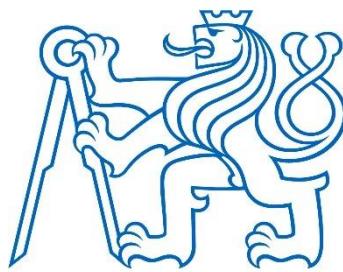


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA JADERNÁ A FYZIKÁLNĚ INŽENÝRSKÁ

Katedra softwarového inženýrství

Studijní program: Aplikace informatiky v přírodních vědách



**Využití GAN neuronových sítí k predikci
pohybu aktiv na finančním trhu**

VÝZKUMNÝ ÚKOL

Vypracoval: Bc. Pavel Ježek

Vedoucí práce: Doc. Ing. Quang Van Tran Ph.D.

Rok: 2024

České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská

Katedra softwarového inženýrství

Akademický rok 2023/2024

ZADÁNÍ VÝZKUMNÉHO ÚKOLU

Student: Bc. Pavel Ježek

Studijní program: Aplikace informatiky v přírodních vědách

Název práce: Využití GAN neuronových sítí k predikci pohybu kurzu aktiv na finančním trhu

Pokyny pro vypracování:

1. Vypracovat literární rešerši na hluboké neuronové sítě, s důrazem na GAN neuronové sítě
2. Prozkoumat současný stav využití GAN sítí pro predikci
3. Vyhodnotit model GAN sítí k predikci pohybu kurzu aktiv na finančním trhu a vyhodnotit získané výsledky

Doporučená literatura:

- [1] Razavi-Far, R., Ruiz-Garcia, A., Palade, V., & Schmidhuber, J. (2022). Generative adversarial learning: architectures and applications. Cham: Springer.
- [2] Aggarwal, C. C. (2018). Neural networks and deep learning. Springer, 10(978), 3.
- [3] Wang, K., Gou, C., Duan, Y., Lin, Y., Zheng, X., & Wang, F. Y. (2017). Generative adversarial networks: introduction and outlook. IEEE/CAA Journal of Automatica Sinica, 4(4), 588-598.
- [4] Kumar, A., Alsadoon, A., Prasad, P. W. C., Abdullah, S., Rashid, T. A., Pham, D. T. H., & Nguyen, T. Q. V. (2022). Generative adversarial network (GAN) and enhanced root mean square error (ERMSE): deep learning for stock price movement prediction. Multimedia Tools and Applications, 1-19.

Jméno a pracoviště vedoucího práce:

Doc. Ing. Quang Van Tran, Ph.D.

Katedra softwarového inženýrství, Fakulta jaderná a fyzikálně inženýrská, ČVUT v Praze

Jméno a pracoviště konzultanta práce:

Datum zadání výzkumného úkolu: 25. 10. 2023

Termín odevzdání výzkumného úkolu: 25. 8. 2024

V Praze dne 25. 10. 2023



vedoucí práce



garant programu



vedoucí katedry

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady uvedené v přiloženém seznamu.

Bc. Pavel Ježek

Poděkování

Děkuji vedoucímu Doc. Ing. Quang Van Tranovi Ph.D. za neocenitelné rady a pomoc při tvorbě výzkumného úkolu.

Bc. Pavel Ježek

Abstrakt

Tato práce se zabývá aplikací generativních nepřátelských sítí (GAN) pro predikci cen vybraných finančních aktiv, konkrétně akcií společnosti Apple, ČEZ, komoditního zlata a kryptoměny Bitcoin. GAN sítě, jako pokročilý nástroj strojového učení, umožňují vytváření realistických predikcí časových řad, což je v této oblasti obzvláště cenné. Práce se zaměřuje na teoretické základy hlubokého učení, GAN sítí, popisu dat, implementaci modelu a následnou analýzu výsledků. Na výsledných modelech pak bylo aplikováno modelové obchodování s výpočtem potenciálního zisku.

Klíčová slova: GAN, generativní nepřátelské sítě, generativní adversiální sítě, akcie, finanční trh, hluboké učení

Abstract

This paper is focused on the application of generative adversarial networks (GANs) to predict the prices of selected financial assets, namely Apple stock, ČEZ, commodity gold and the cryptocurrency Bitcoin. GANs, as an advanced machine learning tool, enable the generation of realistic time series predictions, which is particularly valuable in this field. This paper focuses on the theoretical foundations of deep learning, GANs, data description, model implementation and subsequent analysis of the results. Model trading was applied to the created models with calculations of potential profits.

Key words: GAN, generative adversarial networks, stock, financial market, deep learning

Obsah

Seznam zkratek.....	8
1 Úvod	10
2 Hlubokého učení	11
2.1 Základní stavba sítě	11
2.2 Dělení.....	12
2.2.1 Vícevrstvý perceptron	12
2.2.2 Rekurentní sítě.....	13
2.2.3 Konvoluční sítě	19
3 GAN sítě.....	26
3.1 Struktura.....	26
3.2 Matematická formulace a algoritmus	27
3.3 Varianty.....	30
3.3.1 Podmíněný GAN (CGAN).....	30
3.3.2 Hluboký konvoluční GAN (DCGAN).....	30
3.3.3 Laplacovský GAN (LapGAN)	30
3.3.4 GAN maximalizující informace (InfoGAN)	30
3.3.5 Energy-Based GAN (EBGAN).....	31
3.3.6 Wasserstein GAN (WGAN).....	31
4 Rešerše prací	32
4.1 Text na obrázek.....	32
4.2 Text na hlas	32
4.3 Úprava obrázku.....	33
4.4 Rozpoznávání vlastností obrázku.....	33
4.5 Časové řady	34
5 Data	35
5.1 Apple	35
5.2 ČEZ	40
5.3 Bitcoin.....	43
5.4 Zlato.....	45
5.5 Zpožděné proměnné	46
5.5.1 Klouzavý průměr.....	46
5.5.2 Bollingerova pásmá	47
5.5.3 MACD (Moving Average Convergence Divergence)	48

5.5.4	Index relativní síly.....	49
5.6	Fourierovy transformace	50
6	Model a jeho výsledky	51
6.1	Popis modelu	51
6.2	Výsledky pro Apple	51
6.3	Výsledky pro ČEZ	53
6.4	Výsledky pro zlato.....	55
6.5	Výsledky pro Bitcoin	57
7	Závěr.....	59
	Zdroje.....	61

Seznam zkratek

ADAM	Adaptive Moment Estimation	Adaptivní momentové odhadování
AGAN	Artistic Generative Adversarial Network	Umělecká generativní nepřátelská síť
ANN	Artificial Neural Network	Umělá neuronová síť
ANOVA	Analysis of Variance	Analýza rozptylu
AttnGAN	Attention Generative Adversarial Network	Generativní nepřátelské sítě s pozorností
BPTT	Backpropagation Through Time	Zpětná propagace skrze čas
BPTT-T	Backpropagation Through Time with Truncation	Zpětná propagace skrze čas s ořezáním
CGAN	Conditional Generative Adversarial Network	Podmíněná generativní nepřátelská síť
CNN	Convolutional Neural Network	Konvoluční neuronová síť
DCGAN	Deep Convolutional Generative Adversarial Network	Hluboká konvoluční generativní nepřátelská síť
DDPM	Diffusion Probabilistic Models	Difúzní pravděpodobnostní modely
DF-GAN	Distribution-Free GAN	Bezdistribuční generativní nepřátelská síť
EBGAN	Energy-Based Generative Adversarial Network	Energetická generativní nepřátelská síť
EMA	Exponential Moving Average	Exponenciální klouzavý průměr
GAN	Generative Adversarial Network	Generativní nepřátelská síť
GRU	Gated Recurrent Unit	Bránová rekurentní jednotka
HiFi-GAN	High Fidelity Generative Adversarial Network	Generativní nepřátelská síť s vysokou věrností
InfoGAN	Information Maximizing Generative Adversarial Network	Informačně maximalizující generativní nepřátelská síť
LapGAN	Laplacian Pyramid Generative Adversarial Network	Generativní nepřátelská síť s Laplacovou pyramidou
LSTM	Long Short-Term Memory	Dlouhá krátkodobá paměť
MACD	Moving Average Convergence Divergence	Konvergence a divergence klouzavých průměrů
MirrorGAN	Mirror Generative Adversarial Network	Zrcadlová generativní nepřátelská síť
MLP	Multilayer Perceptron	Vícevrstvý perceptron
ReLU	Rectified Linear Unit	Reálná funkce jedné proměnné
RFE	Recursive Feature Elimination	Rekurzivní eliminace vlastností
RMSE	Root Mean Square Error	Kořen střední kvadratické chyby
RNN	Recurrent Neural Network	Rekurentní neuronová síť
RSI	Relative Strength Index	Index relativní síly
SGD	Stochastic Gradient Descent	Stochastický gradientový sestup
SMA	Simple Moving Average	Jednoduchá klouzavý průměr
SSA-GAN	Semi-Supervised Adversarial Generative Network	Nepřátelská generativní síť s kombinací učení s a bez učitele
StackGAN	Stack Generative Adversarial Network	Zásobníková generativní nepřátelská síť

StyleGAN	Style Generative Adversarial Network	Stylizovaná generativní nepřátelská síť
StyleGAN2	Style Generative Adversarial Network 2	Stylizovaná generativní nepřátelská síť 2
TTS	Text-to-Speech	Převod textu na řeč
VGG	Visual Geometry Group	Skupina pro vizuální geometrii
WGAN	Wasserstein Generative Adversarial Network	Wassersteinova generativní nepřátelská síť

1 Úvod

Umělá inteligence v posledních letech získává na popularitě. Využívá se v celé řadě sektorech a nové způsoby využití stále vznikají. Jedním z poměrně nových způsobů vytváření modelů jsou Generativní nepřátelské sítě, zkráceně GAN. Jedná se partií hlubokého učení. Základním principem je trénování dvou modelů, diskriminátoru a generátoru. Diskriminátor má za cíl rozpozнат, zdali jsou data falešná (generovaná) nebo pravá. Generátor se snaží vytvořit natolik dobrá data, že je diskriminátor nerozezná od původních.

První zaměření těchto modelů bylo spojené s daty založenými na obrázcích, přístup se již modifikuje i na další typy, jako jsou časové řady. Kdy v tomto sektoru je snaha o co největší přesnost s ne vždy úplnými daty, a tak zkoušeji, zdali nové modely nedokážou dosáhnout ještě lepšího výsledku než dosavadní. A to je cílem této práce, použití modelu GAN na predikce na finančních trzích. Použití tohoto přístupu se předvedlo na více druzích artiklů pohybujících se na různých trzích či burzách. Práce je rozdělena do 5 kapitol.

První kapitola pojednává o hlubokém učení. Popisuje základní princip a důvody jejich vzniku. Přibližuje základní strukturu a jak se liší od ostatních modelů umělé inteligence. Dále se zde vyjmenují základní druhy. Těmi jsou vícevrstvý perceptron, rekurentní sítě a konvoluční sítě. U výše jmenovaných je více do hloubky popisována matematická formulace a některé z užívaných variant.

Druhá kapitola se věnuje GAN sítím. Nejdříve se zaměřuje na základní popis a strukturu tohoto přístupu. Pokračuje se matematickou formulací a algoritmem učení. Dále se zde nachází vyjmenování různých variant s popisem, jak fungují, jak se liší a na co se používají.

Třetí kapitolou je rešerše prací. Ta je rozdělena do 5 podkategorií na základě funkce modelů, které dané vědecké skupiny tvořily. Jsou zde jak ty nejvíce populární (rozpoznání, úpravy či generace obrázku), tak i ty méně zastoupené (práce s hlasem a časové řady).

Čtvrtá kapitola se zabývá využitými daty. Popisuje 4 hlavní směry, kterými se data oddělují (jejich výstupní vektor). Uvádí krátké charakteristiky predikovaných cenin. S tím pak popisuje i základní strukturu dat, jaká další byla přidána a jaké jiné charakteristiky byly vytvořeny za účelem lepšího provázání a lepší generalizace.

Pátá a poslední kapitola je zaměřena na popis využívaného modelu a jeho výsledky. Ty jsou reprezentovány, jak výpočtem jejich přesnosti a chyby, tak i příkladem modelového obchodování.

2 Hluboké učení

Jeden z velkých problémů pro mnoho aplikací umělé inteligence je, že problémy z „reálného světa“ (respektive záznamy a data, která o nich vypozorujeme/změříme a zpracujeme) mají mnoho jednotlivých definujících faktorů s různou váhou důležitosti pro výsledné řešení. Ty mohou být i vzájemně provázány a ovlivňovat se navzájem. Příkladem je rozpoznávání snímku s autem z průmyslové kamery. Auta s tmavší barvou budou s přibližujícími se nočními hodinami zdát se blíže k černé, tvar auta bude jiný na základě úhlu zachycení a další.

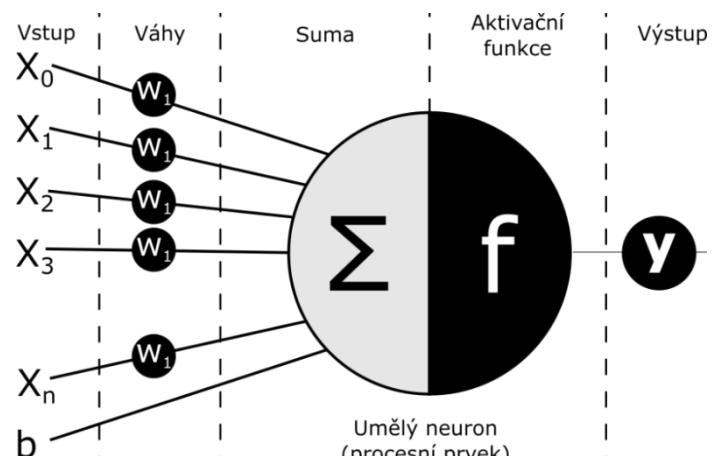
Základní metody strojového učení dokážou velkou řadu těchto problémů vyřešit, ale větší složitost některých z nich (typicky rozpoznávání obrazu) ukázali nedostatečnost těchto přístupů. Velice prominentní výzvou je tzv. *prokletí dimenze*. S vyšším počtem dimenzí se zvětšuje i počet všech možných variant celého problému (může i exponenciálně). Tím pádem pak je možná existence více různých stavů než pozorování v datech. Vytvořený model potom nemá z čeho čerpat a nedokáže přesně určit výsledek pro tyto části dat, tedy správně řešit daný systém.

Dalším problémem jsou předpoklady vznášeny na funkce generalizace, kterou modely vytvářejí – kterou se učí. Nejčastější implicitní předpoklady pak zahrnují hladkosť a lokální konzistence (funkce na malém intervalu je relativně konstantní, moc se nemění). Tyto podmínky fungují dobře pokud máme dostatek dat (myšleno tak, aby byly popsány různé proměny ve výsledku – výšky, hloubky). To opět ale naráží na problém s dimenzemi. Funkce může být dostatečně hladká, ale chová se jinak v každé dimenzi. Potom je určení funkce komplikované [1].

Tyto a jiné problémy pak pokrývá disciplína Hlubokého učení.

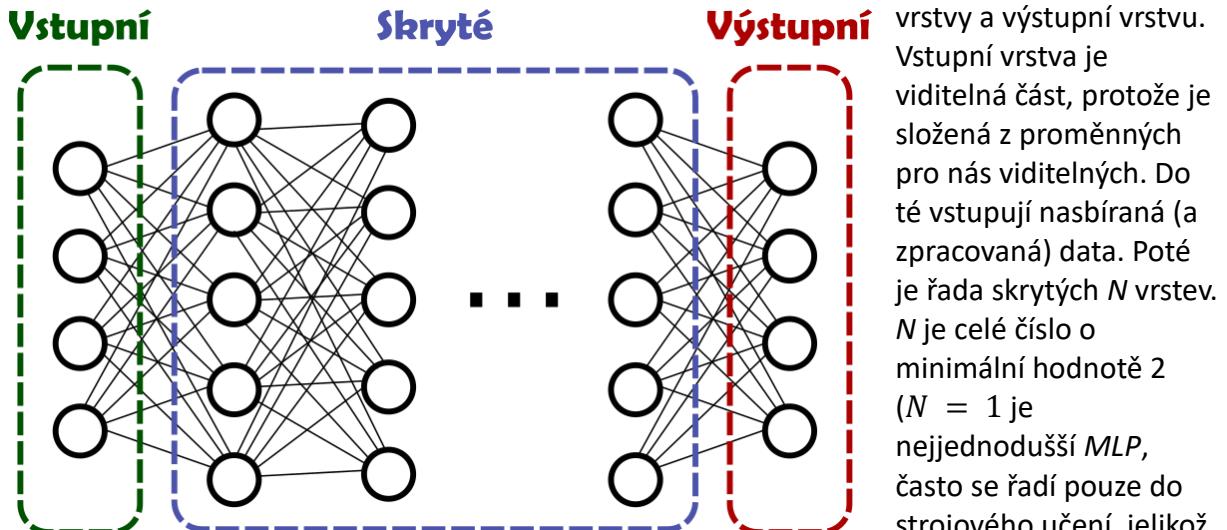
2.1 Základní stavba sítě

Základní struktura je odvozena od umělé neuronové sítě (*Artificial Neural Network*, zkráceně **ANN**). Ta má tvar inspirující se stavbou neuronových sítí v mozku. Skládá se z jednotlivých neuronů, které jsou spojené synapsemi. Do neuronu jako signál vstupuje číslo, to se na základě dané funkce (nazývané aktivační funkce) transformuje a postupuje dále po synapsi (na Obrázku 1 vyznačeno jako hrana vystupující z neuronu) do dalšího neuronu [2].



Obrázek 1: Grafické znázornění neuronu, upravená verze původně pocházející z [2]

Sítě využívající se v hlubokém učení se nazývají hluboké neuronové sítě. Skladba se dá zaobalit do tří částí (graficky znázorněny na Obrázku 2). Obsahuje vstupní vrstvu, skryté



Obrázek 2: Grafické znázornění rozložení neuronů v neuronové síti, rozděleno do tří hlavních částí – vstupní, skryté a výstupní vrstvy či vrstev

vrstvy a výstupní vrstvu. Vstupní vrstva je viditelná část, protože je složená z proměnných pro nás viditelných. Do té vstupují nasbíraná (a zpracovaná) data. Poté je řada skrytých N vrstev. N je celé číslo o minimální hodnotě 2 ($N = 1$ je nejjednodušší *MLP*, často se řadí pouze do strojového učení, jelikož není hluboká). Tyto vrstvy se starají o

rozdelení na jednoduší atributy (abstraktní vlastnosti). Nazývají se skryté, jelikož hodnoty uvnitř neuronů nejsou předem známy, model je musí určit sám. Když se vrátíme k příkladu s obrázky z kamer, tak jedna vrstva se bude zabývat barvou, další zase jednoduchou siluetou auta a tak podobně. Tyto vrstvy na sebe závisí (v první je barva, díky rozlišení barev pak další vrstva rozezná jednoduchou siluetu, ...). Po projití přes celou tuto část pak data vstoupí do výstupní vrstvy, kde se převedou do řešení daného problému. U této části záleží na typu dat (jestli jsou označená nebo ne, je toto označení spojité nebo diskrétní, jeli výstupní třída binární nebo diskrétní s více výstupovými třídami, ...) [2].

2.2 Dělení

Hluboké učení jako takové se dá použít jak pro učení s učitelem (*supervised*), učení bez učitele (*unsupervised*) tak i kombinace těchto dvou (*semi-supervised*). Dále také pro jednotlivé typy se hodí jiné přístupy. Jedním z nich jsou samozřejmě GAN sítě, ty ale budou přiblíženy až v další kapitole, zde bude zaměření na další používané metody.

2.2.1 Vícevrstvý perceptron

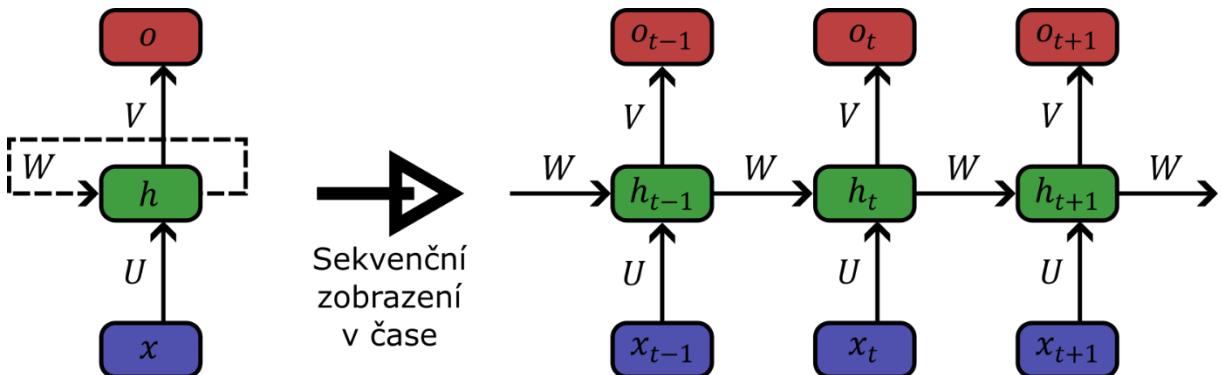
Obecně se tímto názvem označuje základní modely dopředných neuronových sítí. Minimálně má jednu skrytu vrstvu (u té se dá debatovat, zdali patří do hlubokého učení). Všechny vrstvy jsou propojeny se sousední vrstvou plně, tedy z každého neuronu vede synapse do každého neuronu v následující vrstvě. Používá se pro učení s učitelem. Často je pak využívána jako součást složitějších, novějších modelů hlubokého učení (složitější model rozloží data na různém abstraktní ukazatele a MLP pak vytvoří požadovanou strukturu výstupu) [2].

2.2.2 Rekurentní sítě

Jak už bylo zmíněno, data mohou být na sobě závislá. Obecně se tím myslí různá sekvenční závislost, jako jsou časové řady (reálné hodnoty), texty (symbolické značení) či data z prostředí biologie (symboly korespondující se sekvencí aminokyselin v DNA). V praktických aplikacích převažuje využití pro symbolické hodnoty. Rekurentní sítě (*Recurrent Neural Networks*, zkráceně **RNN**) nabízí řešení pro splnění těchto specifických požadavků. V RNN jednotlivé vrstvy odpovídají pozici v sekvenci (času u časové řady). Oproti tradičním neuronovým sítím, mají proměnlivý počet skrytých vrstev. Hlavní funkcí RNN je opakování stejné architektury vrstev skrze čas. Jelikož sekvence jednotlivých pozorování je důležitým parametrem dané části dat. Tím pádem je pro její správné určení potřebný i předchozí stav. Vstupy a výstupy tedy spolu interagují, neuron si „vzpomene“ na předchozí stav (pozorování v časové řadě, symbol v sekvenci). Této „vzpomínce“ se říká skrytý stav či skrytá paměť (*hidden state, hidden memory*). RNN teoreticky mohou, s dostatečným množstvím informací a výpočetní síly, simuloval jakýkoliv algoritmus. Tato vlastnost se nazývá Turingovská kompletnost [1] [3].

2.2.2.1 Základní struktura

Nejjednodušší RNN je zobrazena na obrázku 3. Obecně se ilustruje jako jedna vrstva s cyklickou cestou, na obrázku je také vidět znázornění práce s daty v čase, to se pak podobá dopředné sítí. Je očividné z popisů matic W , U , V , že parametry modelu se nemění v čase.



Obrázek 3: Schéma skryté vrstvy v RNN a jejich rozložení v čase

Tento případ počítá s tím, že každý časový krok má, jak vstup, tak i výstup. Samozřejmě v reálných situacích to není pravidlem (když budeme mít sekvenční problém bude vstup pouze na konci sekvence ale vstupy pokaždé, u popisování obrázku zase budeme mít méně vstupů a ale výstupy po každém kroku, ...).

Jednosměrný základní model

Vztahy si definujeme pro diskrétní časovou jednotku. Vstup v čase t označíme x_t , výstup o_t a skrytý stav h_t . Systém pak vypadá následovně:

$$h_t = f_h(x_t, h_{t-1}), \quad (2.2.2.1)$$

$$o_t = f_o(h_t), \quad (2.2.2.2)$$

kde funkce f_h je funkce přechodu a f_o funkce výstupu. Vezmeme-li tréninková data (sekvenci) ve tvaru $\{(x_1^{(n)}, y_1^{(n)}), \dots, (x_{T_n}^{(n)}, y_{T_n}^{(n)})\}_{n=1}^N$ lze parametry určit jako minimalizace následující nákladové funkce:

$$J(\theta) = \frac{1}{N} \sum_{n=1}^N \sum_{t=1}^{T_n} d(o_t^{(n)}, f_o(h_t)), \quad (2.2.2.3)$$

kde skrytý stav $h_0^{(n)} = 0$. Funkce $d(a, b)$ je metrika pro vzdálenost (Euklidovská, křížová entropie, ...). Po zařazení jednotlivých matic vah, aby model odpovídal znázornění struktury, pak funkce převodu a funkce na výstupu mají tvar:

$$h_t = \phi_h(W^\top h_{t-1} + U^\top x_t), \quad (2.2.2.4)$$

$$o_t = \phi_o(V^\top h_t), \quad (2.2.2.5)$$

kde ϕ_h a ϕ_o jsou nelineární funkce. Obvykle se jedná o nasycující se nelineární funkce, např. hyperbolický tangens nebo logistická funkce. Parametry pro tento systém pak lze určit skrze SGD s gradientem nákladové funkce (2.2.2.3) [3].

Obousměrné síť

Jednou z nevýhod jednosměrných sítí je pro určování stavů pouze minulé. V některých případech dojde k masivnímu vylepšení, pokud použijeme i budoucí stavy (např. rozpoznání psaného písma, vylepšení o budoucí kontextu daného symbolu).

Je potřeba dodefinovat nové značení. Je tady skrytý stav předchozí $h_t^{(b)}$ a skrytý stav následující $h_t^{(f)}$. Stavy mezi sebou interagují pouze v respektovaných skupinách (předchozí s předchozími a naopak). Jak už z názvu vyplívá, stavy $h_t^{(b)}$ interagují ve směru dozadu a $h_t^{(f)}$ dopředu. Oba ale mají stejný vstup x_t a stejný výstup o_t . Dále i jednotlivé matice vah v pořadí vstup, skryté vrstvy a výstup jsou $U^{(f)}$, $W^{(f)}$, $V^{(f)}$ resp. pro obrácený směr $U^{(b)}$, $W^{(b)}$, $V^{(b)}$. Výsledné funkce pak vypadají:

$$h_t^{(f)} = \phi_h(U^{(f)}x_t + W^{(f)}h_{t-1}^{(f)}), \quad (2.2.2.6)$$

$$h_t^{(b)} = \phi_h(U^{(b)}x_t + W^{(b)}h_{t-1}^{(b)}), \quad (2.2.2.7)$$

$$o_t = \phi_o(V^{(f)}h_t^{(f)} + V^{(b)}h_t^{(b)}). \quad (2.2.2.8)$$

Obecně pak T označuje počet jednotlivých časových kroků, v případě textu může být roven počtu znaků. Hodnoty pro zpětný stav v $t = 1$ a pro dopředný směr v $t = T$ jsou udávány konstantou 0,5. Lze je určit i v rámci učení modelu [3].

Vícevrstvá síť

Proces učení je u jednovrstvých sítí jednouchý, ale v praxi nedosahují dostatečné přesnosti. Proto se využívají častěji vícevrstvé. Ty, již podle názvu, obsahují více vrstev. Obecně budou neurony ve vyšších vrstvách získávat vstup od neuronů z nižších. Opět jsou parametry skrz jednu vrstvu stejné. Dojde k upravení funkcí, kde pro skryté stavy platí:

$$h_t = \phi_h \left(Q \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix} \right), \quad (2.2.2.9)$$

kde Q označuje po sloupcích spojenou matici $Q = [U, W]$ a v hranatých závorkách je dlouhý vektor ze vstupu a skryté vrstvy z předchozího kroku. Jelikož zde máme více skrytých vrstev, každý přechod mezi vrstvami bude mít vlastní spojenou matici s indexem daného přechodu. Označení je poté $Q^{(k)}$, kde k označuje, o jaký přechod se jedná. Pro $k = 1$ platí $Q^{(1)} = Q$ z funkce (2.2.2.9). Pro $k \geq 2$ pak platí:

$$h_t^{(k)} = \phi_h \left(Q^{(k)} \begin{bmatrix} h_t^{(k-1)} \\ h_{t-1}^{(k)} \end{bmatrix} \right). \quad (2.2.2.10)$$

Matrice Q pro $k = 1$ má velikost $p \times (d + p)$, kde p je velikost vektoru h_{t-1} a d je velikost x_t . Pro $k \geq 2$ jsou pak matice velikosti $p \times (d + p) = p \times 2p$. Pro přesun ze skryté do výstupu pak platí obdobná funkce. Obecně se používají 2 až 3 skryté vrstvy [3].

