VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
DEPARTMENT OF COMPUTER SYSTEMS

ANALÝZA A PREDIKCE ČASOVÝCH ŘAD POMOCÍ NEURONOVÝCH SÍTÍ

TIME-SERIES ANALYSIS AND PREDICTION BY MEANS OF NEURAL NETWORKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **MARTIN KŇAŽOVIČ**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. MICHAL BIDLO, Ph.D.**
SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

147458

Ústav:	Ústav počítačových systémů (UPSY)
Student:	**Kňažovič Martin**
Program:	Informační technologie
Specializace:	Informační technologie
Název:	**Analýza a predikce časových řad pomocí neuronových sítí**
Kategorie:	Umělá inteligence
Akademický rok:	2022/23

Zadání:

1. Seznamte se s problematikou časových řad, možnostmi jejich analýzy a predikce.
2. Nastudujte základy neuronových sítí se zaměřením na jejich použití v oblasti analýzy a predikce časových řad.
3. Zvolte vhodný typ neuronové sítě a postup pro její trénování na vybraném vzorku časových řad.
4. Navržený systém implementujte pro alespoň dva různé scénáře s cílem dosažení co nejvyšších přesností analýzy a predikce časových řad.
5. Vykonejte sadu experimentů za účelem srovnání schopností systému z bodu 4 řešit danou úlohu.
6. Statisticky vyhodnoťte dosažené výsledky, diskutujte jejich vlastnosti, případně způsoby využití, a možnosti pokračování projektu.

Literatura:

- Dle pokynů vedoucího projektu.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 až 3 zadání a demonstrace implementace prediktoru z bodu 4 formou prototypu.

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

Vedoucí práce:	**Bidlo Michal, Ing., Ph.D.**
Vedoucí ústavu:	Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání:	1.11.2022
Termín pro odevzdání:	10.5.2023
Datum schválení:	31.10.2022

Abstrakt

Táto práca sa zaoberá predikciou cien akcií a to vytvorením predikčných modelov pre vybrané akcie (BRK-A, GOOG a MSFT), ktoré môžu pomôcť investorom pri tvorbe ich investičných rozhodnutí či ako náhrada predikčných modelov v už existujúcich systémoch. V tejto práci sa venujeme tvorbe dvoch typov modelov - jedno-premenného a viac-premenného modelu, pričom obidva sú prezentované vo výslednej podobe v dvoch architektúrach, jednovrstvovej a dvojvrstvovej. Tieto modely sú postavené na princípe neurónových sietí, konkrétne ich podtypu rekurentných neurónových sietí, ktoré využívajú rozšírenie long short-term memory. Výstupom prezentovaných modelov je predikovaná cena nasledujúceho dňa, ktorú je možné použiť na zváženie vhodnosti nákupu, alebo predaja danej akcie. Kvalita jednotlivých predikčných modelov je vyhodnotená na základe strednej kvadratickej chyby (angl. Mean Squared Error) validačnej, prípadne testovacej dátovej sady, ale aj alternatívnym spôsobom na základe predikcie zmeny trendu akcie.

Abstract

This thesis deals with stock price prediction based on the creation of prediction models for selected stocks (BRK-A, GOOG, and MSFT), which can help investors in the creation of their financial decisions or by replacing other stock prediction models in existing prediction systems. Models created in this thesis are presented in two types - univariate model and multivariate model, which are in their final version presented in two architectures, one-layer architecture and two-layer architecture. Discussed models are created by means of neural networks, specifically recurrent neural networks with its extension - Long short-term memory. The output of the presented models is a forecast of the next-day stock price, which can be used for evaluating the right time to buy or sell a given stock. The quality of individual prediction models is evaluated via the mean squared error of the validation or testing dataset or alternatively based on stock price trend prediction.

Kľúčové slová

analýza časových radov, predikcia cien akcií, rekurentné neurónové siete, long short-term memory, predikčné modely, predikcia trendu akcie, krátkodobá predikcia, optimalizácia parametrov modelu, regresia, analýza dát, normalizácia dát

Keywords

time-series analysis, stock price prediction, recurrent neural networks, long short-term memory, prediction models, prediction of stock trend, short term forecasting, model parameter tunning, regression, data analysis, data normalization

Citácia

KŇAŽOVIČ, Martin. *Analýza a predikce časových řad pomocí neuronových sítí*. Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Michal Bidlo, Ph.D.

Analýza a predikce časových řad pomocí neuronových sítí

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedeným Ing. Michala Bidla, Ph.D.. V práci uvádzam všetky literárne zdroje v zozname literatúry a internetové zdroje z ktorých som čerpal formou poznámok pod čiarou.

.....
Martin Kňažovič
4. mája 2023

Poďakovanie

Chcel by som sa poďakovať môjmu vedúcemu práce Ing. Michalovi Bidlovi, Ph.D za všetky cenné rady, ktoré mi dal či už pri technickej realizácii, alebo pri písaní a za jeho ochotu pri konzultovaní všetkých výziev tejto práce. Rovnako by som sa chcel poďakovať mojej rodine a priateľom za to, že mi boli veľmi cennou morálnou oporou.

Obsah

Zoznam použitých skratiek

Skratka	**Význam**
ANN, NN	neurónová sieť
ARIMA	Autoregressive integrated moving average (predikčný model)
ARMA	Autoregressive–moving-average model (predikčný model)
CNN	konvolučné neurónové siete
DS, ds	dátová sada
DWT	diskrétna vlnková transformácia
GSXGB	vylepšená varianta XGB
LSTM	Long short-term memory
MA	pohyblivý priemer (angl. moving average)
MAE	Mean Absolute Error (metrika)
MARS	multivariate adaptive regression spline (forma regresnej analýzy)
max	maximum
min	minimum
MMS	kanditátne riešenie zložené z minima, maxima, štandardnej odchýlky
MSE	Mean Squared Error (metrika)
nds	normalizovaná dátová sada
RF	Random forest (algoritmus strojového učenia)
RMSE	Root Mean Square Error
RNN	rekurenté neurónové siete
RSI	index relatívnej sily
SD, std	štadardná odchýlka
tanh	hyperbolický tangens
VOL, v	obchodovaný objem akcie
XGB	extreme gradient boosting (algoritmus)
GSXGB	vylepšená varianta XGB
XMA	exponenciálny pohyblivý priemer (angl. exponential moving average)

Kapitola 1

Úvod

Obchodovanie na akciových trhoch sa v posledných rokoch značne skomplikovalo, hlavne kvôli zvýšenej volatilite akcií spôsobenej externými faktormi, ktorými sú napr. pandémia COVID 19 či vojna na Ukrajine. Preto je dnes pre investorov obzvlášť dôležité mať nástroje, ktoré môžu pozitívne ovplyvniť ich investičné rozhodnutia. Jedným z takýchto nástrojov sú prediktívne modely, ktoré poskytujú cenné informácie vo forme predpovedí o budúcom správaní akcií.

Cieľom tejto práce preto bude vytvoriť krátkodobé prediktívne modely, ktoré sa budú čo najpresnejšie snažiť predpovedať uzatváraciu cenu akcií nasledujúceho dňa na základe historických cenových dát. Ako prostriedok na vytvorenie spomínaných modelov sme zvolili neurónové siete, ktoré poskytujú nespornú výhodu oproti bežne používaným prediktívným modelom ako napr. ARMA či ARIMA, lebo dosahujú vyššiu presnosť pri predikcii nelineárnych funkcií. [19]

V tejto práci sa konkrétne budeme venovať návrhu dvoch typov modelov - jednopremenného modelu, ktorý bude predikovať uzatváraciu cenu nasledujúceho dňa na základe predošlých uzatváracích cien a viac-premenného modelu, ktorý bude okrem uzatváracích cien využívať niekoľko ďalších faktorov k predikcii uzatváracích cien nasledujúceho dňa.

Tieto modely budú tvorené vo frameworku Tensorflow, pričom najskôr popíšeme všetky potrebné prerekvizity k ich tvorbe, ktorými je napr. charakteristika a predspracovanie dát a následne popíšeme proces ich tvorby v samostatných experimentoch, ktorých výsledky budeme prezentovať vo forme grafov a tabuliek. Vyhodnotenie kvality predikcií bude spočívať v porovnaní výsledkov predikcií modelov s reálnymi hodnotami referenčných akcií (BRK-A, GOOG a MSFT) na validačnej či testovacej dátovej sade podľa potrieb jednotlivých experimentov. Výsledky týchto porovnaní budú vo väčšine prípadov uvádzané pomocou strednej kvadratickej odchýlky, ktorú predstavíme v druhej kapitole. Na záver tejto práce taktiež uvedieme zhodnotenie všetkých dosiahnutých výsledkov a námety na ďalšie pokračovanie práce.

Kapitola 2

Predikcia časových radov

Predikcia časových radov je téma, ktorá je v spoločnosti už dlhodobo skúmaná, lebo je široko využiteľná naprieč rôznymi odvietviami a jej poznanie nám umožňuje sa do istej miery pripraviť na budúcnosť. V tejto práci sa zamerievame na jej podmnožinu, a to predikciu finančných časových radov s cieľom krátkodobej predikcie uzatváracích cien akcií nasledujúceho dňa. Hlavnými aktérmi, na ktorých budeme naše predikcie demonštrovať, budú akcie spoločností Berkshire Hathaway, Alphabet Inc (Google) a Microsoft, prezentované pod skratkami BRK-A, GOOG a MSFT. V rámci tejto kapitoly si predstavíme všeobecný princíp predikcie a charakter dát, ktoré budeme v tejto práci používať, potrebné kroky z hľadiska predspracovania dát, akými sú normalizácia a rozdeľovanie dátových sád a v neposlednom rade, metódy na vyhodnocovanie správnosti predikcií a momentálne trendy v oblasti predikcií finančných dát.

2.1 Princíp predikcie

V súčastnosti existuje viacero spôsobov, ktorými sa obchodníci snažia predikovať cenu akcií, či už sa jedná o manuálne metódy ako napr. technická analýza, alebo sofistikovanejšie spôsoby, akými je napr. tvorba predikčných modelov, ktorou sa v tejto práci zaoberáme. Je preto dôležité správne pochopiť, čo je myslené pod pojmom model. Ako údava encyklopédia Britannica, pojem vedecké modelovanie je možné definovať ako:

Model je fyzická, konceptuálna alebo matematická reprezentácia reálneho fenoménu, ktorý je ťažko pozorovateľný priamo. Vedecké modely sú používané na vysvetľovanie a predikciu správania reálnych objektov alebo systémov a používajú sa v rôznych vedeckých disciplínach, . . . [15],

Takže pod pojmom model budeme v tejto práci rozumieť aproximovanú matematickú reprezentáciu funkcie cien konkrétnej akcie v závislosti na čase, pričom táto aproximácia funkcie vznikne procesom regresie, t.j. „*pokúsime sa odhadnúť krivku, ktorá najlepšie reprezentuje všeobecný trend dátovej sady*“. [4]

Dnes sa v praxi na predikciu častokrát používa podtyp regresných modelov - lineárne regresné modely, ktoré môže reprezentovať jednoduchší ARMA model(detail viď. [18]) či komplexnejší ARIMA model(detail viď. [23]). Problém týchto modelov však spočíva v ich linearite, a teda neschopnosti správne zachytiť nelineárne krivky, ktorými sú napr. aj časové rady cien akcií. Z tohto dôvodu v práci pracujeme s neurónovými sieťami, ktoré používajú nelinárnu regresiu na aproximovanie funkcie ceny akcie. Tieto modely musia byť trénované

zvlášť pre každú akciu, ktorú chceme predikovať. Vstupy jednotlivých modelov sú popísané ďalej pri ich návrhu v kapitole 3 a výstupom týchto modelov je predikovaná cena nasledujúceho dňa. Avšak predtým, ako sa budeme bližšie venovať navrhovaným modelom, sa musíme najskôr zamerať na dáta, ktoré budeme k ich trénovaniu a predikovaniu potrebovať.

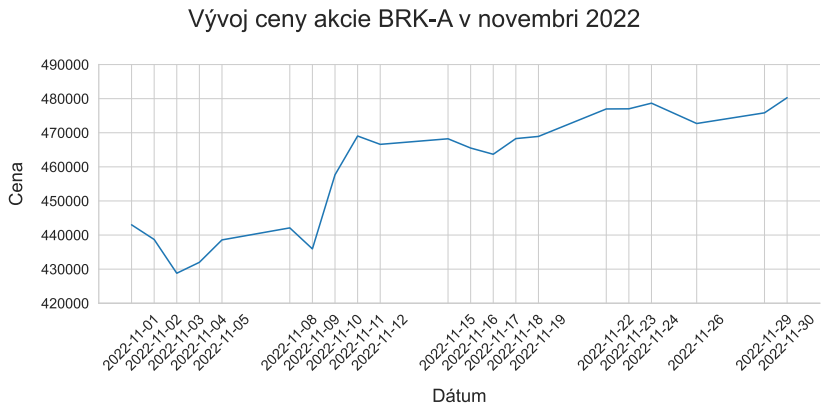
2.2 Charakteristika dát

Pre účely tejto práce boli použité finančné dáta získané z verejného portálu Yahoo Finance[1], ktoré majú nasledovný formát:

Tabuľka 2.1: Ukážka získaných dát pre akciu BRK-A, pričom stĺpce zľava doprava nám znázorňujú: dátum obchodného dňa, prvá, najvyššia, najžnišia, uzatváracia a upravená uzatváracia cena tohto dňa a obchodovaný objem akcie tohto dňa.

Date	Open	High	Low	Close	Adj. Close	Volume
2010-01-04	99350	99910	99201	94800	94800	133500
2010-01-05	99790	100001	99550	99710	99710	118900
2010-01-06	100000	100000	99500	99850	99850	57900
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2019-12-3	338750	340000	337920	339590	339590	100

Ako môžete vidieť v tabuľke 2.1, každý riadok je označený dátumom obchodného dňa, toto je obzvlášť dôležité, lebo s akciami sa neobchoduje každý deň, ale v priemere to je približne 252 dní v roku. Pre účely našej práce je najviac dôležitý šiesty stĺpec (Adj. Close), ktorý obsahuje upravenú zatváraciu cenu akcie, ktorú používame na trénovanie modelov a predikciu. Upravenú uzatváraciu cenu používame preto, lebo lepšie reprezentuje hodnotu danej akcie, keďže oproti uzatváracej cene sú v nej započítané všetky faktory, ktoré by mohli ovplyvniť jej cenu po uzatvorení trhu v daný deň (napr. dividendy, rozdelenia akcií). [6] Ďalším užitočným údajom je posledný stĺpec (Volume), ktorý obsahuje hodnoty obchodovaného objemu akcie, t.j. počet jednotiek akcií, s ktorými sa v daný deň obchodovalo. Stĺpce s otváracou, najvyššou a najnižšou cenou v tejto práci nevyužívame. Pozrime sa teda ďalej ako vizuálne vyzerá priebeh ceny akcie:



Obr. 2.1: Vývoj ceny akcie BRK-A v novembri 2022

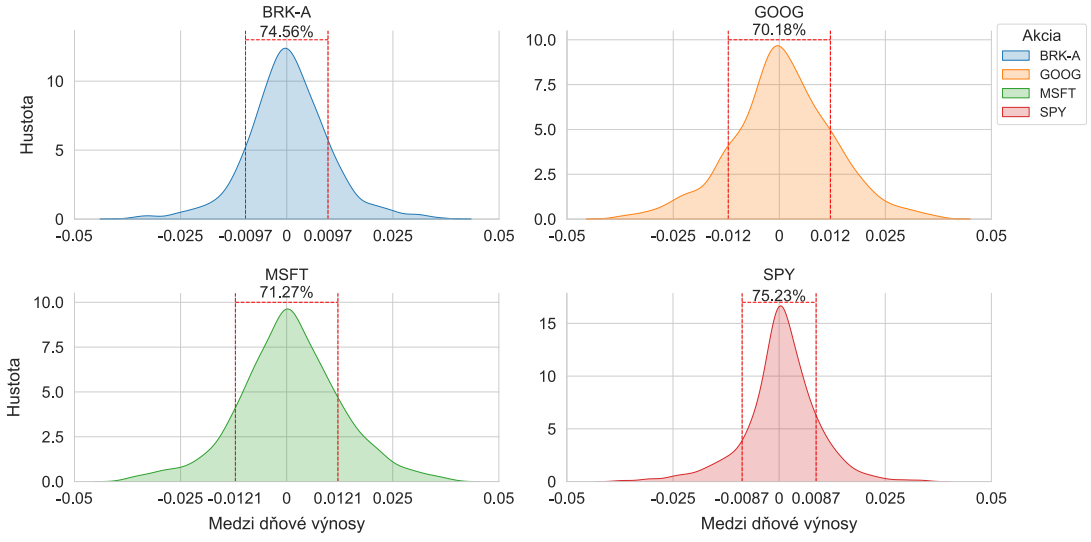
[1] portál Yahoo Finance (https://finance.yahoo.com/)

Pri pohľade na graf 2.1 môžeme vidieť znázornený priebeh akcie BRK-A v novembri 2022, v ktorom bolo 21 obchodných dní. Na prvý pohľad je jasné, že sa jedná o nelineárny časový rad, ktorého charakter nie je monotónny a vykazuje zmenu ceny v desaťtisícoch počas pár dní, čo ho robí ťažko predikovateľným. Ak chceme zistiť o charaktere týchto dát viac, tak sa musíme pozrieť na ich štatistické ukazovatele, pričom je dôležité upozorniť, že sa tieto ukazovatele môžu značne líšiť naprieč akciami. V nasledujúcej tabuľke preto môžete vidieť, ich porovnanie naprieč akciami BRK-A, GOOG, MSFT a SPY.

Tabuľka 2.2: Štatistické ukazateľe pre akcie BRK-A, GOOG, MSFT a SPY

Akcia	**Počet cien**	**Priemer**	**SD**	**SD výnosov**	**Obdobie**
BRK-A	5031	145024.95	77732.56	0.0113	2000 - 2020
GOOG	3868	24.47	16.93	0.0147	2000 - 2020
MSFT	5031	34.95	28.97	0.0147	2000 - 2020
SPY	5031	127.71	62.21	0.0108	2000 - 2020
BRK-A	2516	202706.45	70661.86	0.0097	2010 - 2020
GOOG	2516	32.18	16.27	0.012	2010 - 2020
MSFT	2516	50.97	34.17	0.012	2010 - 2020
SPY	2516	171.60	60.62	0.0087	2010 - 2020

V hornej časti tabuľky 2.2 sa nachádzajú štatistické ukazovateľe pre dátovú sadu tvorenú uzatváracími cenami akcií medzi rokmi 2000 až po rok 2020. V spodnej časti tabuľky sa nachádza menšia dátová sada zachytávajúca uzatváracie ceny akcií v období od roku 2010 po rok 2020. Menšia dátová sada (10 ročná dátová sada) vychádza z referenenčnej práce StockBot: Using LSTMs to Predict Stock Prices [12]. Väčšia dátová sada bola zvolená ako alternatíva k menšej dátovej sade v experimente, kde bol skúmaný vplyv dĺžky trénovacieho obdobia na úspešnosť modelu (trénovacím obdobiam sa budeme bližšie venovať v kapitole 5.5).



Obr. 2.2: Distribúcia medzi dňových výnosov 10-ročnej dátovej sady

Na to, aby sme mohli využiť štatistické ukazovatele prezentované v tabuľke 2.2, tak sa musíme pozrieť bližšie na pojem „volatilita akcie“ [8]. Tento pojem sa používa pri obchodovaní akcií a vyjadruje nám mieru zmeny ceny akcie, na základe ktorej vieme posúdiť risk danej investície. Výpočet volatility ($\sigma * \sqrt{T}$) [8] spočíva v spočítaní štandardnej odchýlky, ktorá sa vynásobí odmocninou zvolenej časovej periódy, pre ktorú volatilitu počítame (v našom prípade sa pozorujeme medzidňovú volatilitu, takže zvolená perióda je jeden deň), pričom väčšia hodnota znamená väčší rozptyl od priemernej ceny akcie, a teda značí väčšiu nestabilitu ceny akcie. Dôležitým faktom je, že keby sme chceli vyrátať volatilitu z uzatváracích cien, ktoré v tejto práci používame, tak by sme dostali hodnoty, ktoré sú uvedené v tabuľke 2.2 pod štandardnou odchýlkou, ale neboli by veľmi užitočné. Ak by sme totiž chceli použiť volatilitu (štandardnú odchýlku) na odhad, v akom rozsahu sa bude cena akcie pohybovať, tak vzniká problém, že ceny akcie nezvyknú mať normálne rozloženie, prípadne ak by sme chceli zvážiť výber akcie, do ktorej chceme investovať na základe porovnania volatility vybraných akcií, ako môžeme vidieť v tabuľke 2.2, tak sú tieto hodnoty diametrálne odlišné naprieč akciami. Tento problém je riešiteľný spôsobom, že nebudeme rátať štandardnú odchýlku z upravených uzatváracích cien, ale vypočítame z nich medzidňovú výnosnosť (dňovú percentuálnu zmenu ceny akcie). Medzidňová výnosnosť sa charakterom viac podobá na normálne rozloženie a jej priemerná hodnota je približne nula. Toto môžeme vidieť demonštrované na obrázku 2.2, kde sme sa pozreli na distribúcie návratností všetkých referenčných akcií a indexu S&P500 (SPY). Na grafoch sú znázornené červenými prerušovanými čiarami štandardné odchýlky medzidňových návratností a ako môžeme vidieť, tak rozsah hodnôt $\langle -SD, SD \rangle$ obsahuje zhruba 70 až 75% všetkých hodnôt, na základe čoho môžeme usúdiť predpokladané rozsahy návratností akcií. Čo je ešte dôležitejšie, je to, že tým, že sú si medzidňové návratnosti podobné, tak ich môžeme medzi sebou porovnávať. Dobrým spôsobom ako porovnávať rôzne faktory akcií, je dať ich do perspektívy voči trhovým indexom, a preto sme na porovnanie zvolili index S&P500 (SPY). [2] Z uvedených hodnôt v tabuľke 2.2 vidíme, že volatilita akcie BRK-A je porovnateľná s hodnotou volatility indexu S&P500, z čoho môžeme usúdiť, že majú podobnú mieru stability návratností akcií. Na druhej strane akcie GOOG a MSFT majú porovnateľne vyššiu hodnotu volatility ako index S&P500, takže ich návratnosti sú nestabilnejšie a investovanie do nich môže byť riskantnejšie. Z týchto zistení vzniká otázka „Má volatilita akcie vplyv na predikovateľnosť cien akcií?“. Túto otázku sa pokúsime zodpovedať v závere tejto práce.

Zaujímavou poznámkou k rozboru dát je, že pri dátach z rozmedzia rokov 2000 až 2020 má akcia GOOG „len“ 3869 uzatváracích cien, ale ostatné akcie majú 5032. Toto je spôsobené tým, že spoločnosti Microsoft a Berkshire Hathaway sú značne staršie oproti spoločnosti Google, ktorá prišla na akciový trh až v roku 2004.

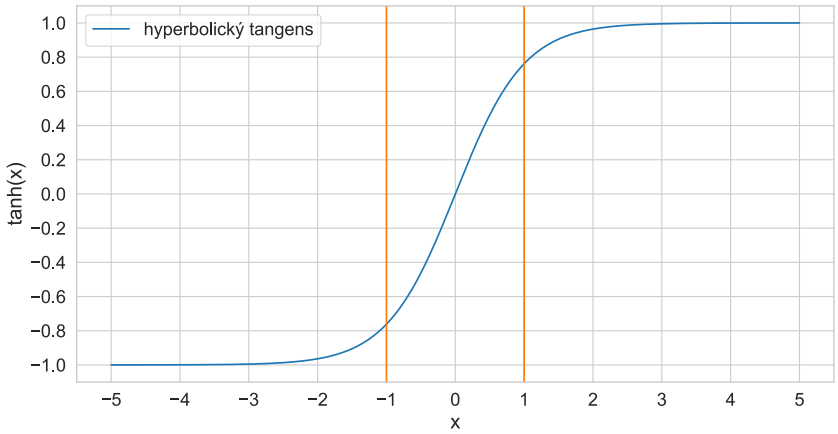
2.3 Predspracovanie dát

Vzhľadom na stanovený charakter dát je ich predspracovanie veľmi dôležité, lebo v prvom rade zlepšuje výslednú kvalitu modelov, ale taktiež zrýchľuje samotný proces trénovania. Dáta sme v tejto práci najskôr predspracovali tzv. **MaxAbsScaler**[2] normalizáciou, aby sme zaručili lepšie výsledky modelov, následne sme ich rozdelili do niekoľkých dátových sád v optimálnom pomere, čo umožnilo výsledné modely validovať a v neposlednom rade sme rozdelené dátové sady pripravili do dátových sád frameworku Tensorflow.

