close Price	Max Open Interest	
3808.78	3801.12	0.131 ETH	
Opened	2024-05-23 21:44:57		
Closed	2024-05-23 21:45:10		

Obr. 67: Ukázka realizovaných obchodů zprostředkovaných neuronovou sítí

Přestože byl bot nastaven tak, aby vyhodnocoval trh sekundu či dvě před uzávěrem svíčky, i tato krátká doba může mít významný vliv na pohyb ceny. Například, v jedné situaci NancyTA\_bot úspěšně nastavil pozici sekundu před uzávěrem svíčky, avšak během této poslední sekundy cena Etherea klesla o 0,1 %, což vedlo k okamžitému spuštění funkce  $stop\_loss$  a zaznamenání ztráty. Po realizaci ztráty cena následně vzrostla.

Prakticky se tak stalo, že podle vizuálních výsledků v grafu vývoje ceny Etherea by bylo možné konstatovat, že jsme dosáhli zisku, ale ve skutečnosti jsme kvůli jednosekundovému rozdílu mezi nastavením pozice a uzavřením svíčky realizovali ztrátu.

Uvažujme tedy, že by byl winrate udržen i nadále a vraťme se ke grafu z minulé kapitoly.



Obr. 68: Nejprofitovější strategie

Graf znázorňuje celý cyklus od minulého halvingu Bitcoinu až po současný halving. Velkou výhodou našich výsledků je relativně nízká volatilita, kterou by ocenili především konzervativní investoři. Při pohledu na vývoj ceny Bitcoinu od začátku našeho obchodování zjistíme, že jeho hodnota vzrostla přibližně 6násobně.



Obr. 69: Změna hodnoty BTC za poslední 4 roky [63]

Na začátku, během období COVID-19, se hodnota našeho portfolia snížila o 15 %, zatímco hodnota Bitcoinu klesla až o 63 %. Dále pak v bearmarketu, kdy Bitcoin ztratil až 75 % své hodnoty, naše portfolio zůstávalo s jemnými výkyvy stabilní.

V porovnání s vývojem ceny Etherea by však náš bot značně zaostával. Ethereum dokázalo od posledního halvingu zvýšit svou hodnotu 19násobně.