Učení

Proces učení bude popsán na nejzákladnějším algoritmu pro RNN, a to *Backpropagation Through Time* (BPTT). Z názvu již vyplívá, že se jedná o algoritmus využívající zpětnou propagaci modelem. Prochází se od výstupu ke vstupu pomocí pravidla o složené derivaci. Ukázka jednotlivých výpočtu na jednoduchém RNN modelu (skryté modely využívají identitu, tedy $\phi(x) = x$) jsou definována následovně:

$$h_t = Wh_{t-1} + Ux_t, \quad (2.2.2.11)$$

$$o_t = Vh_t, \quad (2.2.2.12)$$

kde $x_t \in \mathbb{R}^d$ je vstupní vektor, $h_t \in \mathbb{R}^h$ je skrytý stav, $o_t \in \mathbb{R}^q$ je výstupní vektor, $U \in \mathbb{R}^{h \times d}$, $W \in \mathbb{R}^{h \times h}$ a $V \in \mathbb{R}^{q \times h}$ jsou matice vah parametrů jednotlivých převodů. Účelová funkce popisující ztrátu skrz T časových kroků je tedy:

$$L = \frac{1}{T} \sum_{t=1}^T l(o_t, y_t), \quad (2.2.2.13)$$

kde $l(o_t, y_t)$ označuje ztrátu za časový krok t , y_t je množina výstupů z trénovacích dat (správná hodnota, kterou se model snaží produkovat).

Trénování tohoto modelu probíhá výpočtem parciálních derivací pro jednotlivé parametry, tedy $\frac{\partial L}{\partial U}$, $\frac{\partial L}{\partial W}$ a $\frac{\partial L}{\partial V}$. První se vypočítá derivace podle výstupu:

$$\frac{\partial L}{\partial o_t} = \frac{\partial l(o_t, y_t)}{T \cdot \partial o_t} \epsilon \mathbb{R}^q. \quad (2.2.2.14)$$

Díky tomu se dá vyjádřit i první hledanou parciální derivaci, a to na výstupu modelu:

$$\frac{\partial L}{\partial v} = \sum_{t=1}^T \frac{\partial L}{\partial o_t} h_t^\top. \quad (2.2.2.15)$$

Pro další gradienty je potřeba vypočítat parciální derivaci podle skryté vrstvy. Ta bude mít jiný počet členů na základě, kde se počítá (podle toho v jakém časovém kroku). Obecně ho lze pro $1 \leq t \leq T$ vypočítat jako:

$$\frac{\partial L}{\partial h_t} = \sum_{i=t}^T (W^\top)^{T-i} V^\top \frac{\partial L}{\partial o_{T+t-i}}. \quad (2.2.2.16)$$

Následně lze pak vyjádřit dvě poslední parciální derivace, které procházejí skrz skryté stavy h_1, \dots, h_T . Ty mají pak tvar:

$$\frac{\partial L}{\partial U} = \sum_{t=1}^T \frac{\partial L}{\partial h_t} x_t^\top, \quad (2.2.2.17)$$

$$\frac{\partial L}{\partial W} = \sum_{t=1}^T \frac{\partial L}{\partial h_t} h_{t-1}^\top. \quad (2.2.2.18)$$

Toto řešení má však problémy. Přesněji jsou u výpočtu parciální derivace podle skrytých vrstev (funkce (2.2.2.17) a (2.2.2.18)). Zde se může počítat s velkými maticemi W^\top . Pokud jsou vlastní čísla menší než 1 tak mizí (nazýváno problém mizejícího gradientu) nebo pokud je větší jak 1 tak diverguje (problém explodujícího gradientu). Může se tedy jednat o numericky nestabilní řešení. To se dá vyřešit zkrácením sum ve vzorci k nějakému menšímu číslu. Vytvoří se aproximace na výsledný gradient [4] [5].

Pomocí těchto parciálních derivací poté zoptymalizujeme použitím speciálních metod. Příkladem je SGD nebo ADAM.

2.2.2.2 Jiné modely

RNN mají mnoho způsobu modelování architektury sítě, z jiných oborů zaměření. V této části se budou rozebírat některé z používaných.

Long Short-Term Memory (LSTM)

Problémy BPTT pak lze obejít využitím metody *Long Short-Term Memory* (pro učení používá upravenou BPTT-T, se zkrácenými sumami). Návrh se podobá logickým branám v počítači (graficky je znázorněna na obrázku 4). Využívá tzv. paměťové buňky (*memory cell*). Ty mají podobné vlastnosti jako má skrytý stav, s přídavkem uložení informace. Pro správu těchto buněk se využívají brány. Výstupní brána načítá data z buňky, vstupní brána rozhoduje, kdy načítat daná data a brána zapomnění resetuje stav v dané buňce.

Do jednotlivých neuronů pak vchází vstupní vektor a skrytý stav z předchozího kroku, ty procházejí přes zmíněné brány. Data zpracovávají 3 plně propojené vrstvy s logistickou aktivační funkcí. Výstupy z bran jsou hodnoty z intervalu (0, 1).

Definujeme je matematicky. Skrytých jednotek je h , velikost části dat je n a počet vstupů je d . Vstupní brána, brána zapomnění a výstupní brána jsou v pořadí $I_t \in \mathbb{R}^{n \times h}$, $F_t \in \mathbb{R}^{n \times h}$ a $O_t \in \mathbb{R}^{n \times h}$. Funkce pro brány jsou pak definovány následovně:

$$I_t = \sigma(X_t U_i + H_{t-1} W_i + b_i), \quad (2.2.2.19)$$

$$F_t = \sigma(X_t U_f + H_{t-1} W_f + b_f), \quad (2.2.2.20)$$

$$O_t = \sigma(X_t U_o + H_{t-1} W_o + b_o), \quad (2.2.2.21)$$

kde vstupní data jsou $X_t \in \mathbb{R}^{n \times d}$, skrytý stav z minulého kroku je $H_{t-1} \in \mathbb{R}^{n \times h}$, brány mají pak váhy parametrů pro jednotlivé přesuny informací, jedná se o $U_i, U_f, U_o \in \mathbb{R}^{d \times h}$ a $W_i, W_f, W_o \in \mathbb{R}^{h \times h}$. Dále $b_i, b_f, b_o \in \mathbb{R}^{1 \times h}$ jsou parametry zkreslení. Indexy pak označují, pro kterou z bran dané matice patří.

Dále je součástí struktury paměťová buňka. Nejdříve definujeme kandidáta na paměťovou buňku. Má označení $\tilde{C}_t \in \mathbb{R}^{n \times h}$. Jedná se o obdobnou část buňky, jako jsou brány. Avšak využívá hyperbolický tangens (hodnota je tedy z intervalu (-1, 1)). Přesněji je definována:

$$\tilde{C}_t = \tanh(X_t U_c + H_{t-1} W_c + b_c), \quad (2.2.2.22)$$

kde obdobně jako u bran $U_c \in R^{d \times h}$ a $W_c \in R^{h \times h}$ jsou váhy parametrů a $b_c \in R^{1 \times h}$ parametr zkreslení.

Jejím využitím, společně se vstupní bránou a bránou zapomnění, pak ovládáme, kolik dat využijeme (\tilde{C}_t) a kolik zachováme z paměťové buňky $C_{t-1} \in \mathbb{R}^{n \times h}$ z předešlého časového kroku (využívání brány F_t). To je docíleno funkcí:

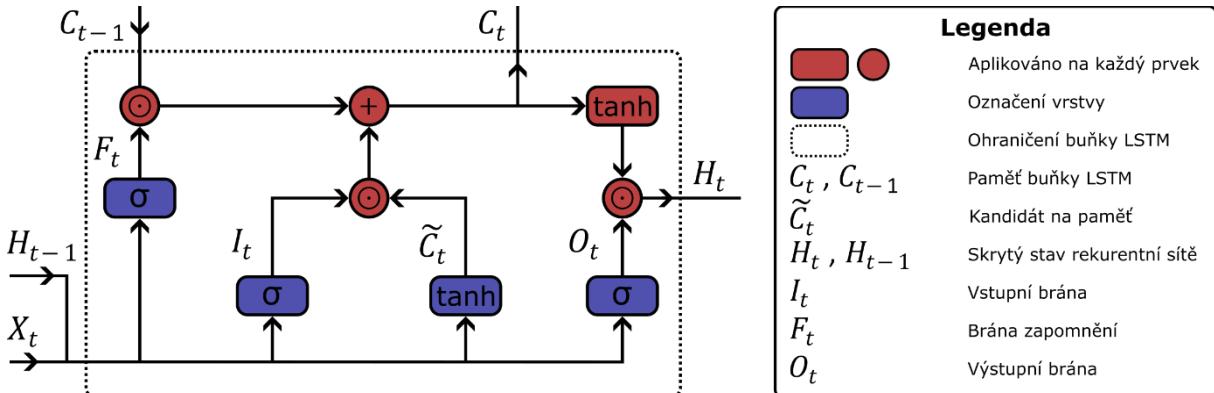
$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t. \quad (2.2.2.23)$$

Symbol \odot označuje násobení po složkách. Pokud bude brána zapomnění okolo 1 a vstupní brána je 0, pak se paměťová buňka předává celá (v neuronu stávajícího kroku).

Poslední je výpočet pro skrytý stav. Ten vypadá následovně:

$$H_t = O_t \odot \tanh(C_t). \quad (2.2.24)$$

Pokud je hodnota z výstupní brány kolem 1 předá se (skoro) celá paměť přijatá z předešlého skrytého stavu. Když je blízko 0, tak zase převezme informace pouze z paměťové buňky [5].



Obrázek 4: Schéma toku informací skrz buňku LSTM

Gated Recurrent Units (GRU)

GRU jsou novější než LSTM a berou se jako jeho zjednodušená verze. Nevyžívá totiž 3 brány, ale jenom 2. Nemá tedy interní paměť jako má LSTM. Díky tomu je rychlejší její trénování a zároveň však nedochází k významnému poklesu přesnosti.

Základem jsou tedy 2 brány: resetovací brána a aktualizační brána. Tyto brány mají jako aktivační funkci nastavenou logistickou (výstup z nich je mezi 0 a 1). Resetovací brána pak ovládá na jak velkou část minulého stavu si budeme chtít „zpomenout“ a aktualizační brána zase jak hodně se ze starého stavu okopíruje do nového. Brány jsou plně propojené sítě.

Matematicky pak resetovací bránu $R_t \in \mathbb{R}^{n \times h}$ a aktualizační bránu $Z_t \in \mathbb{R}^{n \times h}$ v časové kroku t lze definovat:

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r), \quad (2.2.2.25)$$

$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z), \quad (2.2.2.26)$$

kde n je počet pozorování, d je počet vstupů, $X_t \in \mathbb{R}^{n \times d}$ jsou vstupní data, skrytý stav je označen $H_{t-1} \in \mathbb{R}^{n \times h}$ (h v případě dimenze reálných čísel označuje počet skrytých jednotek), váhy pro různé přechody (podle indexu) jsou pak $W_{xr}, W_{xz} \in \mathbb{R}^{d \times h}$ a $W_{hr}, W_{hz} \in \mathbb{R}^{h \times h}$ (index h u matic W označuje pro jaký typ operace se používá, tedy přechody informací ze skrytého stavu), posledními jsou pak vektory $b_r, b_z \in \mathbb{R}^{1 \times h}$, které označují parametry zkreslení.

V LSTM existuje kandidát na paměťovou buňku, zde ho „nahrazuje“ kandidát na skrytý stav \tilde{H}_t . Ten je provázán s resetovací branou následovně:

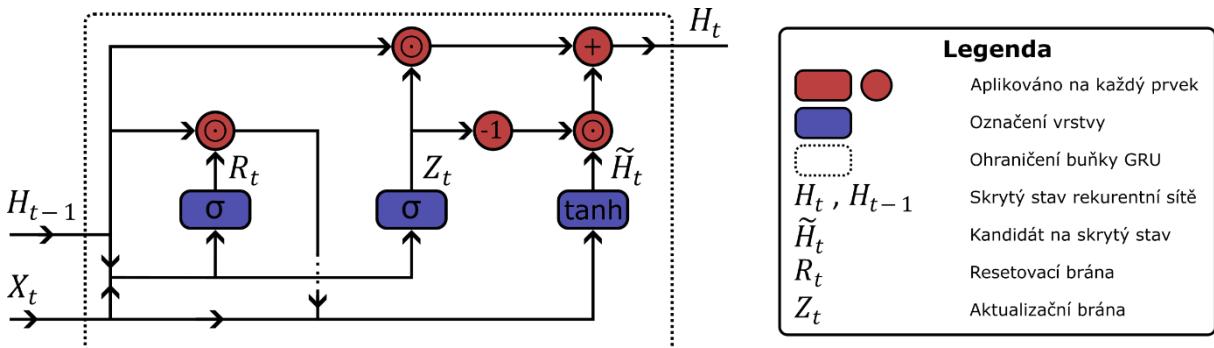
$$\tilde{H}_t = \tanh C x_t W_{xh} + (R_t \odot H_{t-1}) W_{nh} + b_h, \quad (2.2.2.27)$$

kde $W_{xh} \in \mathbb{R}^{d \times h}$ a $W_{nh} \in \mathbb{R}^{h \times h}$ jsou matice vah parametrů a b_h opět zkreslení. Operátor \odot označuje Hadamardův součin. Tento vztah pak ukazuje, jak díky součinu po složkách resetovací brána kontroluje, kolik se z předchozího stavu využije. Pokud je brána hodnotou blízko 1, využije se stav jako při struktuře normální RNN. Pokud je blíže 0, pak se jedná o MLP s klasickým vstupem, tedy nevyužije se skrytý stav a resetuje se.

Využitím kandidáta a aktualizační brány pak výsledný skrytý stav je získána jako:

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t. \quad (2.2.2.28)$$

Pokud se hodnota aktualizační brány pohybuje blízko 1, použije se celý skrytý stav z předchozího kroku. Když je blízko 0, tak se přistoupí k použití kandidáta vytvořeného z rovnice (2.2.2.28) [5].



Obrázek 5: Schéma toku informací skrz buňku GRU

2.2.3 Konvoluční sítě

Obecně (například využitím *MLP*) se data s 2 dimenzemi, to je popis obrázku například (pokud by byl barevný tak i 3 dimenze), museli transformovat na 1 dimenzi a až poté tvořit učební model. Tím se ale ztratí část dat (informace o okolních pixelech v obrázku např.). Konvoluční neuronové sítě (*Convolutional Neural Networks*, zkráceně *CNN*) jsou koncipovány pro práci s daty v mřížce, tedy zachovají se i dodatečné informace ukryté v jejich struktuře.

2.2.3.1 Základní struktura

Hlavní vlastnost CNN již prozrazuje název – konvoluce. Obecně se jedná o označení operace mezi dvěma funkcemi. Výsledkem je pak funkce, která v rámci mezích, reflektuje vlastnosti původních funkcí. Operace je pak dána v 1 dimenzi následovně pro spojitý případ:

$$g(x) = f(x) \odot h(x) = \int_{-\infty}^{\infty} f(s)h(x-s) ds, \quad (2.2.3.1)$$

kde $f(x)$, $g(x)$ a $h(x)$ jsou funkce a s je fiktivní proměnná pro účely integrace (zpravidla mezi 0 a 1). A pro diskrétní:

$$s(t) = (x \cdot w)(t), \quad (2.2.3.2)$$

kde x je vstup, w se nazývá jádro (*kernel*), také zvané filtr (někdy se názvy liší na základě dimenze) a $s(t)$ se nazývá mapa znaků/rysů či mapa jádra. V dalších částech práce pak bude používán název filtr. Pro časové řady (s diskrétním časem) by se následující vztahy formulovaly jako:

$$s(t) = \sum_{-\infty}^{\infty} x(a)w(t-a). \quad (2.2.3.3)$$

V praxi data mají více dimenzí, tyto vztahy by se pouze poupravily na základě počtu těchto dimenzí (počet integrálů, počet sum atp.) [6].

Filtr pak označuje malé matice. Vytváří se během učícího procesu a jsou upravovány následnou zpětnou propagací. Jeho rozměry jsou důležitým faktorem pro určení, jaký typ znaků filtr bude vyhledávat [7].

Byl zde zmíněn termín mapa znaků. To je jedna z klíčových částí modelu. Tyto váhy poskytuje obecný popis toho, jak intenzivně se jednotlivé regiony dat shodují s aplikovanými filtry. Každá hodnota v mapě odpovídá informaci z aktivovaného neuronu popisující část vstupu (např. obrázku). Každý filtr generuje kanál (hloubka v obecném případě s maticemi) mapy, ten pak popisuje daný vzor/vlastnost dat. Pokud je v jedné vrstvě více filtrov, tyto kanály se sdružují a vytvoří mapu, která popisuje různé znaky vstupu. Matematicky se proměnná z mapy znaků definuje:

$$y(i, j, k) = \sum_{l=1}^F \sum_{m=1}^F \sum_{n=1}^{C(i,n)} w(l, m, n, k) x(i + l - 1, j + m - 1, n) + b(k), \quad (2.2.3.4)$$

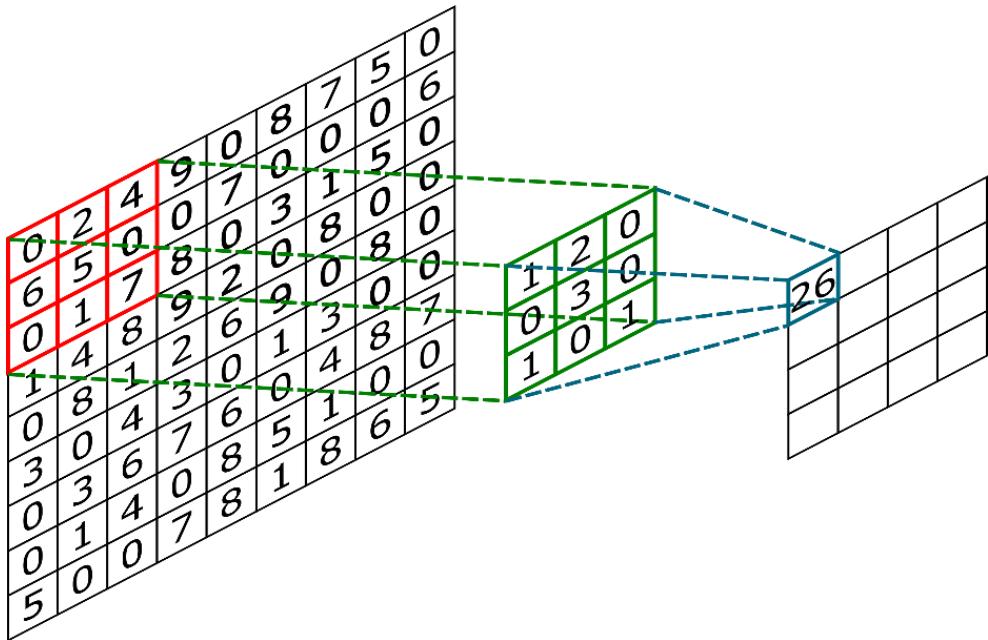
kde $y(i, j, k)$ označuje pozici (i, j) v k -té mapě znaků, $w(l, m, n, k)$ je váha k -tého filtrov na pozici (l, m, n) , $x(i + l - 1, j + m - 1, n)$ je hodnota n -tého vstupu mapy na pozici $(i + l - 1, j + m - 1)$ a $b(k)$ je hodnota zkreslení pro mapu k , F je velikost filtrov a $C(i, n)$ je počet kanálů vstupu [7].

Průběh výpočtu mapy pak probíhá sklouzaváním (*sliding*) filtrem přes vstupní data. Vypočítává se skalární součin mezi filtrem a danou oblastí dat. Výsledky se sečtou a zpracuje je nelineární (aktivační) vrstva (více popsána níže) [7].

Struktura samotná pak bude odlišná od plně propojených neuronových sítí, jako jsou MLP. CNN mají typicky vstupní vrstvu, několik konvolučních vrstev, sdružovací (*pooling*) nebo podvzorkovací (*subsampling*) vrstvy a nelineární vrstvy. Existují i další typy vrstev (některé, ty používanější ve více architekturách, jsou vyjmenovány v rozpisu níže), které se objevují jen v některých typech CNN, příkladem třeba rekurentní vrstva. Některé z vyjmenovaných vrstev se často nezmiňují jako samostatné (typicky se nelineární vrstva bere jako součást konvolučních, nepopisuje se jako samostatná součást sítě) [6].

Shrnutí vrstev:

- Vstupní vrstva:
Příprava dat pro vstup do sítě. Typicky se jedná o pole s více dimenzemi.
- Konvoluční vrstva:
Jedná se o fundamentální část celé sítě. Hlavním cílem je nalezení nejvíce zřetelné znaky v datech. Skládají se z filtrov též zvaných učebních jader (*learnable kernels*). Ty se snaží o extrakci znaků na lokální úrovni a každý filtr vypočítává mapu znaků. Pořadí znaků, které se snaží každá konvoluční vrstva rozpoznat, je od nejjednodušších (hrany, rohy, čáry apod.) po složitější. Filtry pak konvolují na různých vrstvách vstupních dat pomocí posouváním po kroku (*stride*, počet sloupců a řádků, tedy i vícedimenzionální, část či region dat, jedná se o posun filtrov) za časovou jednotku [6].



Obrázek 6: Prostorová reprezentace operace konvoluce pomocí jádra

- Nelineární (aktivační) vrstva:

Tato vrstva provádí nelineární transformaci. Přináší tedy nelineárnost pro lineární aktivační funkce a dovoluje učení složitějších modelů. Standardními funkcemi jsou hyperbolický tangens, logistická funkce nebo rektifikovaná lineární jednotka (ReLU). Základem ReLU je aplikace funkce $\max(0, x)$ [6].

- Sdružovací a vzorkovací vrstva:

Snižuje rozlišení předešlých map znaků kompresí znaků a tím zmenšuje složitost sítě. Další funkcí sdružování vrstev je zvyšování odolnosti vůči šumu a malým změnám pro již naučené znaky. Tím se docílí zaměření na nejdůležitější vzory

1	3	8	7
9	5	2	1
2	8	2	3
3	2	5	9

Krok: 2x2
Funkce: max



9	7
8	9

Obrázek 7: Ukázka postupu sdružovací vrstvy pomocí maximalizační funkce

v datech. Výsledkem je pak verze vstupu se sníženým vzorkováním a méně komplexní mapa znaků s méně parametry pro další vrstvy. Existuje více přístupů, jedním z nich je, že se jako nová hodnota vezme maximum ze čtvercové mřížky dat, další by bylo používání průměru [6].

- Normalizační vrstva:

Normalizuje dávku (batch) z předchozí (myšleno pořadí v síti) vrstvy. Toho docílí odečtením průměru a vydelením směrodatnou odchylkou. Napomáhá to snížení vnitřního kovariantního posunu a zlepšuje konvergenci [7].

- Vrstva odpadnutí (*dropout*):

Tato vrstva náhodně odstraní část neuronů z předchozí vrstvy (opět podle pořadí v síti). Pomáhá to proti jevu zvaného overfitting neboli přeучení, tedy model příliš opisuje tréninková data [7].

- Plně propojená vrstva:

Jedná se o vrstvu s architekturou obdobnou klasické neuronové sítě. Propojuje všechny neurony mezi vrstvami. Typicky používána na konci sítě pro vytvoření požadovaného výstupu [7].

2.2.3.2 Učení

Postup učení (trénink na datech) je obdobný jako u RNN (resp. princip je podobný pro všechny neuronové sítě, obecně je učení u RNN speciální varianta zpětné propagace). Základním přístupem je zpětná propagace (existují i upravené verze pro CNN) a výsledek pak projde optimalizačním algoritmem (SGD, ADAM, ...).

Zpětnou propagaci u CNN provádíme na výstupu ze sítě, kde propočítáme chybu oproti datům. Pak propočítáme parciální derivace a využijeme pravidlo o složené derivaci. Tím nám pak vznikají rovnice:

$$\frac{\partial L}{\partial w_{ij}^{(l)}} = \frac{\partial L}{\partial z_i^{(l)}} \cdot \frac{\partial z_i^{(l)}}{\partial w_{ij}^{(l)}}, \quad (2.2.3.5)$$

$$\frac{\partial L}{\partial b_i^{(l)}} = \frac{\partial L}{\partial z_i^{(l)}} \cdot \frac{\partial z_i^{(l)}}{\partial b_i^{(l)}}, \quad (2.2.3.6)$$

kde L je účelová funkce, $w_{ij}^{(l)}$ je váha neuronu i ve vrstvě $l - 1$ spojeného s neuronem j ve vrstvě l , $b_i^{(l)}$ zkreslení neuronu i ve vrstvě l , $z_i^{(l)}$ je vážená suma vstupů do neuronu i ve vrstvě l [7].

2.2.3.3 Pokročilé architektury

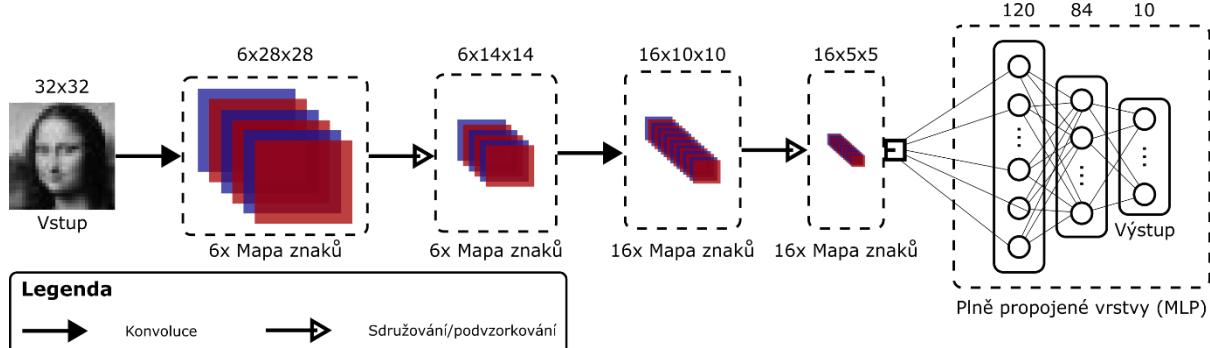
Konvoluční neuronové sítě se, stejně jako celé odvětví, vyvíjí a objevují se stále nové variace. Není tedy možné vyjmenovat všechny. V této části bude pouze výběr z nich.

LeNet

Jedná se o jednu z prvních plně funkční konvoluční sítí. Vznikla v 90. letech minulého století. Má jednu vstupní, 4 skryté a 3 výstupní vrstvy. Mezi skryté vrstvy patří 2 konvoluční a 2 sdružovací. Konvoluční vrstva poté obsahuje i logistickou aktivační funkci (tedy ji nedefinují jako samostatnou vrstvu, ale jako součást konvoluce). Sdružování se vykonává skrze průměry. Filtry jsou rozměrů 5×5 . Ty transformují data do dvoudimenzionálních map znaků, jejich počet pak v případě obrazu koresponduje s počtem kanálů (většinou). První konvoluční vrstva má 6 výstupních kanálů a druhá 16. Sdružovací vrstvy jsou 2×2 (krok je poté 2). Tím snižují dimenzi o faktor 4 (prostorové podzorkování).