[2] zdroj: dokumentácia pre MaxAbsScaler (https://scikit-learn.org/stable/modules/classes.html#module-sklearn.preprocessing)

2.3.1 Normalizácia

Hlavnou ideou normalizácie je transformácia čistých vstupných dát do rovnomernejšie rozloženej dátovej sady, pričom je nevyhnutné škálovať tieto dáta pre potreby vytváraných modelov. [1] Tento proces je obzvlášť dôležitý pri navrhovaní modelov neuronovými sieťami, lebo faktory ako odľahlé hodnoty či škála vstupných hodnôt môžu výrazne ovplyvniť proces trénovania modelu a samotné výsledky. Vhodným príkladom z tejto oblasti sú aktivačné funkcie používané v jednotlivých vrstvách neurónových sietí. Medzi populárne voľby pre normalizáciu patrí funkcia hyperbolický tangens (tanh), ktorú sme taktiež použili pre účely tejto práce.



Obr. 2.3: Funkcia hyperbolický tangens

Ako je vidno na obrázku 2.3, táto funkcia je veľmi citlivá na hodnoty z rozmedzia -1 až 1. Z toho vyplýva, že aj pri malej zmene vstupnej hodnoty z tohto rozmedzia môžeme výrazne ovplyvniť výstup daného neurónu, čo demonštruje potrebu správnej škály vstupných hodnôt. [1] Spôsobov, ktorými môžeme normalizovať dátovú sadu existuje viacero, pričom sa tieto metódy hlavne líšia v rozsahu výstupných hodnôt (niektoré normalizačné metódy garantujú výsledný rozsah normalizovaných hodnôt napr. (0, 1)), schopnosti dobre spracovať odľahlé hodnoty či samotnou komplexnosťou metódy. Pre účely tejto práce bola zvolená metóda normalizácie pomenovaná **MaxAbsScaler**, ktorá je popísaná rovnicou:

$$X_i^s = \frac{X_i}{|max(X)|}$$

kde pod X_i rozumieme hodnotu dátovej sady, ktorú chceme normalizovať, čo môže byť v našom prípade uzatváracia cena danej akcie, X reprezentuje celú dátovú sadu, ktorú chceme v experimente normalizovať, pričom z nej potrebujeme určiť maximálnu absolútnu hodnotu a X_i^s je výsledná normalizovaná hodnota. Veľmi dôležité je však povedať, že pri normalizácii v experimentoch sa hodnota $|max(X)|$ určuje z trénovacej sady, lebo by sme v praxi túto hodnotu z validačnej či testovacej sady nemohli určiť, keďže by si vyžadovala poznať budúce hodnoty, a preto sa validačná a testovacia dátová sada normalizujú na základe škály vytvorenej z trénovacej dátovej sady. Praktické použitie tejto normalizačnej metódy môže vyzerať nasledovne.

Povedzme, že naša dátová sada (ds) pozostáva z nasledujúcich hodnôt:

$$ds = [-15,\ -3,\ -2,\ -1,\ 4,\ 7,\ 10,\ 14]$$

V tomto prípade platí, že: $|max(X)| = 15$ a normalizovaná dátová sada (nds) bude vyzerať nasledovne:

$$nds = [-1,\ -0.2,\ -0.1\overline{33},\ -0.0\overline{66},\ 0.2\overline{66},\ 0.4\overline{66},\ 0.\overline{66},\ 0.9\overline{33}]$$

Dôležité je teda poznamenať, že metóda **MaxAbsScaler**:

- môže nadobúdať kladné aj záporné hodnoty,
- má presne stanovený rozsah hodnôt $\langle -1\,,\ 1 \rangle$.

Metóda **MaxAbsScaler** bola vybraná pre účely tejto práce primárne z dôvodu, že sa počas nášho testovania preukázala ako metóda, ktorej použitie viedlo k vytvoreniu modelov, ktoré tvorili najkvalitnejšie predikcie. Ďalším faktorom, ktorý nás v tomto výbere utvrdil, je, že sa pri tvorbe modelov, ktoré používajú sekvenčné dáta ako napr. ceny akcií, ktoré nie sú z normálneho rozdelenia, odporúča pre dosiahnutie lepších výsledkov radšej vstupné dáta normalizovať ako štandarizovať[3]. Dôležité je ešte zmieniť, že autori referenčnej práce [12] sa rozhodli vstupné dáta štandarizovať tzv. metódou **Z-Score**, ktorú sme v prvom experimente používali aj my, ale kvôli nedostatočným predikčným výsledkom sme sa rozhodli pre normalizáciu metódou **MaxAbsScaler**, pričom tieto výsledky prezentujeme v podkapitole 5.2. Tento rozdiel v predspracovaní vstupných dát má za následok, že porovnávanie nami dosiahnutých výsledkov pomocou normalizovaných chýb z validačnej či testovacej dátovej sady s normalizovanými chybami od autorov referenčného článku nie je úplne korektné, a preto budeme porovnávať naše modely s našimi už dosiahnutými výsledkami a taktiež demonštrovať kvalitu vytvorených predikcií graficky.

2.3.2 Delenie dát do dátových sád

V štatistike vo všeobecnosti platí, že čím viac hodnôt dokážeme získať, tým lepšie dokážeme popísať charakter týchto dát. Mohlo by nám teda prísť adekvátne pri trénovaní predikčných modelov využiť všetky dostupné dáta, čo by zapríčinilo niekoľko problémov. V prvom rade, ak by sme použili všetký dostupné dáta na trénovanie modelu, tak by nám neostali žiadne nezávislé dáta, ktoré by sme mohli použiť na validáciu či testovanie vytvorených modelov. V druhom rade by sme takto mohli zapríčiniť pretrénovanie modelov (angl. overfitting), t.j. nami vytvorené modely by tvorili správne predikcie počas trénovania, ale nedokázali by správne predikovať nové dáta. [7]

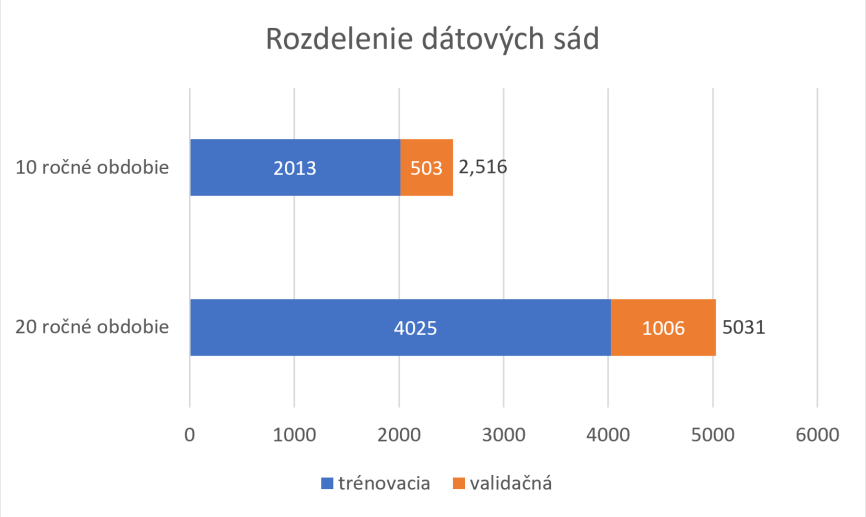
Tento fenomén teda evokuje vyhradenie určitých dát na overenie správnosti modelu. V praxi sa bežne rozdeľujú dáta na trénovaciu dátovú sadu a následne sa podľa potreby vyhradia dáta na testovanie, prípadne na validáciu a testovanie zároveň. V prípade, že sú vyhradené dáta na validáciu aj testovanie, tak by validačné dáta mali byť prvé dáta, ktoré model nevidel pri trénovaní a testovacie dáta by mali byť určené na finálne overenie správnosti modelu.

Vzhľadom na to, že sme si už definovali potrebu dvoch či troch nezávislých dátových sád, aby sme mohli korektne model natrénovať a vyhodnotiť jeho správnosť, nastáva otázka

[3] zdroj: Machine Learning Mastery (https://machinelearningmastery.com) How to Scale Data for Long Short-Term Memory Networks in Python

ako tieto sady správne vytvoriť. Najčastejšie sa k otázke delenia dát v praxi odborníci stavajú tak, že sa celá dátová sada rozdelí v pomere 80:20, pričom 80% dát je použitých na trénovanie jednotlivých modelov a 20% sa používa na samotné testovanie. V prípade, že je žiaduce vytvoriť validačnú dátovú sadu, sa tento pomer upravuje na 80:10:10, pričom 80% dát je stále použitých na trénovanie, nasledujúcich 10% slúži na validáciu a posledných 10% na finálne testovanie. Vhodnou otázkou je taktiež, prečo práve pomer 80 ku 20. Samotný dôkaz tohto faktu je síce nad rámec tejto práce, ale autori článku [7] poskytujú empirický dôkaz, ktorého záverom je, že na trénovanie by malo byť určených zhruba 80% dostupných dát. Moderným trendom v tejto oblasti pri obzvlášť veľkých dátových sadách (rádovo desaťtisíce, stotisíce hodnôt a viac) je zväčšovať tento pomer v prospech trénovacích dát, ako udáva Andrew Ng, známy priekopník v oblasti umelej inteligencii vo svojom online kurze Structuring Machine Learning Projects. [4]

Vzhľadom na počty dostupných dát uvedené v tabuľke 2.2 sme pre účely tejto práce vytvorili dve dátové sady, trénovaciu, tvorenú z približne 80% dostupných uzatváracích cien pre jednotlivé akcie a zvyšných 20% sme použili na validovanie presnosti predikcií. Tento postup bol použitý na obidve dátové obdobia (10-ročné, 20-ročné), ktoré v práci používame. Niektoré experimenty si vyžadovali použitie aj testovacieho obdobia, ktoré sme zvolili v rozsahu 1.1.2020 až 31.12.2022, pričom sa vždy používala len časť tohto obdobia, ktorá je bližšie špecifikovaná v návrhoch samotných experimentov a toto obdobie slúžilo na nezávislé overenie správnosti predikcií.



Obr. 2.4: Rozdelenie použitých dátových sád

Na obrázku 2.4 môžeme vidieť vizuálne demonštrované rozdelenie dátových sád použitých v tejto práci. Horná časť grafu znázorňuje 10-ročné obdobie, t.j. 1.1.2010 až 1.1.2020, ktoré pozostáva z 2516 uzatváracích cien pre každú jednu použitú akciu, pričom modrou farbou je znázornený počet dát použitých na trénovanie a oranžovou farbou je znázornený počet dát použitých na validáciu. V dolnej časti grafu je prezentované 20-ročné obdobie, t.j. 1.1.2000 až 1.1.2020, ktoré pozostáva z 5031 uzatváracích cien pre všetky akcie okrem akcie GOOG (tá má len 3868 uzatváracích cien, lebo prišla na trh až v roku 2004), pričom popis je analogický voči popisu 10-ročného obdobia v hornej časti.

[4] zdroj: Structuring Machine Learning Projects (https://coursera.org/) - Size of the Dev and Test Sets

2.3.3 Pripravenie dát pre modely

Poslednou úlohou v procese predspracovania dát je vytvorenie dátových sád, ktoré bude používať framework Tensorflow pri trénovaní modelov. Jednotlivé vstupné dáta musia dodržiavať predpísaný tvar dvojice `(vstupné hodnoty, správna výstupná hodnota)`. Tieto dvojice sú vytvárané z rozdelených dátových sád z minulého kroku a následne sú pomiešané, pričom tento krok je podstatný preto, aby pri trénovaní optimalizátor neurónovej siete neskončil na lokálnom optimálnom riešení, ale aby optimalizátor dokázal nájsť globálné optimálne riešenie.[5] V neposlednom rade sa z týchto dátových sád vytvoria menšie skupinky (angl. mini batch), ktorých výhodou je, že rovnako ako miešanie vstupných dát podporujú konvergovanie optimalizátora ku globálnemu optimálnemu riešeniu a taktiež zefektívňujú proces trénovania, čo má za následok kratšie trénovacie časy.[6] Výstupom tohto procesu sú dátové sady, ktoré sú pripravené na trénovanie a validáciu vo frameworku Tensorflow.

2.4 Validácia predikčných modelov

Jedným z hľavných zámerov tejto práce je tvorba modelov prostriedkami neuronových sietí, a preto je potrebné definovať spôsob, akým sa bude vyhodnocovať úspešnosť vytvorených modelov. Vzhľadom na to, že takto vytvorené modely je problematické validovať analyticky či teoreticky je nutné použitie chybových metrík. [21] Týchto metrík je niekoľko a medzi najpoužívanejšie sa radia:

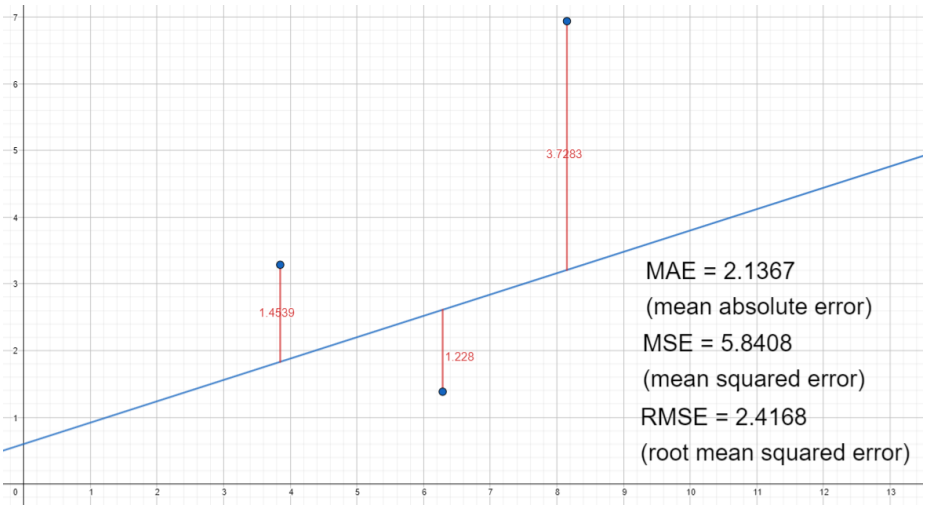
- **Mean Absolute Error**
  - Je definovaná ako priemerná kladná odchýlka predikovanej hodnoty voči referenčnej hodnote.
  - Vzorec výpočtu: $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i^{ref} - y_i^{pred}|$, kde n je počet hodnôt, y_i^{ref} je referenčná hodnota a y_i^{pred} je predikovaná hodnota referenčnej hodnoty y_i^{ref}.
  - Na vstupe je teda n dvojíc (referefenčná hodnota, predikovaná hodnota), z ktorých sa spraví rozdiel, následne sa všetky absolútne odchýlky sčítajú a spriemerujú, takže vo výsledku dostávame jednu hodnotu.
- **Mean Squared Error**
  - V princípe funguje podobne ako MAE, ale namiesto absolútnej hodnoty sa vzdialenosti (rozdiel predikcie od referenčnej hodnoty) umocní na druhú, takže vo výsledku pracuje s kladnými hodnotami.
  - Vzorec výpočtu: $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i^{ref} - y_i^{pred})^2$, kde n je počet hodnôt, y_i^{ref}, je referenčná hodnota a y_i^{pred} je predikovaná hodnota referenčnej hodnoty y_i^{ref}.
  - Napriek nepatrnej zmene vo výpočte sa táto zmena výrazne prejavuje na výsledkoch, keďže umocňovanie vzdialenosti na druhú sa veľmi prejavuje obzvlásť pri odľahlých hodnotách - môžeme povedať, že MSE je viac citlivé na odľahlé hodnoty.

[5] zdroj: DeepwizAI (https://www.deepwizai.com/) - Why Random Shuffling improves Generalizability of Neural Nets

[6] zdroj: Machine Learning Mastery (https://machinelearningmastery.com/) - A Gentle Introduction to Mini-Batch Gradient Descent and How to Configure Batch Size

– Rovnakým spôsobom ako v minulom bode sa môžeme pozrieť na rozdiel dvoch blízkych hodnôt, ktorý je menší ako 1. V takomto prípade by ich umocnenie produkovalo ešte menšie hodnoty, t.j. menej významené hodnoty, a preto hovoríme, že MSE je menej citlivé na blízke hodnoty.

- **Root Mean Square Error**
  - Vychádza z chybovej metriky MSE.
  - Dá sa vypočítať spôsobom $RMSE = \sqrt{MSE}$.
  - Podstata tejto chybovej metriky spočíva v tom, že pri výpočte MSE sa vzdialenosť umocňuje na druhú, čo by v praxi mohlo znamenať, že ak by sme predikovali ceny a rozdiel referenčnej a predikovanej ceny umocníme, tak by sme docielili, že výsledná veličina by bola pri predikcii v dolároch $US\$^2$, čo môže byť pri niektorých prácach nežiaduce. [20]



Obr. 2.5: Porovnanie metrík v praxi v nástroji GeoGebra[7]

Na obrázku 2.5 môžete vidieť porovnanie jednotlivých chybových metrík v praxi, pričom referenčné hodnoty sú udané tromi bodmi a predikované hodnoty sú dané bodmi, kde sa stretáva priamka $f(x) = 0.32x + 0.6$ s jednotlivými zvislicami z referenčných bodov. Zvislice nám na obrázku reprezentujú vzdialenosť týchto dvoch bodov. Dôležité z tejto vizuálicie je to, že vzhľadom na citlivosť chybovej metriky MSE na odľahlé hodnoty, môžeme aj po jej odmocnení vo forme metriky RMSE vidieť značný rozdiel voči metrike MAE.

Pre účely tejto práce bola použítá chybová metrika MSE, a to z dôvodu porovnávania, keďže autori referenčnej práce [12] udávajú svoje výsledky pomocou nej. Taktiež nám táto voľba pripadala rozumná, lebo ak uvažujem fakt, že MSE je citlivé na odľahlé hodnoty, tak v koncepte využitia MSE ako chybovej metriky pri trénovaní neuronových sietí, kde je cieľ túto hodnotu minimalizovať, sa vo výsledku zbavujeme modelov, ktoré produkujú vo svojich predikciach veľa odľahľých hodnôt. Čo sa tyká problému umocnenej predikčnej veličiny, ktorá pri MSE vzniká, tak odchýlky sú v tejto práci používané výlučne na porovnávacie účely, a preto pre nás problém netvorí.

[7] zdroj: GeoGebra, výtvor užívateľa zackakil, dostupné tu (https://www.geogebra.org/m/yybenxjm)

2.5 Trendy v oblasti predikcie finančných dát

Na záver kapitoly dva by sme radi zhrnuli moderné trendy v oblasti predikcie finančných dát, aby sme vytvorili prehľad relevatných techník v tejto oblasti a zároveň poskytli inšpiráciu na budúci výskum.

Článok **StockBot: Using LSTMs to Predict Stock Prices** bol pre našu prácu dôležitý, lebo podnietil jej vzniknutie. Autori tohto článku sa snažili o predikciu cien akcií pomocou rekurentných neuronových sietí na princípe long short-term memory (LSTM). V práci navrhujú niekoľko metód predikcie metódu **ground-truth**, ktorá predikuje cenu akcie jeden deň dopredu za použitia uzatváracích cien akcie z minulosti a metódu **updated-truth**, ktorá umožňuje dlhodobejšiu predikciu na základe tvorenia predikcií z už predikovaných uzatváracích cien. Tieto predikcie následne využívajú v skripte (robotovi) na automatické nakupovanie / predaj akcií, ktorý deklarujú ako veľmi úspešný. V práci taktiež navrhujú alternatívnú architektúru k vytvorenej rekurektnej neurónovej sieti v kombinácii s LSTM, a to enkóder-dekóder model, ktorý nedokázal prekonať svojho predchodcu. [12]

Ďalej prezentujeme článok **Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models**, ktorý porovná a vyhodnocuje dvanásť techník na predikciu cien akcií na základe metriky RMSE. Autori tejto práce vytvorili osem regresných modelov na základe techník strojového učenia: multivariate regression, multivariate adaptive regression spline (MARS), decision trees, Bagging regression, Boosting regression, Random forest (RF) regression, ANN regression, support vector machine (SVM) a štyri regresné modely vytvorené hlbokými neuronovými sieťami na princípe LSTM, ktoré predikujú päť cien budúceho týždňa a to konkrétne dva modely využívajúce: päť otváracích cien minulého týždňa (model 1.), desať otváracích cien posledných dvoch týždňov (model 2.) a dva modely typu enkóder dekóder, ktoré používajú dáta posledných dvoch týždňov, pričom tretí model používa len otvárcie ceny ako vstupné dáta, ale štvrtý model používa obchodné objemy, otvárciu, uzatváraciu, najnižšiu a najvyšiu cenu. Následne boli tieto modely porovnané, pričom najlepšie výsledky dosahovali hlboké neurónové siete, ktoré používali len otváracie ceny v poradí od najlepšieho: prvý model, druhý model, tretí model a štvrtý model. Modely vytvorené technikami strojového učenia dosahovali rádovo horšie výsledky, pričom najlepšie z nich boli: modely viacrozmernej a RF regresie, MARS regresie a Boosting regresie. [11]

Zaujímavým konceptom v predikcii finančných dát je využívanie hybridných modelov, t.j. tvorenie modelov postavených na niekoľkých menších modeloch. Vhodným článkom prezentujúcim tieto koncepty je **Forecasting Method of Stock Market Volatility in Time Series Data Based on Mixed Model of ARIMA and XGBoost**, v ktorom jeho autori používajú diskrétnú vlnovú dekompozíciu na rozdelenie aproximačnej a chybovej zložky vstupnej dátovej sady, pričom tieto zložky predikujú samostatne, aproximačná zložka je predikovaná modelom ARIMA a chybová zložka vylepšeným XGBoost modelom. Vytvorené predikcie sú následne spojené procesom vlnovej rekonštrukcie. V práci autori porovnávajú svoj návrh s modelmi ARIMA, XGB, GSXGB (vylepšený XGB), DWT-ARIMA-XGB a DWT-ARIMA-GSXGB (návrh autorov). Autori deklarujú najnižšiu chybovosť ich navrhovaného modelu oproti porovnávaným štyrom modelom. [22]

Poslednú alternatívnú metódu, ktorú chceme prezentovať je predikcia cien akcií pomocou konvolučných neuronových sietí (CNN). Vhodným článkom z tejto oblasti je **Research on Stock Price Prediction Method Based on Convolutional Neural Network**, v ktorom sa autori snažia predikovať ceny niekoľkých dní definovaných „posuvným oknom“ (angl. sliding window) použitím cien z predošlého posuvného okna. Zaujímavosťou tejto

práce je neštandardný prístup k vstupným dátam CNN, keďže autori používajú 1D dáta, konkrétne finančné časové rady, oproti štandardnému prístupu k vstupným dátam v CNN, a to 2D obrázkom. Autori následne testujú nimi navrhnutý model na troch akciách (BBL, CPALL, PTT) thajského akciového trhu a deklarujú vysokú presnosť modelu, čo oddôvodňujú tým, že ich model nie je závislý na predošlých informáciách, ale tvorí predikcie len na základe cien z minulého posuvného okna. [17]

Na základe prezentovaných populárnych metód sme sa rozhodli založiť náš princíp predikcie na dvoch regresných modeloch, a to na modeli s jednorozmerným vstupom, ktorý využíva rekurektné neurónové siete na princípe LSTM a predikuje uzatváraciu cenu nasledujúceho dňa na základe uzatváracích cien obdobia 60 dní a na modeli s totožnou architektúrou, ale s viacrozmerným dátovým vstupom.

Kapitola 3

Predikcia pomocou rekurektných neurónových sietí

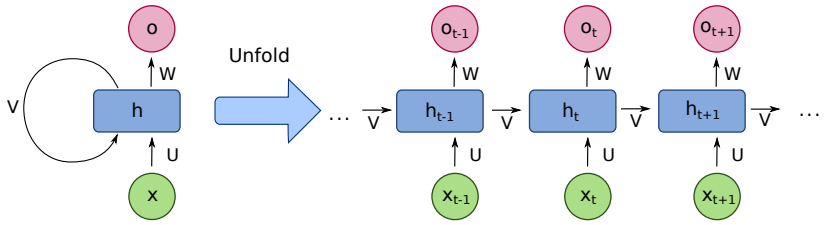
Ako prostriedok predikcie sme zvolili neurónové siete, hlavne kvôli ich schopnosti dobre aproximovať nelineárne funkcie, s ktorými sa pri akciách stretávame. Konkrétne sme sa zamerali na ich podtyp - rekurentné neurónové siete (RNN), lebo sú preferovaným typom neuronových sietí na prácu so sekvenčným dátami. V práci používame ich rozšírenie Long short-term memory, ktoré pomáha riešiť všeobecný problém rekurentných neurónových sietí - miznúci gradient a zároveň dokážu lepšie udržovať dlhodobé závislosti, čím sa zväčšuje pravdepodobnosť správnej predikcie modelu. V tejto kapitole sa teda budeme bližšie venovať princípu rekurektných neuronových sietí a ich rozšíreniu long short-term memory a následne popíšeme návrh našich modelov.