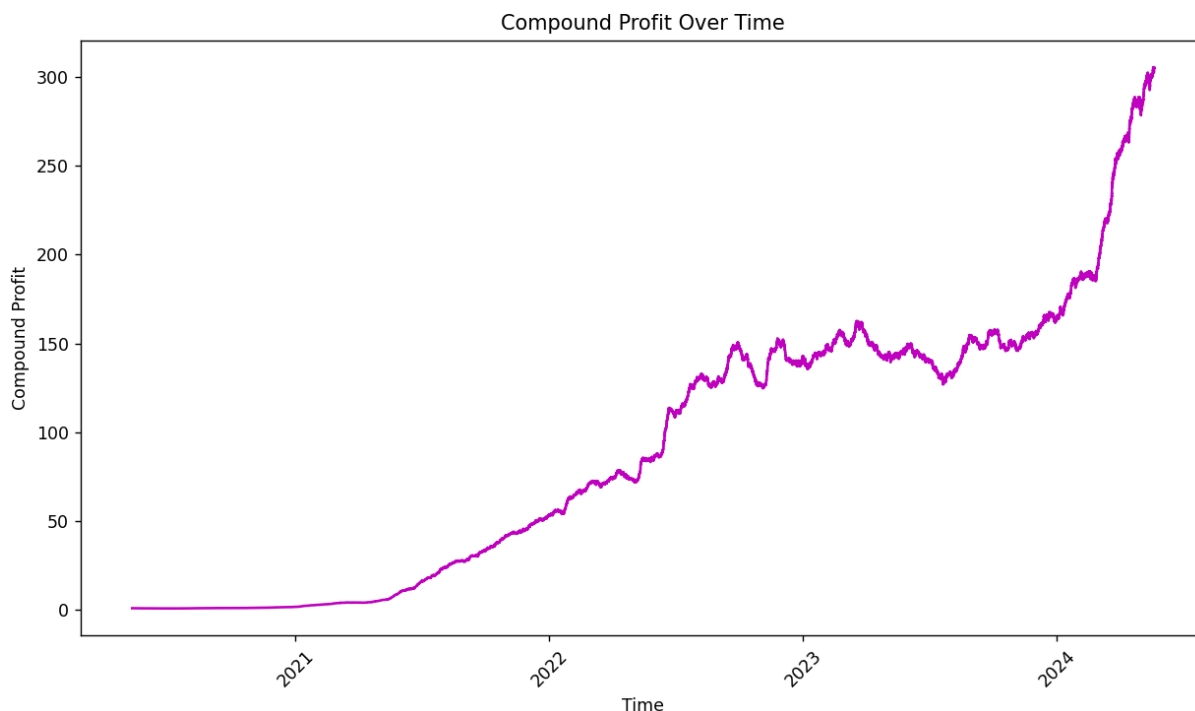


Obr. 70: Změna hodnoty ETH za poslední 4 roky

Nad Ethereum bychom měli navrch pouze ve vrcholení bearmarketu kolem června roku 2022.

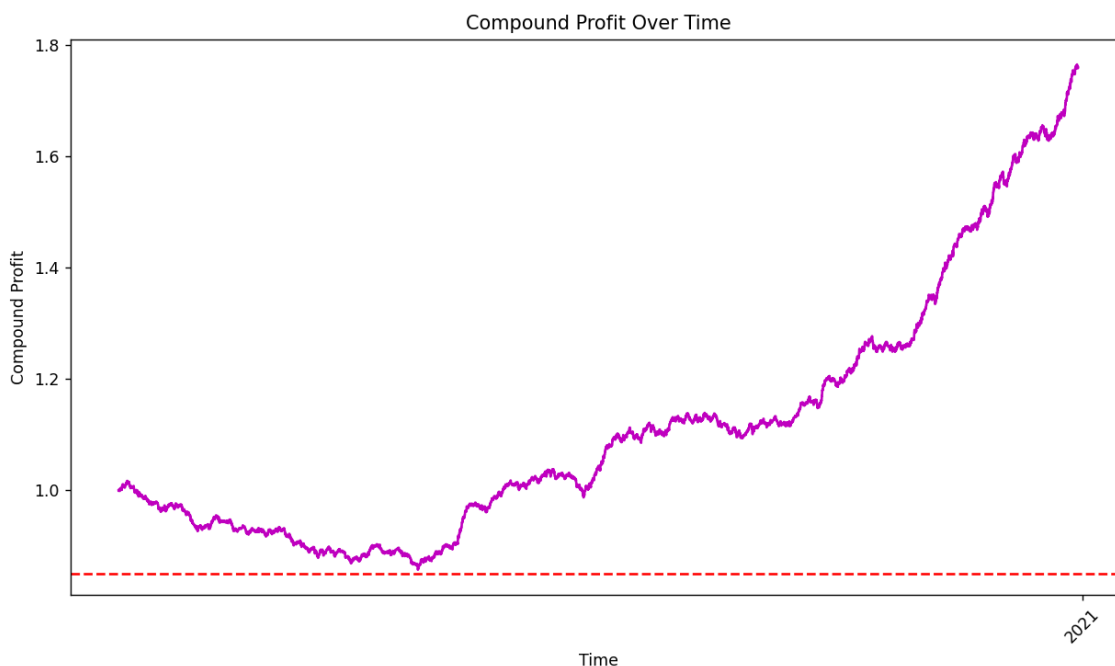
Každopádně máme „ESO“ v rukávu a tím je složené úročení! Následující graf zobrazuje průběh vývoje portfolia spravovaného naším botem za předpokladu, že se přičítá vždy zisk procentuálně z aktuálního zůstatku na účtu a ztráta se také realizuje procentuálně vzhledem k aktuálnímu zůstatku na účtu. Takže při každém zisku pronásobíme svůj aktuální zůstatek hodnotou 1,002 a při ztrátě jej pronásobíme hodnotou 0,999.

Výsledky jsou po téhle úpravě strategie vskutku zajímavé. Tímhle způsobem by bylo poraženo i Ethereum. Následující obrázek ukazuje vývoj hodnoty portfolia se strategií složeného úročení. Výpočet je realizovaný v souboru `vypocet_slozene_uroceni.py`.