Pro vytvoření požadovaného výstupu je potřeba projít plně propojenými sítěmi. Vezme se tedy čtyřdimenzionální vstup a transformuje se na 2 dimenze, první označuje index v dávce (*batch*), přes kterou toto zmenšování je prováděno, a druhý obsahuje vektor reprezentující

dané pozorování. Výstupní vrstvy jsou 3 a to s 120, 84 a 10 neurony (poslední odpovídá počtu výstupních tříd, jelikož se jedná o klasifikaci) [5].



Obrázek 8: Schéma celé sítě LENET-5

AlexNet

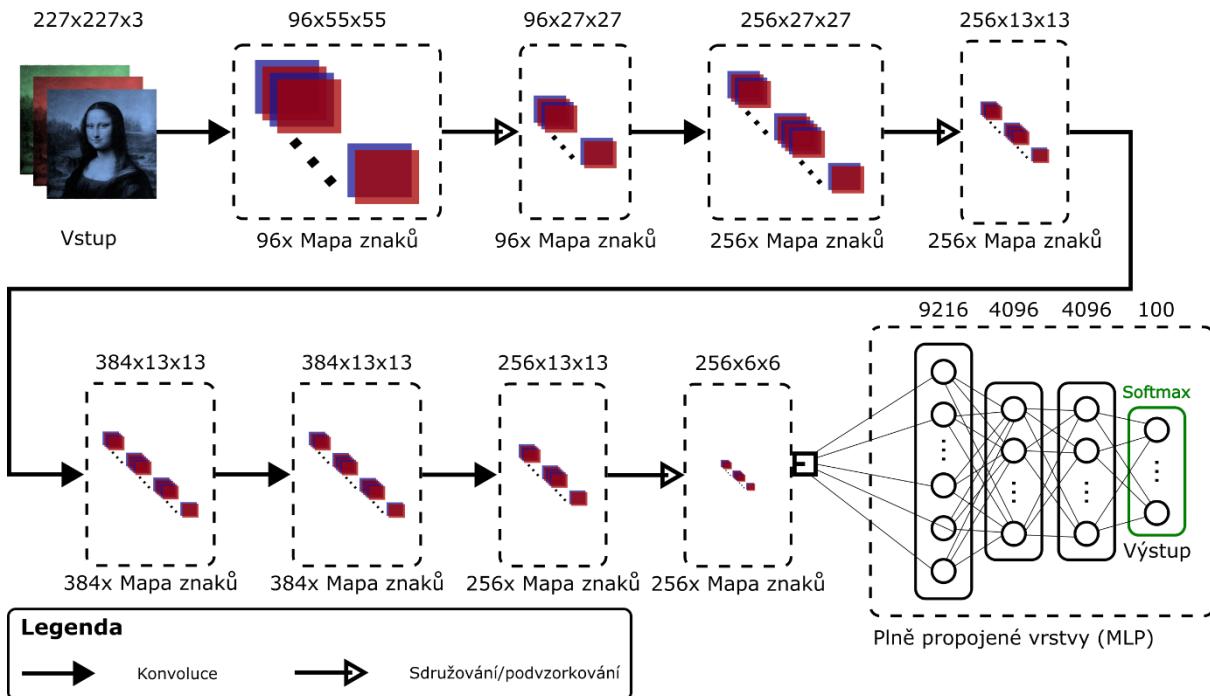
CNN architektura nebyla vždy tak hojně používán jako dnes. Problémy byly jak v množství dat, tak i v síle hardwaru. Konvoluční sítě tak nebyly těmi nejsilnějšími přístupy v oblastech jako bylo rozpoznávání obrazu (opakem je dnešní využívání především na práci s obrazem). To se zlomilo s rokem 2012, kdy AlexNet vyhrál *ImageNet Large Scale Visual Recognition Challenge*. Tím rozobil představy, že učení předem daných vlastností, je lepší než schopnost počítače si dané vlastnosti či znaky najít sám.

Struktura se často srovnává se sítí LeNet díky její podobnosti, má však daleko složitější strukturu. Původní vědecká práce, představující tento model, využívala 2 grafické karty a tím pádem byla částečně paralelní, rozdělovali se data mezi ně. Na Obrázku 9 je zobrazena lineárnější verze.

Její struktura se dá shrnout do 1 vstupní, 8 skrytých a 3 výstupních vrstev (na obrázku 9 je vyobrazena v grafické podobě). Ve skrytých je poté 5 konvolučních a 3 sdružovací (opět se aktivační vrstva bere jako součást konvoluční). Aktivační funkci nepoužívá logistickou ale *ReLU*. Ta dokáže násobně zvýšit rychlosť učení (na počet jednotlivých epoch) a je velice rychlá na výpočet (oproti logistické se nemusí počítat mocniny Eulerova čísla). Dále v tomto modelu používá (není to tedy jako samostatná vrstva, princip je stejný jako vrstva odpadnutí) odhadování některých hodnot. V původním článku nastavili pravděpodobnost 0,5 na to že výstup neuronu ve skryté vrstvě bude 0, tedy se vynechá. Ty se pak dále neberou v potaz při dalším procházení modelem (tedy princip *dropout*). Toto odhadování bylo použito v prvních dvou plně propojených sítích v modelu (tedy druhá a třetí vrstva od konce, obě jsou součástí výstupu).

První konvoluční vrstva má filtr (jádro) velikost 11×11 s krokem 4 pixely. Druhá má filtry velikosti 5×5 , třetí až pátá zase 3×3 . Mezi první a druhou, druhou a třetí, pátou a výstupem (myšleno MLP na konci modelu) jsou sdružovací vrstvy 3×3 s krokem 2.

Sdružování je pak zahrnuto i s překřížením jednotlivých sdružovaných oblastí. Výstup je pak složen ze dvou plně propojených sítí s 4096 výstupy. Kvůli tomu se také dříve rozdělovaly na více grafických karet. Poslední je poté s 1000 výstupy (popřípadě požadovaný výstup) [5] [8].



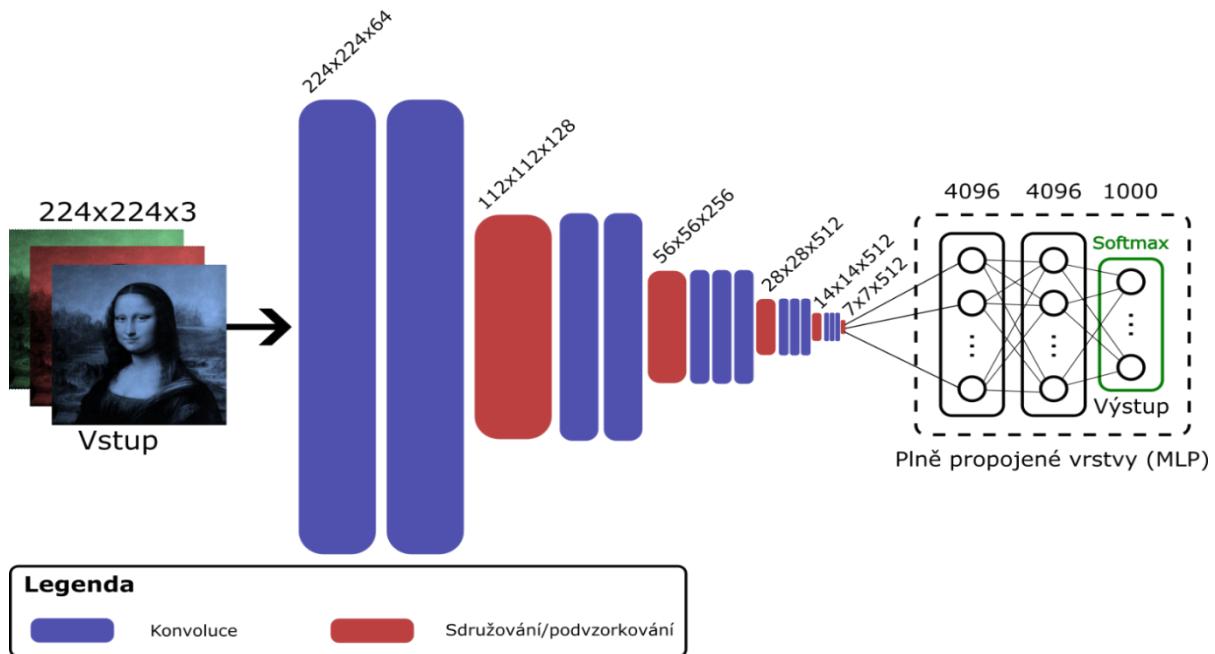
Obrázek 9: Schéma sítě ALEXNET

VGG

Zkratka a název pro tento přístup tvorby modelů vychází z názvu skupiny *Visual Geometry Group*. Jejich práce byla první, co se zaměřila na nápad využití bloků. Ty se pak dají využít v učení v různých cyklech.

Základní stavební blok (část) má následující části:

- Konvoluční vrstvu s nastaveným vyplňováním (padding) pro zachování rozlišení
- Nelineární vrstva (typicky ReLU)
- Sdružovací vrstva (typicky maximalizační, redukuje rozlišení)



Obrázek 10: Schéma sítě VGG-15

Tento přístup má i své limitace, a to se promítá do maximálního počtu konvolučních vrstev. To je nastaveno z důvodu rychlé ztráty dimenzí v datech. Jedná se o číslo $\log_2 d$ (kde d označuje dimenzi). Původní práce se zaměřovala na užití více takovýchto bloků a zda je síť lepší větší do šířky nebo do délky. Ta ukázala, že aplikace dvou 3×3 konvolučních filtrů má podobnou úspěšnost jako jeden 5×5 , zároveň však potřebují méně parametrů (5×5 potřebuje zhruba jako tři 3×3). Obecně se pak skládání 3×3 filtrů stalo standardem. Z toho vyplývá, že bloky jsou tvořeny řadou 3×3 filtrů s hodnotou výplní (*padding*) 1 a 2×2 maximalizační sdružovací vrstva s krokem 2.

Síť jako taková má dvě hlavní části. Jednou jsou samotné bloky a druhou výstupní vrstvy (opět jsou to tři plně propojené sítě, celou síť je možnost vidět na obrázku 10). Často se bloky definují počtem konvolučních vrstev a výstupních vrstev [5].

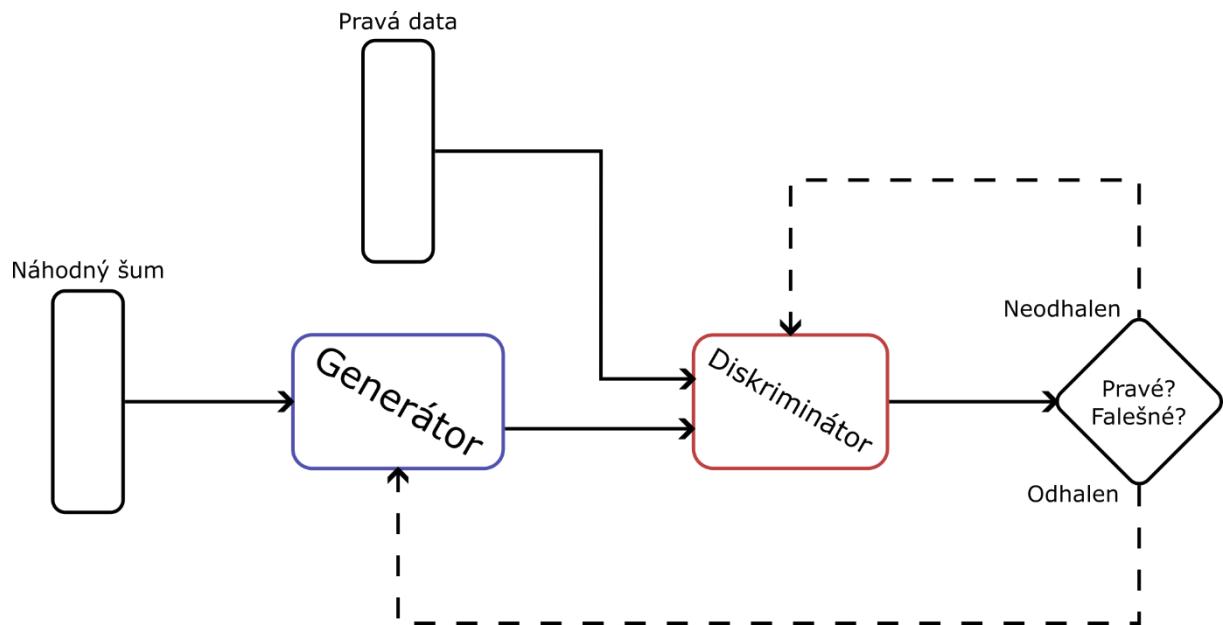
3 GAN sítě

Plným názvem zvané Generative Adversarial Networks. V překladu se jedná o generativní nepřátelské sítě. Jedná se o pokročilou partii hlubokého učení, která se datuje do roku 2014. Nejdříve je dobré vysvětlit, co jsou generativní a diskriminativní modely, jelikož s GAN sítěmi souvisí.

Diskriminativní modely se přímo snaží odhadnout hodnotu pravděpodobnosti $P(y|\bar{X})$ třídy značení y pomocí hodnot \bar{X} . Příkladem může být logistická regrese. Generativní modely, na druhou stranu, odhadují pravděpodobnost $P(\bar{X}|y)$. Tedy snažíme se z dat vytvořit model, který dokáže zachytit všechny jejich různé vlastnosti, aby pak následně vytvořil umělá data, která budou ze stejného rozdělení. Obecně se jedná o modely jak pro učení s učitelem, tak bez něj, na rozdíl od diskriminativních [3] [5].

3.1 Struktura

Základem GAN modelů je princip, že pokud generátor vytvoří data, které nedokážeme rozpoznat od originálu, potom je vhodný jako model. Toho pak dosahuje využitím výhod diskriminativních modelů pro tvorbu generativních modelů. Přístup by se dal přirovnat ke dvouvýběrovému testu ze statistiky (rozhoduje, zda data pochází ze stejného rozdělení). GAN sítě jdou na to více konstruktivně. Máme jeden model, který generuje data a druhý na základě stejných tréninkových dat (společně s daty z právě zmíněných testů) pak rozpoznává, zda se jedná o původní data. Oba tyto modely – sítě spolu soupeří (adversial v názvu). Pokaždé, když se generátor pokusí obelstít diskriminátora, se na základě těchto falešných dat upraví (vylepší formou učení). Zároveň však generátor tyto informace využije také [5].



Obrázek 11: Ilustrační schéma GAN, celá čára označuje tok informací, čárkovaná označuje změnu gradientů – změny modelů

3.2 Matematická formulace a algoritmus

Diskriminátor je binární klasifikátor, který má za úkol rozeznat, zda je vstup x ze skutečného souboru dat, nebo se jedná o falsifikát z generátoru. Nejčastěji je jeho výstupem skalární veličina $o \in \mathbb{R}$. Pro určení předpovídáné pravděpodobnosti se využije logistická funkce. Po dosazení tedy platí pro výpočet pravděpodobnosti $D(x)$:

$$D(x) = \frac{1}{1 + e^{-o}}. \quad (3.2.1)$$

Označíme y třídu značení, zdali jsou data pravá (hodnota 1) nebo falešná (hodnota 0). Potom diskriminátor má při svém učení za cíl minimalizovat křížovou entropii:

$$\min_D \{-y \log D(x) - (1-y) \log(1 - D(x))\}. \quad (3.2.2)$$

Generátor nejdříve vytvoří tzv. latentní proměnné z . Jedná se o náhodná čísla z normálního rozdělení, tedy $z \sim N(0,1)$. Poté vygeneruje podle funkce G falešný vstup $x' = G(z)$. Cílem je obestíti diskriminátor, tedy chceme, aby platilo $D(G(z)) \approx 1$. Jinak řečeno, upravujeme parametry generátoru G tak aby byla maximální hodnota křížové entropie v hodnotě $y = 0$. Matematicky lze vyjádřit jako:

$$\max_G \left\{ -(1-y) \log (1 - D(G(z))) \right\} = \max_G \left\{ -\log (1 - D(G(z))) \right\}. \quad (3.2.3)$$

Pokud generátor tvoří správně, poté je splněno $D(G(z)) \approx 1$ a křížová entropie se blíží 0. To ale vytváří velice malé gradienty (problém srovnatelný s mizejícími gradienty při učení neuronových sítí) a malý postup učení pro diskriminátor. Proto se upravuje rovnost na minimalizační:

$$\min_G \left\{ -y \log (D(G(z))) \right\} = \min_G \left\{ -\log (D(G(z))) \right\}, \quad (3.2.4)$$

kde se falešné x' dává jako vstup do diskriminátoru s označením $y = 1$. Pokud to spojíme dohromady, vznikne minimax hra mezi diskriminátorem a generátorem:

$$\min_D \max_G \left\{ -E_{x \sim p_{Data}} \log(D(x)) - E_{z \sim p_{Gen}} \log(1 - D(G(z))) \right\}, \quad (3.2.5)$$

kde p_{Data} a p_{Noise} jsou rozdělení pro data a latentní proměnné (nebo také šum) [5]. Původní definice od autorů původního článku je lehce odlišná (v podstatě se jedná jenom o jinou variantu stejného řešení):

$$\min_G \max_D \left\{ E_{x \sim p_{Data}} \log(D(x)) + E_{z \sim p_{Gen}} \log(1 - D(G(z))) \right\}. \quad (3.2.6)$$

Z této funkce pak vychází i základní algoritmus. Ten je zobrazen na Algoritmu 1. Pro optimalizaci využívá stochastickou gradientní metodu.

Aby algoritmus byl korektní, je potřeba aby bylo jeho globální optimum ve tvaru $p_{Gen} = p_{Data}$, a zároveň aby k němu algoritmus konvergoval.

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k , is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$ from noise prior $p_g(\mathbf{z})$.
- Sample minibatch of m examples $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ from data generating distribution $p_{data}(\mathbf{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \left[\log D(\mathbf{x}^{(i)}) + \log (1 - D(G(\mathbf{z}^{(i)}))) \right].$$

end for

- Sample minibatch of m noise samples $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$ from noise prior $p_g(\mathbf{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log (1 - D(G(\mathbf{z}^{(i)}))).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Algoritmus 1: Princip učení GAN sítí, původní znění z [9]

Pro dokázání globálního optima je potřeba si přepsat definici problému, kdy pro jakýkoliv generátor G se určují parametry diskriminátoru D maximalizací funkce:

$$\begin{aligned} V(\sigma, D) &= \int_x p_{Data}(x) \log(D(x)) dx + \int_z p_{Gen}(z) (1 - \log(D(z))) dz \\ &= \int_x p_{Data}(x) \log(D(x)) + p_{Gen}(x) (1 - \log(D(x))) dx. \end{aligned} \quad (3.2.7)$$

Označíme si funkci uvnitř integrálu jako $f(y)$. Tato funkce v číslech $p_{Data}(x) = a$, $p_{Gen}(x) = b$, kde $(a, b) \in \mathbb{R}^2 \setminus \{0,0\}$, pak nabývá maxima (na intervalu $(0,1)$) v hodnotě $\frac{a}{a+b}$. Zároveň není potřeba aby byl diskriminátor definován mimo množinu $supp(p_{Data}) \cup supp(p_{Gen})$. Jeho optimální hodnota $(D_G^*(x))$ se poté rovná:

$$D_G^*(x) = \frac{p_{Data}(x)}{p_{Data}(x) + p_{Gen}(x)}. \quad (3.2.8)$$

Cíl tréninku diskriminátoru se dá také interpretovat jako maximalizace funkce log-likelihood pro pravděpodobnost $P(Y = y|x)$, kde Y značí, zdali se jedná o data z p_{Data} ($y = 1$) nebo jsou z p_{Gen} ($y = 0$). Definice „soupeření“ pak může mít tvar:

$$\begin{aligned} C(G) = \max_D V(G, D) &= E_{x \sim p_{Data}} \left(\log \left(\frac{p_{Data}(x)}{p_{Data}(x) + p_{Gen}(x)} \right) \right) \\ &\quad + E_{x \sim p_{Gen}} \left(\log \left(\frac{p_{Gen}(x)}{p_{Data}(x) + p_{Gen}(x)} \right) \right). \end{aligned} \quad (3.2.9)$$

Optimální případ nastává, když $p_{Gen} = p_{Data}$ a $D_G^*(x) = 0,5$ (to vychází z rovnosti rozdělení), tím pádem pak platí $C(G) = -\log 4$. To že se jedná o globální minimum se ověří pomocí odečtení rovnice

$$E_{x \sim p_{Data}}(-\log 2) + E_{x \sim p_{Gen}}(-\log 2) = -\log 4 \quad (3.2.10)$$

od výrazu $C(G) = V(D_G^*, G)$. Tím se získá vztah:

$$C(G) = -\log 4 + KL\left(p_{Data} \left\| \frac{p_{Data} + p_{Gen}}{2}\right.\right) + KL\left(p_{Gen} \left\| \frac{p_{Data} + p_{Gen}}{2}\right.\right), \quad (3.2.11)$$

kde KL označuje Kullbackovu–Leiblerovu divergenci. Jedná se o míru z matematické statistiky. Určuje, jak se odlišuje jedna distribuční funkce od druhé. Je definována následovně:

$$KL(P \| Q) = \int_{-\infty}^{\infty} p(x) \ln\left(\frac{r(x)}{q(x)}\right) dx. \quad (3.2.12)$$

Pokud se rozdělení shodují (skoro všude), pak je hodnota 0, opačně je kladná. Avšak tato metrika není symetrická, tedy nám ještě nedá požadované řešení [10]. K tomu pomůže jiná metrika, která šla vyzozorovat v rovnici (3.2.11) a to Jensenovo-Shannonova divergence. Ta má tvar:

$$JSD(P \| Q) = \frac{1}{2}KL(P \| M) + \frac{1}{2}KL(Q \| M), \quad (3.2.13)$$

kde $M = \frac{1}{2}(P + Q)$ je smíšené rozdělení P a Q . Díky tomu pak výsledná funkce má tvar:

$$C(G) = -\log 4 + 2JSD(p_{Data} \| p_{Gen}). \quad (3.2.14)$$

Tato metrika je již symetrická, kladná a má nulovou hodnotu, pokud jsou rozdělení stejná (skoro všude). Tím pádem je $C^* = -\log 4$ globální minimum $C(G)$ a jediné řešení pro $p_{Gen} = p_{Data}$, a tedy optimální řešení.

O konvergenci mluví následující předpoklad:

„Pokud G a D mají dostatečnou kapacitu a v každém kroku algoritmu diskriminátor má možnost dostát svého optima pro dané G , a p_{Gen} je aktualizováno na základě vylepšování kritéria“

$$E_{x \sim p_{Data}} \log(D_G^*(x)) + E_{z \sim p_{Gen}} \log(1 - D_G^*(x)), \quad (3.2.15)$$

potom p_{Gen} konverguje k p_{Data} [9].“

Pro důkaz tvrzení si je třeba označit si funkci $V(G, D) = U(p_{Gen}, D)$ jako funkci proměnné p_{Gen} , kdy její definice je stejná jako uvedené kritérium. Tato funkce je konvexní. Když je $f(x) = \sup_{\alpha \in \mathcal{A}} f_\alpha(x)$ a $f_\alpha(x)$ je konvexní v x pro každé α , potom $\partial f_\beta(x) \subseteq \partial f$, když se $\beta = \arg \sup_{\alpha \in \mathcal{A}} f_\alpha(x)$. To je ekvivalentní k výpočty gradientní metody pro p_{Gen} pro optimální D společně s odpovídajícím G . Již víme, že $\sup_D U(p_{Gen}, D)$ je konvexní v p_{Gen} s globálním optimem a tedy, s dostatečně malými kroky, p_{Gen} konverguje k p_{Data} . Tím bylo dokázáno jak konvergence, tak optimum daného algoritmu [9] [10] [11].

3.3 Varianty

Základní varianta má však i nevýhody. Na ty se snaží navázat stále vznikající nové varianty řešení pomocí generativních nepřátelských sítí. V této sekci budou některá přiblížena.

3.3.1 Podmíněný GAN (CGAN)

Z názvu vyplívá, že se jedná o podmínkovou verzi. Liší se pouze přidáním informace navíc (třídy označení, nazveme ji c). Generátor poté využije třídu c a přidá jí k latentní proměnným z pro zlepšení vytváření falešných vstupů $x' = G(z|c)$. Samozřejmě tuto informaci má i diskriminátor a ten ji použije na zlepšení rozpoznávání – $D(G(z|c))$. V této variantě lze ovládat generování nových dat, narozdíl od klasického GAN. Výsledná funkce, přes kterou se provádí učení je pak [12]:

$$\min_G \max_D \left\{ -E_{x \sim p_{Data}} \log(D(x|c)) - E_{z \sim p_{Noise}} \log(1 - D(G(z|c))) \right\}. \quad (3.3.1.1)$$

3.3.2 Hluboký konvoluční GAN (DCGAN)

Jedná se o třídu konvolučních GAN sítí. Jde o první síť využívající de-konvolučních neurálních sítí (*de-CNN*). Ty významně vylepšuje stabilitu trénování. Struktura má generátor tvořen konvoluční neurální sítí a diskriminátor naopak de-konvoluční. Změny jsou v CNN struktuře. Jsou odstraněny sdružovací vrstvy a nahrazeny krokovými konvolucemi, generátor i diskriminátor využívají normalizaci jednotlivých dávek (*batch normalization*). Jako aktivační funkce pak generátor využívá ReLU a diskriminátor Leaky – ReLU [12].

3.3.3 Laplacovský GAN (LapGAN)

Byl představen jako sekvenční obrázkový generátor. Kombinuje CGAN a Laplacovskou pyramidu (LP). Pyramidový tvar označuje víceúrovňovou reprezentaci obrázku, kde je obrázek několikrát vyhlazen či podvzorkován, aby se oddělily jednotlivé části (respektive se jedná o úrovně detailů). Jedná se o generování ve více vrstvách, při kterém řada GAN sítí generuje jednotlivé úrovně detailů a řadí je do LP tvaru. Laplacovská pyramida vychází z Gausianské pyramidy (ty využívá pro jednotlivé úpravy Gaussovské vyhlazování) [12].

3.3.4 GAN maximalizující informace (InfoGAN)

Hlavní zaměření je na učící proces bez učitele. Jedná se o nadstavbu GAN, která dokáže rozpoznávat diskrétní i spojité latentní faktory. Dále se dokáže přizpůsobit velikosti dat a nepotřebuje více času na trénink než klasický GAN. Hlavní účelová funkce učení má tvar:

$$\min_G \max_D \{V(D, G) - \lambda I(c; G(z, c))\}, \quad (3.3.4.1)$$

kde λ je hyperparametr učení, z označuje šum (náhodný) který nelze kompresovat, c se nazývá latentní kód a I označuje společnou informaci. Přesněji c je jedna část z původního šumu (rozdelen na nekompressovatelnou část z a popisované c), které míří na významné strukturované sémantické rysy rozdělení dat. Společná informace I je pak unikátní regularizační člen. Definovat ji lze následovně:

$$I(X; Y) = Entropy(X) - Entropy(y). \quad (3.3.4.2)$$

Podobnou variantou je pak Semi-supervised InfoGAN. Ten využívá společnou informaci mezi neoznačenými daty a umělými, kdy reprezentaci latentního kódu zvládne vytvořit z menších neoznačených datových souborů (efektivněji než klasický InfoGAN) [12] [13].

3.3.5 Energy-Based GAN (EBGAN)

Jedná se o kombinaci autoenkóderu (vícevrstvý perceptron, má stejný počet vstupů a výstupů a lichý počet skrytých vrstev) a GAN sítí. Diskriminátor jako funkce energie, kde nízká energie odpovídá pravým datům a vysoká energie poukazuje na falešná. Používá dvě různé funkce učení pro diskriminátor a generátor:

$$L_D(x, z) = D(x) + \max(0, m - D(G(z))), \quad (3.3.5.1)$$

$$L_G(z) = G(z). \quad (3.3.5.2)$$

Z těchto rovnic je zřejmé, že minimalizace L_G podle parametrů G je podobná maximalizaci L_D , kde parametry D mají stejné minimum pro kladné rozpětí a to m [12].