3.1 Rekurektné neurónové siete

Rekurektné neurónové siete boli prvýkrát predstavené v práci **Learning representations by back-propagating errors** [16] prof. D. Rumelharta pred skoro 40 rokmi a ich prínos v oblasti umelej inteligencie bol nesmierny. Tento spôsob návrhu neurónových sietí nám umožnil riešiť problémy ako rozpoznávanie reči, prekladanie a spracovanie textu či predikciu rôznych javov ako napríklad počasia. Uplatnenie tohto podtypu neuronových sietí je široké a všeobecne by sa dalo povedať, že vhodnými problémami pre tento typ sietí sú také, kde dokážeme vstupnú veličinu definovať ako postupnosť (sekvenciu) dát, pričom sú v nej závilosti, ktoré potrebujeme modelovať. Toto si môžeme demonštrovať nasledujúcim príkladom:

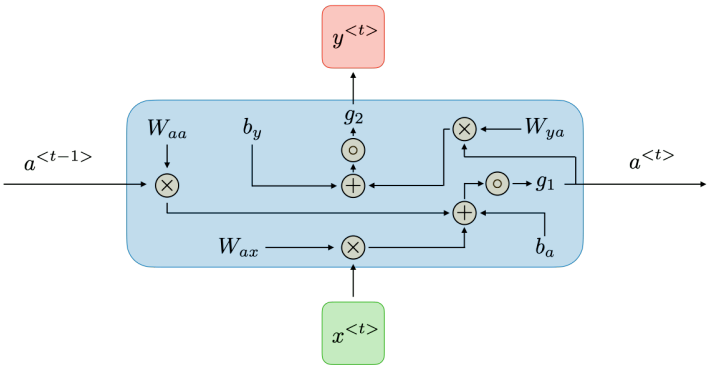
Celý svoj život som žil na Slovensku. Hovorím preto plynule _____ .

Ak by sme mali model, ktorý by spracovával tieto vety slovo po slove, tak by tento model možno detegoval, že niekto žil celý život na Slovensku, ale v druhej vete by mu k určeniu, o aký jazyk sa jedná, chýbal kontext. [10] Pre lepšie pochopenie ako by v prípade rekuretných neuronových sietí model mohol detegovať kontext, budeme demonštrovať ich princíp a štruktúru nasledujúcim obrázkom.



Obr. 3.1: Grafická vizualizácia štruktúry rekurentných neuronových sietí[1]

Na obrázku 3.1 môžeme pozorovať štruktúru rekurektných neuronových sietí. Vľavo je prezentovaný skrátený zápis, kde x reprezentuje vstupné dáta, ktoré vchádzajú do bloku h, ktorý spočíta „skrytý stav“ (angl. hidden state) a z ktorého následne vznikne výstupná hodnota reprezentovaná znakom o. Znak v reprezentuje aktivačné hodnoty predošlých vstupov, čo lepšie pochopíme, ak sa pozrieme na pravú časť obrázku. V prípade, že chce RNN zistiť výstupnú hodnotu pre x_t, tak na jej vypočítanie potrebuje dve zložky, momentálnú hodnotu vstupu x_t a predošlú aktivačnú hodnotu h_{t-1}. V prípade, že treba vypočítať výstupnú hodnotu prvého vstupu x_0, predošlá aktivačná hodnota sa nastavuje buď na vektor núl alebo náhodne. [2] Tento spôsob použitia predošlých aktivačných hodnôt pri počítaní aktuálnej aktivačnej hodnoty je obzvlášť dôležitý, lebo vytvára väzby a dodáva sieti nezbytný kontext, ktorý sme spomínali v minulom príklade a rovnako je dôležitý aj pre potreby našej práce, keďže sú naše vstupné dáta tvorené z usporiadanej postupnosti hodnôt, pričom predpokladáme existenciu závislostí predošlých hodnôt na budúcich.



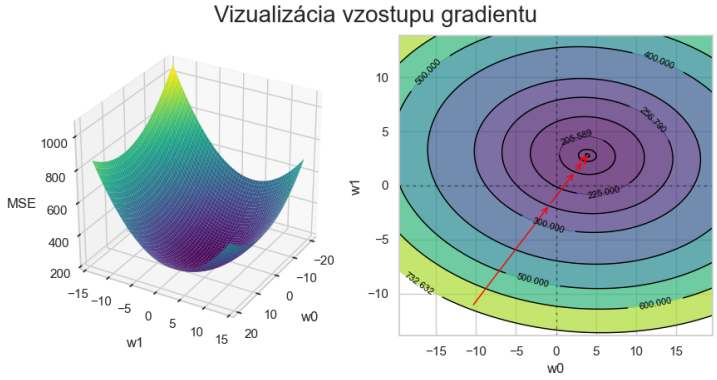
Obr. 3.2: Grafická vizualizácia výpočtu skrytého stavu neuronovej siete[3]

Obrázkom 3.2 by sme sa chceli bližšie venovať samotnému počítaniu skrytého stavu. K vypočítaniu výslednej hodnoty $y^{\langle t \rangle}$ je potrebný vstup v čase t, ktorý je prezentovaný hodnotou x_t a predošlá aktivačná hodnota $a^{\langle t-1 \rangle}$. Prvým krokom je vynásobiť dátový vstup x_t vektorom váh W_{ax} a aktivačný vstup $a^{\langle t-1 \rangle}$ vektorom váh W_{aa}. Následne sa tieto hodnoty sčítajú spolu s vektorom takzvaného biasu a na túto hodnotu sa uplatní aktivačná funkcia g_1, čím dostávame aktivačnú hodnotu $a^{\langle t \rangle}$. Tento proces by sme mohli porovnať s počítaním

[1] zdroj: https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg
[2] zdroj: Sequence Models - Recurrent Neural Network Model (https://www.coursera.org/)
[3] zdroj: https://stanford.edu/ shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

výstupu skrytej vrstvy obyčajnej neurónovej siete, samozrejme, bez pridávania aktivačnej zložky $a^{\langle t-1 \rangle}$. Na vypočítanie výstupnej hodnoty $y^{\langle t \rangle}$ musíme aktivačnú hodnotu $a^{\langle t \rangle}$ prenásobiť vektorom váh W_{ya}, pričítať k nemu bias b_y a uplatniť výstupnú aktivačnú funkciu g_2[4]. Ako môžete pozorovať k vypočítaniu výstupnej hodnoty je potrebné poznať tri vektory váh a dve hodnoty biasu, ktoré neúronová sieť zisťuje v procese trénovania. Voľba aktivačných funkcií je taktiež veľmi dôležitá, ako sme spomínali v kapitole 2.3.1 o normalizácii, tak nám tieto funkcie výrazne ovplyvňujú výsledné hodnoty, rýchlosť trénovania a taktiež uvádzajú potrebný faktor nelinearity do výpočtu.[5] Posledným dvom konceptom, ktorým sa v tejto podkapitole budeme venovať, je proces trénovania rekurektnej neurónovej siete a problémom miznúceho a explodujúceho gradientu. Vo všeobecnosti sa na trénovanie neurónových sietí používa algoritmus spätnej propagácie (angl. backpropagation) a rekurektné neurónové siete nie sú žiadnou výnimkou. Zmenou oproti všeobecným neurónovým sieťam je, že sa v RNN používa obdoba spätnej propagácie - **backpropagation through time**, ktorej funkciu popíšeme po definovaní pojmov **forward** a **back propagation**.

V predošlom odstavci sme opísali, ako funguje počítanie výstupnej hodnoty a skrytého stavu. Keby sme tento proces vykonali pre všetky vstupné vzorky, tak by sme tento proces anglicky pomenovali „**forward propagation**“ a získali by sme výstupnú hodnotu siete. Pre proces trénovania je podstatné, aby sme mohli takto vypočítaný výsledok porovnať so správným výsledkom. Toto porovnanie nám umožní účelová funkcia (angl. loss function), ktorá vypočíta rozdiel výstupnej hodnoty s referenčnou hodnotou podľa kritérií spomenutých v kapitole 2.4. Po zistení hodnoty účelovej funkcie treba spraviť spätný priechod sieťou (angl. **backpropagation**) pre každú vstupnú vzorku s cieľom minimalizovať hodnotu účelovej funkcie.[6] Tento proces realizujú optimalizátory, ktoré vykonávajú rôzne obdoby algoritmu vzostupu gradientu (angl. gradient descent, ukážka obr. 3.3) a na základe počítania derivácií upravujú váhy a bias, ktoré sa používajú na vypočítanie skrytých stavov, pričom táto spätná korekcia by mala zabezpečiť správne výsledky.



Obr. 3.3: Grafická vizualizácia vzostupu gradientu zobrazená vpravo na vrstevniciach účelovej funkcie, kde je demonštrované, že každou jednou iteráciou spätného priechodu (značené šípkami) sa snaží optimalizátor minimalizovať chybu, tzn. nájsť minimum účelovej funkcie. V ľavej časti obrázku je demonštrovaná 3D funkcia dvoch váh a ich chyby vyjadrenej v MSE, ktorú sa optimalizátor snaží minimalizovať.

[4] zdroj: https://stanford.edu/ shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

[5] zdroj: Machine Learning Mastery (https://machinelearningmastery.com/) - Using Activation Functions in Neural Networks

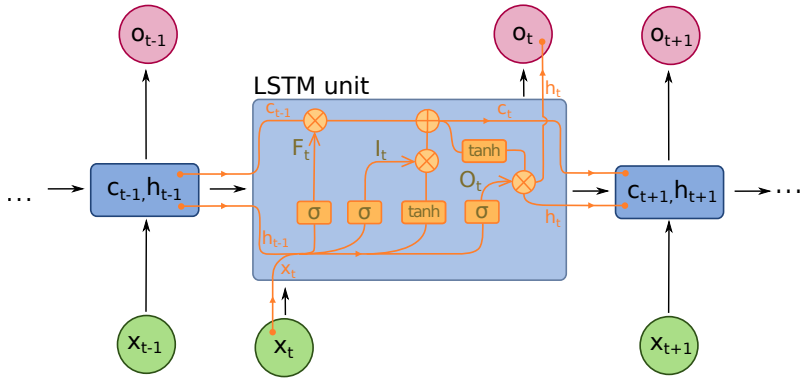
[6] zdroj: Programmathically (https://programmathically.com/) - Understanding Backpropagation With Gradient Descent

Hlavný rozdiel oproti klasickej verzii spätnej propagácie je, že v RNN sa musí vykonať spätná propagácia pre každú vstupnú vzorku vzhľadom nato, že finálny výsledok vznikol po postupnom spracovaní všetkých vzoriek, ale pri všeobecných NN sa tento proces vykonáva len raz pre finálny výsledok. Tento fakt robí RNN obzvlášť náročné na výpočetné prostriedky a voľba vhodného optimalizátora je teda kľúčová. V našej práci sme preto použili Adam optimalizátor, ktorý sa radí medzi jeden z najefektívnejších používaných optimalizátorov.[7]

Dôležité je spomenúť, že proces tvorenia modelov technológiou RNN má aj negatívne stránky. Ako sme v tejto podkapitole uviedli, trénovanie týchto modelov sprevádza vyššia výpočetná náročnosť a taktiež problém explodujúceho a miznúceho gradientu. Pri počítaní vzostupu gradientu je totiž potrebné rátať potenciálne väčšie množstvo derivácií chýb (v závilosti od komplexnosti navrhovanej siete), pričom sa používa reťazové pravidlo (angl. chain rule), ktoré má formu súčinov čiastočných derivácií. V prípade, že sú tieto častočné derivácie menšie ako jedna, tak môžu spôsobiť to, že výsledný súčin sa bude približovať k nule a trénované váhy sa nám vynulujú a teda „zmiznú“. [9] Opačný problém nastáva vtedy, ak by bol súčin týchto čiastočných derivácií väčší ako jedna, čo by viedlo k značnému zväčšeniu hodnôt trénovaných váh, teda k „explodujúcemu gradientu“ a vzostup gradientu diverguje. Tieto problémy sú však riešiteľné a rozšírenie Long short-term memory, ktorému sa budeme ďalej venovať, značne redukuje problém miznúceho gradientu.

3.2 Long short-term memory

Long short-term memory (LSTM) je rozšírenie, ktoré vychádza zo štruktúry RNN, zlepšuje schopnosť zachytávania dlhodobých závislostí a zároveň pomáha riešiť problém miznúceho gradientu. Pre lepšie pochopenie tohto rozšírenia je vhodné sa pozrieť, z čoho sa LSTM jednotka skladá a ako sa počíta jej výstupná hodnota a vnútorný stav.



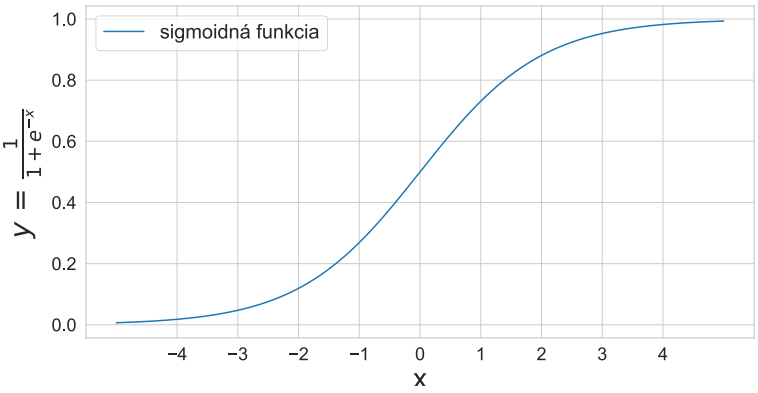
Obr. 3.4: Grafická vizualizácia fungovania LSTM jednotky[8]

Podobne ako samotné RNN, LSTM jednotka používa skrytý stav a vstupnú hodnotu pri tvorbe výstupnej hodnoty. Čím sa však líši od RNN, je používanie nového konceptu stavu jednotky (angl. cell state). Rozdiel medzi skrytým stavom a stavom jednotky je ten, že skrytý stav z RNN prezentuje len výstupnú aktivačnú hodnotu predošlého vstupu, ale

[7] zdroj: Machine Learning Mastery (https://machinelearningmastery.com/) - Gentle Introduction to the Adam Optimization Algorithm for Deep Learning

[8] zdroj: https://commons.wikimedia.org/wiki/File:Long_Short-Term_Memory.svg

vnútorný stav LSTM jednotky prezentuje hodnoty všetkých predošlých stavov. Ak by sme chceli LSTM jednotku rozdeliť na logické celky, tak by sme mohli povedať, že obsahuje tri definujúce komponenty, a to „zabúdaciu“ (angl. forget gate), vstupnú a výstupnú bránu. Prvá brána, ktorej sa budeme venovať, je zabúdacia, pričom jej hlavnou úlohou je filtrovať relevantné informácie zo stavu jednotky definovaného na obrázku 3.4 ako c_{t-1}. Využíva sa k tomu kombinácia vstupných dát x_t a predošlého skrytého stavu h_{t-1}, na ktorú sa použije sigmoidná funkcia, ktorej výstup je definovaný z rozsahu nula až jedna, pričom hodnota blízka nule znamená, že informácie zo stavu jednotky sú irelevantné a hodnota približujúca sa jednej znamená, že sú potrebné. Tento výstup je na obrázku 3.4 značený ako F_t a násobí sa so stavom jednotky, čo spôsobuje buď elimináciu, alebo zachovanie tohoto stavu. [3]



Obr. 3.5: Vizualizácia sigmoidnej funkcie

Po vyfiltrovaní potrebných informácií potrebuje LSTM jednotka modifikovať svoj stav jednotky. Najskôr sa zo vstupných dát vytvorí vektor, ktorý bude tento stav modifikovať, pričom získava kontext z predošlého skrytého stavu a následne je spracovaný funkciou tanh. Táto funkcia sa využíva preto, lebo chceme umožniť redukciu vplyvu stavu jednotky čo umožňujú záporné výstupné hodnoty tejto funkcie. Toto by nebolo možné, ak by sme použili sigmoidnú funkciu, lebo nedosahuje záporné hodnoty. Následne potrebuje LSTM jednotka zistiť, či si tento vektor potrebuje pamätať. Toto je realizované rovnakým spôsobom ako v zabúdacej bráne - pomocou sigmoidnej funkcie (značenej I_t), ktorá reprezentuje vstupnú bránu, pričom výstupná hodnota približujúca sa nule v násobení eliminuje vytvorený vektor a hodnota približujúca sa jednotke ho zachová. Výsledný vektor následne modifikuje stav LSTM jednotky tak, že sa s ním sčíta. Poslednou úlohou LSTM jednotky je vypočítať jej skrytý stav, ktorý je taktiež použitý ako výstup tejto jednotky. Výpočet skrytého stavu prebieha transformovaním celkového stavu jednotky funkciou tanh, ktorý sa filtruje vo výstupnej bráne O_t založenej na sigmoidnej funkcii, pričom proces filtrácie funguje rovnako ako pri predošlých bránach. Výstupom LSTM jednotky teda je nový skrytý stav h_t, upravený vnútorný stav c_t a výstupná hodnota totožná so skrytým stavom. [3]

Na záver podkapitoly je ešte vhodné poznamenať, akým spôsobom rozšírenie LSTM rieši problém miznúceho gradientu. Ako sme v predošlej podkapitole spomínali, problém miznúceho gradientu vzniká pri spätnom prechode v procese trénovania, ak sú hodnototy čiastočných derivácií pri reťazovom pravidle menšie ako jedna. Pri spätnom prechode RNN, ktoré používa LSTM jednotky sa však tieto derivácie počítajú kvôli rozdielom v ich štruktúre odlišne a hodnoty blízke nule tento problém nespôsobujú.

3.3 Metóda ground-truth

Táto metóda bola navrhnutá v práci StockBot: Using LSTMs to Predict Stock Prices [12], pričom funguje na princípe „posúvacích okien“, ktoré sú definované ako fixný počet dní, pričom každý je reprezentovaný jednou uzatváracou cenou, na základe ktorých sa predikuje uzatváracia cena nasledujúceho dňa. Tento model môžeme popísať podľa nasledujúceho vzorca:

$$y_{t+1} = F([x_{t-59}, x_{t-58}, \dots x_t])$$

kde pod F rozumieme funkciu modelu, x_t prezentuje dnešnú uzatváraciu cenu akcie, x_{t-59} je uzatváracia cena 59. predošlého obchodovaného dňa voči dňu t a y_{t+1} je predikovaná uzatváracia cena nasledujúceho dňa. Týmto spôsobom sa nasledovne robí krátkodobá predikcia z jedného dňa na druhý, vždy za použitia dát posledných 60 dní. Čo sa týka technickej realizácie tejto metódy, tak je postavená na technológii rekurentných neurónových sietí s rozšírením LST a preto bola vybraná ako referenčná voči experimentom tejto práce.

Tabuľka 3.1: Parametre neúronovej siete prezentované v referenčnej práci [12]

Parameter	**Hodnota**	**Parameter**	**Hodnota**
past_history	60	Batch size	64
LSTM layer units	20	Epochs	500
Stack depth	2	Steps per epoch	200
Train-test split	80%/20%	Validation steps	50

Týmto parametrom môžeme rozumieť nasledovne:

- **past_history** - počet uzatváracích cien potrebných na predikciu jedného dňa v budúcnosti,
- **LSTM layer units** - počet LSTM jednotiek v každej vrstve neurónovej siete,
- **Stack depth** - počet vrstiev neurónovej siete, konkrétne sa jedná o jednu LSTM vrstvu a jednu „spájajúcu - výstupnú“ (angl. dense) vrstvu,
- **Train-test split** - pomer, v ktorom autori delia trénovaciu a validačnú dátovu sadu,
- **Batch size** - počet jednej „mini dávky“ (mini batch) použitej pri trénovaní,
- **Epochs** - počet trénovacích iterácií,
- **Steps per epoch** - počet dávok z trénovacej dátovej sady, ktoré sa použijú vrámci jednej epochy,
- **Validation steps** - počet dávok z validačnej (testovacej) dátovej sady, ktoré musí validačný generátor spracovať počas trénovania jednej epochy.

Dátové sady použité autormi tejto práce boli taktiež získané z verejného portálu Yahoo Finance[9], pričom používali historické dáta z obdobia 2010 až 2020 rozdelené podľa vyššie spomenutého pomeru na trénovaciu a testovaciu dátovú sadu. Tieto dátové sady boli taktiež

[9] portál Yahoo Finance (https://finance.yahoo.com/)

normalizované pomocou metódy **MaxAbsScaler** uvedenou v kapitole 2.3.1. Na vyhodnotenie výsledkov bola zvolená účelová funkcia priemernej kvadratickej chyby (MSE). Ako optimalizátor neurónovej siete bol použitý algoritmus Adam a parameter miery učenia α nebol uvedený.

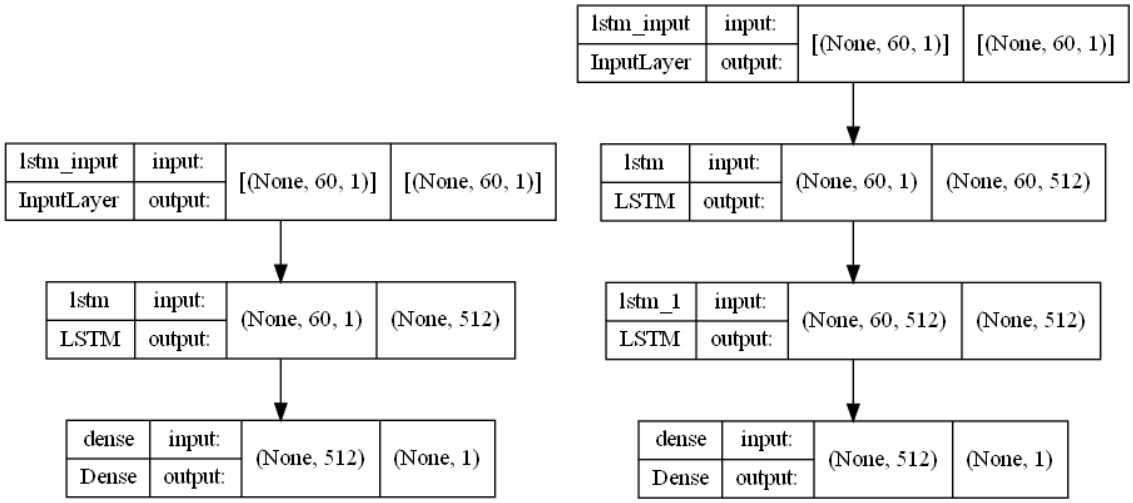
Tabuľka 3.2: Výsledky prezentované autormi referenčnej práce podľa zvolenej architektúry modelov z validačnej dátovej sady (MSE).

Počet vrstiev architektúry	**BRK-A**	**GOOG**	**MSFT**
jednovrstvová	0.0071	0.0044	0.0167
dvojvrstvová	0.0063	0.0080	0.022

Tieto výsledky boli vyhotovené predikciou validačného obdobia o dĺžke 200 dní (2018≈2019) predikovaných z jedného dňa na druhý pri použití parametrov uvedených v tabuľke 3.1 a budú použité na porovnanie výsledkov nami navrnutých modelov.

3.4 Jedno-premenný model

Jedná sa o alternatívny návrh modelu prezentovaného v predošlej podkapitole, pričom sme sa snažili o vylepšenie tohto modelu optimalizáciou zvolených parametrov. Navrhované zmeny parametrov budú oddôvodnené v kapitole 5, kde prezentujeme výsledky experimentov. Principiálne tento model funguje rovnako, takže na vstupe vyžaduje uzatváracie ceny akcií za posledných 60 dní a jeho výstupom je predikovaná uzatváracia cena nasledujúceho dňa. Tento model prezentujeme v dvoch rôznych architektúrach, a to jednovrstvovej (jedna vrstva LSTM jednotiek) a dvojvrstvovej.



Obr. 3.6: Vizualizácia jedno-premenného modelu, pričom vľavo je jednovrstvová architektúra a vpravo viacvrstvová

Pre lepšiu demonštráciu bola vytvorená grafická vizualizácia modelu, ktorú môžete pozorovať na obrázku 3.6. V ľavej časti je graf jedno-premenného jednovrstvového modelu, ktorý sa skladá zo vstupnej vrstvy, ktorá očakáva vstupné dáta v rovnakom formáte ako model ground-truth z predošlej kapitoly, a to $[(None, 60, 1)]$, čo znamená, že pre vytvorenie

predikcie N výstupných dní potrebujeme modelu dodať N (v zápise „None“) krát 60 vzoriek (60 po sebe idúcich dní), pri ktorej uvedieme 1 hodnotu, ktorá je v tomto prípade uzatváracia cenna daného dňa. Tie sú následne spracované v skrytej LSTM vrstve s 512 LSTM jednotkami a odovzdané poslednej spájajúcej vrstve, kde sa dopočíta výstupná hodnota, teda predikovaná uzatváracia cena nasledujúceho dňa. V pravej časti obrázku 3.6 môžete vidieť graf jedno-premenného viacvrstvového modelu, ktorého popis je analogický voči grafu v ľavej časti. Rozdiel medzi nimi je v počte skrytých LSTM vrstiev, pričom variácia vpravo používa dve skryté LSTM vrstvy po 512 jednotkách. Následne budeme prezentovať zvolené parametre týchto modelov:

Tabuľka 3.3: Nami zvolené parametre modelov

Parameter	**Hodnota**	**Parameter**	**Hodnota**
past_history	60	Batch size	30
LSTM layer units	512	Steps per epoch	200
Train-test split	80%/20%	Validation steps	50

Tabuľka 3.4: Parametre závislé od typu modelu

Model	**Epochy**	**Miera učenia α**
Jedno-premenný jedno-vrstvový	65	0.0004
Viac-premenný jedno-vrstvový	58	0.00037
Jedno-premenný dvoj-vrstvový	52	0.00031
Viac-vrstvový dvoj-vrstvový	56	0.00031

V tabuľke 3.3 prezentujeme výsledné všeobecné parametre modelov, ktoré sa od referenčných parametrov z literatúry, uvedených v tabuľke 3.1, líšia počtom LSTM jednotiek, ktorý bol zvýšený na 512, pričom túto zmenu odvôdňujeme v podkapitole 5.2.