Obr. 71: Síla složeného úročení

Z obrázku je patrné, že se naše portfolio k dnešnímu dni zhodnotilo přibližně 300násobně, což je tedy o něco více než zhodnocení Ethereum ve stejném období. Při pohledu na období do roku 2021 vidíme, že jsme při kumulativním zhodnocení zaznamenali pokles hodnoty portfolia na 85 %. Podívejme se, jak by vypadala situace při našem nynějším složeném úročení:



Obr. 72: Maximální propad hodnoty portfolia

Vidíme, že i při složeném úročení jsme hodnotu 85 % původního portfolia, která je v grafu zobrazena červenou čerchovanou přímkou, nepřekročili.

## 5 ZÁVĚR

Tato diplomová práce se zabývala problematikou využití umělé inteligence pro automatizaci obchodování na burze. Cílem bylo prozkoumat současné technologie, které jsou aplikovány v algoritmickeém obchodování, a následně navrhnout a vyvinout automatizovaný obchodní systém využívající umělou inteligenci.

Ve druhé kapitole práce jsme se zaměřili na současné technologie používané v algoritmickeém obchodování, včetně vysokofrekvenčního obchodování (HFT), cloudových řešení, strojového učení, blockchainu a smart contracts. Popsali jsme také různé strategie a modely, které jsou v algoritmickeém obchodování využívány, a diskutovali jsme o výzvách a rizicích, která tento způsob obchodování přináší.

Ve třetí kapitole jsme se podrobně věnovali aplikacím umělé inteligence v obchodování na burze, jako je prediktivní analytika, automatizované obchodní systémy a zpracování přirozeného jazyka. Diskutovali jsme také o etických a regulačních výzvách spojených s nasazením umělé inteligence v této oblasti.

Čtvrtá kapitola se zaměřila na návrh a vývoj automatizovaného obchodního systému. Popsali jsme architekturu systému, výběr programovacích jazyků a nástrojů, a uvedli jsme příklady konkrétních implementací algoritmů pro obchodování. Důležitou součástí této kapitoly bylo také testování a hodnocení výkonnosti navrženého systému.

Výsledky dosažené v této práci ukazují, že využití umělé inteligence může výrazně zvýšit efektivitu a přesnost obchodování na burze. Automatizovaný obchodní systém vyvinutý v rámci této práce dosáhl významného zhodnocení portfolia, což dokládá potenciál umělé inteligence v této oblasti. Nicméně je třeba vzít v úvahu také rizika spojená s nasazením těchto technologií, zejména technologická rizika a etické otázky.

V závěru lze konstatovat, že cíle práce byly splněny a navržený systém představuje přínos k oblasti algoritmickeého obchodování. Pro další výzkum a vývoj by bylo zajímavé se zaměřit na optimalizaci stávajících algoritmů, prozkoumat možnosti integrace dalších technologií, jako je kvantové počítání, a dále rozvíjet etické a regulační rámce pro využití umělé inteligence v obchodování na burze.

Závěrem bych rád poděkoval všem, kteří mi poskytli podporu a pomoc při vypracování této práce, obzvláště mému otci za cenné rady a podporu při nasazování jednotlivých botů na server a svému vedoucímu za psychickou podporu a vedení.