3.3.6 Wasserstein GAN (WGAN)

Tento přístup využívá pro účelovou funkci Wassersteinovu metriku. Oproti základním GAN sítím diskriminátor není binárním klasifikátorem, ale využívá již zmíněnou Wassersteinovu vzdálenost. Obecně trénink netrpí problémy mizejícího gradientu nebo problémy s kolapsem režimu (*mode collapse*). Metrika (také zvaná *Earth-Mover*) je definována:

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} E_{(x,y) \sim \gamma}(\|x - y\|), \quad (3.3.6.1)$$

kde $\Pi(P_r, P_g)$ je množina všech společných rozdělení $\gamma(x, y)$, $\gamma(x, y)$ označuje, jak velké množství má být přesunuto z x do y aby došlo k transformaci z P_r do P_g [12].

4 Rešerše prací

Obecně GAN sítě mají celou škálu využití, a ty se s nově vznikajícími variacemi rozšiřují. Nejčastější jsou však spojovány s oblastí fotografií a videí. Již první práce definující tento postup, myšleno *Generative Adversarial Nets* od Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville a Yoshua Bengio, prezentovala svoje výsledky na generaci různých fotografií (bylo zde více druhů, příkladem psaná čísla). I když se jedná o 10 let starou práci, hlavní zaměření těchto sítí se nezměnilo, pouze se rozšířily aplikace. Proto je tato sekce rozdělena do podkapitol na základě zaměření daných prací. Ty nezahrnují všechny možné oblasti, ve kterých se GAN využívá, ale nastní ty používanější.

4.1 Text na obrázek

Velice diskutovanými modely v oblasti umělé inteligence jsou tvorby obrázků na základě textového vstupu. Toto využití se nevyhýbá ani GAN sítím. Příkladem může být práce *DF-GAN: A Simple and Effective Baseline for Text-to-Image Synthesis*, v té použili variantu Deep Fusion GAN. Ten se skládá z diskriminátoru, generátoru a kódovače textu. První dva jmenované jsou konvoluční sítě s převzorkovacími vrstvami (za diskriminátor jsou přidané speciální prvky, které vylepšují sémantickou kvalitu) a síť LSTM na převod textu. Jako vstupní data využily soubor fotografií ptactva s popisem v 10 jazycích a soubor obecných činností s popisem v 5 jazycích. Výsledky pak porovnávali s dalšími modely (StackGAN, AttnGAN, MirrorGAN, ...) [14].

Další takovou prací je *Text to Image Generation with Semantic-Spatial Aware GAN*. Ta pak popisuje variantu SSA – GAN. V té je generátor vytvořen z SSA bloků a stejný diskriminátor jako práce předchozí. Využívá i stejně datasety a pak s ním (a dalšími) porovnává i výsledky [15].

4.2 Text na hlas

Počítačem „čtený“ text je už poměrně zavedená praxe i v samotných operačních systémech. Dlouhodobě se však jednalo o algoritmickou záležitost využívající nahrávky jednotlivých písmen, slabik, či hlásek. S postupem hlubokého učení se začaly tvořit modely se specializací i na tuto oblast. Příkladem využívající GAN sítě pak je práce *JETS: Jointly Training FastSpeech2 and HiFi-GAN for End to End Text to Speech*. Ta spojuje model FastSpeech2, který převádí text na mel-spektrogram (spektrogram na melové škály – škála tónů zaměřená na vnímání lidského ucha), a HiFi-GAN. Přesněji se HiFi-GAN skládá z jednoho generátoru a 2 diskriminátorů, jeden multiškálový a druhý víceperiodový. První zmiňovaný prozkoumává různé úrovně zvuku a snaží se zachytit po sobě jdoucí vzorce a dlouhodobé závislosti. Druhý je složen z několika menších diskriminátorů, který každý zkoumá část periodických signálů zvukové stopy [16].

Další zajímavou prací je *Diffgan-tts: High-fidelity and efficient text-to-speech with denoising diffusion gans*. Ten aplikuje model DDPM (Denoising Diffusion Probabilistic Model) aby byl využitelný v reálném čase, a jsou dostačující pouze 4 kroky pro odstranění šumu. Učení tohoto modelu se provádí pomocí 2 fází. První vytvoří jednoduchý TTS (zkratka pro *Text-to-Speech*) model, který se využije pro trénování DPPM ve druhé fázi [17].

4.3 Úprava obrázku

Toto téma pak zpracovává například práce *Spatially-Adaptive Multilayer Selection for GAN Inversion and Editing*. To využívá StyleGAN2 a přidává k němu vlastní rozpoznávání latentních vrstev pro zlepšení invertibility a možností úprav. Díky více vrstvám model lépe rozpozná, co přesně za změny a v jakých místech má provést (přidávání různých doplňků na fotografii obličeje, změna barvy včí či zvířat atp.). Jedná se o model, který invertuje obrázek pomocí předem vytrénovaného StyleGAN2 a zároveň zachovává možnost ho editovat. StyleGAN jako takový je kombinací *Progressive GAN* a *neural style transfer*. StyleGAN2 je pak vylepšená verze, které používá modulace nahledu místo instanční normalizace. Dále také má vylepšené schéma progrese učebního plánu [18].

Stejný typ modelu (myšleno StyleGAN2) využívá i práce *Drag Your GAN: Interactive Point-based Manipulation on the Generative Image Manifold*. Ta se však zaměřuje na grafický vstup uživatele, přesněji body tahu, přes které se obrázek upraví. To docílili pomocí vytvořeného modelu s učitelem, který rozpoznává, co dané tahy mají znamenat (jestli se jedná o posun, nebo třeba jinou úpravu jako je otevření apod.). Nezaměřili se pouze na tahy, ale lze přidat i masky obrázku, ve kterých se dané změny budou provádět [19].

Do této oblasti by se dali zahrnout i překlady obrázků do obrázků. Obecně se jedná také o úpravy, ale často ovlivňují celkový vzhled. Může to být různé dobarvování, dokreslování nebo vylepšování kvality. Prací zabývající se tímto druhem převodu je *Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation*. Autoři využívali StyleGAN a obohatili převod na *pixel2style2pixel*. Ten v základu převádí pomocí autoenkového obrázku na stylové vektory, které potom pokračují do StyleGAN generátoru, kde vytváří rozšířený latentní prostor. Díky tomu není potřeba aby byl obrázek vždy reprezentován v množině využívající StyleGAN (invertování a až poté úpravy) a není potřeba přesná *pixel-to-pixel* reprezentace. Ten pak zvládne celou řadu úkonů jako je: dobarvování, dokreslování, kreslení ze skici, vylepšování detailů či větší úpravy jako otočení [20].

4.4 Rozpoznávání vlastností obrázku

Odvětví rozpoznávání obrazu se spojuje s tvorbou autonomních rozpoznávacích strojů. Jako velice přínosné se to projevuje při pomoci rozpoznávání různých útvarů, například v lékařství. Tím se zabývala práce *Deep neural system for supporting tumor recognition of mammograms using modified GAN*. Pomocí varianty AGAN vytváří dodatečnou reprezentaci snímků mamografů s cílem odhalit normální či abnormální snímky neboli ty co neobsahují nebo obsahují zhoubné shluky. AGAN přesněji označuje *autoencoder-generative adversarial network*, tedy variantu spojující GAN a autoenkového (druh neuronových sítí využívající kódování dat, které zmenšuje jejich dimenzi s malou ztrátou informace) [21].

Další prací v této oblasti je *A Novel Visual Representation on Text Using Diverse Conditional GAN for Visual Recognition*. Model má pak za cíl získat různé vlastnosti ze synteticky generovaných obrázků (pomocí DCGAN), ty pak převede na textový výstup. Oba tyto výstupy, společně se vstupem do generátoru (myšleno data na kterých byl DCGAN trénován), pak vytvoří jednotný výsledek, který popisuje daný vstupní obrázek [22].

4.5 Časové řady

Jak už bylo zmíněno, práce s časovými řadami není hlavní oblast, na kterou se GAN sítě zaměřují. To však neznamená, že práce v této kategorii nevznikají a nemají dobré výsledky. Prací s tímto tématem je *Sentiment-Guided Adversarial Learning for Stock Price Prediction*. Z názvu je patrné využití sentimentálních dat, přesněji příspěvků na síti twitter.com (dnešní x.com), společně s daty burzovních cen (klasické členění na Open, High, Low a Close). Jaký sentiment jednotlivé příspěvky přináší určili pomocí předtrénovaného modelu VADER (s upraveným výstupem, jelikož neutrální zprávy nejsou pro tento typ modelu důležité). K výsledků pak připočítali váhy jednotlivých profilů (autorů příspěvků), resp. počet sledujících. Jednalo se o 9 společností, přesněji Apple, Nvidia, Electronic Arts, Eastbay, Endo International, Kandy Hotels, Starbucks, Skyworks Solutions a Cisco Systems. Časové období pak bylo od 1. 4. do 10.6. v roce 2016. Tato číselná data pročistili od velkých extrémů (nahrazením průměry s nejbližším okolím) a transformovali pomocí Fourierových transformací (tím redukovali šum a získali více sinusoidní tvar, vhodný pro RNN). Poslední úpravou pak byla přibližná normalizace (aby se podobalo normálnímu rozdělení). Hlavním modelem je GAN skládající se z LSTM generátoru a MLP diskriminátoru, případně ještě CGAN, ale pro porovnání vytvořili i další modely (LMR, KNN, ARIMA). Výstupní třídu v jednotlivých modelech se pak lišily (přesněji se pak všechny variace tvořily 36 modelů). Obecně pak nejlepších výsledků dosáhly KNN modely a CGAN modely. GAN a CGAN pak dohromady byly nejlepším modelem v 11 z 36 modelů [23].

Další prací v této kategorii je *Stock price prediction using BERT and GAN*. Opět využívá sentimentální analýzu trhu, přesněji se jedná o zprávy a jejich titulky z webu seekingalpha.com zpracované modelem finnBERT. Ostatní data jsou pak ceny z burzovního trhu, technické indikátory (SMA, BB, ...), ceny amerických indexů (NASAQ, NYSE a S&P 500), indexy některých burz (Londýn, Indie, Tokio, ...), historické ceny zásadních komodit (ropa, zlato, ...) a historické ceny některých z ostatních firem z technologické oblasti (Microsoft, Google, ...). Data jako taková pak byly normalizovány. Model samotný se pak skládal z diskriminátoru CNN a generátoru GRU. Výsledky pak byly porovnány s dalšími modely (GRU, LSTM, ...). Hodnota RMSE pro výsledný model poté činila 1,827 a v porovnání s ostatními ji měla nejmenší [24].

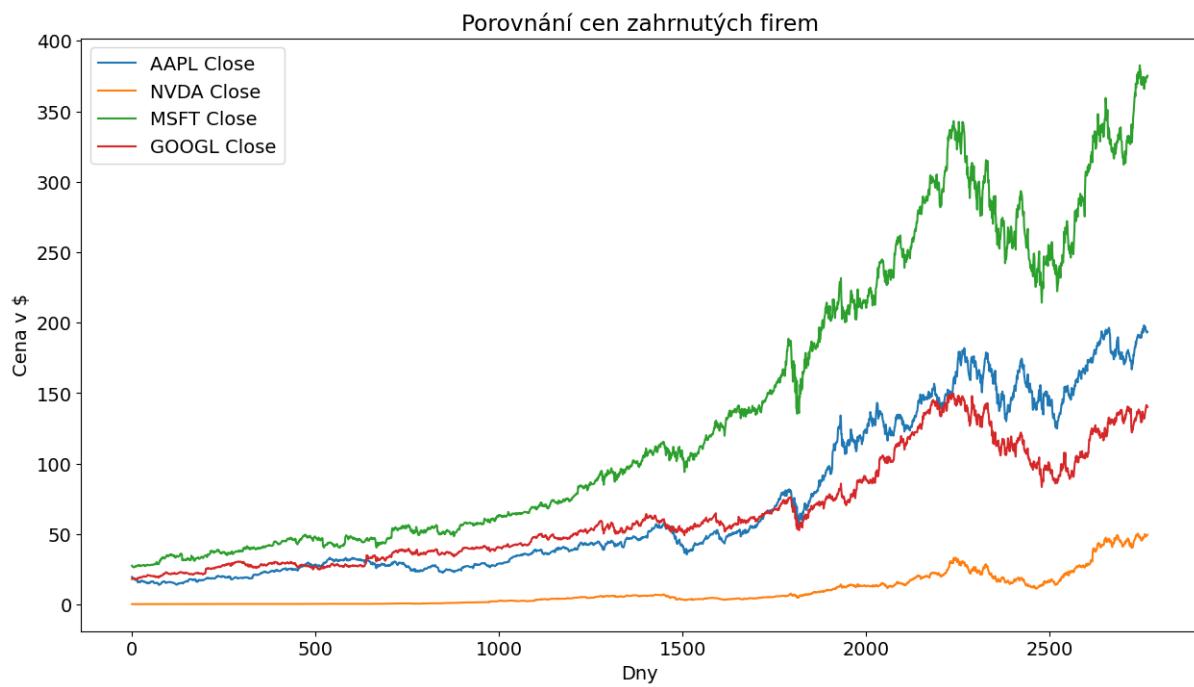
5 Data

GAN model byl použit pro předpověď pro vývoj ceny akcií Apple, akcií skupiny ČEZ, zlata a Bitcoinu. Většina z uvedených cen byla získána ze stránky *yahoo.com* nebo *investing.com*. U ostatních bude zdroj zmíněn. K obecným cenám různých akcii a komodit (kryptoměn), byly přidány hodnoty vyhledávání určitých klíčových výrazů, které poskytuje společnost Google v rámci služby *Google Trends*. Oblast dat zahrnuje období od 3.1.2013 do 28.12.2023 (hodnoty nejsou pro každý den, objevuje se zde mnoho různých akcií a komodit prodávaných na různých burzách, a tedy jiné otevírací doby). U předpovědí hladiny kryptoměny Bitcoin je období kratší, jedná se o interval od 10.3.2016 do 28.12.2023. K této datům byly pak přidány doplňkové zpožděné ukazatele, blíže budou zmíněny později v této kapitole.

5.1 Apple

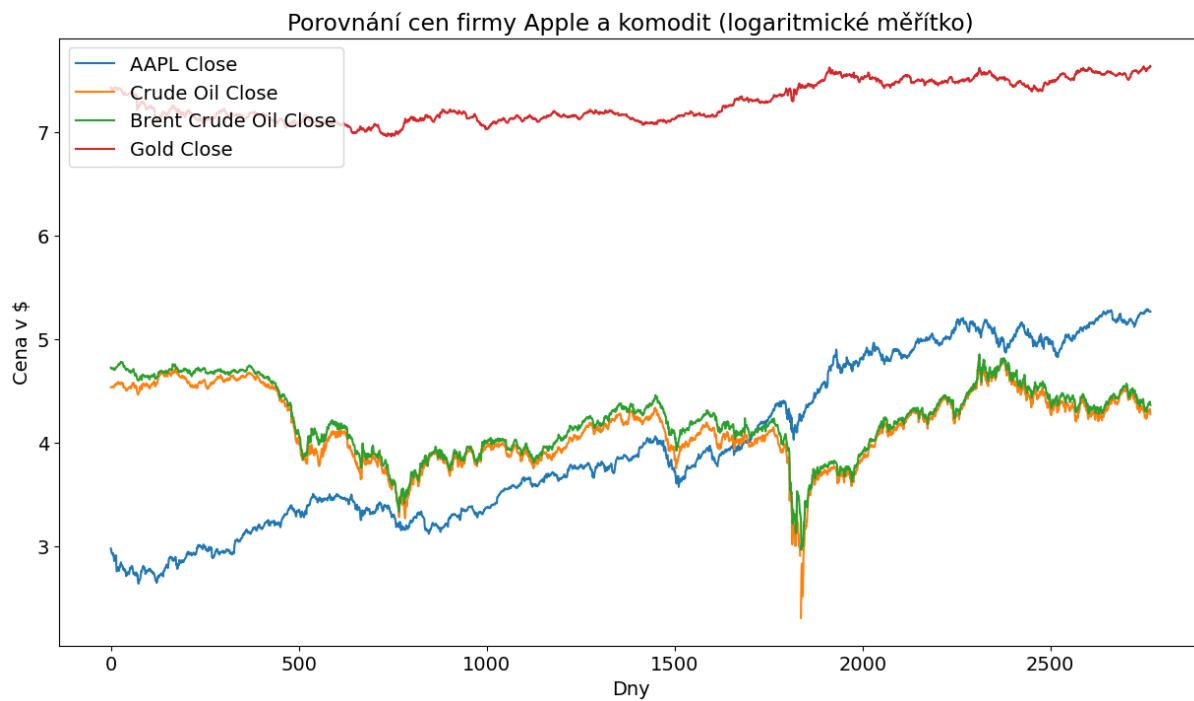
Společnost Apple je jedna ze společností s největší hodnotou na světě. Dlouhodobě se pohybuje v žebříčcích mezi prvními třemi pozicemi. Byla založena roku 1976 Stevem Jobsem a Stephenem G. Wozniakem. Jejich prvním produktem byly domácí počítače. Velice úspěšným byl počítač Apple II, který započal růst této značky. Firma vstoupila na akciový trh v roce 1980. Společnost poté pokračovala ve výrobě počítačů, avšak jejich série Mac nedosahovala takových úspěchu jako předchozí stroje. Velký vzrůst firmy nastal po návratu Steva Jobse v roce 1997 (v roce 1985 po neshodách s vedením opustil firmu a založil si vlastní s názvem *NexT*) s příchodem výdělečnějších značek jako je iMac. V pozdějších letech začala firma přicházet s revolučními nápady v odvětví technologií, díky kterým získávala na hodnotě. V roce 2001 přišla se službou iTunes a zařízením pro poslech hudby iPod, čímž oživila trh s hudebními médií. S čím je však nejvíce firma spojovaná dnes je její řada telefonů. První vydala v roce 2007, a od té doby se jedná o její nejúspěšnější značku, která dodnes ovlivňuje její cenu na trhu (s vydáním či ohlášením nové řady se spojuje i zvýšení ceny akcií). V přítomnosti se firma zabývá celou škálou technologických a softwarových výrobců a služeb. Stále vyrábí osobní počítače iMac, tablety iPad, provozuje cloudové úložiště iCloud, zprostředkovává platby skrz platební bránu ApplePay a mnoho dalších [25].

Dlouhodobě se jedná o firmu s rostoucí základnou zákazníků a je proto vhodná jako investiční akcie. I v sledovaném období je snadno vidět její stoupající trend. Obecně, jelikož se jedná o obrovský celosvětový korporát, ovlivňuje jí celá řada různých proměnných. Zahrnutá data obsahovala ceny při otevření a zavření burzy, nejvyšší a nejnižší denní hodnotu, množství obchodovatelné na burze, procentuální změna konečné ceny a přepočítaná konečná cena vzhledem k prodanému množství (jedná se o schéma Open, Low, High, Close, Adj. Close, Volume s přidáním změny – Change). Dále byli přidány firmy s podobně velkou kapitalizací. Jedná se Nvidia, Microsoft a Google. Nvidia je technologická firma která se převážně zabývá grafickými kartami a architekturou jejich čipů (a ovládací softwarů kolem nich). Microsoft je dnes již převážně softwarově zaměřenou firmu (v minulosti se pokoušeli o celou řadu produktů, jako třeba telefony – dnešní nejúspěšnější, z řad hardwaru, jsou herní konzole Xbox). A poslední jmenovaná je Google, opět převážně softwarová firma, která konkuруje firmě Apple ve sféře mobilních telefonů a služeb kolem nich. Všechny mají stejnou strukturu informací, vyjmenovanou u cen akcií Apple.



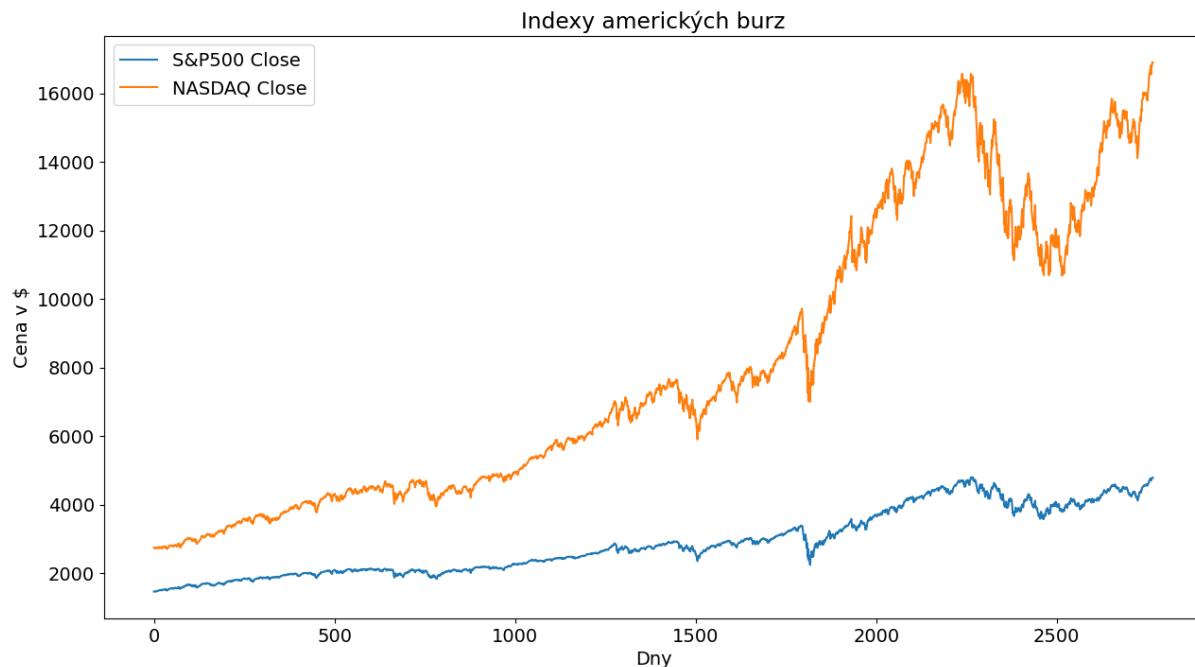
Obrázek 12: Graf popisující cenu jednotlivých firem při uzavírání burzy

K tomu byly přidány ceny komodit jako je zlato a ropa. U nich chybí pouze přepočet konečné ceny na základě množství komodity prodané na trhu. U ropy se přesněji jedná o Brent Crude Oil (ropa ze Severního moře) a Crude Oil z USA (Texas apod.). Ceny zlata pochází ze stránky Kaggle, přesněji z datasetu *Gold, Silver & Precious Metals Futures Daily Data* [26].



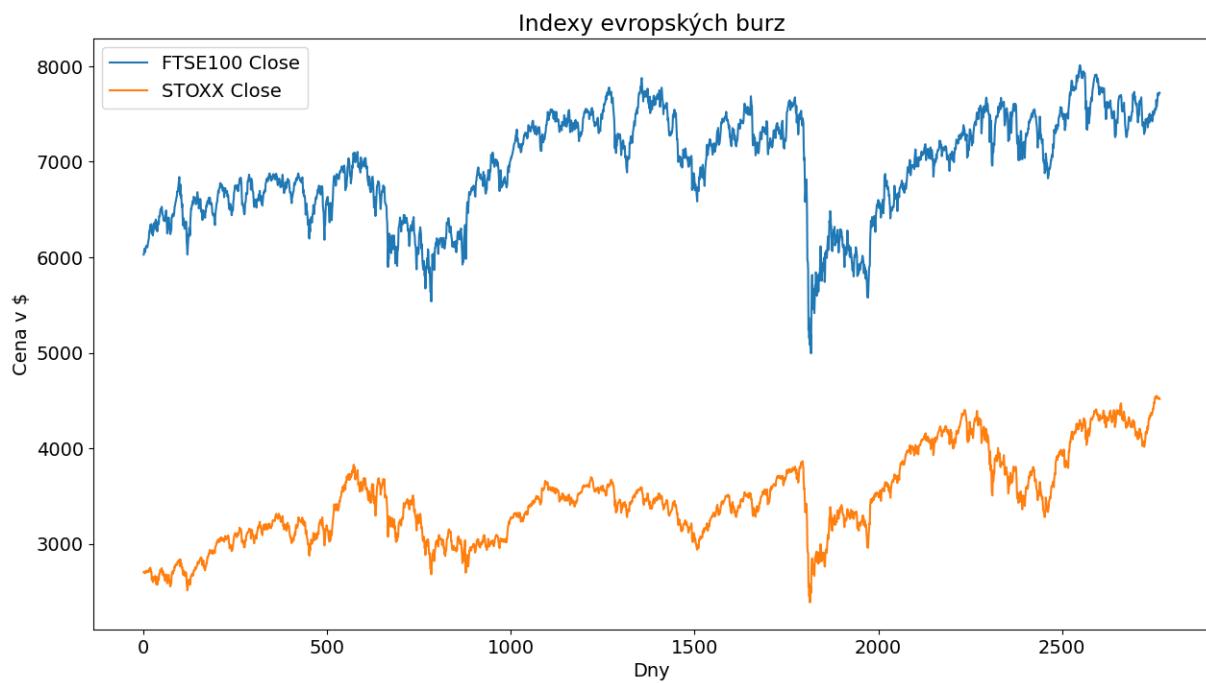
Obrázek 13: Graf cen opcí zahrnutých komodit, zobrazen v logaritmickém měřítku

Dalšími komponentami datasetu jsou indexy z různých burz celého světa. První zahrnutý je S&P 500 obsahující 500 nejsilnějších firem Spojených států amerických (a tedy zahrnují i Apple, Microsoft, Google a Nvidii). Ten v posledních letech měl obrovský nárůst. V USA působí i další silný index, a to NASDAQ. Jedná se o druhý největší (po S&P 500) index na světě. Je svázán s New Yorskou burzou se stejným názvem. Obsahuje firmy, kterou jsou exkluzivně obchodované na ní (bez prioritních akcií, ETF a pár dalších). Jejich počet se v čase mění. Oba tyto indexy obsahují výše zmíněné vlastnosti a zároveň i průměr množství prodaných akcií za posledních 20 dní. Oba tyto indexy pochází také ze stránky kaggle.com z datasetu *S&P 500 and NASDAQ 100: Daily Data* [27].



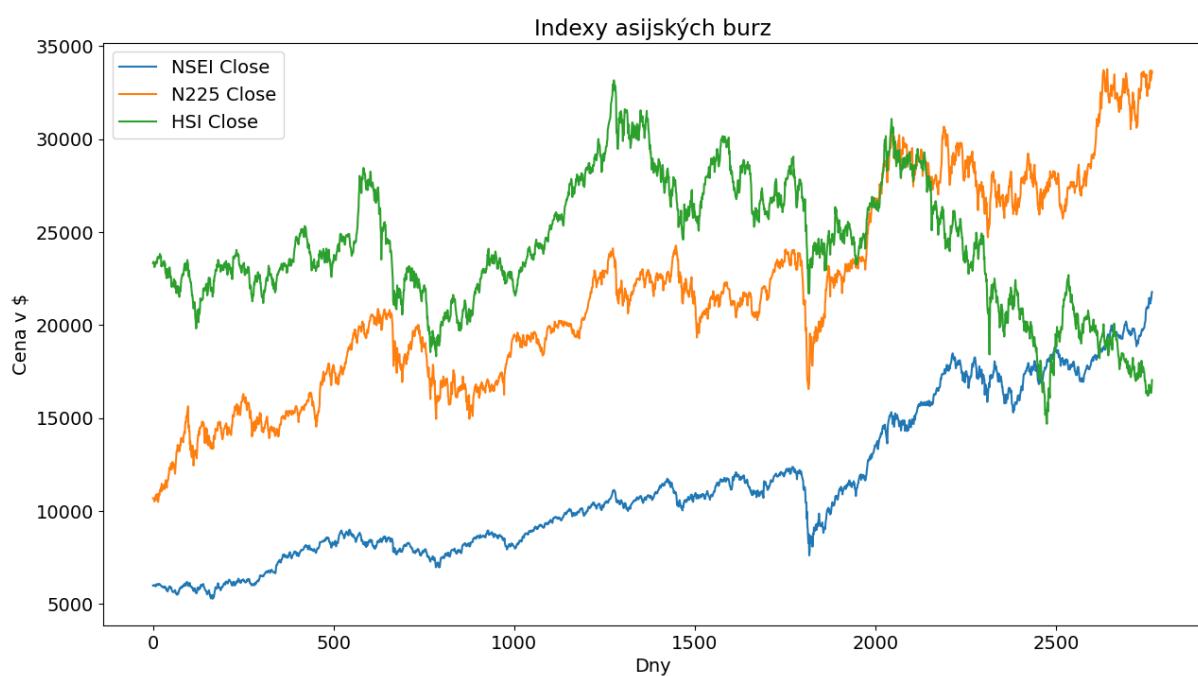
Obrázek 14: Graf cen Indexů nalézajících se v USA

Následují indexy ze zbytku světa. FTSE 100 je index londýnské burzy, zahrnuje 100 největších britských firem. Následuje STOXX 50. Jedná se o index eurozóny obsahující 50 společností z 11 zemí, které vedou ve svých respektovaných odvětvích. Jsou obchodované v Eurex burze se sídlem v Německu. Většina společností pochází z Francie a Německa.