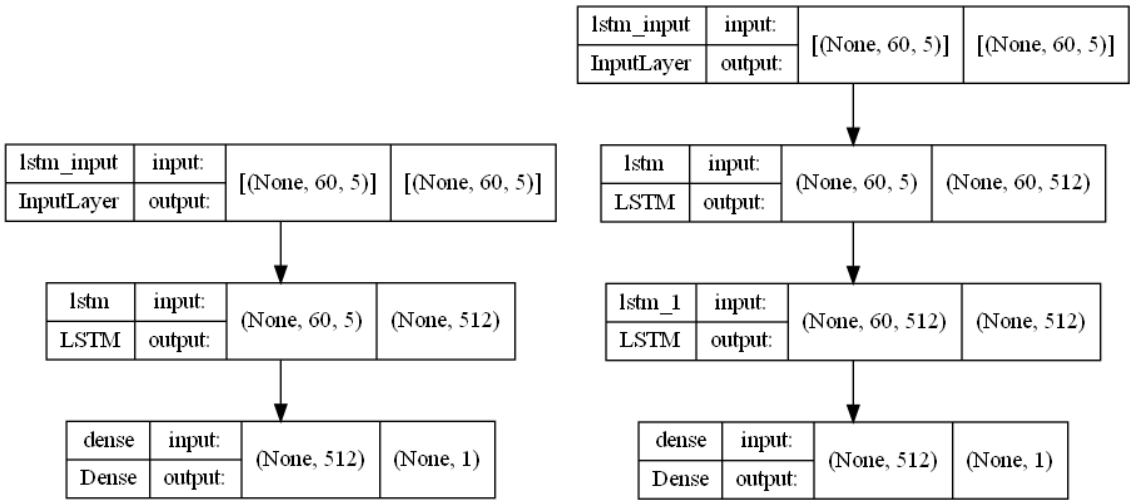
V tabuľke 3.4 uvádzame parametre, ktoré sú priamo závislé od typu modelu, keďže optimálny počet epoch a miery učenia sa líší v skoro všetkých prípadoch. Ako optimalizátor neurónovej siete bol použitý algoritmus Adam, pričom sme ako vyhodnocovaciu metriku zvolili strednú kvadratickú chybu. Na trénovanie, validáciu a testovanie boli použité dátové sady uvedené v kapitole 2 podľa potrieb konkrétnych experimentov. Celkový počet prezentovaných finálnych modelov tohto typu je šesť vzhľadom na to, že boli v práci použité tri referenčné akcie a sú prezentované dve rozdielne architektúry.

3.5 Viac-premenný model

Tento model vznikol ako vlastný prínos tejto práce a vychádza z idey - „Ako by sa zmenila úspešnosť jedno-premenného modelu, ak by nepracoval len s uzatváracími cenami, ale aj s rôznymi štatistickými či finančnými ukazovateľmi.“ Realizácia tohto modelu je o niečo komplikovanejšia oproti jedno-premennému modelu, lebo okrem toho, že berie na vstup uzatváracie ceny posledných 60 dní, tak používa denný obchodovaný objem a taktiež sa pre každý jeden deň vytvorí nové časové okno 30 dní minulých dní k danému dňu a zistia sa štatistické ukazovatele - minimum, maximum a štandardná odchýlka. Tento model by sme teda mohli definovať takto:

$$y_{t+1} = F\left(\begin{bmatrix} x_{t-59}, & v_{t-59}, & max((x_{t-89}, x_{t-59}\rangle), & min((x_{t-89}, x_{t-59}\rangle), & std((x_{t-89}, x_{t-59}\rangle) \\ x_{t-58}, & v_{t-58}, & max((x_{t-88}, x_{t-58}\rangle), & min((x_{t-88}, x_{t-58}\rangle), & std((x_{t-88}, x_{t-58}\rangle) \\ \vdots & & & & \\ x_t, & v_t, & max((x_{t-30}, x_t\rangle), & min((x_{t-30}, x_t\rangle), & std((x_{t-30}, x_t\rangle) \end{bmatrix}\right)$$

Pričom pod F rozumieme funkciu modelu, x_t prezentuje dnešnú uzatváraciu cenu, pod v_t rozumieme dnešný obchodovaný objem, pod x_{t-59} rozumieme uzatváraciu cenu 59 predošlého dňa, pod $max((x_{t-89}, x_{t-59}\rangle)$ rozumieme najvyšiu uzatváraciu cenu z časového rozmedzia posledných 59 až 89, pričom posledný deň neberieme do úvahy. Funkcie *min* a *std* pracujú analogicky voči popísanej funkcii *max*. Všetky parametre modelov spomínaných v tabuľke 3.3 vrátane parametrov uvedených v tabuľke 3.4 sú totožné s parametrami jedno-premenných modelov. Taktiež tento model prezentujeme v dvoch architektúrach, a to jednovrstvovej a dvojvrstvovej.



Obr. 3.7: Vizualizácia viac-premenného modelu, pričom vľavo je jednovrstvová architektúra a vpravo viacvrstvová

Na obrázku 3.7 vľavo môžeme vidieť, že jednovrstvový model sa skladá z rovnakých častí ako jedno-premenný model, konkrétne zo vstupnej, skrytej LSTM a spájajúcej vrstvy. Rozdielne sú v tom, že viac-premenný model prijíma vstupné dáta vo formáte $[(None, 60, 5)]$, čo znamená, že ak chceme predikovať N výstupných dní, tak musíme modelu dodať N (prezentovaných vo formáte ako „None“) krát 60 vzoriek (60 po sebe idúcich dní) krát 5 vstupných premenných modelu pre každú jednu vzorku. Nasledovaný je skrytou vrstvou 512 LSTM jednotiek, ktorá využíva na predikciu všetky vstupné premenné a výstup z tejto vrstvy sa spája vo výstupnej vrstve. V pravej časti sa nachádza viac-premenný viacvrstvový model, ktorého popis je analogický voči modelu v ľavej časti, ale využíva dve skryté LSTM vrstvy po 512 LSTM jednotiek. Rovnako ako pri jedno-premennom modeli v tejto práci prezentujeme šesť variácií viac-premenného modelu, dve rôzne architektúry pre tri rôzne akcie.

Kapitola 4

Návrh experimentov

V tejto kapitole uvedieme prehľad experimentov, ktorému sme sa v rámci tejto práce venovali, počnúc implementačnou stránkou získavania finančných dát, replikáciou referenčného či návrhu vlastného modelu, vytvorenými analýzami a optimalizáciami modelov. V neposlednom rade sa budeme venovať procesu štandarizácie týchto experimentov do jednotného postupu, ktorý bude použitý pri realizácii všetkých experimentov. Konkrétne výsledky experimentov budeme následne prezentovať v nasledujúcej kapitole.

4.1 Získanie finačných dát

Predtým, ako sme mohli začať so samotným návrhom a realizáciou experimentov, sme museli získať potrebné finančné dáta a vzhľadom na to, že všeobecný popis a charakteristika dátových sád už bola uvedená v kapitole 2.2, teraz sa môžeme zamerať na implementačnú časť získavania dát. Ako prvé je dôležité podotknúť, že naša práca nevyužíva jednu centrálnu dátovú sadu, ale dáta sú agregované podľa potrieb jednotlivých experimentov, odlišujúce sa počtom obchodných dní a sú ukladané do samostatných csv súborov. Tieto dátove sady teda delíme rovnako, ako vidno v tabuľke 2.2, kde dátovú sadu definuje rozmedzie dátumov [1] a identifikátor patričnej akcie. Vzhľadom na individuálne požiadavky na dáta pre jednotlivé modely bola napísaná trieda `Stock`, ktorej rozhranie je definované nasledovne: `Stock(name, start_date, end_date, force_fetch_stock)`, pričom `name` je identifikátor akcie, `start_date` a `end_date` reprezentujú obdobie, pre ktoré je potrebné získať dáta a `force_fetch_stock` je voliteľný parameter, ktorý zabezpečí, že ak by sa na disku už nachádzala potrebná dátová sada, tak bude napriek tomu znovu stiahnutá a uložené dáta sa prepíšu. Používanie tejto triedy je jednoduché, keďže po dodaní spomínaných parametrov rozhrania trieda zavolá svoju metódu `fetch()`, ktorá získa požadované dáta pomocou balíčku `yfinance`, ktorý sťahuje tieto dáta priamo z portálu Yahoo Finance. Získané dáta sú následne uložené do atribútu `df` triedy `Stock` vo formáte dátového rámcu (angl. dataframe) z knižnice `pandas`. Pri implementácii tejto triedy sme mysleli aj na efektivitu, a preto ak sa získané dáta na disku ešte nenachádzajú, tak sa uložia na lokálny disk do zložky `stocks` v koreňovom adresári a v prípade, že sa vyžaduje rovnaké dátové obdobie, tak sa dátová sada rovno načíta z disku.

[1] Niektoré experimenty môžu potrebovať aj testovaciu sadu, ale môžeme povedať, že sa približne delia na 10 a 20-ročné obdobia.

4.2 Experiment 1 - Tvorba jedno-premenného modelu

V úvodnom experimente tejto práce sme sa zamerali na vytvorenie jedno-premenného modelu, ktorý mal slúžiť ako odrazový mostík do problematiky predikovania cien za použitia neurónových sietí. Keďže sme v tej dobe ešte nepoznali optimálne parametre pre tento model, tak sme použili viacero parametrov z referenčnej práce [12] uvedené v tabuľke 3.1.

Avšak existovali aj parametre, ktoré sa líšili od začiatku implementácie a jedným z nich bol počet epoch modelu. V referenčných parametroch je uvedené, že autori práce použili 500 trénovacích epoch, ale náš prvotný model nedosahoval pri takomto veľkom počet trénovacích epoch dobré výsledky, a preto sme ho zredukovali na 50. Ďalší podstatný parameter, miera učenia α tiež nebola definovaná autormi práce, a preto bola ponechaná na predvolenú hodnotu optimalizátora Adam vo frameworku Tensorflow.

4.3 Experiment 2 - Vylepšenie jedno-premenného modelu

Po úspešnej implementácie jedno-premenného modelu v prvom experimente, kde sme dosiahli veľmi podobné grafy, ako uvádzajú autori referenčného článku [12], sme sa rozhodli overiť správanie vytvoreného modelu na dátovej sade pozostávajúcej z nezávislého časového obdobia. Vytvorili sme preto novú testovaciu dátovú sadu pozostávajúcu z 200 dní po skončení validačných dát (rok 2020), pričom sme zistili rozdielnu kvalitu predikcií medzi modelmi referenčných akcií. Následne sme analyzovali vytvorené predikcie a hľadali príčinu neuspokojúcich výsledkov. Tento problém sa nám podarilo vyriešiť pomocou experimentovania s rôznymi počtami LSTM jednotiek modelov a otestovaním rôznych metód normalizácie.

4.4 Experiment 3 - Tvorba viac-premenného modelu

Jednou z ďalších ideí, ktoré sme sa rozhodli v rámci tejto práce preskúmať, bol námet na vytvorenie modelu, ktorý bude využívať niekoľko vstupných premenných na predikciu nasledujúceho dňa a následné vyhodnotenie, či nastáva zlepšenie predikcií oproti jednopremennému modelu. Dôležitou podmienkou týchto vstupných parametrov bolo aj to, že všetky museli byť dopočítateľné z uzatváracej ceny akcie pre prípad, že by sme sa v budúcnosti rozhodli, že chceme, aby model dokázal predikovať cenu nie len nasledujúceho dňa, ale povedzme N dní do budúcnosti. Takáto predikcia by mohla fungovať tak, že by sa k dnešnému dňu vytvorila predikcia na zajtrašok, z predikovanej hodnoty zajtrajška by sa vypočítali všetky vstupné parametre a znova by sa vykonala predikcia. Každopádne sme tento prístup kvôli zlým výsledkom zavrhli a predkladáme ho na preskúmanie v budúcej práci.

4.5 Experiment 4 - Analýza predikcií modelov

V tomto štádiu práce sme mali vytvorené dva typy predikčných modelov, ktoré generovali na prvý pohľad pekné grafy na rovnakom časovom období, aké zvolili autori referenčného článku, t.j. 200 dní. Avšak vzhľadom na princíp predikcie týchto modelov, a to konkrétne na fakt, že vždy predikujeme hodnotu budúceho dňa, sme sa rozhodli pozrieť bližšie na vývoj predikcií v rámci jedného mesiaca. Vo výsledkoch prezentujeme pohľad na predikcie relatívne nových dát z roku 2021, konkrétne za mesiac december. V rámci tohto expe-

rimentu bola vykonaná analýza alternatívneho vyhodnocovania výsledkov predikcií, a to redukovaním problému predikcie cien na problém binárnej klasifikácie zmeny trendu akcie nasledujúceho dňa, ktorá vyprodukovala zaujímavé výsledky aj napriek tomu, že modely neboli priamo trénované na binárnu klasifikáciu zmeny trendu ceny akcie. V tomto experimente taktiež prezentujeme porovnanie validačných a testovacích odchýliek 60 modelov rozdelených podľa typu (jedno a viac-premenný) a prislúchajúcej akcie modelu.

4.6 Experiment 5 - Optimalizácia navrhnutých modelov

Záverečný experiment tejto práce spočíval v optimalizácii výsledkov jednotlivých modelov, a to konkrétne vyhodnotením odchýliek jednotlivých modelov pri rôznych trénovacích parametroch. Veľmi dôležitým bodom tohto experimentu bolo hľadanie optimálneho parametru miery učenia α a trénovacích epoch, ktorý by minimalizoval odchýlku na validačnej dátovej sade. Preto sme pre účely tohto experimentu natrénovali 72 modelov rozdelených do štyroch hlavných konfigurácií podľa typu modelu (jedno a viac-premenné modeli) a podľa počtu LSTM vrstiev (jedno a viac-vrstvové modeli), z čoho nám vzniklo 18 modelov v rámci jednej skupiny. Tieto skupiny obsahovali modeli všetkých troch referenčných akcií, pre ktoré sme vykonali tri nezávislé trénovania a keďže sme chceli v rámci tohto experimentu testovať aj vplyv dĺžky trénovacích období, tak sme použili dve dátové sady rôznej dĺžky - 10 a 20 ročné dátové sady definované v kapitole 2.2 z čoho nám vzniklo 18 modelov v rámci jednej skupiny. Z týchto výsledkov sme následne určili parameter miery učenia α a vhodný počet trénovacích epoch, pričom sme chceli vykonať záverečné vyhodnotenie úspešnosti s týmito novými parametrami, a preto sme na základe prezentovaných štyroch konfigurácií natrénovali 480 rôznych modelov rovnakým spôsobom ako pri hľadaní optimálnej miery učenia a trénovacích epoch, len sme modely jednotlivých referenčných akcií trénovali desaťkrát a nie trikrát. Taktiež v tomto experimente prezentuje voliteľné vylepšenie vo forme návratu k najlepšej epoche, ktoré zabezpečí, že po dokončení trénovania si model uchová váhy z epochy, v ktorej dosiahol najnižšiu validačnú odchýlku. Získané výsledky boli následne použité na porovnanie navrhovaných konfigurácií predikčných modelov pomocou ich validačných odchýliek na základe rôznych kritérií ako úspešnosť modelov podľa referenčnej akcie, podľa konfigurácie modelu či podľa použitej dátovej sady.

4.7 Štandarizácia procesu experimentovania

Vzhľadom na podobnosť činností, ktoré bolo treba vykonať v jednotlivých experimentoch, sme sa rozhodli tento proces štandarizovať nasledovným všeobecným postupom:

- **Definícia parametrov** - pri každom experimente sa musia definovať parametre, ktoré chceme otestovať, dôležitá je obzvlášť dĺžka trénovacej / validačnej dátovej sady, či sa používa testovacia sada, počet trénovacích epoch, typ modelu, atď..
- **Načítanie dát** - načítanie dát prebieha pomocou triedy `Stock`, ktorej stačí poskytnúť patričné parametre definované v podkapitole 4.1 a môžeme s dátami okamžite pracovať.
- **Normalizácia dát** - pre dosiahnutie dobrých výsledkov sa dáta musia najskôr normalizovať postupom popísaným v podkapitole 2.3.1.

- **Príprava dátových sád** - dáta je následne treba pripraviť do správneho formátu definovaného typom modelu nasledovne:

  jedno-premenný model - dáta musia dodržiavať formát Nx60x1, pričom N je počet dní, ktoré chceme predikovať a pre každý tento deň je potrebné poskytnúť modelu 60 predošlých uzatváracích cien,

  viac-premenný model - dáta musia dodržiavať formát Nx60x5, pričom N je počet dní, ktoré chceme predikovať a pre každý z poskytnutých 60 dní je potrebné vyrátať všetky vstupné premenné daného modelu.

- **Trénovanie modelu** - následne musí prebehnúť trénovanie modelu, pričom validácia prebieha automaticky vo frameworku Tensorflow. Alternatíva tohto kroku je načítanie už existujúceho modelu.

- **Variabilná časť experimentov** - táto časť sa môže meniť pri každom experimente podľa jeho potrieb, obsahovať môže napr. vykreslenie grafov za požadované obdobia, vyhodnotenie testovaných parametrov, zápis výsledkov do .csv súboru, atď..

Štandarizácia tohto procesu mala za následok zlepšenie konzistentnosti kódu a uľahčenie realizácie samotných experimentov, keďže boli všetky podprocesy experimentu dopredu definované. V neposlednom rade by sme chceli priblížiť formu, ktorou budeme prezentovať výsledky v jednotlivých experimentoch. Všetky výsledky, ktoré v tejto práci predkladáme budú vo forme grafov či tabuliek, ktoré budú porovnávané na základe dosiahnutých odchýlok modelu a preto chceme objasniť nasledujúce pojmy.

- **trénovacia odchýlka** - udáva chybu, ktorú model dosiahol na trénovacom období,

- **validačná odchýlka** - udáva chybu, ktorú model dosiahol na validačnom období, pričom ju používame na porovnávanie najčastejšie,

- **testovacia odchýlka** - udáva chybu, ktorú model dosiahol na testovacom období a je udaná len vtedy, ak sa v rámci experimentu porovnáva validačné obdobie s testovacím.

Všetky vyššie uvedené odchýlky budú udané pomocou metriky MSE.

Kapitola 5

Výsledky experimentov

V tejto kapitole sa budeme venovať realizácii experimentov z minulej kapitoly, pričom popíšeme všetky dosiahnuté výsledky práce, ktoré budú prevažne demonštrované grafmi predikcií jednotlivých akcií. Výsledky následne vyhodnotíme a budeme porovnávať na základe dosiahnutých odchýlok modelov pomocou metriky MSE. Celkové zhodnotenie výsledkov budeme prezentovať v poslednej kapitole spolu s námetmi budúcej práce.

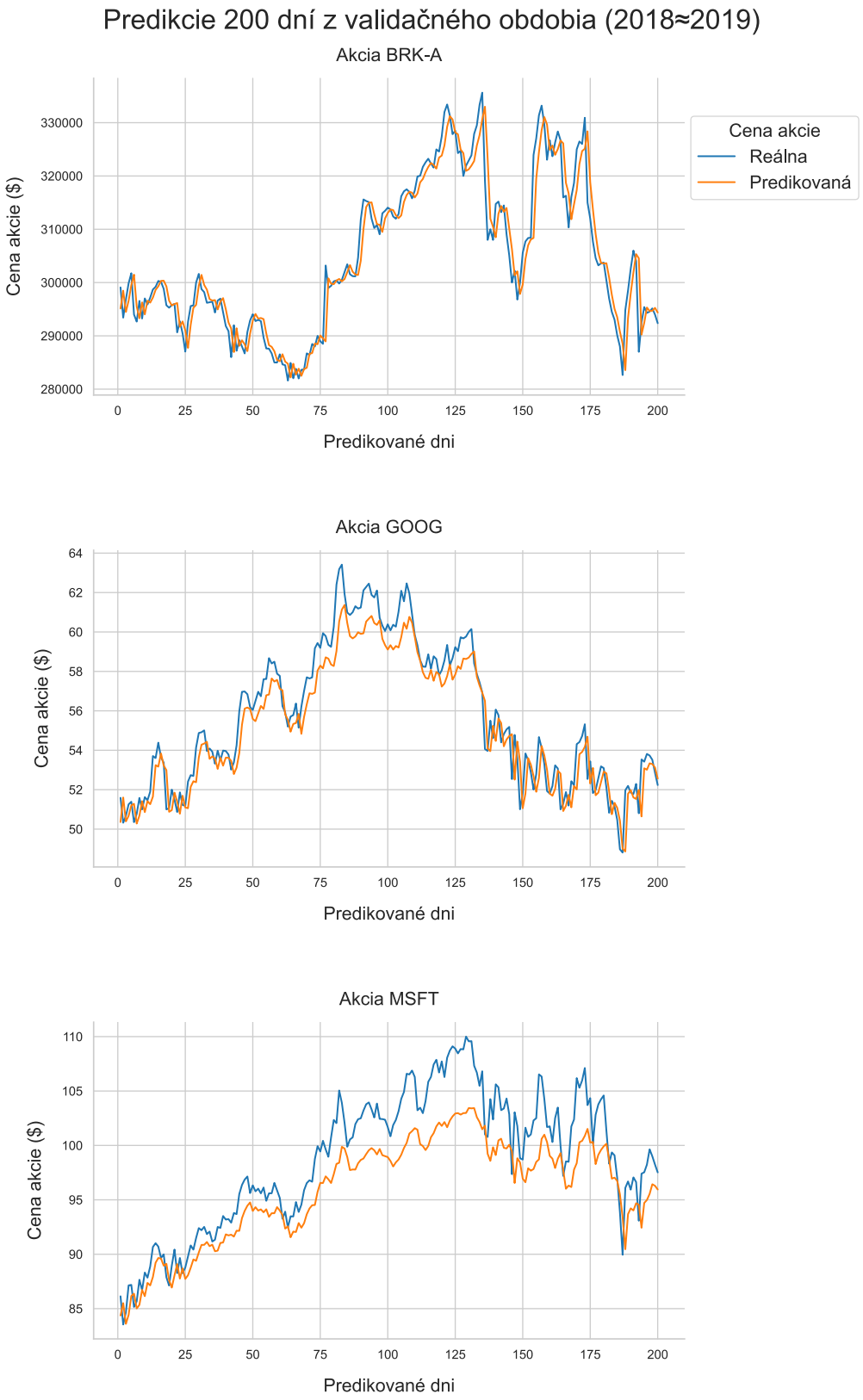
5.1 Experiment 1 - Tvorba jedno-premenného modelu

Podstatou prvého experimentu bolo vytvoriť základ tejto práce, na ktorom sa bude ďalej stavať a vytvoriť prvú verziu jedno-premenného modelu definovaného podľa špecifikácie v podkapitole 3.4. V tejto fáze už bola vytvorená potrebná abstrakcia na získavanie finančných dát a definovaný všeobecný postup experimentovania, ale na vytvorenie jedno-premenného modelu bolo potrebné najskôr definovať jeho architektúru vo frameworku Tensorflow a pripraviť dáta do potrebného formátu patričných dátových sád.

Vzhľadom na to, že sme vedeli, že rozličné experimenty budú mať rozličné požiadavky na architektúru modelov, tak sme sa rozhodli vytvoriť konfigurovateľnú nadstavbu nad jednotlivými metódami frameworku Tensorflow, ktorú sme definovali v triede `Model`. Toto nám značne uľahčilo proces experimentovania s rozličnými architektúrami, lebo všetky potrebné parametre ako napr. počet vrstiev, epoch, LSTM jednotiek, atď. boli nastaviteľné ako parametre tejto triedy. Taktiež sme mohli vytvorené modely jednoducho uložiť vo forme H5 súboru a následne ich neskôr načítať a pracovať s nimi.

Následne sme potrebovali transformovať finančné dáta na trénovaciu a validačnú dátovú sadu, čo podnietilo vznik metódy `prepare_dataset_single_model`, ktorá tvorí zo vstupných dát dvojice (vstup, správna hodnota) vo forme pohyblivých okien spomínaných v podkapitole 3.4 a je používaná vo všetkých nasledujúcich experimentoch, ktoré pracujú s jedno-premenným modelom. Tieto dvojice sú ďalej transformované do dátových sád frameworku Tensorflow spôsobom definovaným v podpkapitole 2.3.3.