## SEZNAM POUŽITÉ LITERATURY

- [1] MORDOR INTELLIGENCE. Algorithmic Trading Market - Market Trends. Online. Dostupné z: <https://www.mordorintelligence.com/industry-reports/algorithmic-trading-market/market-trends> [cit. 2024-05-23].
- [2] GRAND VIEW RESEARCH. Algorithmic Trading Market Report. Online. Dostupné z: <https://www.grandviewresearch.com/industry-analysis/algorithmic-trading-market-report> [cit. 2024-05-23]
- [3] GLOBENEWSWIRE. Algorithmic Trading Global Market Report 2023: Pioneering Innovations Drive Algorithmic Trading Market, Empowering Traders with Automation. Online. Dostupné z: <https://www.globenewswire.com/news-release/2023/09/11/2740562/0/en/Algorithmic-Trading-Global-Market-Report-2023-Pioneering-Innovations-Drive-Algorithmic-Trading-Market-Empowering-Traders-with-Automation.html> [cit. 2024-05-23].
- [4] WJAETS. Algorithmic Trading in Financial Markets. Online. Dostupné z: <https://wjaets.com/sites/default/files/WJAETS-2024-0054.pdf> [cit. 2024-05-23].
- [5] FORTUNE BUSINESS INSIGHTS. Algorithmic Trading Market. Online. Dostupné z: <https://www.fortunebusinessinsights.com/algorithmic-trading-market-107174> [cit. 2024-05-23].
- [6] SCHWAB. High-Frequency Algorithmic Trading. Online. Dostupné z: <https://www.schwab.com/learn/story/high-frequency-algorithmic-trading> [cit. 2024-05-23].
- [7] STARTUS INSIGHTS. Algorithmic Trading: News Brief. Online. Dostupné z: <https://www.startus-insights.com/innovators-guide/algorithmic-trading-news-brief/> [cit. 2024-05-23].
- [8] QUANTIFIED STRATEGIES. Algo Trading Strategies. Online. Dostupné z: <https://www.quantifiedstrategies.com/algo-trading-strategies/> [cit. 2024-05-23].
- [9] WALL STREET ZEN. Algorithmic Trading Strategies. Online. Dostupné z: <https://www.wallstreetzen.com/blog/algorithmic-trading-strategies/> [cit. 2024-05-23].
- [10] TRADING CANYON. Algorithmic Trading Strategies. Online. Dostupné z: <https://www.tradingcanyon.com/trading/algorithmic-trading-strategies/> [cit. 2024-05-23].
- [11] ALPHAEX CAPITAL. Algorithmic Trading in Forex. Online. Dostupné z: <https://www.alphaexcapital.com/algorithmic-trading-in-forex/> [cit. 2024-05-23].
- [12] ADMIRAL MARKETS. Trading Strategies. Online. Dostupné z: <https://admiralmarkets.com/education/articles/forex-strategy/trading-strategies> [cit. 2024-05-23].
- [13] ROBOTS.NET. How Does Algorithmic Trading Work?. Online. Dostupné z: <https://robots.net/fintech/how-does-algorithmic-trading-work/> [cit. 2024-05-23].

- [14] HYSCALER. Financial Trading Success with AI Innovations. Online. Dostupné z: <https://hyscaler.com/insights/financial-trading-success-with-ai-innovations/> [cit. 2024-05-23].
- [15] NURP. The Future of Algorithmic Trading: Trends to Watch Out For. Online. Dostupné z: <https://nurp.com/wisdom/the-future-of-algorithmic-trading-trends-to-watch-out-for/> [cit. 2024-05-23].
- [16] CARMATEC. AI in Fintech: Role, Opportunities, and Use Cases. Online. Dostupné z: <https://www.carmatec.com/blog/ai-in-fintech-role-opportunities-and-use-cases/> [cit. 2024-05-23].
- [17] TECHSMILY. The Future of Algorithmic Trading: Emerging Trends and Technologies. Online. Dostupné z: <https://techsmily.com/2024/01/16/the-future-of-algorithmic-trading-emerging-trends-and-technologies/> [cit. 2024-05-23].
- [18] CRYPTSY. Crypto Trading Platforms 2024. Online. Dostupné z: <https://cryptsy.com/crypto-trading-platforms-2024/> [cit. 2024-05-23].
- [19] BENZINGA. Artificial Intelligence in Stock Trading: How AI is Revolutionizing Investment Strategies. Online. Dostupné z: <https://www.benzinga.com/general/23/11/35775521/artificial-intelligence-in-stock-trading-how-ai-is-revolutionizing-investment-strategies> [cit. 2024-05-23].
- [20] ANALYTICS VIDHYA. Machine Learning for Stock Market Prediction with Step-by-Step Implementation. Online. Dostupné z: <https://www.analyticsvidhya.com/blog/2021/10/machine-learning-for-stock-market-prediction-with-step-by-step-implementation/> [cit. 2024-05-23].
- [21] MDPI. Deep Reinforcement Learning for Financial Trading. Online. Dostupné z: <https://www.mdpi.com/2673-4591/56/1/254> [cit. 2024-05-23].
- [22] BUILT IN. AI Trading: How Artificial Intelligence is Shaking Up the Stock Market. Online. Dostupné z: <https://builtin.com/artificial-intelligence/ai-trading-stock-market-tech> [cit. 2024-05-23].
- [23] ANALYST ANSWERS. Best AI Trading Software Bots Reviewed: Free & Paid. Online. Dostupné z: <https://analystanswers.com/best-ai-trading-software-bots-reviewed-free-paid/> [cit. 2024-05-23].
- [24] EMB GLOBAL. AI Trading and Its Uses. Online. Dostupné z: <https://blog.emb.global/ai-trading-and-its-uses/> [cit. 2024-05-23].
- [25] SPROUT SOCIAL. Sentiment Analysis Tools. Online. Dostupné z: <https://sproutsocial.com/insights/sentiment-analysis-tools/> [cit. 2024-05-23].
- [26] GITHUB. Sentiment Analysis Topics in Python. Online. Dostupné z: <https://github.com/topics/sentiment-analysis?l=python> [cit. 2024-05-23].
- [27] ARXIV. Sentiment Analysis with Transformer Models: A Comprehensive Review. Online. Dostupné z: <https://arxiv.org/abs/2305.14842> [cit. 2024-05-23].