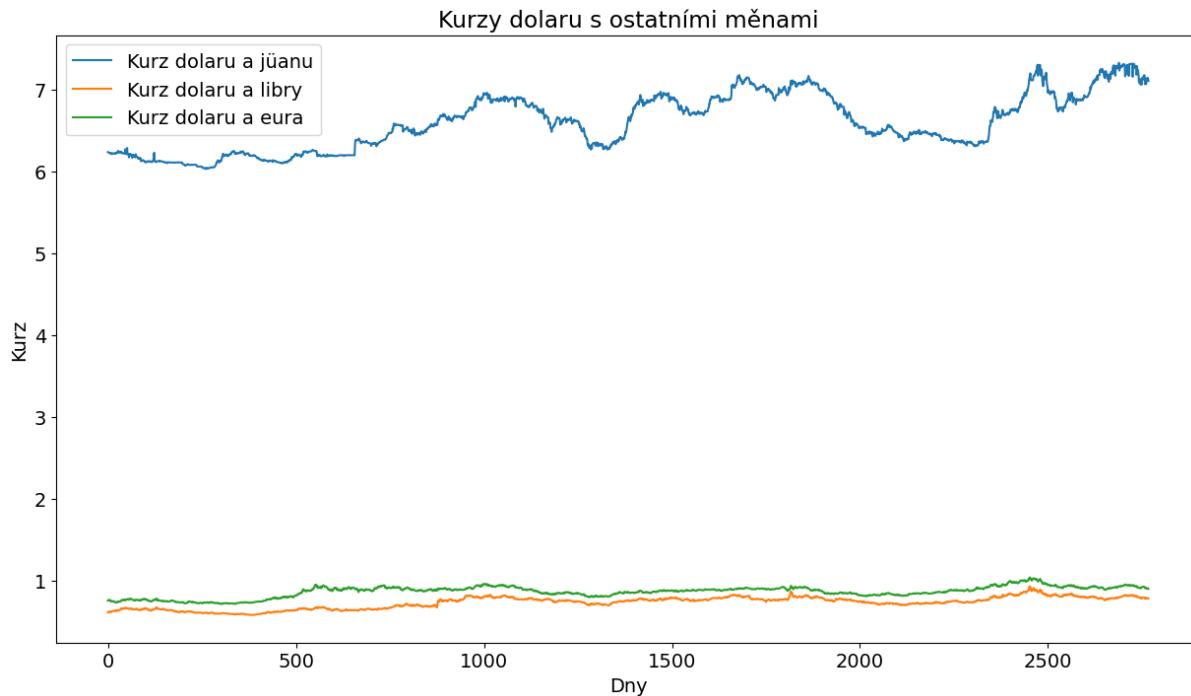
Obrázek 15: Konečné ceny indexů z evropských burz

Pokračují indexy z Asie. NIFTY 50 je index indické burzy, obsahuje 50 největších Indických společností a pokrývá hodnotu blížící se 60 % volného trhu akcií na NSE. Nikkei 225 je japonský akciový index a zaujímá 225 největších společností na území Japonska (obchodovaných na tokijské burze). Posledním je HSI, index Hongkongské burzy a obsahuje 82 firem.



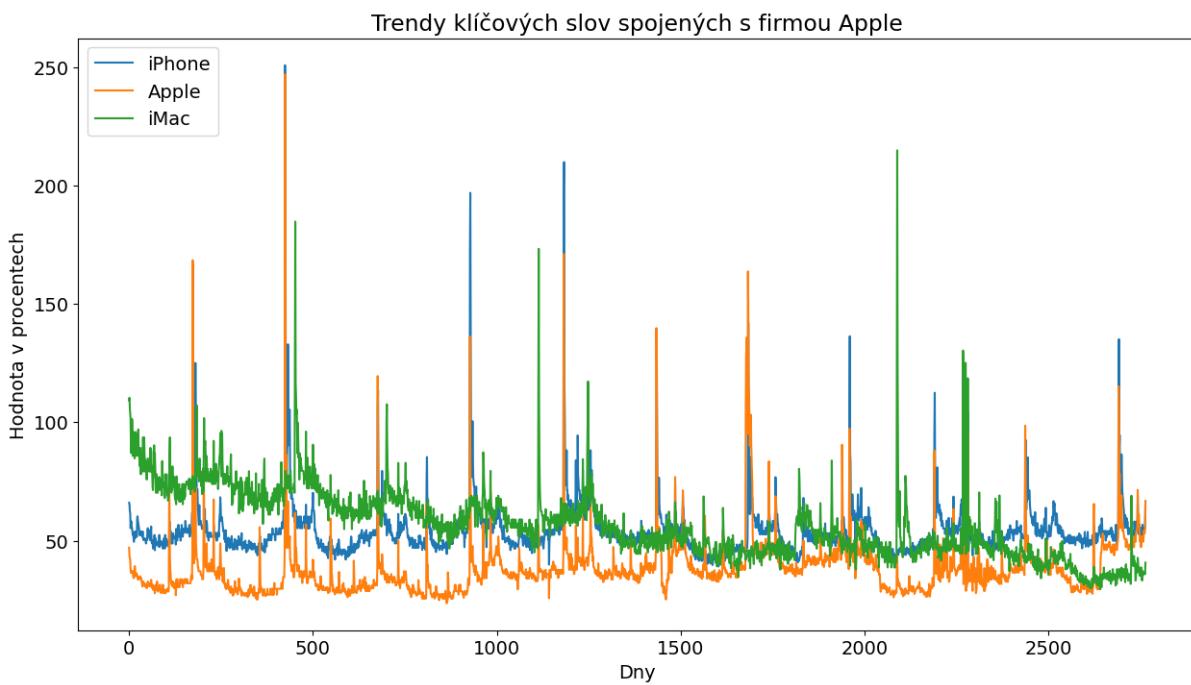
Obrázek 16: Hladiny cen indexů burz pocházejících z Asie

Dalšími byly přidanými kurzy mezi dolarem a eurem, dolarem a čínským jüanem a dolarem a librou. Ty byly ve stejném rozložení, jak bylo popsáno výše, s výjimkou množství (volume) a upravenou koncovou cenou (adj. close).



Obrázek 17: Kurzy dolaru s měnami z Evropy a Asie

Poslední jsou pak výše zmínovaná data z Google Trends, kde byly použity klíčová slova Apple, iPhone a iMac.

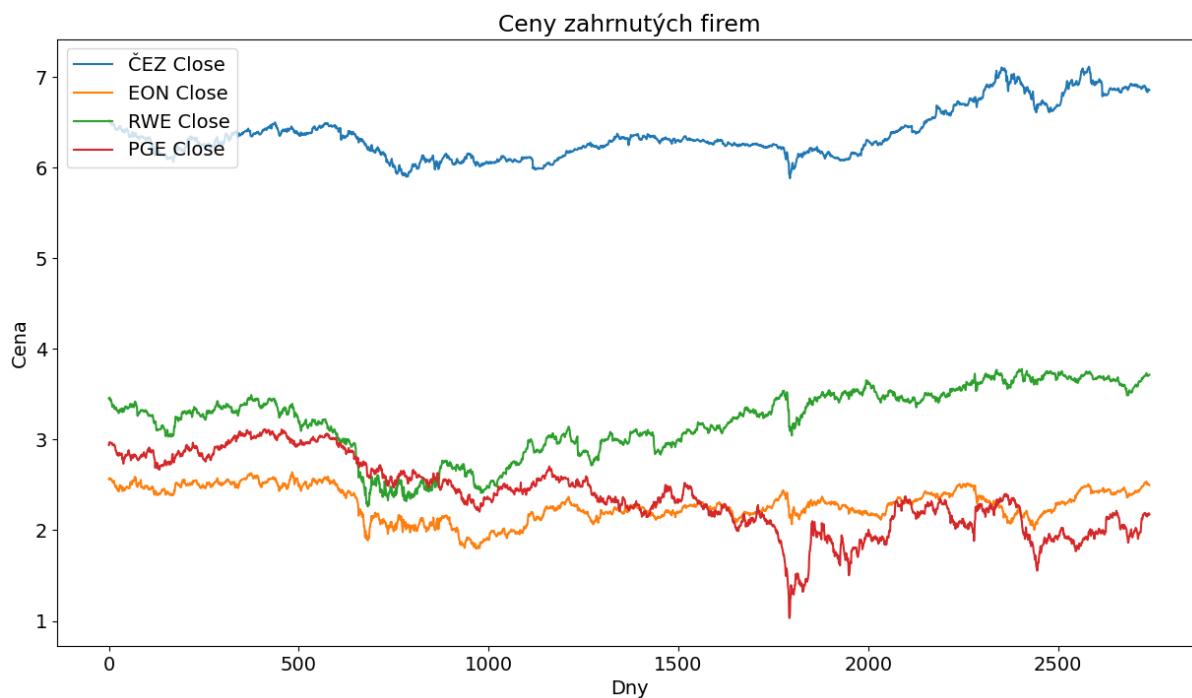


Obrázek 18: Poměrové zobrazení vyhledávání klíčových slov ve vyhledávači google.com

5.2 ČEZ

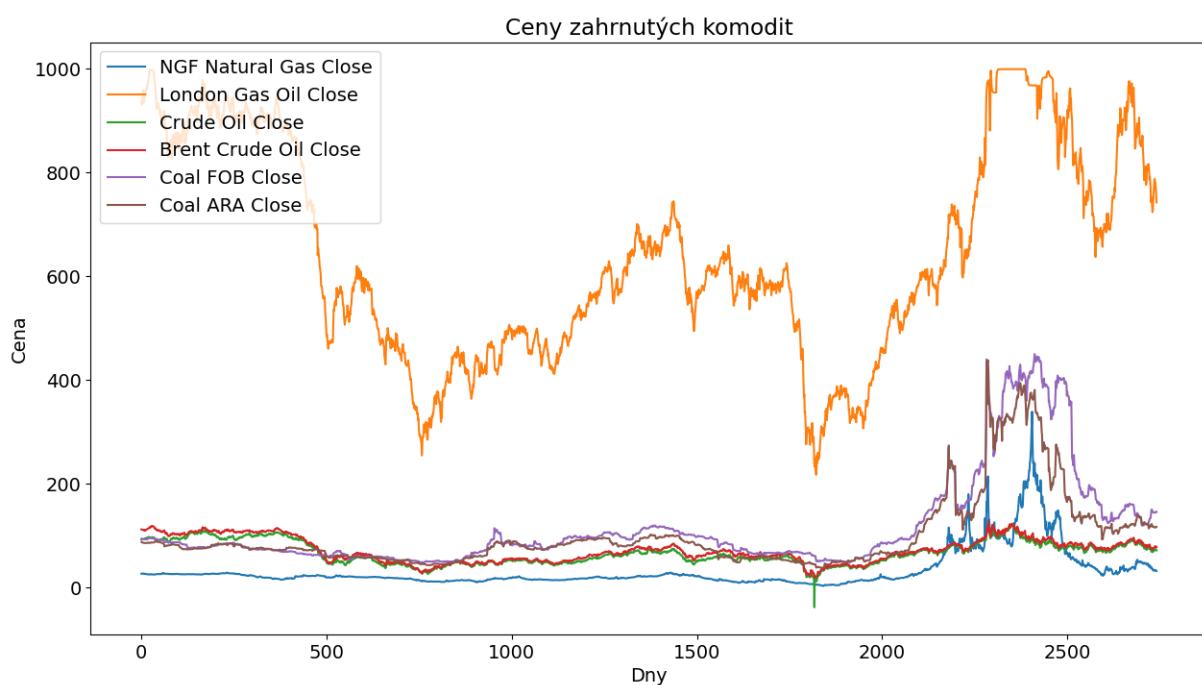
Založení předchůdce skupiny ČEZ se datuje do roku 1945, kdy se tehdy samostatných 1300 firem spojilo a přešlo pod správu státu a tím vznikly České energetické závody. Všestrannou společností zastřešující výrobu, vedení i služby zákazníkům se stala až v osmdesátých letech, v té době stále státní podnik. První velká změna pak přišla porevolučním rozdelením distribučních sítí do menších krajských částí. První kolo privatizace přišlo v roce 1992, kdy se provádělo i různé převádění majetku a vysokonapěťových sítí. Plán byl, že stát si ponechá 67 % akcií, aby mohl ovlivňovat a regulovat ceny elektřiny, zbytek putoval do privatizace. Až při druhé vlně se státní podíl zastaví na 67,5 %. Zbytek vlastní z 26,5 % právnické osoby, 5 % fyzické osoby a zbylá procenta vlastní Restituční investiční fond. Výše zmíněné distribuční společnosti byly zprivatizované při druhé vlně kupónové privatizace (stát si ponechá zhruba 80 %). Díky Evropským směrnicím se oddělila přenosová struktura do nově vznikající společnosti ČEPS, která má za cíl se starat a vylepšit přenosovou síť vysokého napětí. Na začátku nového milénia se Česká republika snažila o plný prodej celé společnosti soukromému vlastníkovi, narazila však na neochotu ze strany zájemců plnit podmínky vlády (například záruky okolo jaderné elektrárny Temelín) a i velký rozdíl mezi představami státu a soukromníku v rámci cen (italská společnost Enel nebo francouzská Edf nabízeli kolem 140 miliard Kč, stát však očekával alespoň 200 miliard Kč). Cíle se tedy změnily na vybudování nadnárodního korporát v rukou státu. K samotnému zformování skupiny ČEZ dochází až v roce 2003, čímž si tato společnost zafixovala svou silnou pozici jako výrobce tak i distributor energie v Čechách. V dalších letech už posilovala svou pozici v Centrální Evropě, Východní Evropě a na Balkánském poloostrově v rámci různých akvizic elektráren a jiných částí a služeb v rámci energetiky. To se změnilo v roce 2019 s novou vizí společnosti a začalo prodávání zahraničních aktiv. Výjimkou byly moderní energetické služby, které naopak chtějí posílit jak České republice, tak v zahraničí. Začátkem roku 2021 zároveň zrychluje přechod na bezemisní provoz, kdy nízkoemisní se chtějí stát v 2030 a plně bez uhlíkových emisí v roce 2050. Zároveň se v budoucích letech započne rozšíření jaderné elektrárny Dukovany a spekuluje se i o investicích do jaderné elektrárny Temelín [28][29].

Tato akciová společnost je z principu velice stabilní (vyrábí a distribuuje kritickou surovinu, kde poptávka se s přibývajícími lety zvětšuje). Energetika jako taková je v posledních letech ve společnosti velice diskutovaným tématem. Oproti firmě Apple se však jedná o lokálněji laděnou firmu, působící hlavně v Čechách a střední Evropě. Tím se odvíjí i jaké další faktory jsou zahrnuty do dat určených pro predikci. Prvními jsou nejpřímější konkurenti v odvětví v dané oblasti. Nejsilnější by se dala označit společnost E.ON působící převážně v Německu. Dalšími jsou RWE původně z Německa, PRE z České republiky a PGE z Polska. Ceny ČEZ jsou zahrnuty také.

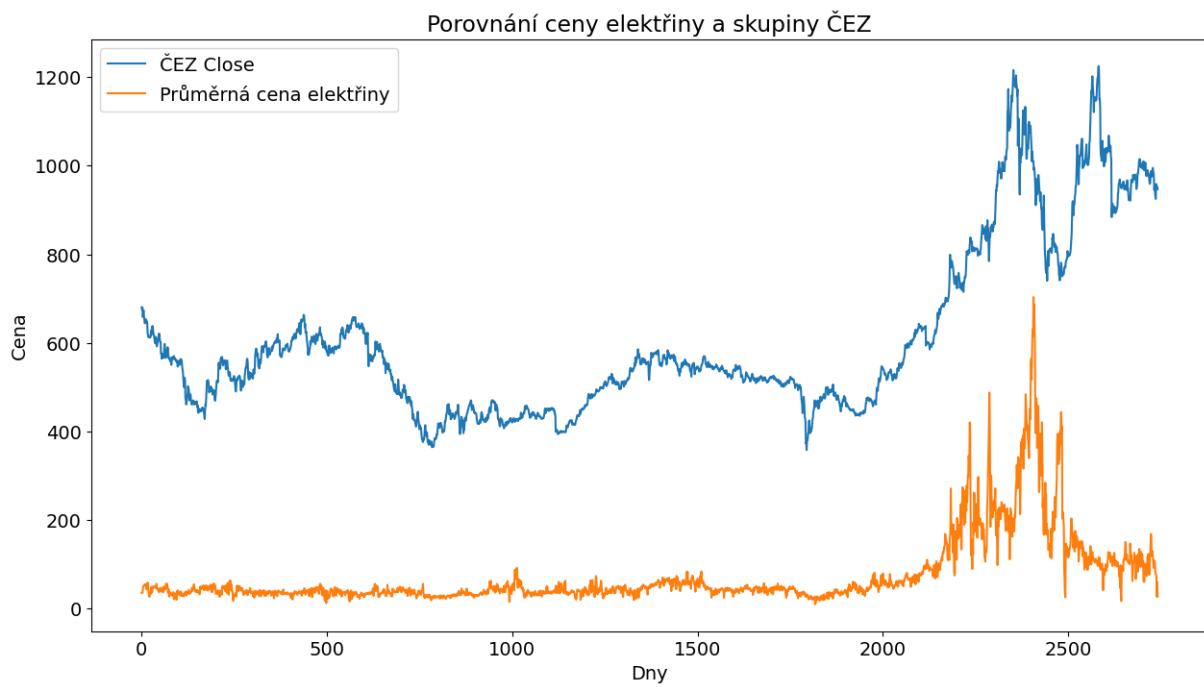


Obrázek 19: Ceny zahrnutých firem pro predikce skupiny ČEZ (společně s její cenou) v logaritmickém měřítku

V datech se také nachází i různé komodity. Jedná se o plyn, topný olej, ropu, uhlí a elektřinu. Plyn je zastoupen opcí NG=F, topný olej London Gas Oil, ropa má stejně jako v předchozí sekci dva participanty – Brent Crude Oil a Crude Oil, uhlí má také dva – FOB Newcastle a ARA CIF Coal. Poslední sekce, elektřina, je reprezentována spotovou cenou (Base Load, Peak Load, Offpeak Load) a průměrnou cenou elektřiny v daný den s přidanou procentuální změnou.

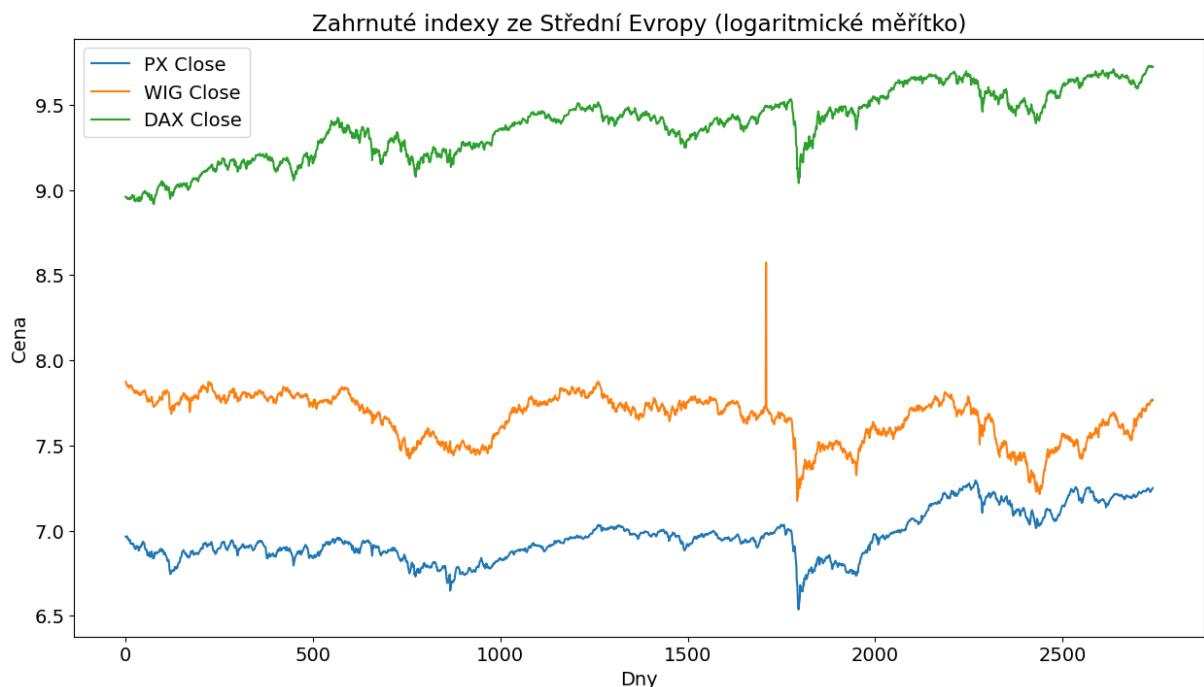


Obrázek 20: Ceny komodit použitých u předpovědi ceny společnosti ČEZ



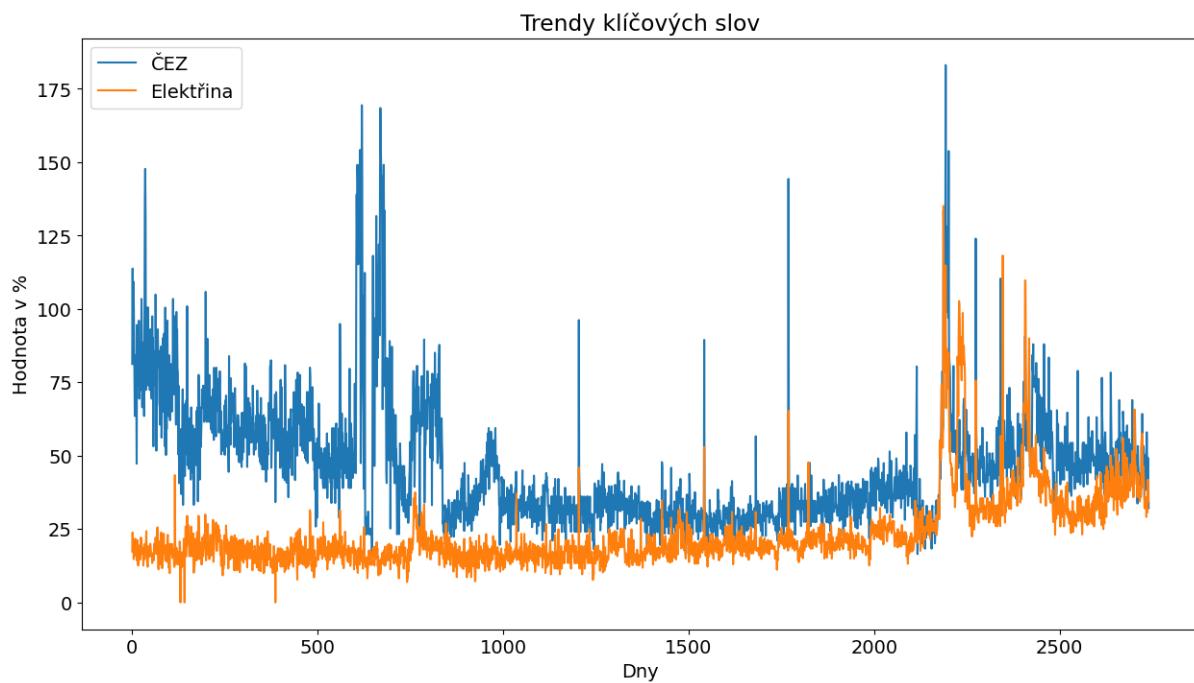
Obrázek 21: Porovnání ceny skupiny ČEZ a průměrná cena elektřiny v daný den

Následují indexy lokálních burz. Pražský index PX je přímo svázán s ČEZ (je jeho součástí a největším přispěvatelem co do kapitalizace), a proto je součástí dat. Dalším je WIG, index Varšavské burzy. Obsahuje 330 polských společností. Velice důležitý ekonomický gigant Střední Evropy (i celé Evropy) je Německo, a i index jeho burzy DAX (také GDAXI) je zahrnut. Vypočítává se na základě 40 největších firem.



Obrázek 22: Ceny indexů použitých v datasetu pro ČEZ

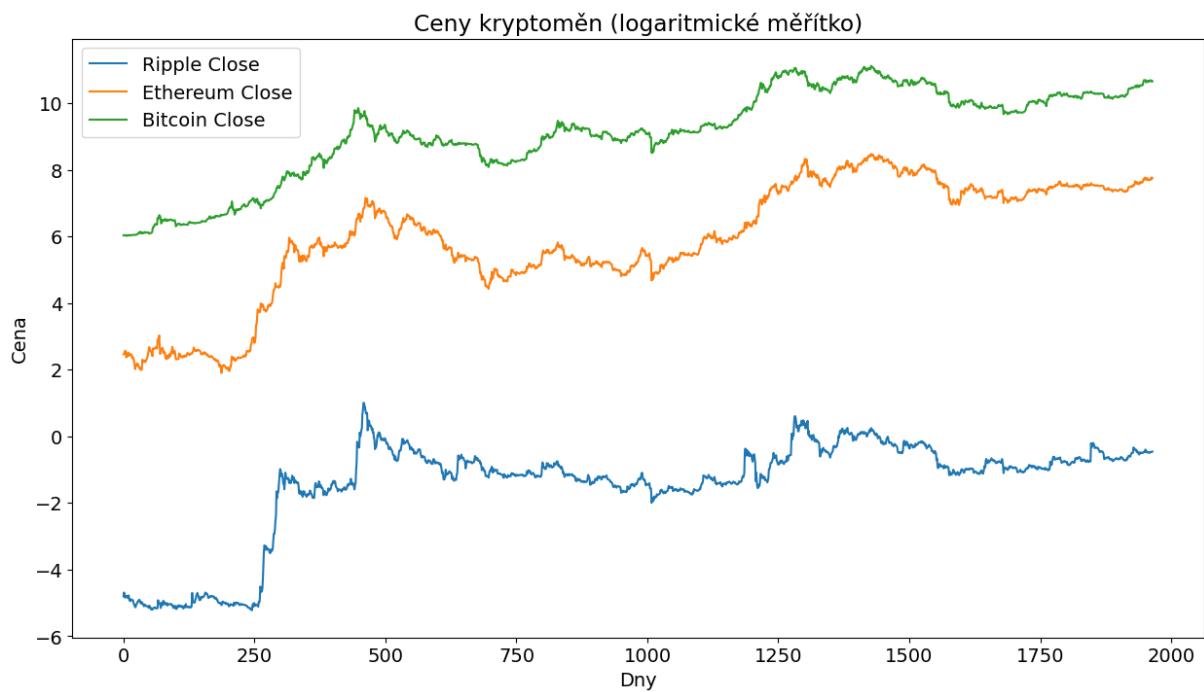
Dále jsou také zahrnuty kurzy měn česká koruna na dolar, česká koruna na euro a česká koruna na polský zlotý. K tomu jsou pak přiřazeny data o vyhledávání klíčového slova ČEZ a Elektřina z Google Trends.