Po dokončení potrebnej abstrakcie sme začali s implementáciou jedno-premenného modelu. Na trénovanie a validáciu bola použitá štandardná 10-ročná dátová sada definovaná v podkapitole 2.3.2 a tieto dáta boli normalizované spôsobom definovaným v podkapitole 2.3.1. Následne sme tento model trénovali s parametrami definovanými v tabuľke 3.3, ale s originálnym počtom LSTM jednotiek (20) a epoch (50). Takto sa nám podarilo vytvoriť prvý jednovrstvový jedno-premenný model, ktorý sme otestovali tak, že sme vytvorili graf predikcií validačného obdobia, ktorý sme porovnali s grafmi referenčnej práce [12].



Obr. 5.1: Výsledky predikcie 200 dní z validačného obdobia

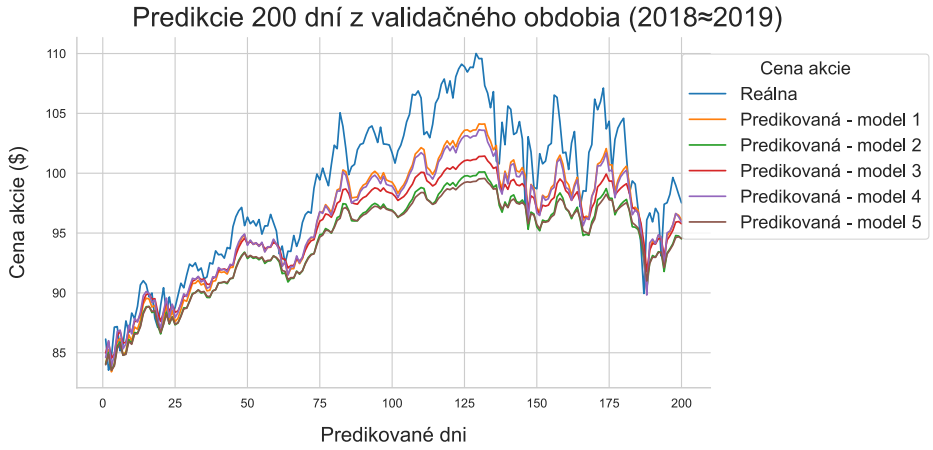
Na obrázku 5.1 sú prezentované výsledky troch predikčných modelov, ktoré predikovali 200 obchodných dní z validačného obdobia, konkrétne bolo predikované obdobie 1.1.2018 až 15.1.2019, pričom sa vždy predikovala uzatváracia cena nasledujúceho dňa na základe predošlých uzatváracích cien. Každý jeden graf je vytvorený na základe predikcií jedného samostatného modelu, pričom oranžová krivka je predikovaná cena akcie a modrá krivka je reálna cena akcie použitá na referenciu. Obidve uvedené krivky prezentujú cenu v dolároch.

V prípade, že sa zameriame na konkrétne modely, tak môžeme pozorovať, že model vytvorený pre akciu BRK-A opisuje referenčnú krivku a dosahuje najnižšiu priemernú kvadratickú odchýlku 0.0045[1]. Modely vytvorené pre akcie GOOG a MSFT dosahujú horšie priemerné odchýlky oproti modelu pre akciu BRK-A, a to konkrétne GOOG - 0.0165 MSE a MSFT - 0.835 MSE. Rádová odlišnosť odchýliek pri modeli akcie MSFT je pravdepodobne spôsobená neschopnosťou modelu presne zachytiť väčšie výkyvy ceny ako napr. nárast ceny v stom dni predikcie či prudký pád ceny zhruba v 185. dni predikcie. Vzhľadom na to, že rovnaký fakt uvádzajú taktiež autori referenčného článku a výsledky vytvorených predikčných modelov sú zhruba podobné, sa v ďalšom experimente pozrieme na to, ako by sa dali tieto modely vylepšiť.

5.2 Experiment 2 - Vylepšenie jedno-premenného modelu

V druhom experimente sme chceli bližšie preskúmať kvalitu predikcií týchto modelov na dátach, ktoré sme ešte nepoužívali a vzhľadom na to, že štandardná 10-ročná dátová sada končí pred rokom 2020, tak sme sa rozhodli použiť ako testovaciu sadu dáta z obdobia 1.1.2020 až 12.1.2021, ktoré reprezentujú 200 obchodných dní, čo uľahčuje porovnanie s výsledkami predikcií z minulého experimentu. Voľba tohto testovacieho obdobia je obzvlášť zaujímavá, lebo môžeme pozorovať výsledky predikcií modelov v čase pandémie COVID19, kde bol zaznamenaný značný nárast trhovej volatility [5] a sledovať, či zvýšená volatilita akcií bude mať výrazný vplyv na kvalitu ich predikcií.

Modely boli pre tento experiment trénované štandarizovaným procesom za použitia rovnakých parametrov ako v predošlom experimente, pričom kvalita predikcií týchto modelov by mala byť porovnateľná s kvalitou predikcií modelov z minulého experimentu. Na nasledujúcom grafe budeme demonštrovať možné rozdiely v kvalite počas piatich trénovaní.



Obr. 5.2: Na grafe môžeme pozorovať malé rozdiely v kvalite medzi modelmi.

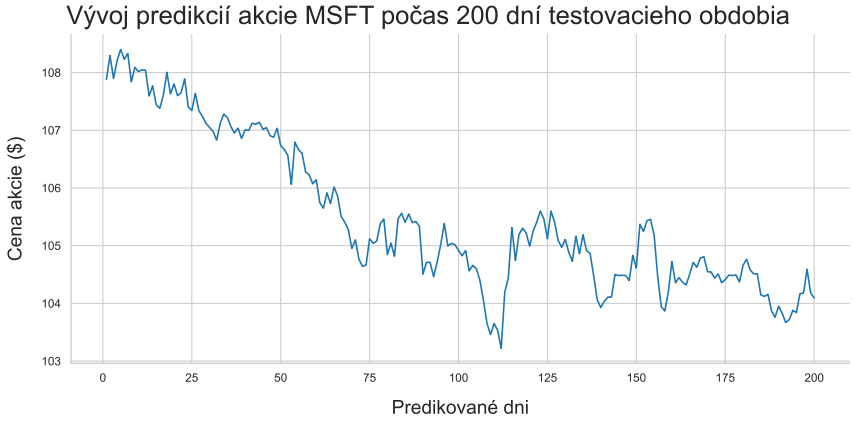
[1]Všetky uvedené odchýlky budú počítané z normalizovaných dát.

Obr. 5.3: Výsledky predikcie 200 dní z testovacieho obdobia

Na obrázku 5.2 môžeme pozorovať menšie rozdiely medzi predikciami jednotlivých modelov, z čoho môžeme usúdiť, že kvalita týchto predikcií je približne rovnaká. Výsledkom

tohto pozorovania sa budeme riadiť aj v ďalších experimentoch, a preto sa nebudeme snažiť vytvoriť jeden perfektný model, ktorý dosiahne najnižšiu validačnú odchýlku, ale budeme sa zameriavať na zistenie optimalných parametrov modelu, ktoré budú produkovať najlepšie všeobecné výsledky.

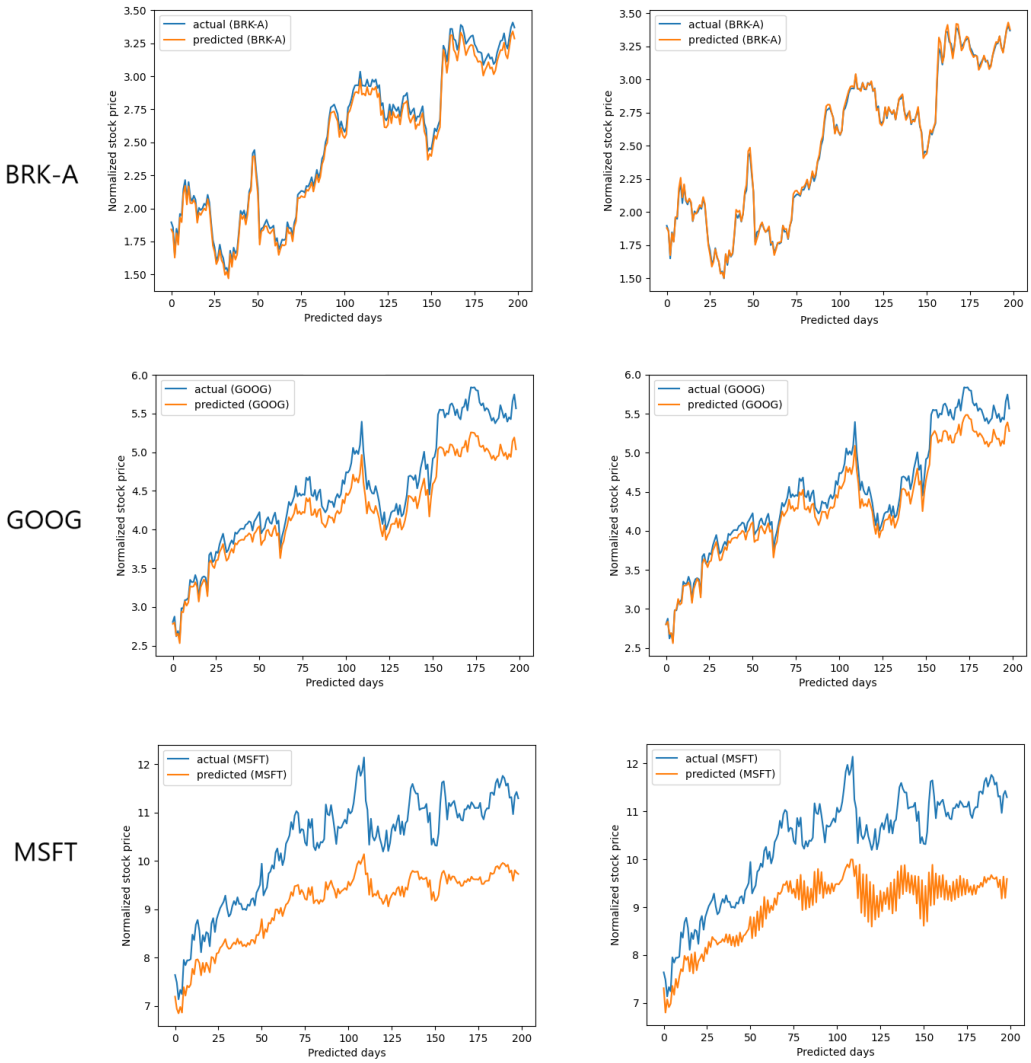
Na obrázku 5.3 pozorujeme predikcie modelov z druhého experimentu porovnané s referenčnou cenou na 200 obchodných dňoch testovacieho obdobia. Na prvý pohľad je jasné, že úspešnosť týchto modelov sa veľmi líší naprieč prezentovanými modelmi. V najvrchnejšom grafe akcie BRK-A vidíme, že sú jej predikcie testovacieho obdobia rovnako kvalitné ako predikcie validačného obdobia, pričom krivka vytvorená z predikcií relatívne dobre opisuje referenčnú krivku ceny akcie. Na strednom grafe akcie GOOG je viditeľné zhoršenie kvality predikcií testovacieho obdobia oproti predikciám validačného obdobia aj napriek tomu, že predikcie testovacieho obdobia udržiavajú trend referenčnej ceny akcie približne do 150. predikovaného dňa. Najväčšie zhoršenie môžeme pozorovať na spodnom grafe akcie MSFT, kde nie sme schopní porovnať predikcie modelu s referenčnou krivkou ceny akcie.



Obr. 5.4: Výsledky predikcie 200 dní z testovacieho obdobia pre akciu MSFT

Práve kvôli veľkému rozdielu v škále predikcií a referenčnej ceny sme sa rozhodli vizualizovať predikcie akcie MSFT zvlášť na obrázku 5.4. Z tohto grafu sú očividné dva problémy a to, že v prvej rade nesedí škála predikcií, ktorá sa pohybuje na hodnotách 103 - 108 dolárov, pričom škála referenčných cien akcie sa v tomto časovom období pohybuje na rozmedzí 140 - 225 dolárov a v druhej rade predikcie tohto modelu neopisujú trend referenčnej ceny akcie. Z týchto zistení vyplýva, že buď parametre, ktoré sme zvolili pre našu neurónovú sieť nám nedokážu zabezpečiť vytvorenie modelu, ktorý by dokázal produkovať dostatočne kvalitné predikcie v testovacom období, alebo sa v procese experimentovania vyskytla chyba.

Rozhodli sme sa preto túto hypotézu overiť a jednou z prvých zmien referenenčných parametrov, ktorú sme vykonali, bolo zvýšenie počtu LSTM jednotiek, lebo originálny počet 20 LSTM jednotiek nám prišiel nedostatočný. Ostávalo však zistiť, aký počet LSTM jednotiek by bol optimálny z hľadiska kvality výsledných predikcií, ale aj náročnosti trénovania modelu, keďže vyšší počet LSTM jednotiek značne zvyšuje trénovaciu dobu modelu. Vykonali sme preto niekoľko pokusov, kde sme porovnávali grafy predikcií modelov na testovacom období, pričom prezentujeme nasledujúce:



Obr. 5.5: Grafy predikcií 200 dní z testovacieho obdobia s rôznym počtom LSTM jednotiek

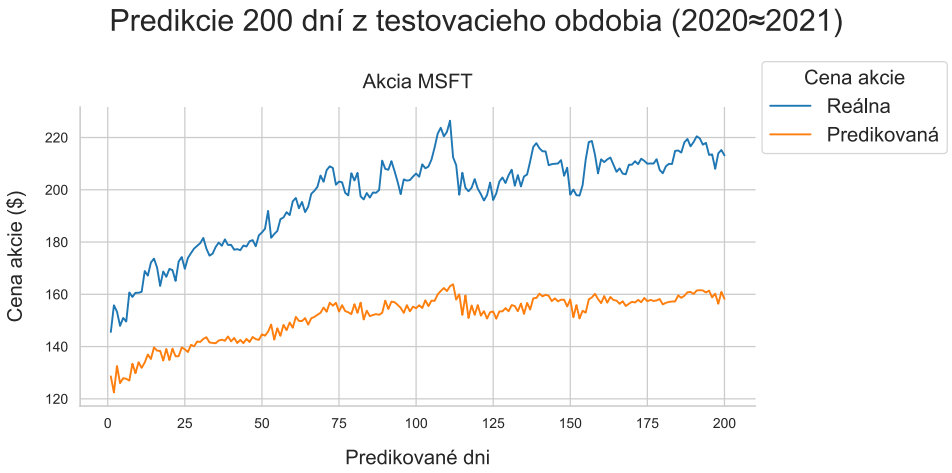
Vzhľadom k tomu, že originálny počet LSTM jednotiek pre jednu vrstvu bol 20, tak sme sa rozhodli najskôr skúsiť zdvojnásobovať tento počet. V rámci tohto experimentu sme preskúmali jednovrstvovú architektúru so 40, 80, 160, 320, 640 a 1280 LSTM jednotkami, pričom na obrázku 5.5 môžete vidieť vývoj predikcií s referenčnou cenou akcie[2] na 200 obchodných dňoch testovacieho obdobia. V ľavej časti grafu sa nachádzajú tri modely, ktoré využívajú 640 LSTM jednotiek a v pravej časti sa nachádzajú ďalšie tri modely, ale s 1280 LSTM jednotkami.[3] V prípade, že porovnáme kvalitu týchto predikcií s predikciami prezentovanými na obrázku 5.3, tak môžeme pozorovať, jemné zlepšenie na modeli akcie BRK-A a výrazne lepšie výsledky pri modeloch akcií GOOG a MSFT. Pri pohľade na obzvlásť problémovu škálu modelu akcie MSFT je toto zlepšenie najvýraznejšie, pričom sa výsledky približujú k škále referenčnej ceny a zároveň opisujú trend akcie.

[2]Všetky krivky uvedené v grafe 5.5 používajú normalizovanú cenu.
[3]Ostatne modely nebudú prezentované vzhľadom na nedostatočnú kvalitu ich predikcií.

Taktiež je dôležité zhodnotiť rozdiely medzi prezentovanými architektúrami a určiť, ktorý počet LSTM jednotiek dosahuje lepšie výsledky. Ak sa rozhodneme porovnávať výsledky pre akcie BRK-A a GOOG, tak môžeme povedať, že by bolo vhodnejšie používať 1280 LSTM jednotiek, lebo ich predikcie dosahujú mierne lepšie výsledky ako ich náprotivky s 640 LSTM jednotkami. Avšak v tomto prípade sme sa rozhodli radšej zamerať na model akcie MSFT, lebo pri ňom vznikalo viac problémov ako pri ostatných modeloch referenčných akcií. Model akcie MSFT, ktorý používal 640 LSTM jednotiek dosiahol výrazné zlepšenie aj napriek tomu, že škála jeho predikcií nebola ideálna. Čo sa týka jeho náprotivku s 1280 jednotkami, môžeme pozorovať ďalší problém, a to, že krivka jeho predikcií osciluje s vysokou frekvenciou, takže toto nastavenie by predikovalo veľmi nespoľahlivé výsledky.

Ďalšími faktormi, ktoré sme pri voľbe výsledného modelu zvažovali, bola aj značne vyššia náročnosť trénovania modelov s 1280 LSTM jednotkami, ale aj to, že sme odhadovali, že problém horšej škály modelu MSFT vznikol pri chybe v normalizácii dát a je riešiteľný, pričom riešiteľnosť problém oscilácie predikcií nám nebola známa.

Po zhrnutí týchto informácií sme sa preto rozhodli nepoužiť architektúru s počtom 1280 LSTM jednotiek a keďže sme vtedy porovnávali úspešnosť aj modelu s 320 LSTM jednotkami, ktorý nemal tak dobré výsledky ako model so 640 LSTM jednotkami, tak sme sa rozhodli pre kompromis, a to architektúrú s 512 LSTM jednotkami.



Obr. 5.6: Predikcie modelu MSFT s 512 LSTM jednotkami na 200 obchodných dňoch testovacieho obdobia, ktoré môžeme porovnať s grafmi modelov MSFT s 640 a 1280 LSTM jednotkami uvedených na obrázku 5.5. Výhodami tejto architektúry sú jej relatívne kvalitné výsledky pri opisovaní trendu akcie a nižšia výpočetná náročnosť trénovania.

V poslednej časti experimentu dva sme sa zamerali na riešenie problému škály modelov. Doposiaľ sme dáta pre modely normalizovali metódou **Z-Score**, ktorú používali aj autori referenčnej práce[12], čo nám síce umožňovalo jednoduché porovnanie výsledkov medzi prácami, ale ako demonštrujeme aj na grafe 5.6, k dosiahnutiu lepších výsledkov bude v prvom rade potrebné zlepšiť škálu výsledných predikcií, ktorá sa určuje v procese normalizácie. Rozhodli sme sa preto vyskúšať niekoľko normalizačných metód (MaxAbsScaler, MinMax, tanh-estimator, atď.) a chceme prezentovať výsledky po normalizovaní metódou MaxAbsScaler.

Obr. 5.7: Výsledky predikcie 200 dní z validačného obdobia

Výsledky predikcií jednotlivých modelov po zmene normalizačnej metódy môžeme vidieť na obrázku 5.7, pričom boli doposiaľ najpresnejšie, aké sa nám podarilo vytvoriť. Krivky predikovaných cien akcií BRK-A a GOOG opisujú referenčnú krivku cien a aj na modely akcie MSFT pozorujeme výrazné zlepšenie kvality predikcií. Záverom tohto experimentu bolo, že sme upravením počtu LSTM jednotiek a spôsobu normalizácie dosiahli kvalitnejšie výsledky predikcie. Vhľadom na narastajúci počet potrebných vizualizácií sme sa rozhodli generovať výstupy trénovaných modelov do samostatných csv súborov, ktoré sa nachádzajú v zložke `experiment-results` v koreňovom adresári projektu. Spracovanie výsledkov experimentov do grafickej reprezentácie je realizované skriptom `vizualizer.py`, ktorý načítava výsledky experimentov z týchto csv súborov, ktoré následne spracuje a tvorí z nich grafy prezentované v tejto práci.

5.3 Experiment 3 - Tvorba viac-premenného modelu

Hlavnou myšlienkou experimentu tri bola otázka, aké výsledky by sme dosiahli v prípade, že by sme vytvorili model, ktorého vstupné dáta by boli komplexnejšie, t.j. tvorené z niekoľkých vstupných premenných a či by takéto modely dokázali tvoriť kvalitnejšie predikcie. Prvú úlohu, ktorú bolo potrebné pri konštrukcii tohto modelu vyriešiť, bolo zvolenie správnych vstupovných premenných. Potenciálnych kandidátnych riešení bolo viac a teraz budeme prezentovať zvolené:

5.3.1 Cena

Voľba ceny ako potenciálneho vstupného parametru modelu bola jednoznačná z hľadiska prezentovaných dosiahnutých výsledkov jedno-premenného modelu. Avšak táto vlastnosť akcie prináša do predikcie aj jeden problém, a to, že nezvykne byť stacionárna. Pre lepšie pochopenie, prečo je nestacionárnosť ceny problém, treba najskôr pochopiť, čo vôbec znamená stacionarita v koncepte analýzy časových radov.

Stacionaritu dátovej sady by sme mohli vysvetliť tak, že štatistické vlastnosti dátovej sady sa nebudú meniť v čase. Samozrejme, to neznamená, že sa nebudú meniť jednotlivé hodnoty, ale to, že správanie časového radu ostáva vo všeobecnosti rovnaké. Tieto „zmeny správania“ môžeme popísať faktormi, ako sú napr. trendy pri časových radoch alebo sezónnosť - t.j. pravidelne sa opakujúce vzory či obrazce. Stacionaritu môžeme v dátach pozorovať jednoduchými spôsobmi, ako sú napr. vykreslenie si strednej hodnoty a rozptylu a sledovania, či sa tieto hodnoty výrazne menia po čase, trendovosť a sezónnosť je ľahko pozorovateľná na grafoch cien akcií a v prípade, že si chceme byť istí môžeme použiť štatistické testy - ADF test alebo KPSS test. [14]

Prečo je teda stacionarita respektíve nestacionarita nežiadúca pri predikovaní časových radov? Jednoduchá odpoveď bude taká, že táto nestacionarita nám značí, že sa daný časový rad môže meniť nepredvídateľne, a preto je veľmi ťažké zostrojiť model, ktorý by dokázal túto nepredvídateľnosť spoľahlivo predikovať.

5.3.2 Návratnosť ceny akcie

Jeden zo spôsobov, ktorými sa môžeme vyhnúť nestacionarite dátovej sady, sú transformácie dát. Riešení sa nám tu ponúka niekoľko ako napr. diferenčná transformácia či logaritmická transformácia. Počas realizácie tohto experimentu sme sa rozhodli použiť percentuálnu zmenu medzi jednotlivými dňami, inak povedané vyrátali sme percentuálnu návratnosť

medzi jednotlivými dňami v dátovej sade a použili sme ich ako alternatívu voči samotnej ceny akcie.

5.3.3 Pohyblivé priemery

Jedná sa o štatistický ukazovateľ, ktorý častokrát používajú investori a obchodníci s akciami pre lepšie zachytenie trendov ceny. Tento ukazovateľ nám totiž poskytuje iný pohľad na vývoj akcie ako jej samotnú cenu. Pohyblivých priemerov existuje viacero druhov a my sme sa zamerali na jednoduchý pohyblivý priemer (skr. MA) a exponenciálny pohyblivý priemer (skr. XMA).

Jednoduchý pohyblivý priemer funguje na princípe vytvorenia „časového okna“, t.j. zvolí sa časové obdobie napr. 30 dní a následne sa získajú ceny tohto časového obdobia a vyráta sa z nich priemer. Tento proces sa aplikuje na celú časovú radu, isteže, len do momentu, kým máme dosť dní v dátovej sade, aby sme mohli pre daný deň získať dostatočný počet cien.

Exponenciálny pohyblivý priemer funguje na podobnom princípe ako jednoduchý pohyblivý priemer, ale prináša koncept váh pre jednotlivé dni, ktoré sa do priemeru zarátavajú a týmto spôsobom umožňuje prioritizovať niektoré hodnoty. Príklad z praxe by bol, ak by sme odhadovali, že staršie hodnoty majú menší vplyv na budúcu cenu akcie, a preto by ich váha v priemere bola nižšia ako váha novších hodnôt, takže by výslednú hodnotu exponenciálneho priemeru ovplnyvnili značne menej.

5.3.4 Minimum, Maximum, Štandardná odchýlka

Kanditáne riešenie, ktoré vzniklo z podobnej myšlienky ako pohyblivé priemery, kde sme vytvárali časové okná, ale tentokrát namiesto priemeru používame minimálnu a maximálnu cenu akcie a taktiež štandardnú odchýlku (skr. MMS). Dĺžka časového okna, z ktorého sú tieto štatistické hodnoty počítané, je rovnaká ako pri pohyblivých priemeroch, a to 30 dní.

5.3.5 Obchodovaný objem akcie

Jednou z dôležitých informácií, ktorými disponujeme pri jednotlivých dňoch akcie, je aj hodnota obchodovaného množstva akcií (skr. VOL) v daný deň. Chceli sme preto zistiť, či by sme vedeli využiť túto informáciu na zlepšenie predikcií vzhľadom na to, že v praxi investori, ktorí obchodujú s akciami, túto informáciu využívajú vo forme RSI indexu, ktorý sa počíta práve z množstva obchodovaných akcií.