- [28] SWISS FINTECH. AI-Powered Portfolio Optimization: Future of Asset Management. Online. Dostupné z: <https://swissfinte.ch/ai-powered-portfolio-optimization-future-of-asset-management/> [cit. 2024-05-23].
- [29] KSOLVES. Future of AI in Portfolio Management. Online. Dostupné z: <https://www.ksolves.com/blog/artificial-intelligence/future-of-ai-in-portfolio-management> [cit. 2024-05-23].
- [30] SECURE PRIVACY. Navigating Data Privacy 2024. Online. Dostupné z: <https://secureprivacy.ai/blog/navigating-data-privacy-2024> [cit. 2024-05-23].
- [31] ISACA. Challenges of AI and Data Privacy and How to Solve Them. Online. Dostupné z: <https://www.isaca.org/resources/news-and-trends/newsletters/atisaca/2021/volume-32/challenges-of-ai-and-data-privacy-and-how-to-solve-them> [cit. 2024-05-23].
- [32] WHARTON. AI in Finance: The Promise and Potential Pitfalls. Online. Dostupné z: <https://knowledge.wharton.upenn.edu/article/ai-in-finance-the-promise-and-potential-pitfalls/> [cit. 2024-05-23].
- [33] STANFORD UNIVERSITY. AI Index Report 2021 - Chapter 5. Online. Dostupné z: <https://aiindex.stanford.edu/wp-content/uploads/2021/03/2021-AI-Index-Report-Chapter-5.pdf> [cit. 2024-05-23].
- [34] INTRINIO. AI in the Stock Market: Uses, Trends, and Impact. Online. Dostupné z: <https://intrinio.com/blog/ai-in-the-stock-market-uses-trends-and-impact> [cit. 2024-05-23].
- [35] NASDAQ. AI Trading Software: Here's What You Need to Know. Online. Dostupné z: <https://www.nasdaq.com/articles/ai-trading-software-heres-what-you-need-to-know> [cit. 2024-05-23].
- [36] IBKR CAMPUS. AI Trading Software: Here's What You Need to Know. Online. Dostupné z: <https://ibkr-campus.com/traders-insight/ai-trading-software-heres-what-you-need-to-know/> [cit. 2024-05-23].
- [37] PYPI. Binance Futures Connector 1.8.0. Online. Dostupné z: <https://pypi.org/project/binance-futures-connector/1.8.0/> [cit. 2024-05-23].
- [38] BINANCE. Change Log. Online. Dostupné z: <https://binance-docs.github.io/apidocs/spot/en/#change-log> [cit. 2024-05-23].
- [39] DOUBLOIN. News & Events Affecting Bitcoin Price. Online. Dostupné z: <https://www.doubloin.com/learn/news-events-affecting-bitcoin-price> [cit. 2024-05-23].
- [40] THE CURRENCY ANALYTICS. Ethereum Surges: Factors Driving ETH Price Increase Explained. Online. Dostupné z: <https://thecurrencyanalytics.com/altcoins/ethereum-surges-factors-driving-eth-price-increase-explained-95245> [cit. 2024-05-23].
- [41] CRYPTONEWS. How to Analyze the Cryptocurrency Market: Key Factors to Consider. Online. Dostupné z: <https://cryptonews.com/news/how-analyze-cryptocurrency-market-key-factors-consider.htm> [cit. 2024-05-23].
- [42] COINTELEGRAPH. Bull vs. Bear Crypto Market: What is the Difference and How to Handle Both. Online. Dostupné z: <https://cointelegraph.com/learn/bull-vs-bear-crypto-market-what-is-the-difference-and-how-to-handle-both> [cit. 2024-05-23].