Obrázek 23: Trendy pro klíčová skupina spjatými se skupinou ČEZ

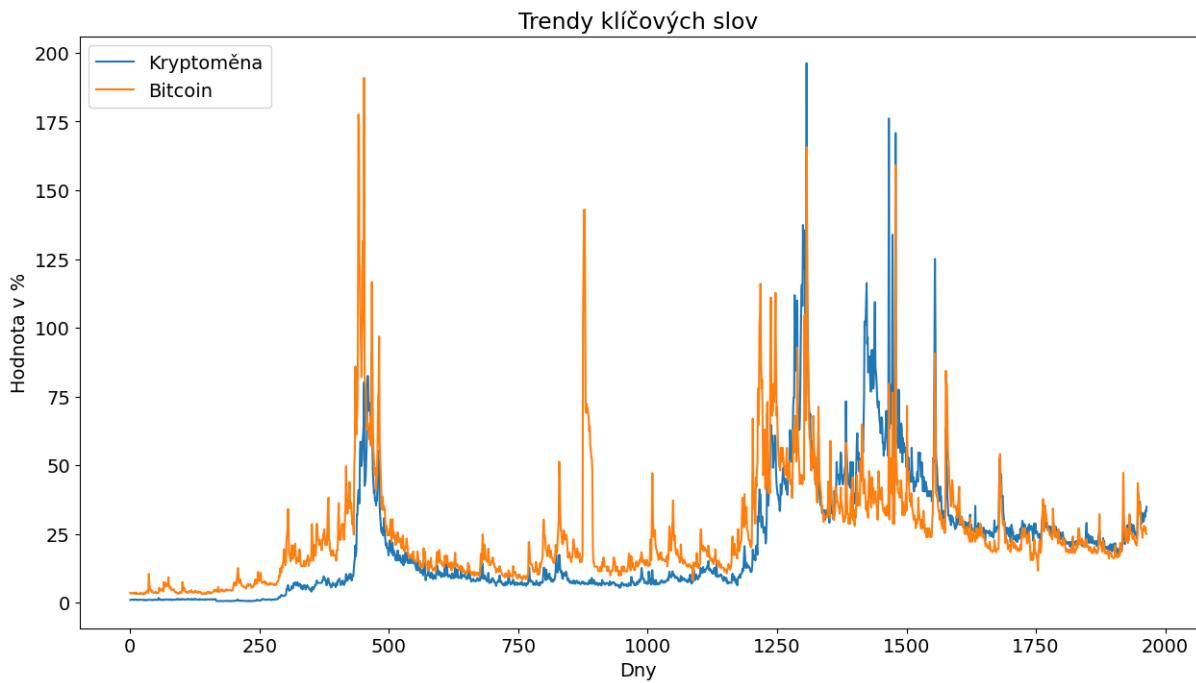
5.3 Bitcoin

Kryptoměny jsou fenoménem hlavně posledního desetiletí. Do této kategorie patří mnoho různých druhů digitálních měn, jeho nejsilnějším zástupcem je Bitcoin. Nejedná se však o prvního průkopníka ve vytvoření digitální měny, žádná ale nespadala do dnes známé kategorie, a hlavně jejich životnost byla celkem krátká. Bitcoin byl vytvořen roku 2008 Satoshi Nakamotem, neznámou osobou (či skupinou osob), kde autorovo jméno (pseudonym) pramení z článku, který byl vydán při vydání této měny. Článek se zabýval peer-to-peer sítí která měla za cíl být systémem pro elektronické transakce bez vyžadování důvěry obou stran. První blok mincí byl ale vytvořen až roku 2009 s objemem 50 kusů. Jedni z prvních uživatelů byly i autoři předchůdců Bitcoinu, jednalo se tedy o poměrně uzavřenou komunitu ve svých začátcích. První reálnou transakcí byl nákup v síti Papa John's. Jednalo se o dvě pizzy za 10000 bitcoinů, v té době za zhruba 40 dolarů. V roce 2011 vyšel BitPay jako alternativa pro mobilní placení, důležitý pro oživení myšlenky pro používání většími společnostmi. V témež roce se také měna začala využívat pochybnými stránkami na tzv. Dark webu, kde díky své anonymizaci získala na velké popularitě. Dlouho dobu taky byla spojovaná hlavně s nelegálními aktivitami. Časem se však přenesla nad tuto proslulost a více se začala dostávat do obyčejného prostoru placení. Jedním z velkých milníků bylo její přijímání Microsoftem. V 2017 burza CME začala prodávat opce Bitcoinu na své burze. Prvním státem, který ji uvedl jako oficiální podporovanou měnou byl pak El Salvador v roce 2021. Satoshi Nakamoto předal vedení nad Bitcoinem Gavinu Andresenovi, který dnes vede Bitcoin Foundation. V dalších letech se pouze přidávali další firmy podporující platby v této měně a její hodnota vyrostla do astronomických výšin [30].



Obrázek 24: Ceny kryptoměn, společné s Bitcoinem v logaritmickém měřítku

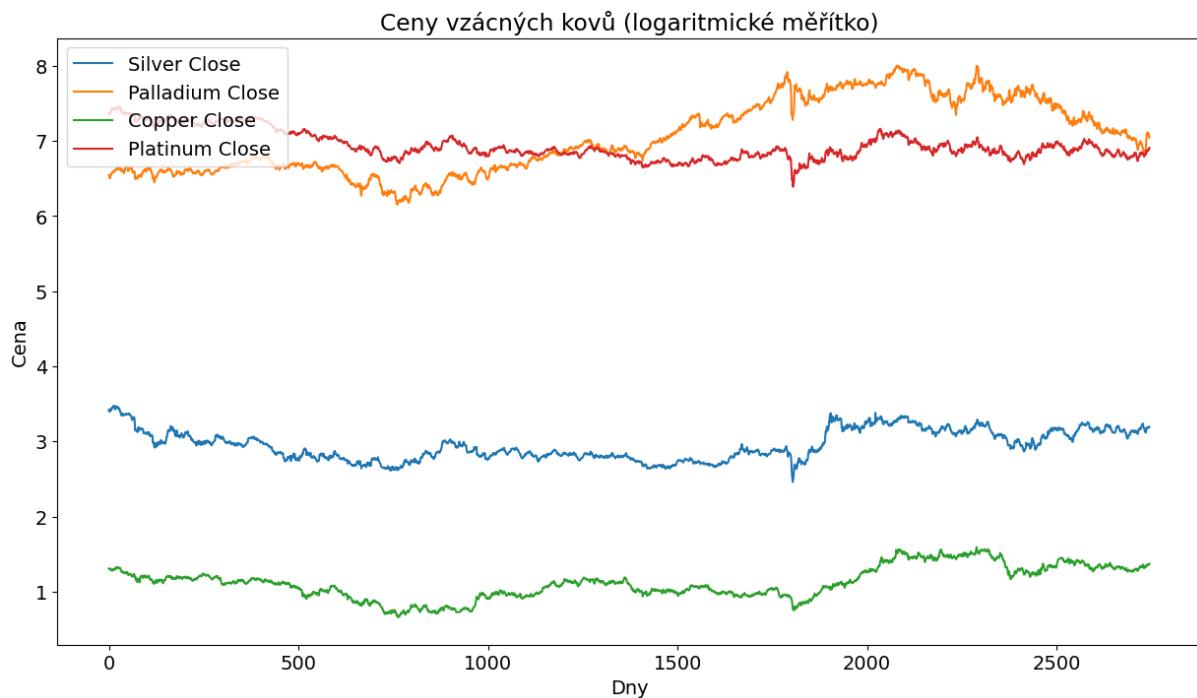
Jedná se dnes o světový fenomén, a tedy dataset je obdobný jako pro Apple. Přidané byly hodnoty dalších kryptoměn. Jedná se o Ethereum a Ripple(XRP). Také jsou zde údaje o vyhledávání klíčových slov Bitcoin a kryptoměna.



Obrázek 25: Trendy klíčových slov spojených s kryptoměnou Bitcoin

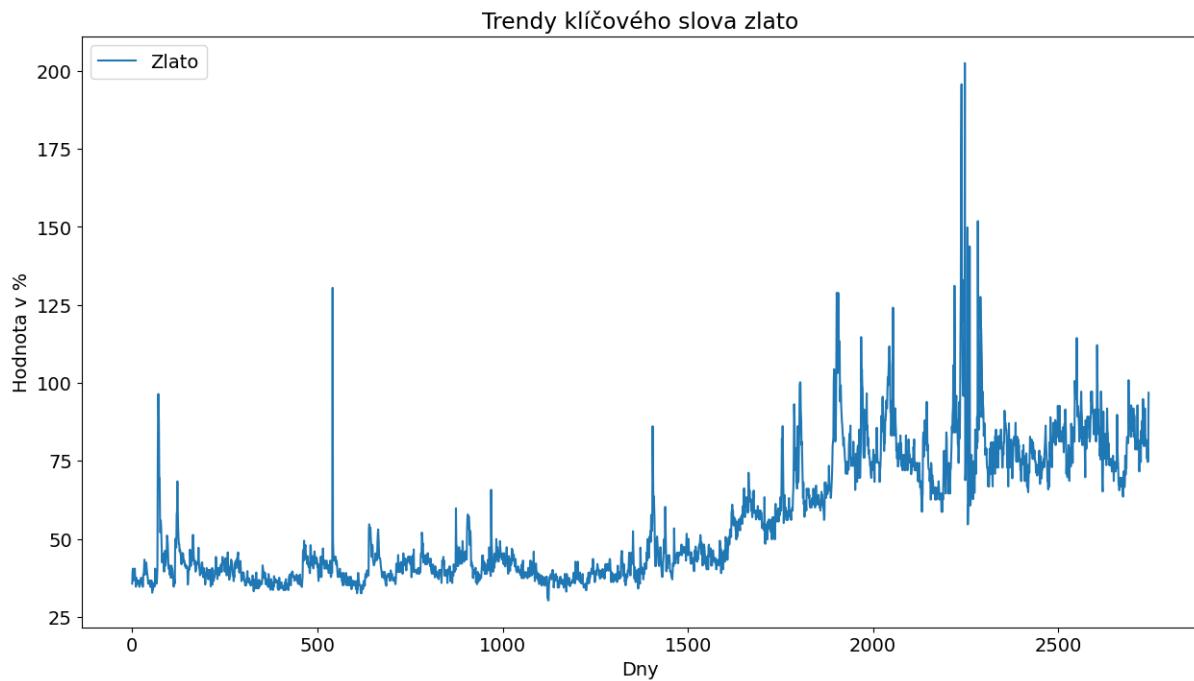
5.4 Zlato

Zlato je velice ceněná komodita už tisíce let. Využívána dříve jako komponent pro výrobu duchovních pomůcek, ukazatel společenského statusu, nebo později jako směnný artikl, vždy si uchovávala určitou hodnotu napříč naší historii. Dnes je stále využíván jako ozdoba, s nástupem technologií ale došlo k nalezení i využití díky jejím fyzikálním a chemickým vlastnostem.



Obrázek 26: Ceny zlata a jiných vzácných kovů v logaritmickém měřítku

Z investičního hlediska se v dlouhodobém horizontu jedná o solidní artikl a s větším využíváním technologií jeho cena neklesá. Data pro predikce jeho ceny jsou opět v širším, globálním, měřítku a jsou tedy obdobná Apple datům. K nim byly přidány ceny ostatních cenných kovů, jako je stříbro, platina a paladium ze stejného datasetu, ze kterého byly ceny zlata zmíněny výše [26]. Přidány byly i hodnoty vyhledávání klíčového slova zlato.



Obrázek 27: Trendy klíčového slova zlato

5.5 Zpožděně proměnné

Ke všem 4 typů predikčních dat byly přidány i zpožděné faktory (výjimka jsou Fourierovy transformace). Ty mají napomoci generalizaci a snížit jeho zpoždění. Byly vypočítány z konečné ceny (MACD jsou z cen High, Open, Low, Close) daného výstupu. Dohromady přispívají do velikosti souboru 14 faktory.

5.5.1 Klouzavý průměr

Výpočetně nejjednodušším je jednoduchý klouzavý průměr:

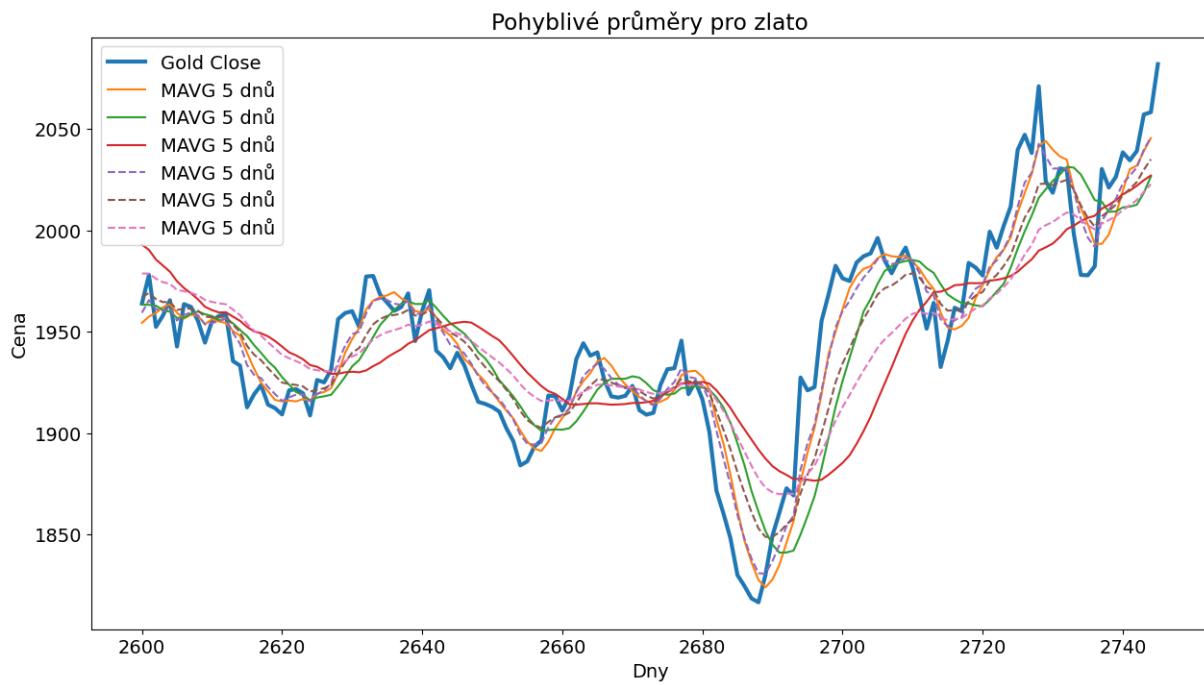
$$SMA = \frac{A_1 + \dots + A_n}{n}, \quad (5.5.1.1)$$

kde A_1 až A_n jsou prvky časové řady a n je velikost (časová délka) úseku. V datech byly použity úseky 5 dnů, 10 dnů a 20 dnů [31].

Dalším byl pak exponenciální klouzavý průměr:

$$EMA_t = \left(A_t \cdot \left(\frac{s}{1+n} \right) \right) + EMA_{t-1} \cdot \left(1 - \left(\frac{s}{1+n} \right) \right), \quad (5.5.1.2)$$

kde t označuje časový krok, s je hodnota vyhlazení (typicky 2) a n je velikost (časová délka) úseku. Opět se v datech počítá 5, 10 a 20 dnů [32].



Obrázek 28. Pohyblivé průměry pro předpověď ceny zlata, přiblížení na posledních zhruba 140 dnů

5.5.2 Bollingerova pásla

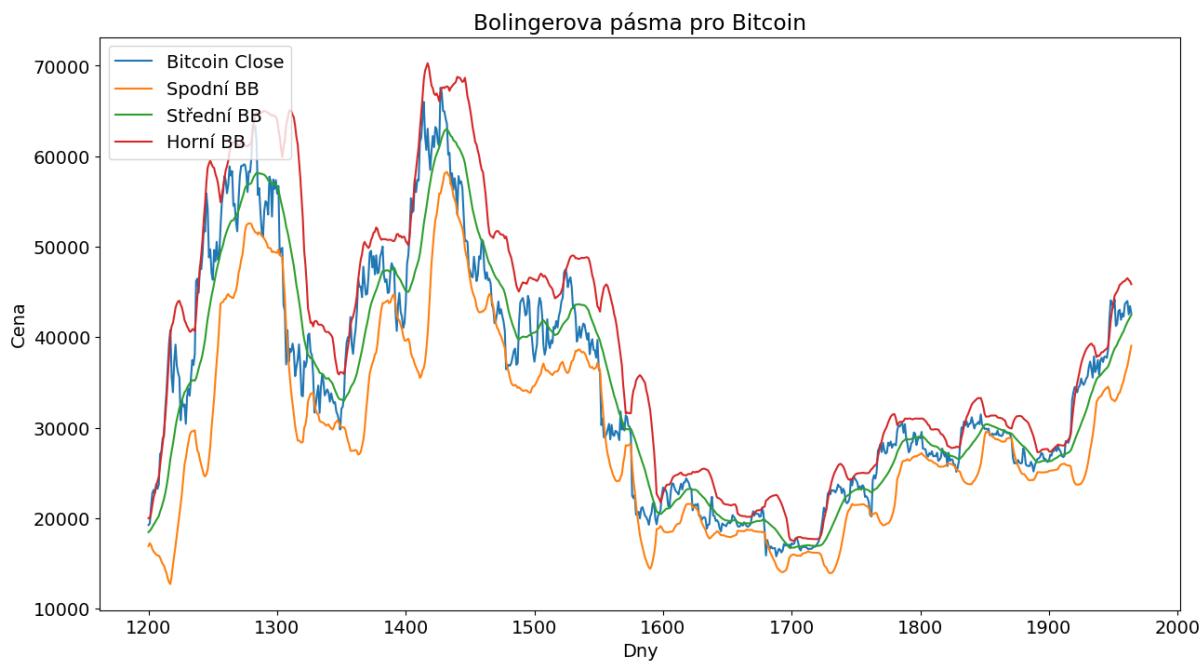
Jedna z metod technické analýzy. Populární pro určení volatility ceny cenných papírů. Obecně podle Johna Bollingera, autora této metody, ji lze využít i pro komodity, opce a kurzy měn. V dnešní době je velmi oblíbená i u kryptoměn. Skládá se ze horního pásla (Upper), středního pásla (Middle) a dolního pásla (Lower). Ty mají výpočet založený na kombinaci jednoduchého klouzavého průměru a standartní odchylky:

$$UpperBB = SMA_n + r \cdot SD_n, \quad (5.5.2.1)$$

$$MiddleBB = SMA_n, \quad (5.5.2.2)$$

$$LowerBB = SMA_n - r \cdot SD_n, \quad (5.5.2.3)$$

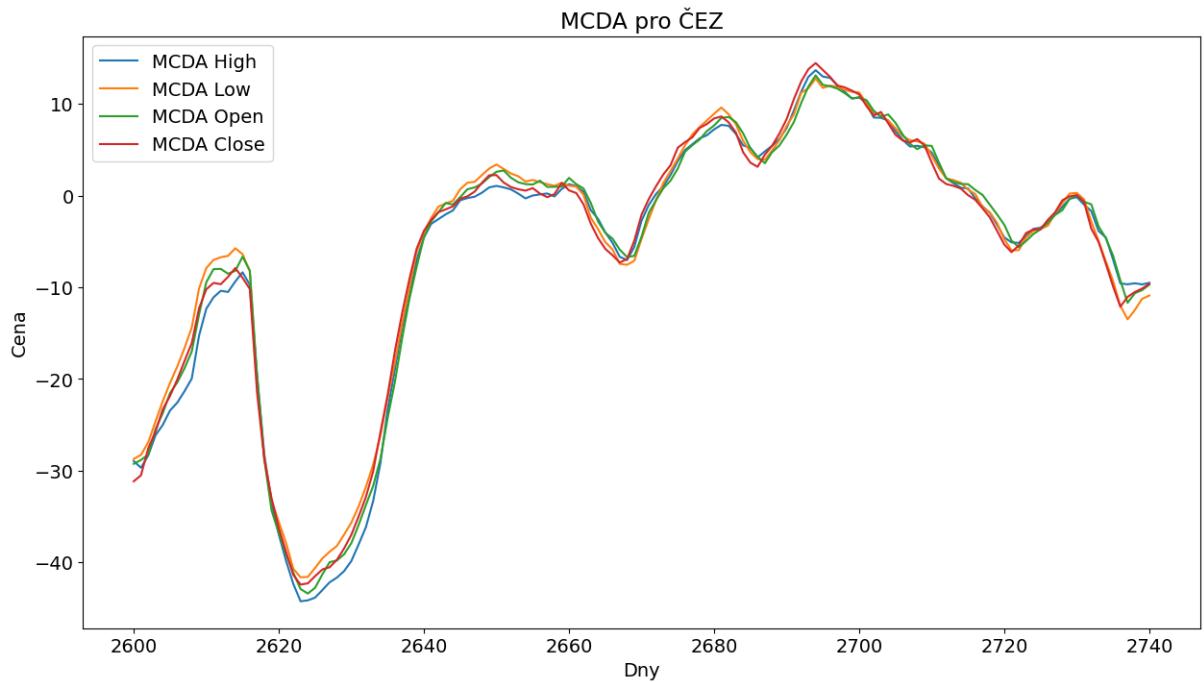
kde SMA_n označuje jednoduchý klouzavý průměr v období n dnů, SD_n je standartní odchylka opět v období n dnů, r je pak koeficient. Koeficienty n a r jsou libovolné, typicky se jedná o období 20 dnů a koeficient násobků odchylky je 2 [33].



Obrázek 29: Bolingerova pásma pro cenu Bitcoinu, přiblžení na posledních zhruba 800 dní

5.5.3 MACD (Moving Average Convergence Divergence)

Jedná se o další indikátor technické analýzy. Tento se zaměřuje na odhalování trendů ceny a jejich momentu, tím pak hledat vstupní body. Vymyslel ho Gerald Appel v 70.letech minulého století. Základní výpočet MACD průběhu (line) je odečtením dvou výpočtů exponenciálních klouzavých průměr s délkou období 12 a 26 [34].



Obrázek 30: Hodnoty MACD pro cenu akcií skupiny ČEZ

5.5.4 Index relativní síly

Další technický indikátor. Tentokrát je jeho zaměření na výpočet rychlosti a velikosti změny ceny a tím odhadnout, jestli dochází k přehnanému nakupování či prodávání daného cenného papíru (pomáhá „předpovědět“ chování trhu). Poprvé ho publikoval J. Welles Wilder v roce 1978. Výpočet tohoto indexu má více částí, první je výpočet hodnot kladné a záporné změny. Ten je následovný pro směr nahoru:

$$U = \begin{cases} A_t - A_{t-1} & A_t > A_{t-1}, \\ 0 & jinak \end{cases} \quad (5.5.4.1)$$

a pro směr dolu:

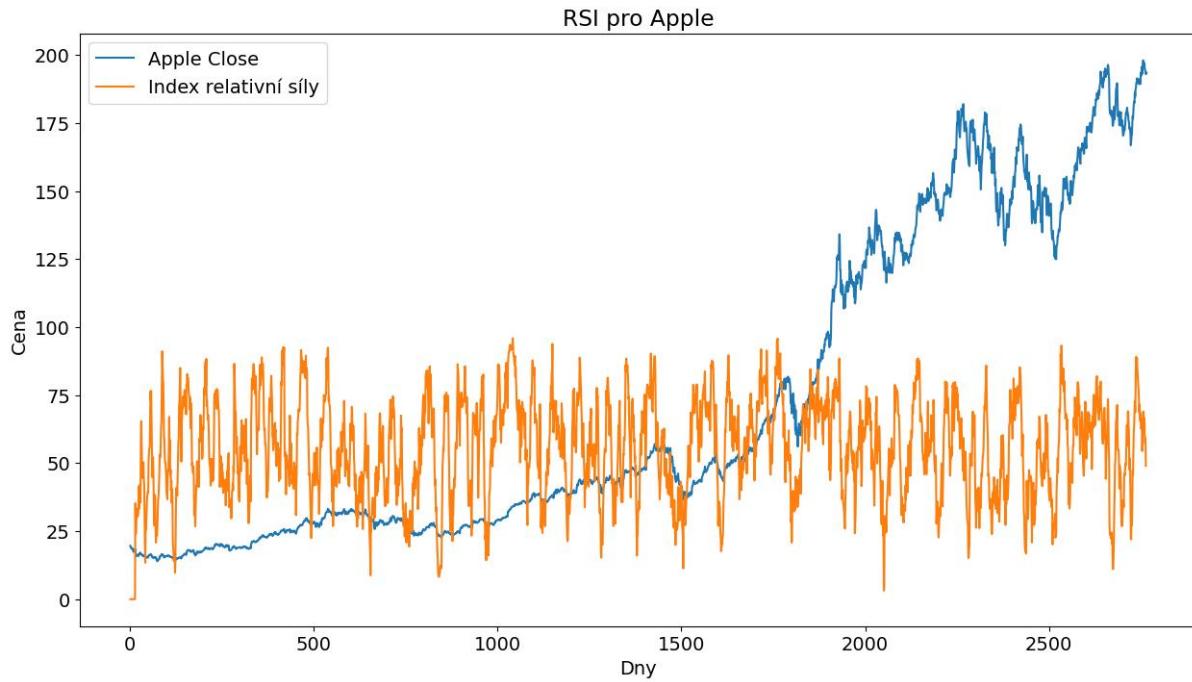
$$D = \begin{cases} A_{t-1} - A_t & A_{t-1} > A_t, \\ 0 & jinak \end{cases} \quad (5.5.4.2)$$

kde A_t je cena v časovém kroku t (den). Obě tyto proměnné pak jsou nezáporné. Pokud by cena v den výpočtu a den o časový krok předtím byla stejná, pak obě budou 0. Relativní sílu pak lze vypočítat jako:

$$RS = \frac{EMA_n(U)}{EMA_n(D)}, \quad (5.5.4.3)$$

kde EMA_n označuje exponenciální klouzavý průměr v období n dnů. Na to se pak naváže výpočtem faktoru navázaných na hodnotu mezi 0 a 100 [35]:

$$RSI = 100 - \frac{100}{1 + RS}. \quad (5.5.4.4)$$



Obrázek 31: Graf indexu relativní síly v porovnání s cenou akcie pro Apple dataset

5.6 Fourierovy transformace

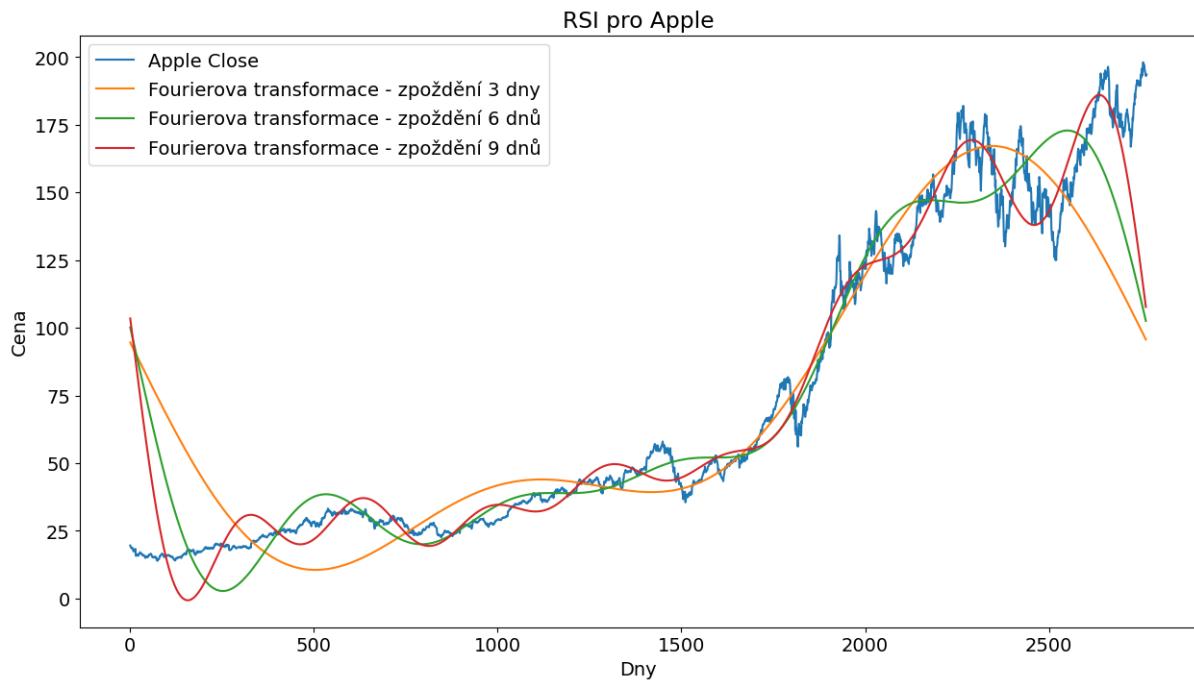
K datům ještě byly přidány i transformace skrze Fourierovu funkci. Ty mají napomáhat globální generalizaci. Pro tu se využívá Diskrétní Fourierova transformace:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-i2\pi\frac{kn}{N}}, \quad (0.1)$$

společně s korespondující inverzní variantou:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{i2\pi\frac{kn}{N}}, \quad (0.2)$$

kde $x(n)$ označuje časovou řadu, n je pak časový index, N je její velikost, k označuje frekvenční koš jehož hodnota je mezi 0 a $N - 1$ korespondující s pozorováním v časové řadě, $X(k)$ je Fourierův koeficient pro frekvenci koše k , i je imaginární jednotka. Pro data byly pak využívány zpožděné varianty těchto transformací a to o 3, 6 a 9 dnů [36].