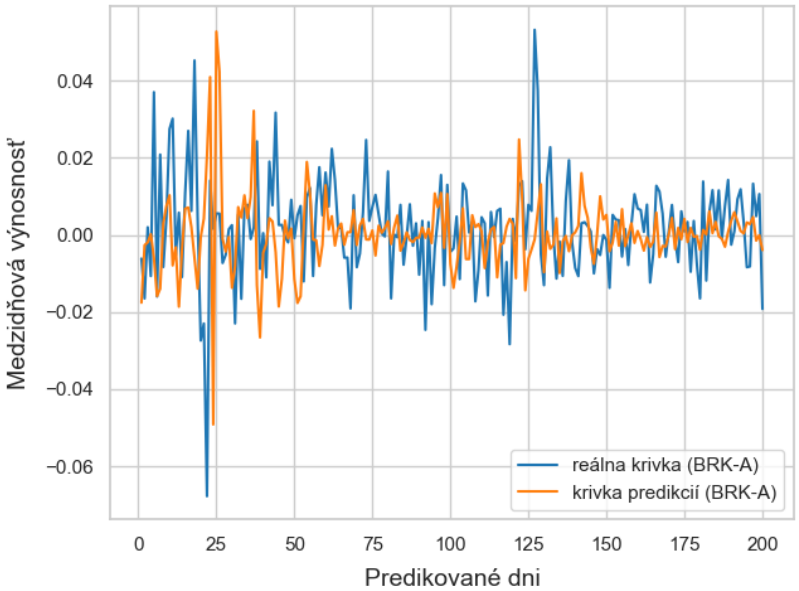
5.3.6 Voľba vhodných parametrov modelu

Následne po ustanovení všetkých kandidátnych riešení bolo treba vyhodnotiť kvalitu predikcií pri rôznych vstupných parametroch modelu. Koncepcia tohto experimentu spočíva v tom, že sme sa rozhodli tvoriť sady vstupných parametrov zložených z jednotlivých kandidátnych riešení. Každá sada musela obsahovať jedno povinné kandidátne riešenie, a to cenu alebo návratnosť, nakoľko pre nás predstavovali základ tohto modelu. Následne sme vytvorili osem kombinácií zo zostávajúcich kandidátnych riešení, ktorých kvalitu predikcií sme vyhodnotili na základe validačných odchýliek získaných trénovaním za použitia 10-ročnej dátovej sady.

Tabuľka 5.1: Výsledky porovnania vstupných parametrov viac-premenného modelu s poradím podľa najnižšej priemernej validačnej odchýlky, kde „ÁNO“ značí prítomnosť kandidátneho riešenia vo vstupných parametroch. Kandidátne riešenia sú prezentované nasledovne z ľava do prava: pohyblivý priemer, exponenciálny pohyblivý priemer, maximum-minimum-štandardná odchýlka a denný obchodovaný objem.

MA	XMA	MMS	VOL	Validačné MSE	Poradie
ÁNO	NIE	ÁNO	NIE	0.000361	2.
ÁNO	NIE	NIE	ÁNO	0.000363	3.
ÁNO	NIE	ÁNO	ÁNO	0.000422	7.
NIE	ÁNO	ÁNO	NIE	0.000393	4.
NIE	ÁNO	NIE	ÁNO	0.000428	8.
NIE	ÁNO	ÁNO	ÁNO	0.000414	6.
NIE	NIE	ÁNO	NIE	0.000412	5.
NIE	NIE	ÁNO	ÁNO	0.000352	1.

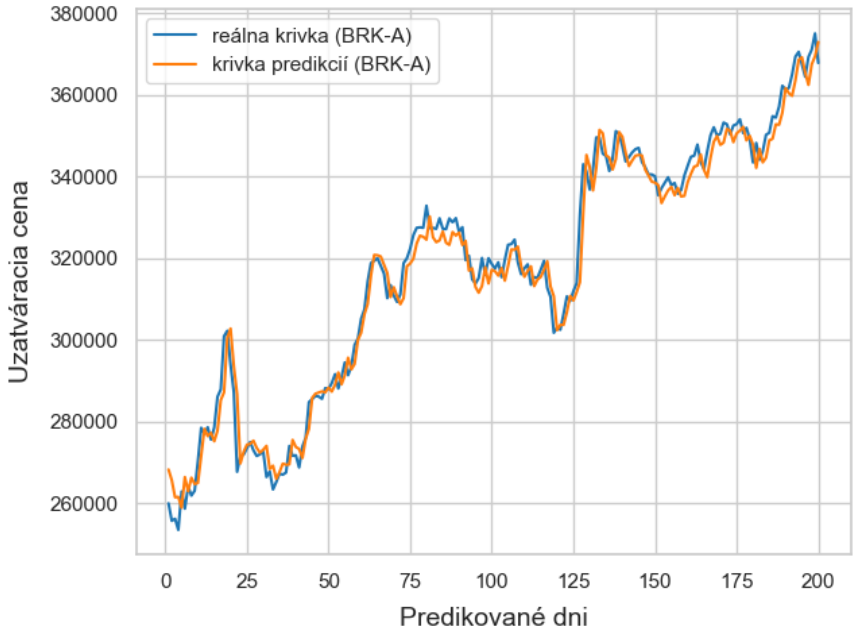
V tabuľke 5.1 prezentujeme výsledné porovnanie nami vytvorených kombinácií kandidátnych riešení s cenou akcie ako hlavným kandidátnym riešením. Poradie týchto sád bolo určené podľa najnižšej priemernej validačnej chyby všetkých troch referenčných akcií, pričom validačné chyby jednotlivých akcií boli vypočítané ako priemerná validačná chyba z troch nezávislých trénovaní. Z uvedených výsledkov je jednoznačné, že najlepšie výsledky dosahuje model, ktorý používa exponenciálny pohyblivý priemer a denný obchodovaný objem akcie ako vstupné parametre.



Obr. 5.8: Výsledok predikcií modelu akcie BRK-A využívajúcej medzidňovú návratnosť a MMS, na ktorej môžeme pozorovať nedostatočnú predikčnú schopnosť modelu (odchýlka na validačnom období bola 0.0248 MSE).

Validačné odchýlky s medzidňovou návratnosťou ako hlavným kandidátnym riešením neuvádzame, lebo porovnanie validačných odchýliek s validačnými odchýlkami prezentovanými

v tabuľke 5.1 nie je z dôvodu odlišných škál možné. Taktiež deklarujeme, že modely, ktoré používali medzidňovú návratnosť ako hlavné kandidátne riešenie v našej práci nedokázali vytvoriť kvalitné predikcie, a preto ich nechávame ako predmet ďalšieho experimentovania.



Obr. 5.9: Výsledok predikcií modelu akcie BRK-A využívajúcej uzatváraciu cenu, denný obchodovaný objem a maximum-minimum-štadardnú odchýlku, ktorá dosiahla najlepšie priemerné výsledky z porovnávaných kandidátnych riešení.

Po zhrnutí všetkých prezentovaných faktov sme dospeli k záveru, že viac-premenný model, ktorý používal uzatváraciu cenu akcie, maximum, minimum, štandardnú odchýlku z 30-dňového pohyblivého okna a denný obchodovaný objem akcie dosiahol najlepšie výsledky, a preto budeme túto kombináciu vstupných parametrov používať v nasledujúcich experimentoch.

Tabuľka 5.2: Porovnanie validačných odchýliek jedno a viac-premenných modelov.

Typ modelu	**BRK-A**	**GOOG**	**MSFT**
jedno-premenný	0.00056	0.00038	0.00011
viac-premenný	0.00020	0.00030	0.00070

V neposlednom rade by sme chceli ešte chceli porovnať dosiahnuté výsledky jedno-premenného a viac-premenného modelu, ktoré môžete vidieť v tabuľke 5.2. Na základe týchto výsledkov môžeme povedať, že viac-premenné modely pre akcie BRK-A a GOOG dosahujú lepšie validačné výsledky, ale jedno-premenný model akcie MSFT dosahuje výrazne lepšie výsledky predikcií ako jeho viac-premenný náprotivok. Finálne porovnanie obidvoch navrhnutých typov modelov vykonáme až po ich optimalizácii v podkapitole 5.5.

5.4 Experiment 4 - Analýza predikcií modelov

Vzhľadom na nami zvolený princíp predikcie, kde modely akcií tvoria predikcie v krátkodobom časovom horizonte z jedného dňa na druhý, sme sa neuspokojili len s prezentáciou výsledkov na dlhšom časovom horizonte 200 dní, ale rozhodli sme sa analyzovať kvalitu predikcií v rámci jedného mesiaca. Vytvorili sme preto tento nezávislý experiment, ktorý trénoval desať modelov pre každý typ modelu a referenčnú akciu s použitím 10-ročnej dátovej sady, takže vo výsledku vzniklo 60 samostatných modelov. Následne sme s každým modelom vytvorili 499 predikcií na testovacom období od 31.12.2020 do 22.12.2022. Kvalitu týchto predikcií sme sa rozhodli vyhodnocovať podľa alternatívneho kritéria, a to správnej predikcie zmeny trendu ceny nasledujúceho dňa. Vyhodnocovanie výsledkov predikcie podľa tohto kritéria budeme demonštrovať príkladom.

Povedzme, že dnešný dátum je 1.12.2022 a už boli zverejnené dnešné uzatváracie ceny akcií BRK-A a GOOG, ktorých uzatváracie ceny nasledujúceho dňa budeme pomocou jednopremenného modelu predikovať. Predikčnému modelu teda pripravíme dnešnú uzatváraciu cenu a ďalších 59 predošlých uzatváracích cien, vytvoríme z nich predikcie a porovnávame ich s reálnou uzatváracou cenou nasledujúceho dňa.

Tabuľka 5.3: Predikcia cien akcií z 1.12.2022 na 2.12.2022 jedno-premenným modelom

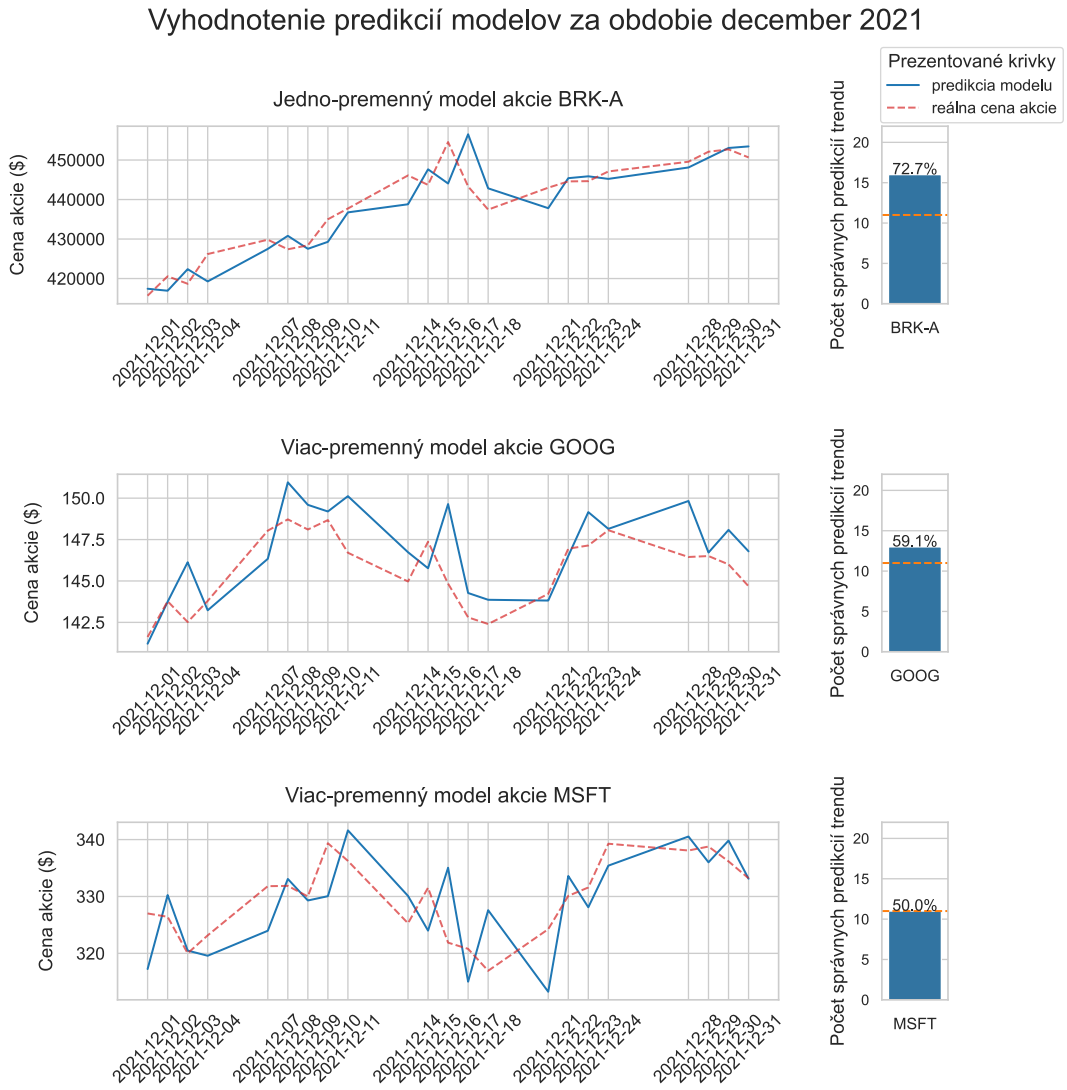
Model akcie	**BRK-A**	**GOOG**
Cena 1.12.2022 ($)	477,085	101.28
Cena 2.12.2022 ($)	477,403	100.83
Predikovaná cena 2.12.2022 ($)	481,720.83	100.99
Rozdiel cien	316 $ / 0.06 %	-0.45 $ / -0.44 %
Rozdiel ceny a predikcie	4,317.83 $ / 0.97 %	-0.29 $ / 0.29 %

V tabuľke 5.3 uvádzame uzatváracie ceny akcií, predikcie pre deň 2.12.2022, rozdiel uzatváracích cien, ktorý udáva nárast alebo poklesy ceny danej akcie uvedený v dolároch a percentách a taktiež rozdiel uzatváracej ceny akcie z 2.12.2022 a predikcie pre tento deň, taktiež uvedených v dolároch a percentách. Dôležitým zistením z týchto čísel sú rozsahy samotných cien, ktoré sú rádovo odlišné, keďže cena akcie GOOG je približne 100 dolárov, ale ceny akcie BRK-A sa v týchto dňoch pohybovali v rozmedzí 475,000 až 480,000 dolárov, pričom medzidňové zmeny ceny sa u akcie BRK-A môžu pohybovať v tisícoch dolárov, ale zmeny cien akcie GOOG sa zvyknú pohybovať v niekoľkých dolároch. Vzhľadom na tieto zistenia je teda ťažké určiť prípustnú hranicu rozdielu reálnej a predikovanej ceny akcie. Teoreticky by sa táto hranica dala určiť ako percentuálna hodnota, ale v našej práci sme sa rozhodli tento problém riešiť prakticky a to na základe týchto tvrdení.

1. V prípade, že sa model nemýli a predikuje zajtrajší rast ceny akcie, tak akciu kúpime dnes a môžeme ju zajtra predať zo ziskom.
2. V prípade, že sa model nemýli a predikuje zajtraší pokles ceny akcie, tak bude správne investičné rozhodnutie akciu dnes nekúpiť.

Ak vyššie uvedené tvrdenia dodržíme, tak budú vytvorené modely v praxi profitovateľné a nemusíme zisťovať prípustnú hranicu chyby predikcií. Preto sme sa rozhodli v rámci tohto experimentu redukovať problém presnej predikcie ceny na problém binárnej klasifikácie zmeny trendu ceny akcie v nasledujúcom dni. Dôležité je ešte podotknúť, že naše modely sme stále trénovali na základe metriky MSE, takže sa modely pri trénovaní snažili minimalizovať

rozdiel medzi predikovanou a reálnou cenou akcie, ale proces trénovania by bolo možné modifikovať tak, aby vytvorené modely predikovali trend ceny akcie čo predkladáme ako námet na ďalšie experimenty.



Obr. 5.10: Výsledky predikcií najlepších modelov všetkých referenčných akcií za mesiac december 2021.

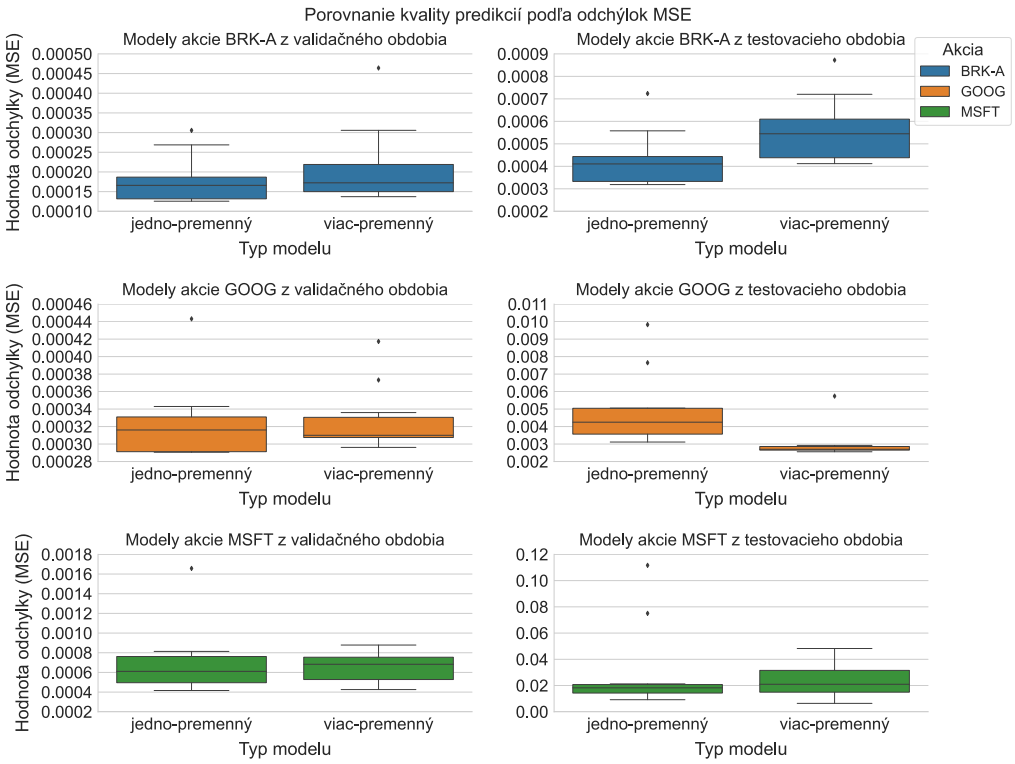
Na obrázku 5.10 prezentujeme tri grafy predikcií jednotlivých referenčných akcií, pričom modré krivky prezentujú vytvorené predikcie najlepších modelov jednotlivých akcií, vybraných na základe dosiahnutého počtu správnych predikcií zmeny trendu ceny akcie a červené krivky prezentujú reálnu cenu akcie v danom čase. V pravej časti týchto grafov sa nachádza ukazateľ počtu správnych predikcií trendu. V mesiaci december 2021 bolo 22 obchodovaných dní, takže sme znázornili 50% úspešnosť oranžovou priamkou. Pohľad na tieto grafy nám odhalil jednu skutočnosť, ktorá nebola taká očividná na grafoch dlhšieho časového horizontu, a to, že predikčné krivky sa síce väčšinu času držia blízko skutočných hodnôt, ale v prípade, že cena akcie mení trend v časovom okamžiku t, tak sa táto skutočnosť prejaví na predikčnej krivky až v čase $t + 1$. Avšak ak by sme vyhodnocovali kvalitu

týchto predikcií na základe metriky MSE na testovacom období, nadobúda nízke hodnoty, konkrétne 0.0003 pre akciu BRK-A, 0.0028 pre akciu GOOG a 0.0064 pre akciu MSFT. Z týchto výsledkov vyplýva, že model akcie BRK-A je najpresnejší a modely ostatných akcií sú aj napriek dobrým výsledkom v porovnaní s modelom akcie BRK-A horšie o jeden rád.

Tabuľka 5.4: Vyhodnotenie predikcií trendov vybraných modelov

Obdobie	**Akcia**	**Správny počet predikcií**	**Celkový počet predikcií**
December 2021	BRK-A	16 (72.7 %)	22
December 2021	GOOG	13 (59.1 %)	22
December 2021	MSFT	11 (50.0 %)	22
Celkové obdobie	BRK-A	279 (55.9 %)	499
Celkové obdobie	GOOG	258 (51.7 %)	499
Celkové obdobie	MSFT	264 (52.9 %)	499

V tabuľke 5.4 prezentujeme dosiahnuté výsledky vybraných modelov za obdobie december 2021 zobrazené na grafe 5.10 a za celkové testovacie obdobie, ktorého dĺžka je 499 obchodných dní, teda približne dva kalendárne roky. Vybrané modely napriek tomu, že neboli priamo trénované na predikciu trendov na relatívne dlhom časovom období, dosahujú nadpolovičnú úspešnosť predikcií, pričom existujú aj mesiace, kde bola úspešnosť modelov výrazne nadpriemerná. Obzvlášť dobré sú výsledky modelu akcie BRK-A, ktorému sa za celkové obdobie darilo najviac v porovnaní s ostatnými prezentovanými modelmi a v decembri 2021 dosiahol úspešnosť až 72.7 %.



Obr. 5.11: Porovnanie validačných a testovacích odchýliek všetkých modelov

Po vyhodnotení úspešností modelov na základe správnej predikcie trendu nás ďalej zaujímali celkové výsledky všetkých trénovaných modelov z hľadiska validačných a testovacích odchýliek, ktoré prezentujeme na obrázku 5.11. Celkovo sme vytvorili šesť grafov, pričom v ľavej časti sa nachádzajú výsledky modelov z validačného obdobia a vpravo z testovacieho obdobia. Každý jeden graf sa skladá z dvoch „krabicových grafov“ (angl. boxplot), pričom vľavo sú prezentované výsledky jedno-premenného modelu a vpravo viac-premenného modelu. Každý jeden krabicový graf je tvorený z desiatich hodnôt odchýlok z patričného obdobia.

Porovnanie výsledkov validačného obdobia s testovacím prináša očakávane výsledky, a to, že môžeme pozorovať ich mierne zhoršenie na testovacom období. Toto zhoršenie je najmenšie pri akci BRK-A, kde sa pohybuje v desattisícinách MSE. Pri modeli GOOG môžeme pozorovať zhoršenie hodnôt o celý rád, kde hodnoty MSE prechádzajú z desaťtisícin pri validačnom období na tisíciny v testovacom období a pri modeloch akcie MSFT sa hodnoty MSE na testovacom období pohybujú v rádoch stotín. Zhodnotenie z hľadiska validačných a testovacích odchýliek je preto jednoduché, lebo v ňom vedú s prehľadom modely akcie BRK-A.

Rozdiely z hľadiska typov modelov nie sú príliš významné, vzhľadom na to, že vo výsledkoch z testovacieho obdobia pre akciu BRK-A dosahuje o niečo lepšie výsledky jedno-premenný model, pri akcii GOOG dosahuje o niečo lepšie výsledky viac-premenný model a pri akcii MSFT sú výsledky oboch typov modelov podobné, kde výsledky viac-premenného modelu majú trochu väčší rozpytyl hodnôt, ale vo výsledkoch jedno-premenného modelu sa nachádzajú dve odhľahlé hodnoty. Zaujímavé je však pozorovanie predikcií viac-premenných modelov akcie GOOG a jedno-premenných modelov akcie MSFT na testovacom období, keďže dosahujú malý rozptyl odchýliek MSE, čo vypovedá o konzistentných výsledkoch naprieč viacerými trénovaniami.