- [43] E\*TRADE. Behind the Crypto Bear Market. Online. Dostupné z: <https://us.etrade.com/knowledge/library/cryptocurrency/behind-the-crypto-bear-market> [cit. 2024-05-23].
- [44] THE CRYPTION. Bullish vs. Bearish: What it Means for Crypto's Future. Online. Dostupné z: <https://thecryption.com/guides/bullish-vs-bearish-what-it-means-for-cryptos-future/> [cit. 2024-05-23].
- [45] BEINCRYPTO. New Crypto Listings. Online. Dostupné z: <https://beincrypto.com/learn/new-crypto-listings/> [cit. 2024-05-23].
- [46] DAILY HODL. Crypto Trader Says 14 Small-Cap Projects Have Massive Potential in January. Online. Dostupné z: <https://dailyhodl.com/2021/01/14/crypto-trader-says-14-small-cap-projects-have-massive-potential-in-january/> [cit. 2024-05-23].
- [47] MUSK, Elon. [@elonmusk]. Online. Twitter. Dostupné z: <https://twitter.com/elonmusk> [cit. 2024-05-23].
- [48] WATCHER GURU. [@WatcherGuru]. Online. Twitter. Dostupné z: <https://twitter.com/WatcherGuru> [cit. 2024-05-23].
- [49] BITCOIN MAGAZINE. [@BitcoinMagazine]. Online. Twitter. Dostupné z: <https://twitter.com/BitcoinMagazine> [cit. 2024-05-23].
- [50] CZ BINANCE. [@cz\_binance]. Online. Twitter. Dostupné z: [https://twitter.com/cz\\_binance](https://twitter.com/cz_binance) [cit. 2024-05-23].
- [51] TWITTER DEVELOPER. About Twitter API. Online. Dostupné z: <https://developer.x.com/en/docs/twitter-api/getting-started/about-twitter-api> [cit. 2024-05-23].
- [52] WEB AUTOMATION. WebAutomation.io. Online. Dostupné z: <https://webautomation.io/> [cit. 2024-05-23].
- [53] TENSORFLOW. TensorFlow. Online. Dostupné z: <https://www.tensorflow.org/> [cit. 2024-05-23].
- [54] ANDROID GUIAS. Co je Tasker?. Online. Dostupné z: <https://androidguias.com/cs/co-je-tasker/> [cit. 2024-05-23].
- [55] MACSPOTS. Tasker: What It Is & How To Use It. Online. Dostupné z: <https://cze.macspots.com/tasker-what-it-is-how-use-it> [cit. 2024-05-23].
- [56] GOOGLE PLAY. Tasker. Online. Dostupné z: <https://play.google.com/store/apps/details?id=net.dinglich.android.taskerm&hl=en> [cit. 2024-05-23].
- [57] TELEGRAM. FAQ: Why doesn't my bot see messages from other bots?. Online. Dostupné z: <https://core.telegram.org/bots/faq#why-doesn-39t-my-bot-see-messages-from-other-bots> [cit. 2024-05-23].
- [58] PYTHON TELEGRAM BOT. Python Telegram Bot Documentation. Online. Dostupné z: <https://docs.python-telegram-bot.org/en/v21.2/> [cit. 2024-05-23].

- [59] WIN SCP. WinSCP. Online. Dostupné z: <https://winscp.net/eng/index.php> [cit. 2024-05-23].
- [60] PUTTY. PuTTY. Online. Dostupné z: <https://putty.org/> [cit. 2024-05-23].
- [61] INVESTOPEDIA. Candlestick. Online. Dostupné z: <https://www.investopedia.com/terms/c/candlestick.asp> [cit. 2024-05-23].
- [62] DAS, Tithi. Lesson 6: Candlestick Session. Online. SlideShare. Dostupné z: <https://www.slideshare.net/TithiDas30/lesson6-candle-stick-sessionppt> [cit. 2024-05-23].
- [63] TRADINGVIEW. TradingView. Online. Dostupné z: <https://www.tradingview.com/> [cit. 2024-05-23].
- [64] STOCKCHARTS. Relative Strength Index (RSI). Online. Dostupné z: [https://school.stockcharts.com/doku.php?id=technical\\_indicators:relative\\_strength\\_index\\_rsi](https://school.stockcharts.com/doku.php?id=technical_indicators:relative_strength_index_rsi) [cit. 2024-05-23].