Obrázek 32: Fourierova transformace pro ceny akcií Apple

6 Model a jeho výsledky

Model byl vytvořen pro všechny zmíněné - 2 akcie, 1 komoditu a 1 kryptoměnu. Základní struktura modelu byla pro všechny stejná, liší se pouze v hyperparametrech. Pro vytvoření modelu byl použit programovací jazyk Python a knihovny Pytorch, Pandas, Numpy, sklearn a statsmodel. Pro grafy pak knihovna matplotlib.

6.1 Popis modelu

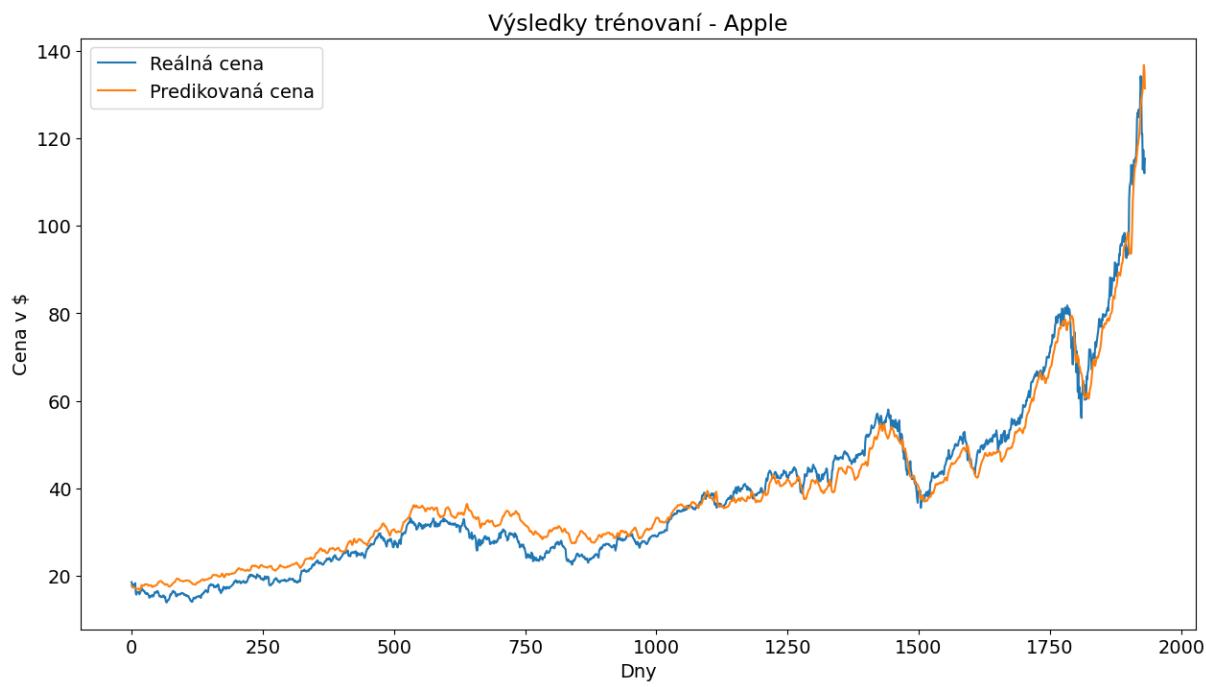
Jak již bylo zmíněno v kapitole GAN sítě, model má dvě hlavní části – generátor a diskriminátor. Generátor je síť GRU skládající se ze tří vrstev GRU (velikost 1024, 512 a 256) s nastaveným odpadnutím (hodnota je 0,2) a třemi plně propojenými vrstvami (velikost 128, 64 a 1) ukončená logistickou funkcí. Diskriminátor je pak složen ze tří konvolučních vrstev, kde velikost jádra je pro první dvě 5 a pro poslední 3, krok mají všechny nastavený na 1, všechny využívají funkci Leaky ReLU a velikost činí 32, 64 a 128. Následuje je vrstva převádějící data na 1 dimenzi a vše ukončují tři plně propojené vrstvy (využívající pořadí funkce Leaky ReLU, ReLU a výstup obsluhuje logistická funkce).

Všechny modely byly trénovány na 165 epoch, s rychlosťí učení 0,00016 a velikost dávky byla 128. Data jako taková byla upravená pomocí funkce MinMaxScaler na interval -1 až 1 do a rozdělena do bloků po 3 dnech, podle kterých se pak predikoval/generoval 4. den. Jako optimalizační algoritmus byl pro generátor i diskriminátor použit Adam, kde první parametr byl mezi 0 až 0,5 a druhý na 0,9. K tomu bylo přidáno i zpomalování rychlosti učení využívající kosinový anealing s nastavenými 165 kroky.

Data byla nejdříve zpracována za účelem se zbavením redundantních faktorů. Na to byla nejdříve využita analýza rozptylu (ANOVA), kde se odstranily vlastnosti s hodnotou p menší jak 0,75. Dále pak byla využita analýza pomocí RFE (recursive feature elimination) využívající lineární regresi. Ta ohodnotila jednotlivé vlastnosti a rozhodla o jejich pořadí podle důležitosti, podle kterého se následně některé z nich vyneschali. Data pak byli rozděleny na 70% tréninková a 30% testovací.

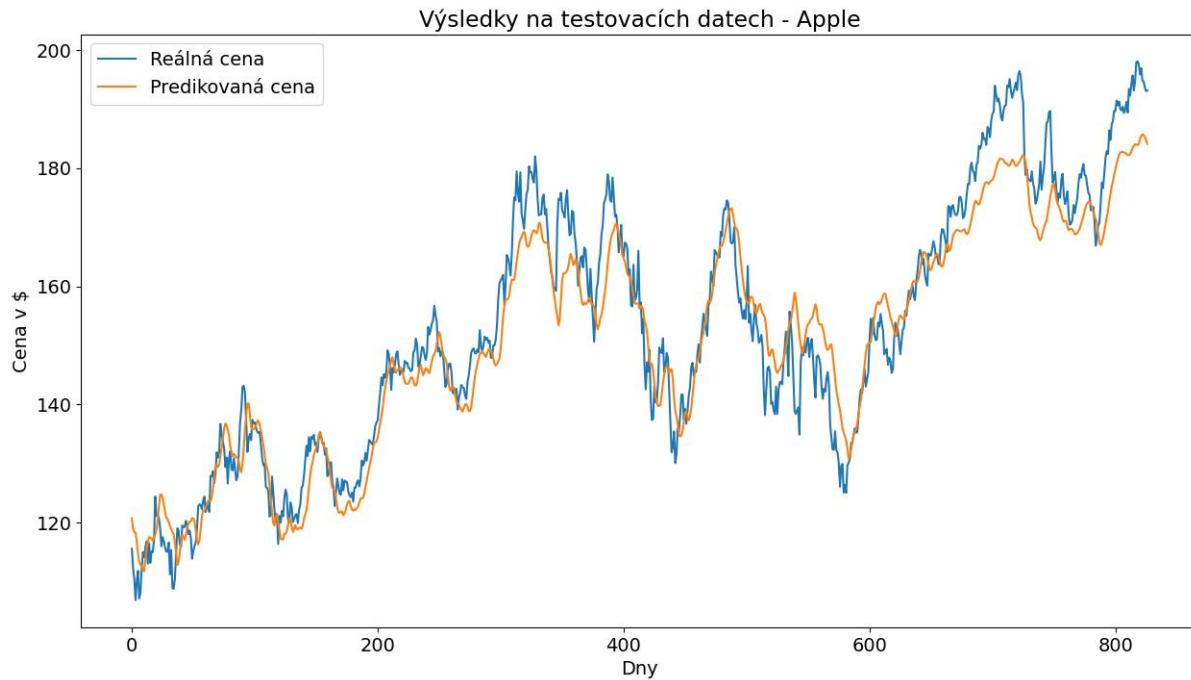
6.2 Výsledky pro Apple

Po výše popsané eliminaci vlastností pak činí velikost dat 2765x50 (data bez přidaných vypočítaných proměnných a trendů). Pro výpočet byli použity Fourierovy transformace. Generátor vytrénovaný na popsaném modelu pak má výsledky v tréninku znázorněné na Obrázek 33. Střední kvadratická chyba modelu pak činí 3,57.



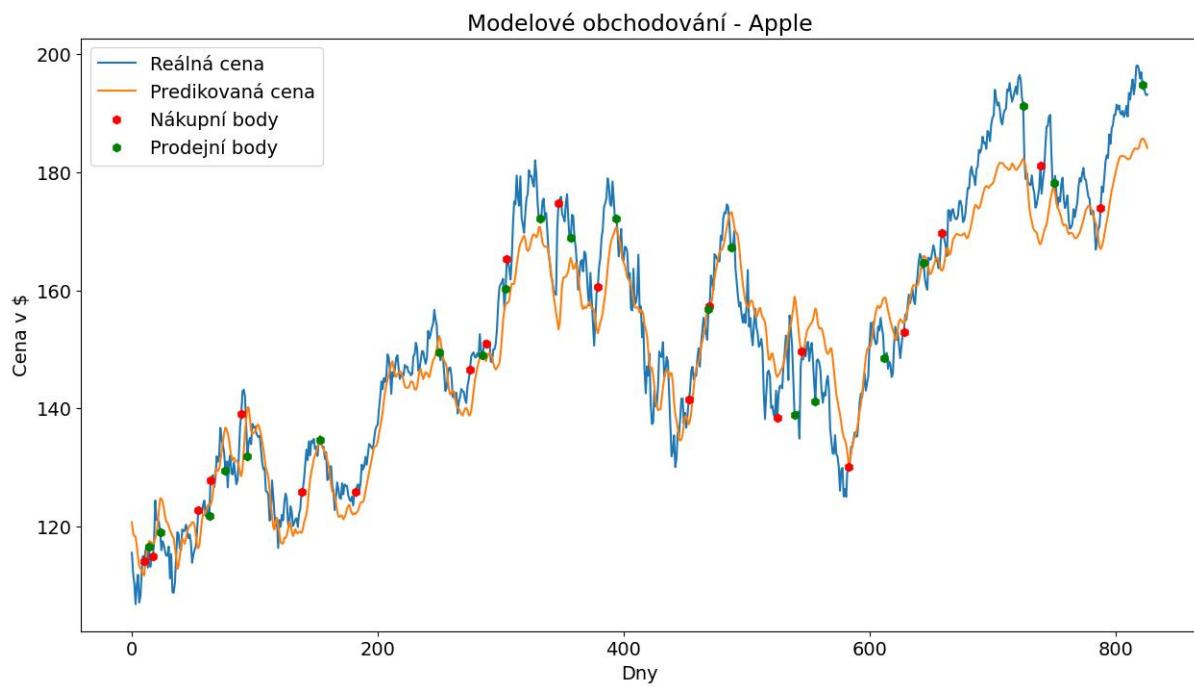
Obrázek 33: Graf porovnání modelu pro akcie Apple na tréninkových datech

Na testovacích datech pak činí střední kvadratická chyba 6,51, Zároveň test PICP pro 5 % vychází na 75 % (pravděpodobnost že se predikce trefí do rozmezí).



Obrázek 34: Graf porovnávající predikci modelu akcií Apple na testovacích datech

Pro vyhodnocení využitelnosti bylo provedeno modelové obchodování. Pro lepší reprezentaci se berou v potaz pouze změny větší než 5 %. Výsledné body jsou pak zobrazeny na Obrázek 35.

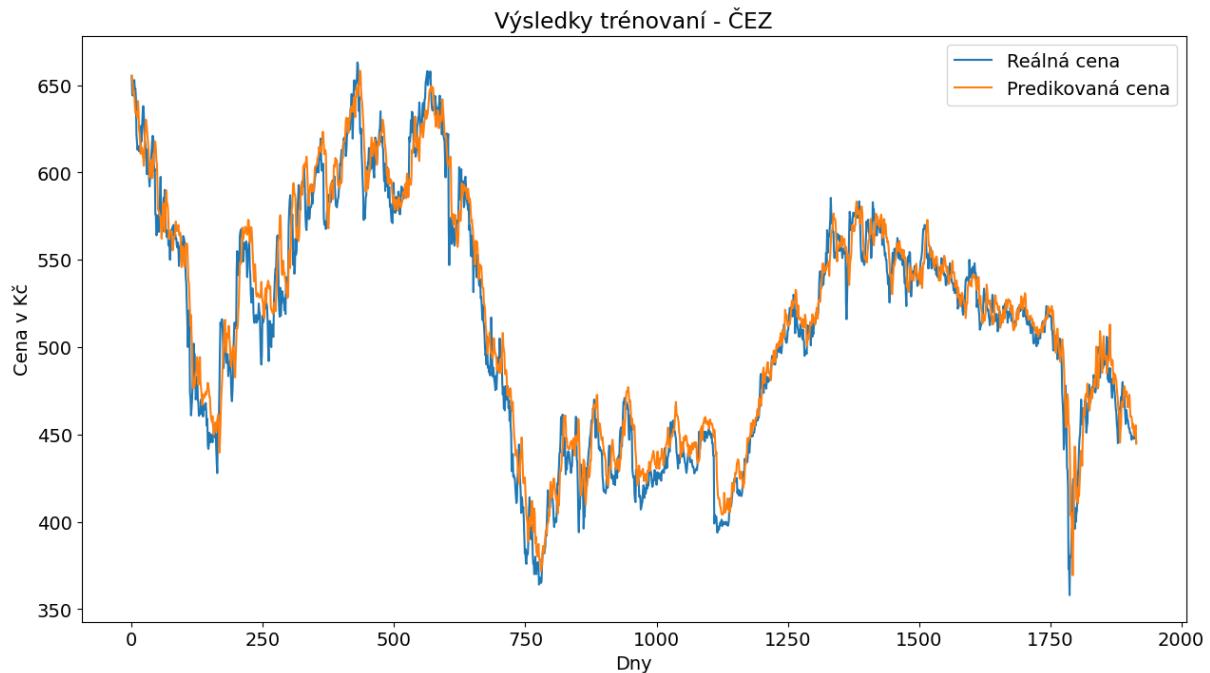


Obrázek 35: Modelové obchodování s modelem akcií Apple, 5% omezení

Pokud by se provádělo obchodování na tomto omezenějším výstupu, tak při vstupním kapitálu 10000 Kč, by částka na konci období činila 25297 Kč, teoretický zisk by tak činil 15297 Kč. Tato částka by ale nebyla konečná, museli by se započítat i poplatky spojené s obchodováním (které se liší na základě kdo a jaké služby poskytuje). Ty jsou od fixních částeck až po procenta z celkového kapitálu.

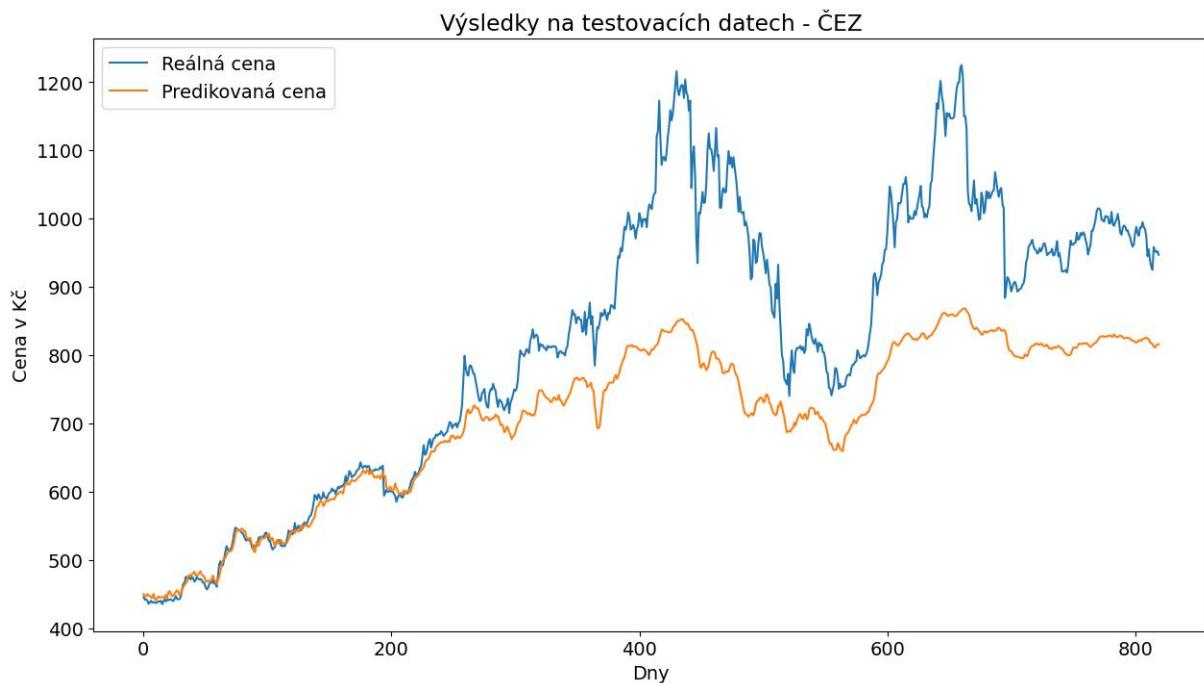
6.3 Výsledky pro ČEZ

Data použitá, po úpravě na základě ANOVA a RFE, mají velikost 2741x50 (opět bez přidaných proměnných a trendů). Pro předpověď ceny akcií skupiny ČEZ nebyly použity Fourierovy transformace. Trénink vychází s chybou RMSE s hodnotou 15,58.



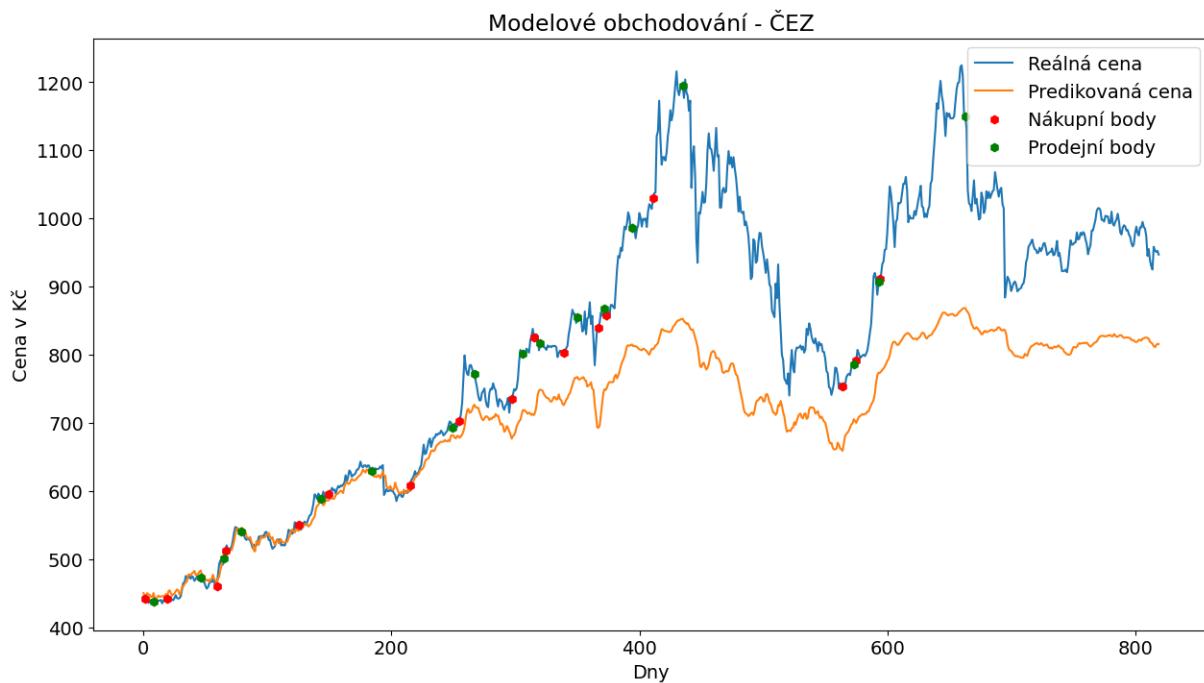
Obrázek 36: Výsledek tréninku na datech pro predikce skupiny ČEZ

Testování má již větší chybu a to 148,53. To je nejvíce způsobeno velkým nárustem v druhé polovině testovacího souboru (roky 2022 a 2023). To souvisí s celoevropskými událostmi (Válka na Ukrajině s tím spojená energetická krize, cena elektřina na burze prudce stoupala). Ty pohnuli s cenami všech energetických společností v Evropě, tedy i skupiny ČEZ. Model rozpoznal růsty v daném období, ale neodhadl jejich velikost.



Obrázek 37: Výsledek testování modelu pro skupinu ČEZ

Využitím tohoto modelu bylo vytvořeno modelové obchodování, opět s omezením 5%. Body obchodování jsou pak zobrazeny na Obrázek 38.

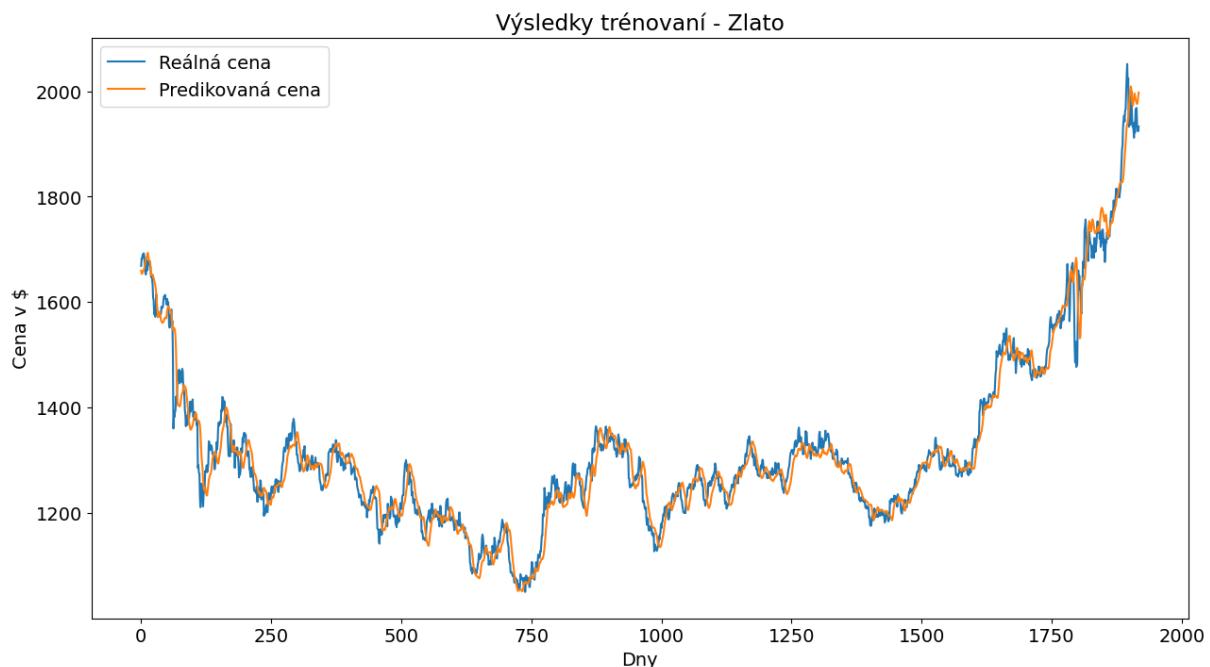


Obrázek 38: Modelové obchodování pro skupinu ČEZ

Za předpokladu vstupního kapitálu 10000Kč tak s vybraným modelovým obchodováním je na konci sledovaného období suma zvýšena na 41019 Kč. Tedy teoretický zisk činí 31019 Kč. Opět však nejsou zahrnuty poplatky.

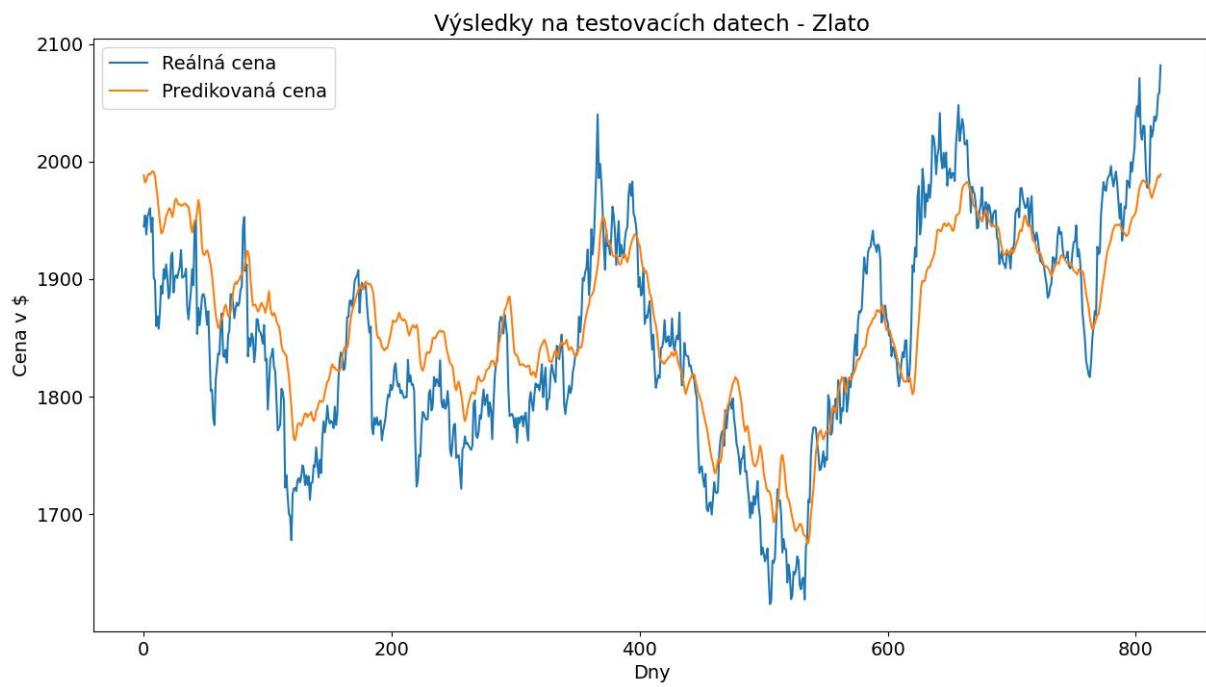
6.4 Výsledky pro zlato

Model předpovídá ceny zlato měl dataset o velikosti 2745x70 (bez přidaných věcí). Nebyli pro jeho výpočet přidány Fourierovy transformace. Výsledek po trénování RMSE je pak roven 32,4.