Tabuľka 5.5: Prezentácia štatistických ukazovateľov odchýliek z obrázku 5.11 v MSE

Obdobie	**Akcia**	**Typ modelu**	**Maximum**	**Minimum**	**Priemer**
Validačné	BRK-A	jedno-premenný	0.000306	0.000126	0.000179
Validačné	BRK-A	viac-premenný	0.000464	0.000137	0.000211
Validačné	GOOG	jedno-premenný	0.000443	0.000291	0.000324
Validačné	GOOG	viac-premenný	0.000417	0.000296	0.000327
Validačné	MSFT	jedno-premenný	0.001657	0.000416	0.000707
Validačné	MSFT	viac-premenný	0.000878	0.000426	0.000644
Testovacie	BRK-A	jedno-premenný	0.000724	0.000319	0.000429
Testovacie	BRK-A	viac-premenný	0.000872	0.000412	0.000561
Testovacie	GOOG	jedno-premenný	0.009826	0.003115	0.004954
Testovacie	GOOG	viac-premenný	0.005740	0.002559	0.003025
Testovacie	MSFT	jedno-premenný	0.111646	0.009199	0.031434
Testovacie	MSFT	viac-premenný	0.048235	0.006405	0.023102

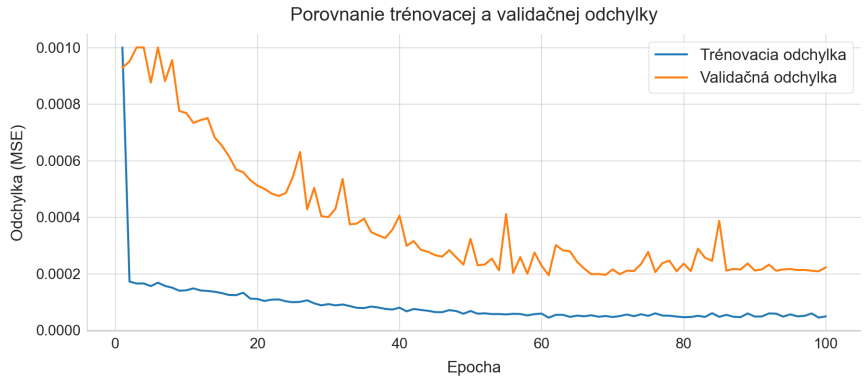
Na základe výsledkov prezentovaných v tabuľke 5.5 usudzujeme, že najlepších výsledkov dosahovali jedno-premenné modely akcie BRK-A, keďže dosahovali najnižšej priemernej validačnej aj testovacej odchýlky. Ak by sme hodnotili kvalitu podľa typu modelu, tak na základe odchýliek z testovacieho obdobia môžeme povedať, že viac-premenné modely dosahovali lepšie výsledky, keďže boli lepšie ako ich náprotivky pri akciách GOOG a MSFT.

5.5 Experiment 5 - Optimalizácia navrhnutých modelov

V poslednom experimente tejto práce sa budeme venovať optimalizácii výsledkov modelov na základe hľadania optimálneho parametru učenia a počtu trénovacích epoch s cieľom zabrániť pretrénovaniu modelov. Taktiež vyhodnotíme vplyv dĺžky trénovacieho obdobia na kvalitu vytvorených predikcií, kde budeme porovnávať výsledky modelov na základe ich validačných odchýliek naprieč 10 a 20 ročným trénovacím obdobím.

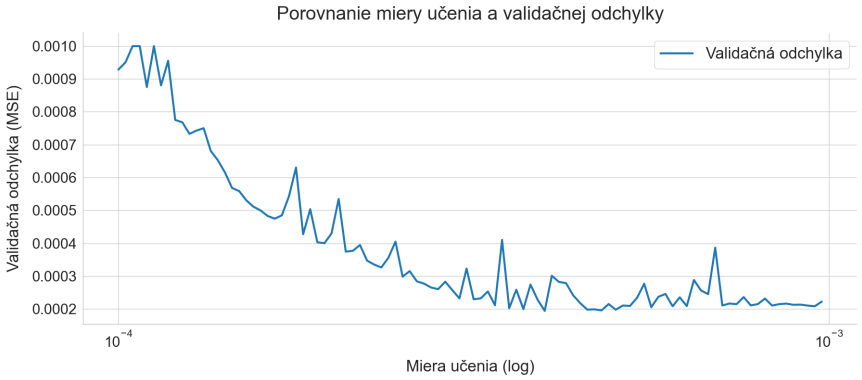
Predtým, ako sa dostaneme k samotnému experimentu, je potrebné bližšie upresniť, čo je to miera učenia optimalizátora a počet trénovacích epoch. Proces trénovania rekurentných neuronových sietí sme popísali v podkapitole 3.1, kde sme hovorili o tom, že v procese trénovania vykonáva optimalizátor variantu vzostupu gradientu. Pod počtom epoch teda môžeme rozumieť počet trénovacích cyklov, ktoré optimalizátor vykoná v procese hľadania minima účelovej funkcie a miera učenia nám definuje mieru zmeny váh modelov, ktorá sa na konci cyklu vykoná. Zlé nastavenie týchto parametrov by teda mohlo viesť k tomu, že pri príliš malom počte epoch by sme v procese trénovania nemali dostatok trénovacích cyklov na nájdenie dostačujúceho riešenia, prípadne ak by bola nevhodne zvolená miera učenia, tak by mohli byť zmeny váh príliš veľké a algoritmus by sa nevedel dostať k minimu účelovej funkcie.

Doposiaľ sme mali nastavený počet trénovacích epoch na 50 a miera učenia nebola špecifikovaná, a preto sa použila predvolená hodnota optimalizátoru Adam - 0.001. Pre potreby tohto experimentu sme sa rozhodli zvýšiť počet trénovacích epoch na 100 aj napriek väčším výpočetným nárokom a rozhodli sme, že vytvoríme štyri konfigurácie, ktoré budeme testovať. Tieto konfigurácie sa odlišovali na základe typu modelu, čiže jedno a viac-premenné a počtom vrstiev, keďže sme chceli zistiť, aký vplyv by mala na výsledky architektúra s dvomi výpočetnými LSTM vrstvami. Ďalej sme sa rozhodli skúmať vplyv dĺžky trénovacej dátovej sady na kvalitu predikcií, a preto sme tieto modely trénovali aj za použitia 20-ročnej dátovej sady. Poslednou vecou, ktorú potrebujeme spomenúť pred prezentáciou prvých výsledkov, je, že sme pre účely tohto experimentu použili LearningRateScheduler z frameworku Tensorflow, ktorý na základe nami definovanej funkcie: $e^4 * 10^{epoch/100}$, nahradí momentálnu trénovaciu epochu v tejto rovnici a tento výsledok nastavuje ako mieru učenia α pre optimalizátor. [4] Toto bolo potrebné spraviť preto, aby sme mohli nájsť optimálnu mieru učenia, čo vysvetlíme na nasledujúcich grafoch.



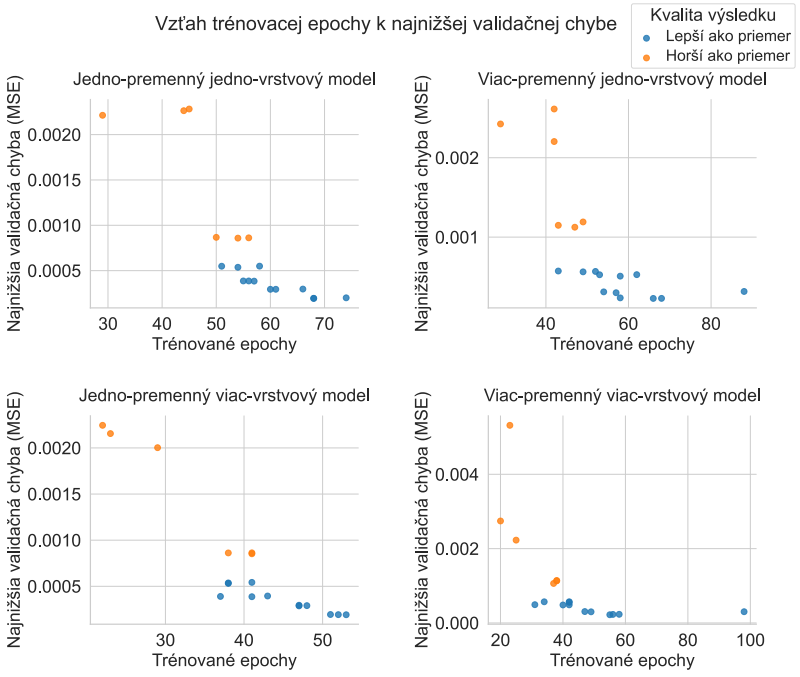
Obr. 5.12: Porovnanie trénovacej a validačnej odchýlky modelu BRK-A

[4] zdroj: Towards Data Science - How to Optimize Learning Rate with TensorFlow, autor: Dario Radečić

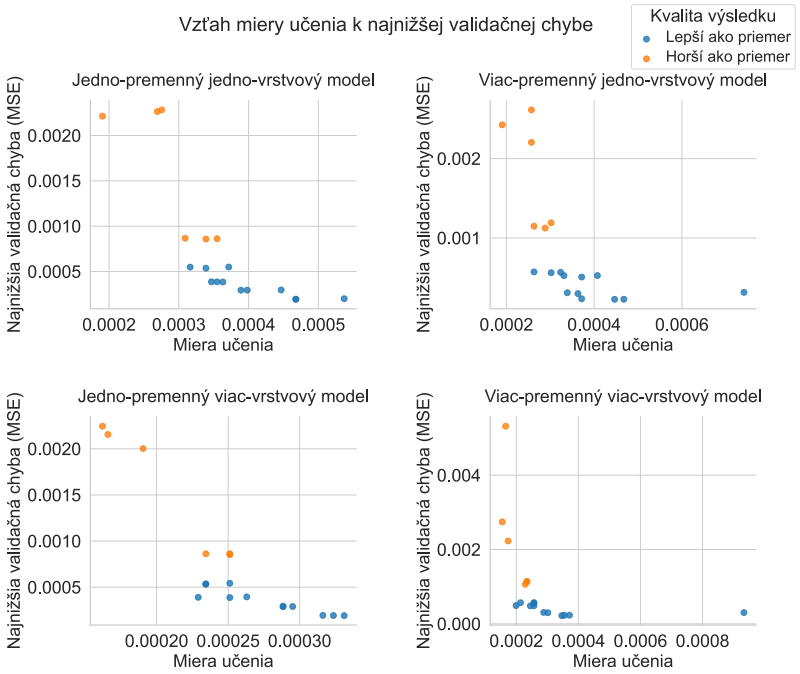


Obr. 5.13: Porovnanie validačných a testovacích odchýliek všetkých modelov

Na obrázku 5.12 môžete vidieť porovnanie trénovacej (modrej) a validačnej (oranžovej) odchýlky v MSE, kde sme nechali trénovať model akcie BRK-A po dobu 100 epoch. Ako môžete z grafu vidieť, tak trénovacia odchýlka naprieč epochami klesá aj keď sa miera jej klesania výrazne spomaľuje okolo 50. epochy, pričom krivka validačnej odchýlky dosahuje svoje minimum v 61. epoche. Z týchto faktov môžeme teda usúdiť, že proces trénovania tohto konkrétneho modelu dosiahol optimálne výsledky v 61. epoche a trénovanie mohlo byť zastavené. Pre tento istý model sme taktiež vytvorili graf na obrázku 5.13, ktorý má na osi x mieru učenia (v logaritmickej stupnici) a na osi y validačnú odchýlku v MSE. Toto nám umožnilo vyhľadať minimálnu hodnotu validačnej odchýlky a na základe nej určiť optimálny parameter miery učenia α, ktorá bola v tomto prípade 0.0004.



Obr. 5.14: Porovnanie validačných a testovacích odchýliek všetkých modelov



Obr. 5.15: Porovnanie validačných a testovacích odchýliek všetkých modelov

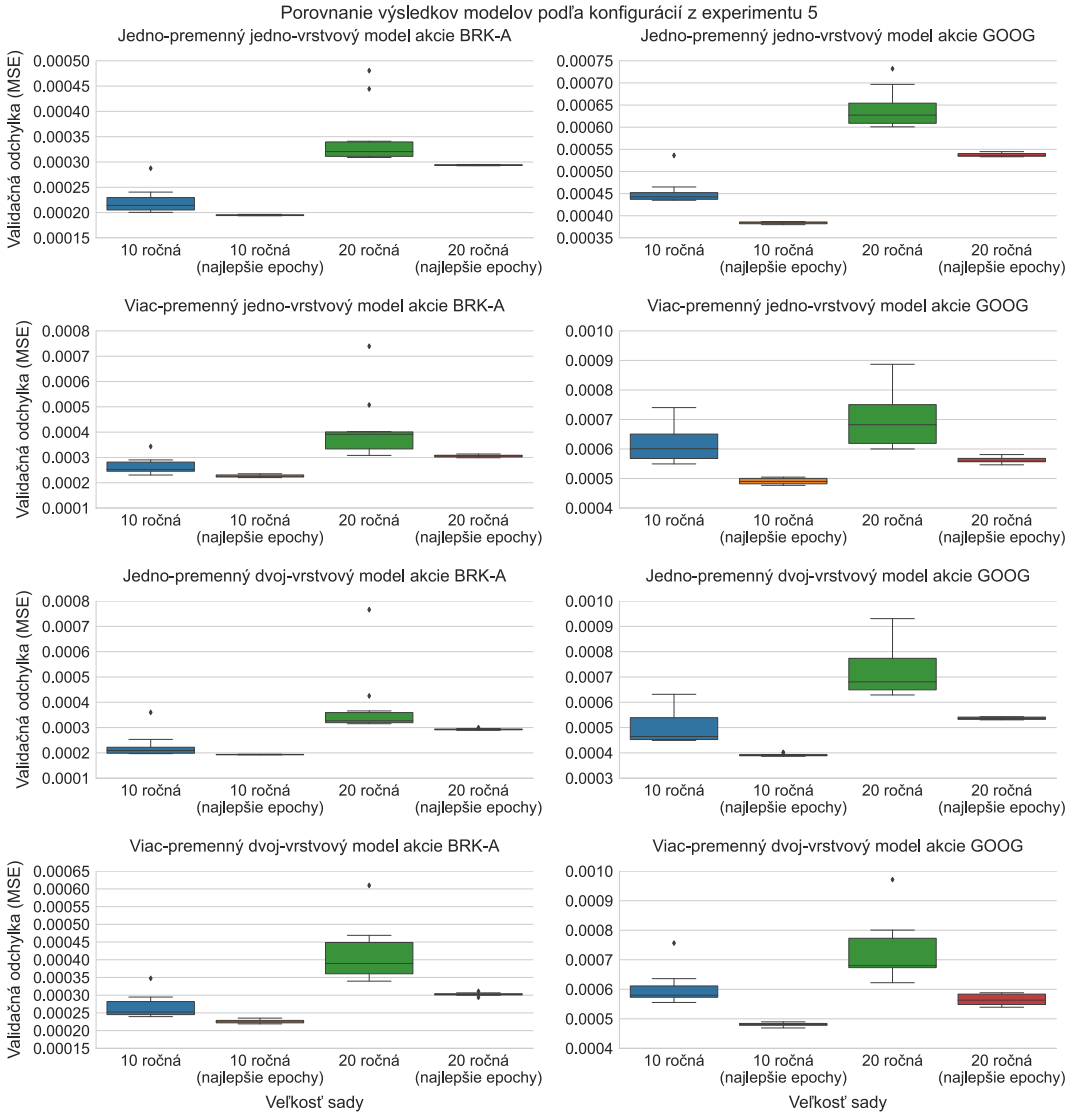
Tento proces sme samozrejme museli vykonať pre všetky spomínané konfigurácie, takže sme natrénovali 72 modelov (18 modelov pre každú konfiguráciu), pričom sme model každej akcie trénovali trikrát pre dosiahnutie všeobecnejších výsledkov, ktoré prezentujeme na obrázkoch 5.14 a 5.15. Obidve štvorice grafov majú rovnakú os y, a to najlepšiu nájdenú validačnú odchýlku, pričom sa líšia v osi x, kde sú na obrázku 5.14 zobrazené epochy a na obrázku 5.15 rôzne miery učenia. Zobrazenie epoch na osi x nám umožní nájsť epochu, v ktorej boli nájdené najmenšie validačné chyby a rovnaký princíp používame na grafoch 5.15 na nájdenie optimálnej miery učenia. Vzhľadom na to, že sme pre jednu konfigúraciu trénovali 18 modelov, tak sme sa snažili určiť najlepšiu priemernú hodnotu epoch a miery učenia. Toto vyhodnotenie sme zjednodušili tak, že sme si vyrátali priemernú hodnotu z najlepších validačných výsledkov celej konfigurácie a ak bola najlepšia validačná odchýlka modelu menšia alebo rovná tejto hodnote, tak bude tento bod označený modrou farbou a opačne oranžovou.

Tabuľka 5.6: Porovnanie zvolených hodnôt konfigurácií

Konfigurácia	**Epochy**	**Miera učenia**
Jedno-premenná jedno-vrstvová	65	0.00040
Viac-premenná jedno-vrstvová	58	0.00037
Jedno-premenná viac-vrstvová	52	0.00031
Viac-premenná ciac-vrstvová	56	0.00031

V rámci tohto experimentu sme sa rozhodli vylepšiť spôsob trénovania, a to spôsobom, že ak by sa v procese trénovania vyskytol model, ktorého kvalita predikcií bola lepšia ako kvalita modelu po poslednej trénovacej epoche, tak na konci trénovania nahradíme váhy trénovaného modelu, čo vo výsledku zabezpečí viac kvalitnejších modelov. Toto vylepšenie

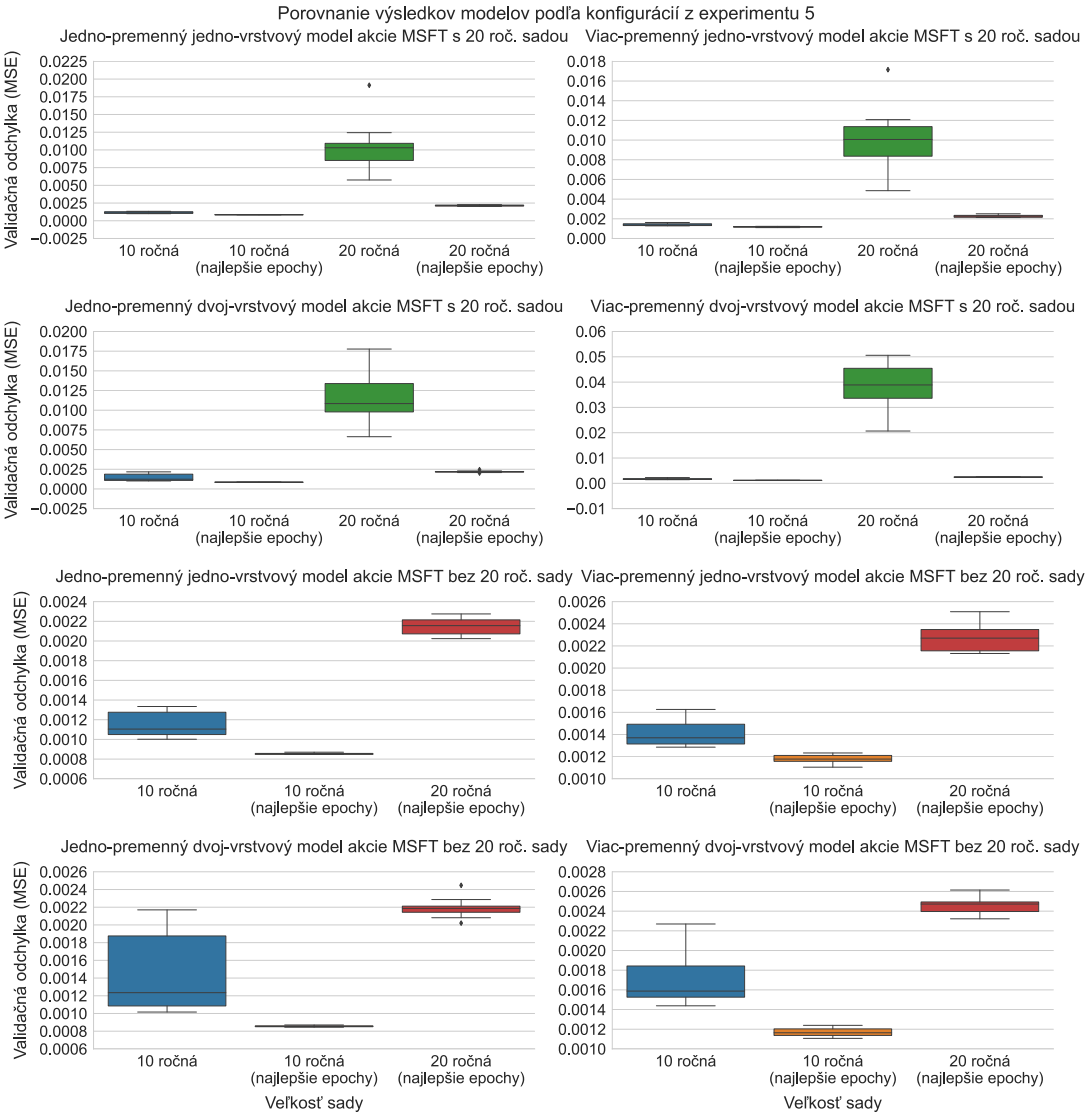
použijeme pre všetky štyri spomínané konfigurácie a obidve dátové sady, takže budeme môcť porovnať k akému zlepšeniu dochádza na všetkých validačných obdobiach.



Obr. 5.16: Porovnanie validačných odchýliek modelov akcie MSFT bez 20 roč. sady

Na obrázku 5.16 prezentujeme porovnanie validačných všetkých typov modelov trénovaných na 10 a 20-ročnej dátovej sade s návratom k najlepšej epoche modelu a bez, pričom tieto výsledky prezentujú 480 samostatných trénovaní na základe týchto parametrov: 4 konfigurácie, 2 dátové sady, bez / s návratom k najlepšej epochy, 3 referenčné akcie a 10 nezávislých trénovaní pre každú akciu. Prezentovaný graf sa skladá z ôsmich častí, pričom stĺpce rozdeľujú jednotlivé konfigurácie, a teda vľavej časti nachádzame výsledky pre akciu BRK-A v pravej pre akciu GOOG. Grafy v ľavej časti od vrchného po spodný prezentujú konfigurácie v nasledovnom poradí: jednopremenný jedno-vrstvový model, viac-premenný jednovrstvový model, jedno-premenný dvojvrstvový model a viac-premenný dvojvrstvový model.

Zaujímavé výsledky z týchto grafov sú, že všetky grafy modelov, kde bol použitý návrat k najlepšej epoche, dosahujú približne rovnaké validačné odchýlky, čo v prvej rade vypovedá o konzistentnosti týchto výsledkov a zároveň o tom, že tieto validačné odchýlky sú očividne najmenšie, aké sa dajú dosiahnuť pri použití týchto parametrov. Ďalším zaujímavým faktom je, že modely trénované na 10-ročnom období bez návratu k najlepšej epoche dosahujú menšie validačné odchýlky ako modely trénované na 20-ročných dátových sadách.



Obr. 5.17: Porovnanie validačných odchýliek modelov akcie MSFT bez 20 roč. sady

Výsledky konfigurácie pre akcie MSFT sú uvedené samostatne na obrázku 5.17, ktorého usporiadenie sa od grafu 5.16 líší z dôvodu, že výsledky trénovania modelov, ktoré používali 20 ročnú dátovú sadu produkovali vyššie odchýlky a skreslovali ostatné výsledky, tak sme sa rozhodli, že horné 4 grafy budú obsahovať všetky štyri skupiny trénovania a z dolných štyroch grafov sme odstránili výsledky modelov, ktoré boli trénované 20 ročnou dátovou sadou. Na týchto grafoch môžeme taktiež pozorovať, že modely trénované na 10 ročnej

dátovej sade dosahujú menšie validačné odchýlky ako modely trénované na 20 ročnej dátovej sade a že rozptyl modelov s návratom k najlepšej epoche je minimálny.

Tabuľka 5.7: Porovnanie výsledkov jednotlivých konfigurácií na základe validačnej odchýlky podľa použitej dátovej sady (DS) pre modely akcie BRK-A

Konfigurácia modelu	**DS**	**odchýlka (MSE)**	**Poradie**
Jedno prem. 1 vrst.	10 roč.	0.000223	3.
Jedno prem. 1 vrst.	10 roč. (naj.)	0.000195	2.
Jedno prem. 1 vrst.	20 roč.	0.000348	13.
Jedno prem. 1 vrst.	20 roč. (naj.)	0.000294	10.
Viac prem. 1 vrst.	10 roč.	0.000265	7.
Viac prem. 1 vrst.	10 roč. (naj.)	0.000227	5.
Viac prem. 1 vrst.	20 roč.	0.000412	15.
Viac prem. 1 vrst.	20 roč. (naj.)	0.000306	12.
Jedno prem. 2 vrst.	10 roč.	0.000227	6.
Jedno prem. 2 vrst.	10 roč. (naj.)	0.000193	1.
Jedno prem. 2 vrst.	20 roč.	0.000382	14.
Jedno prem. 2 vrst.	20 roč. (naj.)	0.000293	9.
Viac prem. 2 vrst.	10 roč.	0.000267	8.
Viac prem. 2 vrst.	10 roč. (naj.)	0.000226	4.
Viac prem. 2 vrst.	20 roč.	0.000414	16.
Viac prem. 2 vrst.	20 roč. (naj.)	0.000302	11.

Na záver tohto experimentu by sme chceli ešte analyzovať kvalitu predikcií modelov na základe rôznych faktorov, a preto tieto dáta budeme prezentovať formou tabuliek, kde kvalitu modelov určíme podľa priemernej dosiahnutej validačnej odchýlky.

Prvé porovnanie, ktoré chceme realizovať touto formou, je nájdenie najlepšej konfigurácie pre jednotlivé akcie, a preto v tabuľke 5.7 prezentujeme porovnanie validačných odchýliek modelov akcie BRK-A pre všetky trénované konfigurácie a dátové sady. V prvej trojici najlepších konfigurácií pre akciu BRK-A sa nachádzajú iba jedno-premenné modely, pričom prvý model je dvojvrstvový a druhé a tretie miesto obsadili jedno-vrstvové modely.