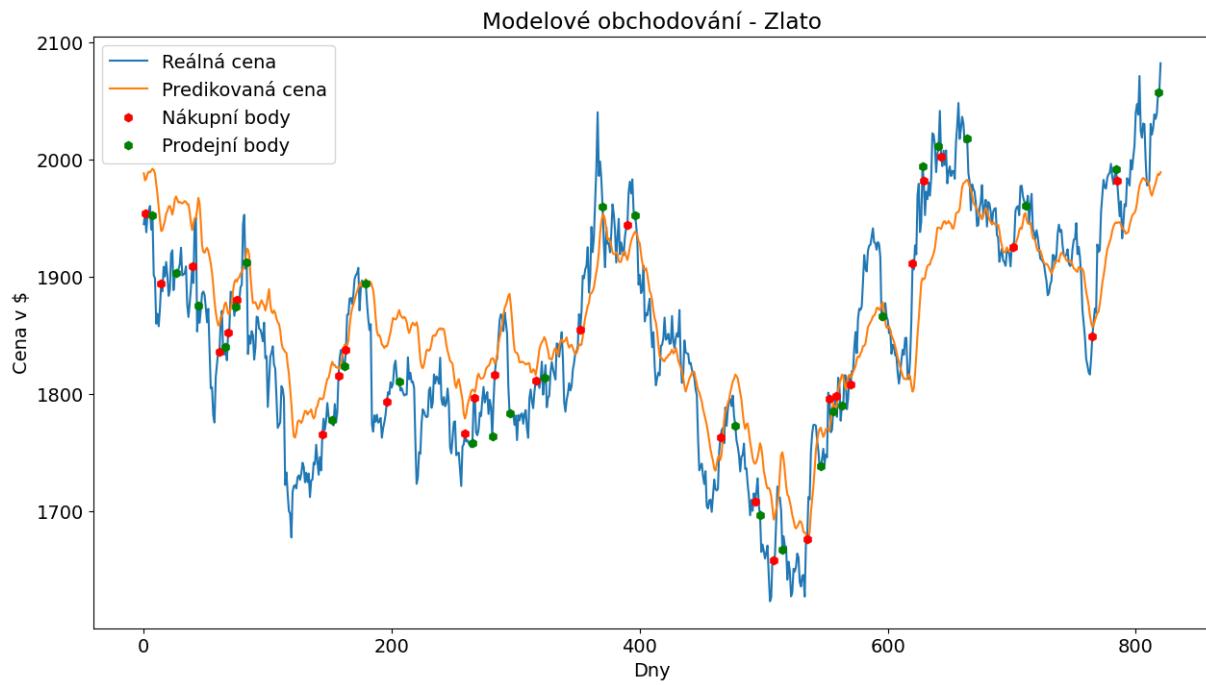
Obrázek 39: Výsledky trénování pro zlato

Pro testovací část dat pak RMSE činí 46,57 a PICP na 5 % se hodnota dostane s pravděpodobností 95,7 % (to je způsobeno lehký zpomalením bodů, které v modelu dělají velkou chybu, proto je pravděpodobnost dost velká).



Obrázek 40: Výsledky na testovacích datech pro Zlato

Na tomto modelu bylo také provedeno modelové obchodování. Tentokrát je omezení pouze 1 % (cena nemá tak velké změny, změny pro 5 % nejsou tak časté). Body je pak možné vidět na obrázku 41.

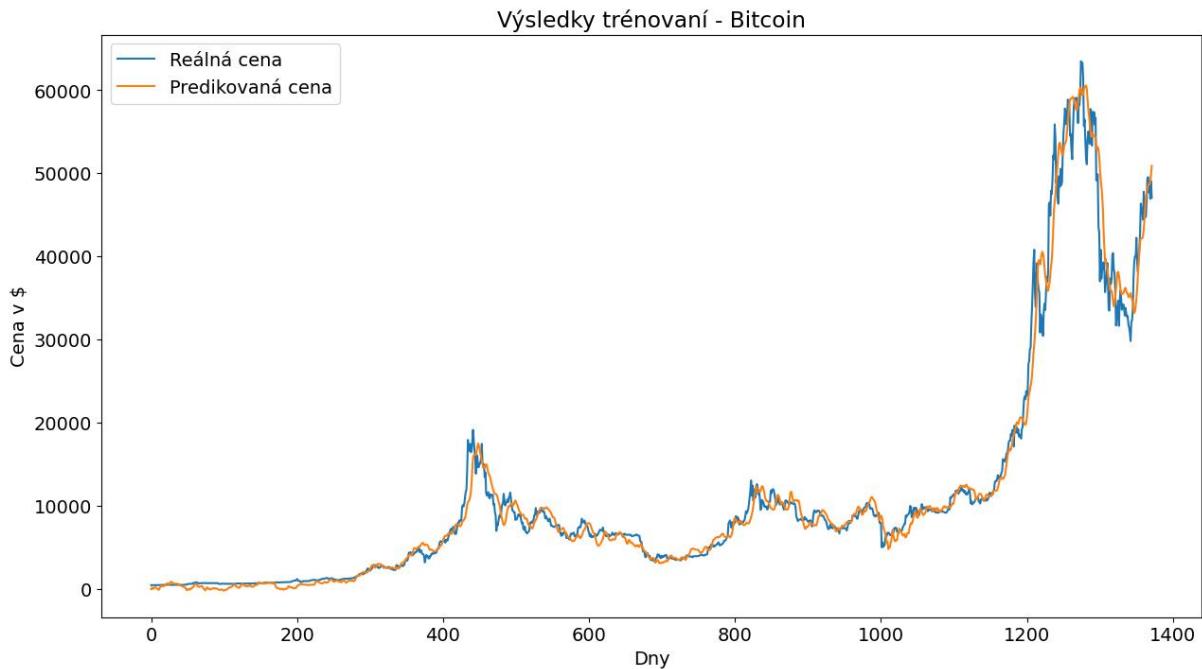


Obrázek 41: Modelové obchodování pro zlato

Na omezení pro 1 % se ze vstupního kapitálu 10000 Kč proinvestovalo 10381 Kč. Pokud by se opět posunulo omezení na 5 % tak tam výsledek již činí 12641 Kč, dojde však jenom ke čtyřem obchodům (což může být i dobrá zpráva, pokud se platí za každou provedenou transakci poplatek).

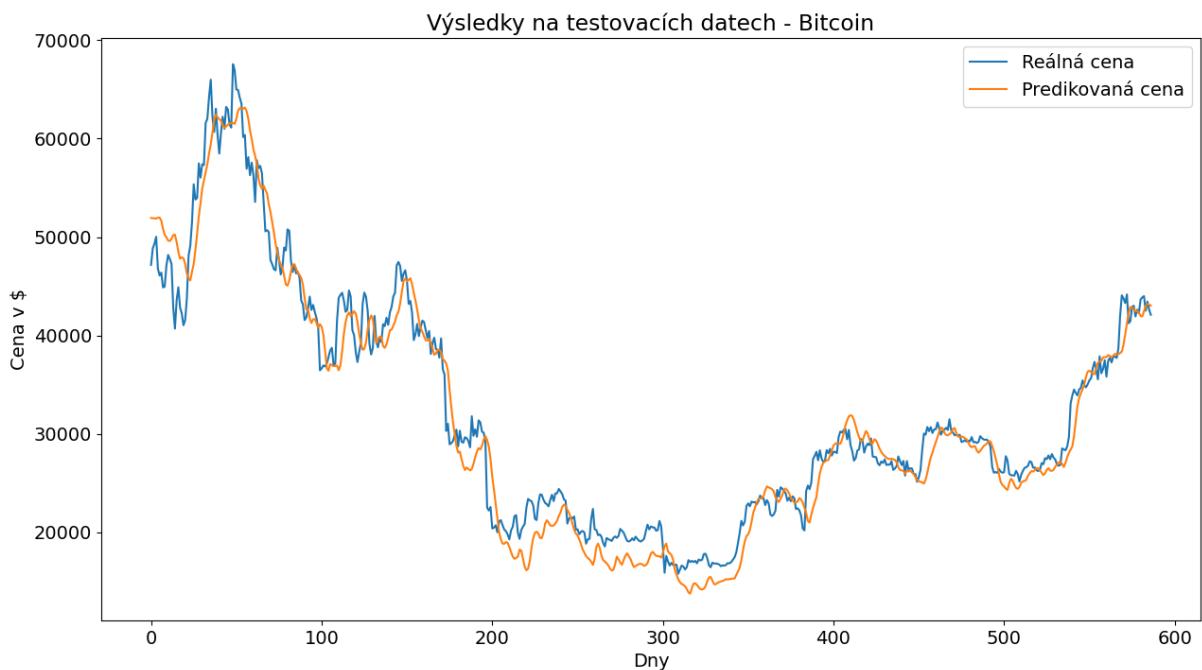
6.5 Výsledky pro Bitcoin

Dataset pro model předpovědi Bitcoinu po úpravách vlastností činí 1965×75 , bez přidaných proměnných. V tomto případě je využíváno Fourierových transformací. Trénovaní bylo provedeno s RMSE o hodnotě 1949.



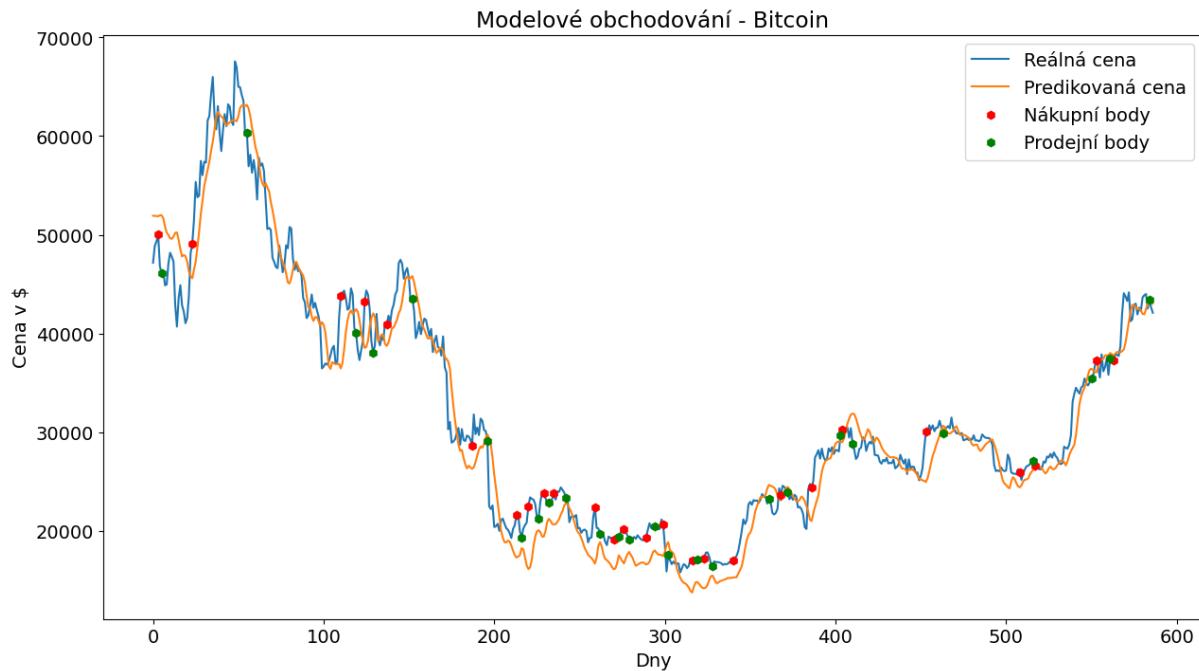
Obrázek 42: Výsledky trénování pro cenu Bitcoinu

Chyba i při trénování je nejvíce způsobena zpožděním modelu a velkou místní volatilitu. Každý menší posun už znamená velkou chybu, jelikož cena kolísá poměrně drasticky (pokud to porovnáme se zlatem, nebo akcemi). Chyba při testování pak dosáhla 2604, PICP na 5 % pak je 47,53 %.



Obrázek 43: Výsledek na testovacích datech pro cenu Bitcoinu

I v tomto případě došlo na modelové obchodování. Kryptoměny pak obecně dosahují větších změn, a tedy bylo omezení nastaveno na 5 %. Body korespondující s obchodní taktikou jsou vyobrazeny na Obrázek 44.



Obrázek 44: Modelové obchodování pro Bitcoin

Pokud by vstupní kapitál činil 10000 Kč, tak na výstupu modelu ve sledovaném období by výstup činil 15167 Kč, a to i když model nemá tak velkou přesnost. Vysoká volatilita je negativum i pozitivum.

7 Závěr

GAN neuronová sítě jsou partií hlubokého učení. Základní princip je na bázi minmax hry. Skládá se ze dvou částí, diskriminátoru a generátoru. Generátor tvoří data na základě vstupního souboru. Diskriminátor se snaží rozeznat, zdali se jedná o data generovaná nebo reálná. Při každém průchodu se pak modely navzájem vylepšují.

Použitý GAN se skládal z generátoru GRU a diskriminátoru CNN. Generátor obsahoval 3 vrstvy GRU s odpadnutím 0,2, pokračované 3 plně propojenými vrstvami. CNN pak má 3 konvoluční vrstvy, první dvě mají velikost jádra 5, poslední má 3. Ty následuje srovnávací vrstva, které data převádí do jedné dimenze. Ukončují ho 3 plně propojené vrstvy.

Model GAN sítí byl použit na predikce akcií Apple, skupiny ČEZ, komoditu zlato a kryptoměnu Bitcoin. Pro první 3 jmenované se jedná o období od 3.1.2013 do 28.12.2023 a pro kryptoměnu Bitcoin pak od 10.3.2016 do 28.12.2023. Každý z daných cílových vektorů měl dataset obsahující k nim relevantní vlastnosti. Obecně, hlavně u světových akcií a komodit, jsou data volně dostupná i v dobrém formátu. Pokud se jedná o více lokální věci (např. firma PRE) nebo některé specifické komodity (cena elektřiny, ceny uhlí) jsou také veřejně dostupné ale jejich získání je obtížnější a tím pádem i případná automatizace se tím ztěžuje (cena elektřiny je dostupná pouze v souborech po jednom roce v souborech se všemi jiným věcmi souvisejícími s trhem elektřiny). Google trendy, které byly využity jako „hodnota popularity“. Poskytují informace o velkých změnách (příkladem můžou být ceny elektřiny v letech 2022 a 2023, téma bylo velice diskutované a firmy s nimi obchodujícími jako je ČEZ to ovlivní) a jsou tedy poměrně jednoduchým nástrojem, jak přidat charakter chování trhu. I když firma Google tyto data dává volně a zdarma k dispozici, nejedná se přímo o hodnoty vyhledávání, ale o poměrová data v procentech v daném vybraném úseku. Zároveň pro úseky delší, jak půl roku, je předává v průměrech přes 3 dny až měsíc (podle velikosti úseku). Data se tedy musí nabírat postupně pro každý půl rok sledovaného období a poté se vynásobí průměrnými hodnotami z dat za celé sledované období.

I přes obtížnější získání jsou data poměrně přínosným ukazatelem, kdy s jejich pomocí má například model Apple akcií chybu pouze 6,5. Funkčnost modelu tím není ale degradována, což je možno vidět na modelovém obchodování při kterém se i s omezením dá dosáhnout velkých teoretických zisků. Porovnání s výsledky ostatních prací se nejedná taky o špatný výsledek. Jainendra Kumar Jain a Ruchit Agrawal měli ve své práci několik modelů od obyčejného GAN až po jejich vylepšenou variaci zvanou FB-GAN. Základní struktura je obdobná té, která je použitá v práci. Jedná se GRU generátor s CNN diskriminátorem. Skladba byly stejná jako v práci. Součástí jejich modelu byla sentimentální analýza zpráv z trhu. Výsledky v rámci RMSE byli pro Apple akcie 6,53 pro čistý GAN a 4,35 pro FB-GAN se sentimentální analýzou zahrnující jak nadpis, tak shrnutí [37]. Jiná práce s názvem *Multi-factor stock price prediction based on gan-trellisnet* využívající GAN-LSTM, StockGan a GAN-Trellisnet měla nejlepší výsledek u poslední jmenované a ten činil 1,034. Ta má strukturu z generátoru na bázi Trellisnet a diskriminátor je CNN. Trellisnet je variací na CNN s dvěma hlavními změnami: má propojené všechny váhy skrze všechny vrstvy a vstupní data vchází v daný časový krok do všech vrstev najednou [38]. *Stock price prediction using BERT and GAN* je další prací zabývající se stejným modelem. Stavba je v tomto případě skoro stejná, liší se hlavně složení dat, kde kromě dat z finančních trhů a zpozděných proměnných využili i

sentimentální analýzu titulků novin. S ní pak dosáhli chyby RMSE 1,827 [24]. Tím ukazují, že velkým zlepšením se dá dokázat jejím využitím v modelu predikce.

Model pro předpověď ceny ČEZu měl horší výsledky než pro Apple, ale v modelovém obchodování měl větší teoretický zisk (dokonce největší ze všech uvažovaných variant). Chyba modelu byla však díky externalitám vysoká (již zmíněno výše). Tomu by mohla pomoci sentimentální analýza, která by mohla více podtrhnou vážnost dané situace. Zahrnuté trendy ji částečně také odhalí ale nedokážou poznat, jestli je pro akcie dobrá nebo špatná (případně kolik z těch vyhledáváních je „záporných“ a kolik „kladných“).

Zlato mělo teoretickou chybu menší než model předchozí, ale v modelovém obchodování už neobstál tak dobře. Jedním z důvodu, že zlato nebylo zdaleka tak volatilní jako předchozí akcie a zároveň model nedokázal tak dobře předpovědět dost přesně vývoj ceny v poměrně turbulentním období. Dal by se vylepšit dalšími globálnějšími ukazateli. Sentimentální analýza globálních událostí by mohla mít největší přínos (zlato se stále bere jako „stabilní“ investice pro mnohé, pokroky v technologiích využívajících tento kov ovlivňují sentiment také).

Poslední uvažovaný model byl pro Bitcoin. Ten měl nejhorší výsledky s ohledem na RMSE, ale to nemělo tak negativní vliv na výsledek modelového obchodování. U této predikce bylo nejvíce vidět zpomalení modelu za reálnými daty, a to byl hlavní přispěvatel k chybě. Jedná se o známý problém pro GAN modely a předpověď časových řad. Vylepšení by mohlo přinést využití jiné stavby modelu, větším počtem zpozděných proměnných nebo přidáním jiných dat přímo nesouvisejících s kryptoměnami (např. zprávy týkajících se světových bank). Pro porovnání v práci *Short-term Prediction of Bitcoin Price Based on Generative Adversarial Network* využívali model se stejnou strukturou jako používaný v této práci, ale jako data zahrnuli mnohem více faktorů. Bylo jich 738 a jednalo se od fundamentálních dat a technických indikátorů až po počty tweetů nebo i Google trendy využívané v této práci. Jejich výsledek pak činil 0,06 v hodnotách RMSE. Kdy jejich model nedosahoval velkých zpozdění a spíše špatně odhadnul výškovou hladinu ceny. Proto je také hladina chyby tak nízká [39].

Práce tedy navrhla model schéma dat pro predikce na finančních trzích. Ty byly prováděny na více druzích artiklů pohybujících se na nich. Přesnost tohoto řešení pak byla porovnána s několika dalšími pracemi (hlavně v rámci akcií Apple, které jsou jedny z nejpoužívanějších u predikcí finančních trhů). Tím byl cíl práce splněn.

Zdroje

- [1] GOODFELLOW, Ian; BENGIO, Yoshua a COURVILLE, Aaron. *Deep learning. Adaptive computation and machine learning series*. Cambridge, MA: MIT press, [2016]. ISBN 0262035618.
- [2] SARKER, Iqbal H. *Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions*. SN computer science, 2021, vol. 2, no. 6. doi: 10.1007/s42979-021-00815-1.
- [3] AGGARWAL, Charu C. *Neural networks and deep learning: a textbook*. Second Edition. Cham: Springer Nature Switzerland, [2023]. ISBN 978-3-031-29641-3.
- [4] PASCANU, Razvan; GULCEHRE, Caglar; CHO, Kyunghyun a BENGIO, Yoshua. *How to Construct Deep Recurrent Neural Networks*. 2014, v. 5.
- [5] ZHANG, Aston; LIPTON, Zachary C.; LI, Mu a SMOLA, Alexander J. *Dive into Deep Learning*. Cambridge University Press, 2023. ISBN 978-1009389433.
- [6] NAMATĒVS, Ivars. *Deep Convolutional Neural Networks: Structure, Feature Extraction and Training*. Information Technology and Management Science. 2017, vol. 20, no. 1, s. 40–47. ISSN 2255-9094.
- [7] KRICHEN, Moez. *Convolutional Neural Networks: A Survey*. Computers. 2023, vol. 12, no. 8.
- [8] KRIZHEVSKY, Alex; KRIZHEVSKY, Sutskever a HINTON, Geoffrey E. *ImageNet Classification with Deep Convolutional Neural Networks*. Advances in Neural Information Processing Systems. 2012, vol. 25.
- [9] GOODFELLOW, Ian, et al. *Generative adversarial nets*. Advances in neural information processing systems, 2014, 27.
- [10] SOCH, Joram, et al. *The Book of Statistical Proofs*. Zenodo. 2023. doi: 10.5281/ZENODO.4305949
- [11] FRANK, Nielsen. *On the Jensen–Shannon Symmetrization of Distances Relying on Abstract Means*. Entropy. 2019, vol. 23, no. 4. doi: 10.3390/e21050485
- [12] ABDUL, JABBAR; XI, LI a BOURAHLA, OMAR. *A Survey on Generative Adversarial Networks: Variants, Applications, and Training*. ACM Computing Surveys (CSUR). 2021, vol. 54, no. 8, s. 1-46.
- [13] CHEN, Xi, et al. *Infogan: Interpretable representation learning by information maximizing generative adversarial nets*. Advances in neural information processing systems, 2016, vol. 29.
- [14] TAO, Ming, et al. *Df-gan: A simple and effective baseline for text-to-image synthesis*. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, s. 16515-16525.
- [15] LIAO, Wentong, et al. *Text to image generation with semantic-spatial aware gan*. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, s. 18187-18196.

- [16] LIM, Dan; JUNG, Sunghee a KIM, Eesung. *JETS: Jointly Training FastSpeech2 and HiFi-GAN for End to End Text to Speech*. INTERSPEECH 2022. 2022.
- [17] LIU, Songxiang; SU, Dan a YU, Dong. *DiffGAN-TTS: High-Fidelity and Efficient Text-to-Speech with Denoising Diffusion GANs*. Online. 2022. Dostupné z: <https://doi.org/10.48550/arXiv.2201.11972>.
- [18] PARMAR, Gaurav, et al. *Spatially-adaptive multilayer selection for gan inversion and editing*. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, s. 11399-11409.
- [19] PAN, Xingang, et al. *Drag Your GAN: Interactive Point-based Manipulation on the Generative Image Manifold*. Online. SIGGRAPH '23: ACM SIGGRAPH 2023 Conference Proceedings. no. 78, s. 1-11.
- [20] RICHARDSON, Elad, et al. *Encoding in style: a stylegan encoder for image-to-image translation*. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, s. 2287-2296.
- [21] SWIDERSKI, B.; GIELATA, L.; OLSZEWSKI, P.; OSOWSKI, S. a KOŁODZIEJ, M. *Deep neural system for supporting tumor recognition of mammograms using modified GAN*. Expert Systems with Applications. 2021, vol. 164. ISSN 0957-4174.
- [22] HU, Tao; LONG, Chengjiang a XIAO, Chunxia. *A Novel Visual Representation on Text Using Diverse Conditional GAN for Visual Recognition*. Online. Expert Systems with Applications. 2021, vol. 30, s. 3499-3512.
- [23] ZHANG, Yiwei; LI, Jinyang; WANG, Haoran a CHOI, Sou-Cheng T. *Sentiment-Guided Adversarial Learning for Stock Price Prediction*. Front. Appl. Math. Stat. 2021, vol. 7.
- [24] SONKIYA, Priyank; BAJPAI, Vikas; BANSAL, Anukriti. *Stock price prediction using BERT and GAN*. arXiv preprint arXiv:2107.09055, 2021.
- [25] *Apple Inc. summary*. Online. Encyclopedia Britannica. Dostupné z: <https://www.britannica.com/summary/Apple-Inc>. [cit. 2024-08-21].
- [26] *Gold, Silver & Precious Metals Futures Daily Data*. Online. Kaggle. Dostupné z: <https://www.kaggle.com/datasets/guillemservera/precious-metals-data>. [cit. 2024-08-21].
- [27] *S&P 500 and NASDAQ 100: Daily Data*. Online. Kaggle. Dostupné z: <https://www.kaggle.com/datasets/guillemservera/sp500-nasdaq-spy-qqq-ohlcv-data>. [cit. 2024-08-21].
- [28] *Zrození impéria. Skutečný příběh společnosti ČEZ*. Online. Aktuálně.cz. 2013. Dostupné z: <https://zpravy.aktualne.cz/denik-insider/zrozeni-imperia-skutecny-pripeh-spolocnosti-cez/r~i:insider:article:17913/>. [cit. 2024-08-21].
- [29] ČEZ. Online. Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation. Dostupné z: <https://cs.wikipedia.org/wiki/%C4%8CEZ>. [cit. 2024-08-21].
- [30] *Bitcoin*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation. Dostupné z: <https://en.wikipedia.org/wiki/Bitcoin>. [cit. 2024-08-21].
- [31] CHANTARAKASEMCHIT, Orawan; NUCHITPRASITCHAI, Siranee a NILSIAM, Yuenyong. *Forex Rates Prediction on EUR/USD with Simple Moving Average Technique and*

Financial Factors. 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). s. 771-774.

- [32] CHEN, James. *What is EMA? How to Use Exponential Moving Average With Formula.* Online. Investopedia. Dostupné z: <https://www.investopedia.com/terms/e/ema.asp>. [cit. 2024-08-21].
- [33] LUND, Brian. *Bollinger Bands: A powerful technical tool for traders.* Online. Encyclopedia Britannica. Dostupné z: <https://www.britannica.com/money/bollinger-bands-indicator>. [cit. 2024-08-21].
- [34] MONTERVIGEN, Karl. *Navigating the markets with moving average convergence/divergence (MACD).* Online. Encyclopedia Britannica. Dostupné z: <https://www.britannica.com/money/macd-moving-average-convergence-divergence>. [cit. 2024-08-21].
- [35] RODRÍGUEZ-GONZÁLEZ, et al, Alejandro. *CAST: Using neural networks to improve trading systems based on technical analysis by means of the RSI financial indicator.* Expert Systems with Applications. 2011, vol. 38, no. 9, s. 771-774.
- [36] Multirate Signal Processing. Online. In: *Multirate and Wavelet Signal Processing. Wavelet Analysis and Its Applications.* Elsevier, 1998, s. 1-28. ISBN 9780126775600. Dostupné z: [https://doi.org/10.1016/S1874-608X\(98\)80046-0](https://doi.org/10.1016/S1874-608X(98)80046-0). [cit. 2024-08-19].
- [37] JAIN, Jainendra Kumar; AGRAWAL, Ruchit. *FB-GAN: A Novel Neural Sentiment-Enhanced Model for Stock Price Prediction.* Proceedings of the Joint Workshop of the 7th Financial Technology and Natural Language Processing, the 5th Knowledge Discovery from Unstructured Data in Financial Services, and the 4th Workshop on Economics and Natural Language Processing@ LREC-COLING 2024. 2024. s. 85-93.
- [38] LIU, Wenjie; GE, Yebo; GU, Yuchen. *Multi-factor stock price prediction based on gan-trellisnet.* Knowledge and Information Systems, 2024, s. 1-22.
- [39] MOOSAKHANI, M.; JAHANGARD-RAFSANJANI, A.; ZARIFZADEH, S. *Short-term Prediction of Bitcoin Price based on Generative Adversarial Network.* Journal of Electrical and Computer Engineering Innovations (JECEI), 2024, vol.12, no.2, s. 485-496.