Ďalšou zaujímavosťou je, že všetky tri najlepšie modely boli trénované za použitia 10-ročnej dátovej sady a dve z troch konfigurácií používali návrat k najlepšej epoche. Modely umiestnenené na 4. až 8. priečke boli všetky taktiež trénované za použitia 10-ročnej dátovej sady, pričom lepšie umiestnenia používali návrat k najlepšej epoche.

Záver pre porovnanie konfigurácií akcie BRK-A je, že je vhodné tieto modely trénovať za použitia 10-ročnej dátovej sady a taktiež je vhodné používať princíp vrátenia sa k najlepšej epoche.

Tabuľka 5.8: Porovnanie výsledkov jednotlivých konfigurácií na základe validačnej odchýlky podľa použitej dátovej sady (DS) pre modely akcie GOOG

Konfigurácia modelu	DS	odchýlka (MSE)	Poradie
Jedno prem. 1 vrst.	10 roč.	0.000453	3.
Jedno prem. 1 vrst.	10 roč. (naj.)	0.000384	1.
Jedno prem. 1 vrst.	20 roč.	0.000641	13.
Jedno prem. 1 vrst.	20 roč. (naj.)	0.000538	8.
Viac prem. 1 vrst.	10 roč.	0.000614	12.
Viac prem. 1 vrst.	10 roč. (naj.)	0.000491	5.
Viac prem. 1 vrst.	20 roč.	0.000697	14.
Viac prem. 1 vrst.	20 roč. (naj.)	0.000561	9.
Jedno prem. 2 vrst.	10 roč.	0.000501	6.
Jedno prem. 2 vrst.	10 roč. (naj.)	0.000392	2.
Jedno prem. 2 vrst.	20 roč.	0.000730	16.
Jedno prem. 2 vrst.	20 roč. (naj.)	0.000537	7.
Viac prem. 2 vrst.	10 roč.	0.000603	11,
Viac prem. 2 vrst.	10 roč. (naj.)	0.000481	4.
Viac prem. 2 vrst.	20 roč.	0.000725	15.
Viac prem. 2 vrst.	20 roč. (naj.)	0.000565	10.

Ďalej by sme sa chceli pozrieť na výsledky validačných odchýliek konfigurácií modelov akcie GOOG, ktoré sme uviedli v tabuľke 5.8. Prvú vec, ktorú sme si na týchto výsledkoch všimli, je, že oproti výsledkom akcie BRK-A uvedeným v tabuľke 5.7, sú výsledky akcie GOOG o niekoľko desaťtisícin väčšie, pričom výsledky modelov akcie BRK-A sa pohybovali v rozmedzí 0.000193 - 0.000414 a výsledky akcie GOOG sa pohybujú v rozmedzí 0.000384 až 0.000730, z čoho vidíme, že výsledky modelov akcie GOOG pochádzajú aj z väčšieho rozmedzia hodnôt.

Čo sa týka samotných umiestnení modelov, tak 1. a 3. miesto pripadlo jedno-premenným a jednovrstvovým modelom a 2. miesto pripadlo jedno-premenným, ale dvojvrstvovým modelom, čo je podobná situácia ako pri výsledkoch modelov akcie BRK-A. Z hľadiska dátových sád konfigurácie umiestnené na prvých troch priečkach boli všetky trénované za použitia 10-ročnej dátovej sady a prvé dve konfigurácie používali návrat k najlepšej epoche.

Konfigurácie umiestnené na 4. až 8. priečke používali návrat k najlepšej epoche (okrem 6. konfigurácie) a prvá konfigurácia, ktorá používala 20-ročnú dátovú sadu a prekonala vo výsledkoch konfiguráciu používajúcu 10-ročnú dátovú sadu, bola až na 7. priečke.

Záver pre konfigurácie modelov akcie GOOG je podobný ako záver pri akcii BRK-A, a to, že pri trénovaní týchto modeloch doporučujeme používať 10-ročnú dátovú sadu a taktiež návrat k najlepšej epoche.

Tabuľka 5.9: Porovnanie výsledkov jednotlivých konfigurácií na základe validačnej odchýlky podľa použitej dátovej sady (DS) pre modely akcie MSFT

Konfigurácia modelu	DS	odchýlka (MSE)	Poradie
Jedno prem. 1 vrst.	10 roč.	0.001146	3.
Jedno prem. 1 vrst.	10 roč. (naj.)	0.000854	1.
Jedno prem. 1 vrst.	20 roč.	0.010295	14.
Jedno prem. 1 vrst.	20 roč. (naj.)	0.002150	9.
Viac prem. 1 vrst.	10 roč.	0.001408	6.
Viac prem. 1 vrst.	10 roč. (naj.)	0.001180	5.
Viac prem. 1 vrst.	20 roč.	0.010021	13.
Viac prem. 1 vrst.	20 roč. (naj.)	0.002281	11.
Jedno prem. 2 vrst.	10 roč.	0.001459	7.
Jedno prem. 2 vrst.	10 roč. (naj.)	0.000856	2.
Jedno prem. 2 vrst.	20 roč.	0.011376	15.
Jedno prem. 2 vrst.	20 roč. (naj.)	0.002191	10.
Viac prem. 2 vrst.	10 roč.	0.001696	8.
Viac prem. 2 vrst.	10 roč. (naj.)	0.001171	4.
Viac prem. 2 vrst.	20 roč.	0.037781	16.
Viac prem. 2 vrst.	20 roč. (naj.)	0.002454	12.

V neposlednom rade sa budeme venovať výsledkom konfigurácií modelov akcie MSFT, prezentovaných v tabuľke 5.9. Z prezentácie výsledných grafov konfigurácií z obrázku 5.17 vieme, že 20-ročné dátové sady dosahovali značne horšie odchýlky ako 10-ročné dátové sady, pričom nám to výsledky prezentované v tejto tabuľke len potvrdili.

Čo sa týka obsadenia prvých troch priečok tak nastáva podobná situácia ako pri výsledkoch konfigurácií modelov akcií BRK-A a GOOG, keďže prvé a tretie miesto obsadili jedno-premenné a jedno-vrstvové modely a druhé miesto pripadlo jedno-premenným dvoj-vrstvovým modelom. Rovnako ako aj v predošlých výsledkoch aj tentokrát sa na prvých priečkach nachádzajú modely trénované na základe 10-ročnej dátovej sady.

Všetky konfigurácie, ktoré sa umiestnili na 4. až 8. priečke taktiež používali 10-ročnú trénovaciu sadu, takže ani jedna konfigurácia, ktorá bola trénovaná na 20-ročnej dátovej sade, sa nedostala do prvej polovice.

Záver pre konfigurácie modelov akcie MSFT je identický ako závery predošlých výsledkov, a to, že prvým miestam dominovali modely trénované na 10-ročných dátových sadách a použitie návratu k najlepšej epoche sa vyplatilo.

Celkové zhrnutie týchto výsledkov je, že na trénovanie odporúčame použiť 10-ročné dátové sady a návrat k najlepšej epoche. V prípade, že by sme si chceli porovnať priemerné výsledky podľa jednotlivých akcií, dospejeme k záveru, že modely akcie BRK-A dosahovali najlepšie výsledky, modely akcie GOOG dosahovali mierne horšie výsledky ako modely akcie BRK-A, ale modely akcie MSFT dosahovali v priemere rádové zhoršenie výsledkov.

Tabuľka 5.10: Celkové porovnanie výsledkov konfigurácií

Konfigurácia modelu	Priemerná odchýlka (MSE)	Poradie
Jedno-premenný 1-vrstvový	0.001460	1.
Viac-premenný 1-vrstvový	0.001538	2.
Jedno-premenný 2-vrstvový	0.001595	3.
Viac-premenný 2-vrstvový	0.003890	4.

Následne sme sa rozhodli analyzovať výsledky podľa jednotlivých konfigurácií modelov bez akcií a dátových sád, ktoré uvádzame v tabuľke 5.10. Z týchto výsledkov je jednoznačné prvenstvo jedno-premenných jednovrstvových modelov. Na druhom mieste skončili viac-premenné jednovrstvové modely, len s mierne menšou priemernou validačnou odchýlkou ako modely jedno-premennej dvojvrstvovej konfigurácie, ktoré obsadili tretie miesto. Na poslednom mieste s výrazne horším výsledkom sa umiestnili viac-premenné dvoj-vrstvové modely, ktoré značne zaostávajú za ostatnými prezentovanými konfiguráciami.

Tabuľka 5.11: Porovnanie priemernej validačnej odchýlky podľa dátovej sady

Dátová sada	**Priemerná odchýlka (MSE)**	**Poradie**
10 ročná	0.000738	2.
10 ročná (naj)	0.000554	1.
20 ročná	0.006152	4.
20 ročná (naj)	0.001039	3.

Ďalším zaujímavým porovnaním, ktoré sme vykonali, bolo porovnanie priemerných validačných odchýliek podľa využitej dátovej sady, ktoré prezentujeme v tabuľke 5.11. Z tohto porovnania vyplýva, že 10-ročná dátová sada s návratom k najlepšej epoche dosahuje najlepšie výsledky, čo potvrdzuje záver z predošlých zistení. Druhé miesto obsadili modely trénované pomocou 10-ročnej dátovej sady bez návratu k najlepšej epoche. Toto porovnanie je obzvlášť zaujímavé z dôvodu, že modely, ktoré používali 20-ročnú dátovú sadu s návratom k najlepšiemu výsledku, dosahovali o niečo horšie výsledky ako modely používajúce 10-ročnú dátovú sadu bez návratu k najlepšej epoche, ale modely, ktoré používali 20-ročnú dátovú sadu bez návratu k najlepšej epoche, dosahujú výrazne horšie výsledky ako ostatné modely. Z tohto vyplýva, že optimalizácia trénovania návratom k najlepšej epoche výrazne zlepšuje dosiahnuté validačné výsledky modelov.

Tabuľka 5.12: Porovnanie priemernej validačnej odchýlky podľa dátovej sady

Typ modelu	**Priemerná odchýlka (MSE)**	**Poradie**
Jedno-premenný	0.001527	1.
Viac-premenný	0.002714	2.

Posledným porovnaním, ktoré sme v tomto experimente vykonali, bolo porovnanie priemernej validačnej odchýlky obidvoch základných typov vytvorených modelov. Z výsledkov uvedených v tabuľke 5.12 je jednoznačné, že jednopremenný model dosahuje lepšie výsledky ako viac-premenný model s nami zvolenými parametrami.

Kapitola 6

Záver

Na záver tejto práce by sme chceli zhodnotiť výsledky, ktoré sme dosiahli pri realizácii všetkých experimentov a taktiež ponúknuť niekoľko námetov budúceho experimentovania.

Prvým modelom, ktorý v tejto práci prezentujeme, je jedno-premenný, jednovrstvový model, ktorého princíp bol uvedený v referenčnej práci [12], pričom sa v prvom experimente snažíme replikovať výsledky z uvedeného článku. V druhom experimente analyzujeme kvalitu predikcií na nezávislom testovacom období po roku 2020, kde sa nám podarilo tento model zlepšiť zväčšením počtu LSTM jednotiek na 512 a zmenou normalizačnej metódy z tzv. **Z-Score** normalizácie na normalizáciu použitím **MaxAbsScalera** z knižnice **scikit-learn** [13].

V experimente tri sa venujeme tvorbe vlastného viac-premenného jedno-vrstvového modelu, kde vykonávame výber vhodných vstupných premenných na základe výsledkov z validačného obdobia prezentovaných kandidátnych modelov, pričom sme zistili, že najlepšie výsledky dosiahol model, kde vstupné premenné tvorila uzatváracia cena akcie, denný obchodovaný objem akcie a maximum, minimum a štandardná odchýlka vypočítaná na základe 30-dňového pohyblivého okna.

V štvrtom experimente bližšie analyzujeme výsledky obidvoch modelov v rámci jedného mesiaca, kde sme zistili, že napriek relatívne malej vzdialenosti medzi predikovanou a reálnou krivkou ceny akcie majú modely problém predikovať zmenu trendu ceny akcie a tá sa na predikciách prejavuje až v nasledujúci časový okamih. Taktiež v tomto experiemnte prezentujeme a vyhodnocujeme výsledky modelov na základe alternatívneho hodnotiaceho kritéria daného schopnosťou modelu správne určiť zmenu trendu akcie nasledujúceho dňa. Výsledky modelov vyhodnotené podľa tohto kritéria dosahujú nadpolovičnú úspešnosť pri predikcii testovacieho obdobia o dĺžke 499 dní, kde najlepší model dosahuje 55.9 % úspešnosť (279 správnych predikcií) aj napriek tomu, že nebol trénovaný pre toto kritérium.

V poslednom experimente sa venujeme hľadaniu optimálneho počtu trénovacích epoch, miery učenia α a porovnávame výsledky 480 modelov na základe dĺžky ich trénovacieho obdobia a typu. Taktiež prezentujeme jedno- a viac-premenné modely v dvojvrstvovej architektúre a porovnávame ich s a bez vylepšenia, ktoré si uloží váhy modelu z jeho najlepšej epochy na základe validačnej odchýlky. Výsledky tohto experimentu preukázali, že jedno-premenné jedno-vrstvové modely trénované za použitia 10-ročnej dátovej sady s návratom k najlepšej epoche dosahujú najnižšiu priemernú validačnú odchýlkyu s tým, že porovnanie prezentovaných konfigurácií jednotlivých akcií preukázalo že modely akcie BRK-A dosahujú najnižšie priemerné validačné odchýlky, nasledovaných s menším zhoršením modely akcie GOOG, ale modely akcie MSFT preukázali výrazné priemerné zhoršenie. Všeobecné porovnanie modelov na základe ich typu dokázalo, že jedno-premenné modely dosahujú vo

všeobecnosti lepšie priemerné validačné odchýlky ako viac-premenné modely, pričom odhadujeme, že toto je hlavne spôsobené horšími výsledkami viacpremenných dvoj-vrstvových modelov.

Počas realizácie tejto práce sme mali niekoľko nápadov, ktoré sme sa rozhodli predložiť ako námety budúcej práce. Prvým takýmto námetom je zmena princípu predikcie uzatváracích cien akcií na predikciu medzidňovej návratnosti. Táto zmena by mala pomôcť v zabezpečení kvalitnejších výsledkov predikcií, keďže medzidňové návratnosti by mali byť stacionárnejšie dáta ako uzatváracie ceny, a teda ľahšie predikovateľné. Ďalším námetom je použitie nami prezentovaného alternatívneho kritéria vyhodnocovania predikcí modelov z kapitoly 5.4, pričom by sa mohli použiť vhodné modely pre binárnu klasifikáciu. Posledným nápadom pre budúce experimenty, ktorý chceme predložiť je otestovanie kvality predikcií viac-premenného modelu s inými vstupnými premennými, pričom by zväčšenie robustnosti tohto modelu mohlo zlepšiť kvalitu jeho predikcií.

Na záver tejto práce by sme chceli zhodnotiť vytvorené modely z praktického hľadiska. Z uvedených výsledkov nám vyplýva, že rozdiel medzi kvalitou modelov jednotlivých akcií môže byť veľký, a preto je správny výber akcie pred investíciou kľúčový. Ak porovnávame dosiahnuté výsledky modelov na základe ich validačnej či testovacej odchýlky, tak by sme vo všeobecnosti odporúčali z nami vytvorených modelov, modely akcie BRK-A, ktorej odchýlky boli najnižšie a neodporučili by sme modely akcie MSFT, keďže dosahovali najhoršie odchýlky. Čo sa týka modelov akcie GOOG, tak tie dosahovali o niečo horšie výsledky ako modely akcie BRK-A, ale značne lepšie výsledky ako modely akcie MSFT, a preto ich vhodnosť nechávame na zvážení čitateľa. V prípade praktického použitia modelov na predikciu zmeny trendu ceny akcie by sme taktiež odporučili modely vytvorené pre akciu BRK-A, keďže dosahovali najlepšie výsledky s primernou úspešnosťou 55.9 % na období dvoch rokov a najvyššou mesačnou úspešnosťou 72.7 %, ale vo všeobecnosti by sme navrhovali zamerať sa na optimalizáciu týchto modelov uvedenú v námetoch budúcej práce. V neposlednom rade odporučujeme zvážiť použitie modelov trénovaných pomocou kratších dátových sád, vzhľadom na to, že modely, ktoré boli trénované za použitia 10-ročnej dátovej sady dosahovali lepšie výsledky ako tie, ktoré boli trénované na základe 20-ročnej dátovej sady.

Literatúra

[1] Brabazon, A. a O'Neill, M. *Biologically Inspired Algorithms for Financial Modelling (Natural Computing Series)*. 1. vyd. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 3540262520.

[2] Butler, R. A. *How to Evaluate Stock Performance* [online]. 30. September 2022 [cit. 2023-03-24]. Dostupné z: https://www.investopedia.com/articles/investing/011416/how-evaluate-stock-performance.asp.

[3] Dolphin, R. *LSTM Networks | A Detailed Explanation* [online]. 21. Október 2020 [cit. 2023-04-17]. Dostupné z: https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9.

[4] Encyclopaedia Britannica, T. E. of. *Regression*. 18. November 2022 [cit. 2023-3-22]. Dostupné z: https://www.britannica.com/topic/regression-statistics.

[5] Ganie, I. R., Wani, T. A. a Yadav, M. P. Impact of COVID-19 Outbreak on the Stock Market: An Evidence from Select Economies. *Business Perspectives and Research*. 0, zv. 0, č. 0, s. 22785337211073635. DOI: 10.1177/22785337211073635. Dostupné z: https://doi.org/10.1177/22785337211073635.

[6] Ganti, A. *Adjusted Closing Price* [online]. 28. December 2020 [cit. 2023-03-24]. Dostupné z: https://www.investopedia.com/terms/a/adjusted_closing_price.asp.

[7] Gholamy, A., Kreinovich, V. a Kosheleva, O. Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation. In:. 2018.

[8] Hayes, A. *Volatility: Meaning In Finance and How it Works with Stocks* [online]. 23. August 2022 [cit. 2023-03-24]. Dostupné z: https://www.investopedia.com/terms/v/volatility.asp.

[9] Khuong, B. *The Basics of Recurrent Neural Networks (RNNs)* [online]. 24. Jún 2019 [cit. 2023-04-15]. Dostupné z: https://pub.towardsai.net/whirlwind-tour-of-rnns-a11effb7808f.

[10] Koehrsen, W. *Recurrent Neural Networks by Example in Python* [online]. 05. November 2018 [cit. 2023-04-10]. Dostupné z: https://towardsdatascience.com/recurrent-neural-networks-by-example-in-python-ffd204f99470.

[11] Mehtab, S., Sen, J. a Dutta, A. Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models. In: Thampi, S. M., Piramuthu, S., Li, K.-C., Berretti, S., Wozniak, M. et al., ed. *Machine Learning and Metaheuristics Algorithms, and Applications*. Singapore: Springer Singapore, 2021, s. 88–106. ISBN 978-981-16-0419-5.

[12] Mohanty, S., Vijay, A. a Gopakumar, N. StockBot: Using LSTMs to Predict Stock Prices. *Journal of Banking and Financial Technology*. 1. vyd. Júl 2022. DOI: 10.48550/arXiv.2207.06605.

[13] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011, zv. 12, s. 2825–2830.

[14] Rasheed, R. *Why Does Stationarity Matter in Time Series Analysis?* [online]. 11. Júl 2020 [cit. 2023-04-20]. Dostupné z: https://towardsdatascience.com/why-does-stationarity-matter-in-time-series-analysis-e2fb7be74454.

[15] Rogers, K. *Scientific modeling*. 21. Máj 2012 [cit. 2023-3-22]. Dostupné z: https://www.britannica.com/science/scientific-modeling.

[16] Rumelhart, D. E., Hinton, G. E. a Williams, R. J. Learning representations by back-propagating errors. *Nature*. Oct 1986, zv. 323, č. 6088, s. 533–536. DOI: 10.1038/323533a0. ISSN 1476-4687. Dostupné z: https://doi.org/10.1038/323533a0.

[17] Sayavong, L., Wu, Z. a Chalita, S. Research on Stock Price Prediction Method Based on Convolutional Neural Network. In: *2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*. 2019, s. 173–176. DOI: 10.1109/ICVRIS.2019.00050.

[18] Shetty, C. *Time Series Models (AR, MA, ARMA, ARIMA)* [online]. 22. September 2020 [cit. 2023-03-24]. Dostupné z: https://towardsdatascience.com/time-series-models-d9266f8ac7b0.

[19] Siami Namini, S., Tavakoli, N. a Siami Namin, A. A Comparison of ARIMA and LSTM in Forecasting Time Series. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2018, s. 1394–1401. DOI: 10.1109/ICMLA.2018.00227.

[20] Trevisan, V. *Comparing Robustness of MAE, MSE and RMSE* [online]. 11. Január 2022 [cit. 2023-04-04]. Dostupné z: https://towardsdatascience.com/comparing-robustness-of-mae-mse-and-rmse-6d69da870828.

[21] Twomey, J. a Smith, A. Validation and Verification. *Artificial Neural Networks for Civil Engineers: Fundamentals and Applications*. September 1999.

[22] Wang, Y. a Guo, Y. Forecasting method of stock market volatility in time series data based on mixed model of ARIMA and XGBoost. *China Communications*. 2020, zv. 17, č. 3, s. 205–221. DOI: 10.23919/JCC.2020.03.017.

[23] Yiu, T. *Understanding ARIMA (Time Series Modeling)* [online]. 26. Apríl 2020 [cit. 2023-03-28]. Dostupné z: https://towardsdatascience.com/understanding-arima-time-series-modeling-d99cd11be3f8.

Prílohy

Príloha A

Popis a stromová štruktúra odovzdanej bakalárskej práce.

```
koreňový priečinok
├── docs/                ... Zložka s bakalárskou prácou
│                            a jej zdrojovými súbormi.
│   ├── bib-styles/      ... Štýly pre zoznam literatúry.
│   ├── obrazky-figures/ ... Obrázky použité v bakalárskej práci.
│   ├── template-fig/    ... Logá VUT.
│   ├── Makefile         ... Súbor pre preloženie zdrojových
│                            súborov bakalárskej práce do
│                            výsledného pdf súboru.
│   ├── fitthesis.cls    ... Základné LaTeX balíčky bakalárskej práce.
│   ├── zadani.pdf       ... Zadanie bakalárskej práce.
│   ├── xknazo01.pdf     ... Súbor s bakalárskou prácou.
│   ├── xknazo01.tex     ... Zdrojový súbor bakalárskej práce.
│   ├── xknazo01-kapi.tex ... Súbor s kapitolami práce.
│   ├── xknazo01-lite.bib ... Súbor použitej literatúry.
│   └── xknazo01-pril.tex ... Súbor s prílohami.
├── experiment-results/  ... Výsledky všetkých
│                            experimentov(.csv, obrázky).
│   ├── exp-1-2/         ... Výsledky experimentu 1 a 2.
│   ├── exp-3/           ... Výsledky experimentu 3.
│   ├── exp-4/           ... Výsledky experimentu 4.
│   └── exp-5/           ... Výsledky experimentu 5.
├── models/              ... Priečinok na ukladanie
│                            natrénovaných modelov(.h5).
│   ├── exp-1-2/         ... Modely z experimentu 1 a 2.
│   ├── exp-3/           ... Modely z experimentu 3.
│   ├── exp-4/           ... Modely z experimentu 4.
│   └── exp-5/           ... Modely z experimentu 5.
├── src/                 ... Zložka so zdrojovými
│                            súbormi(.py).
├── stocks/              ... Priečinok na ukladanie stiahnutých
│                            finančných dát(.csv).
├── experiment-1-2.py    ... Skript na trénovanie a uloženie
│                            výsledkov prvého a druhého experimentu.
├── experiment-3.py      ... Skript na trénovanie a uloženie
│                            výsledkov tretieho experimentu.
├── experiment-4.py      ... Skript na trénovanie a uloženie
│                            výsledkov štvrtého experimentu.
├── experiment-5.py      ... Skript na trénovanie a uloženie
│                            výsledkov piateho experimentu.
└── vizualizer.py        ... Skript na vytvorenie prezentovaných grafov
                             a analýzu výsledkov experimentov.
```

Príloha B

V tejto prílohe uvedieme presný postup na spustenie experimentov prezentovaných v tejto bakalárskej práci, požadované knižnice a spôsob generovania výsledných grafov.

Tabuľka 1: Vyžadované balíčky s verziami použitými pri tvorení práce

Baličeк	**Verzia**
cairosvg	2.6.0
cudnn	8.1.0.77
cudatoolkit	11.2.2
matplotlib	3.5.3
numpy	1.21.5
pandas	1.4.4
pandas_market_calendars	4.1.2
python	3.9
scipy	1.9.3
scikit-learn	1.1.3
seaborn	0.12.2
tensorflow	2.1
yfinance	0.2.18

Prvým krokom je získanie všetkých balíčkov uvedených v tabuľke 1 a všetkých závilostí, ktoré tieto balíčky vyžadujú. Pri každom z týchto balíčkov uvádzame presnú verziu, ktorú sme použili počas tvorby tejto práce. Pre preloženie zdrojových súborov bakalárskej práce je taktiež potrebné prostredie schopné prekladu súborov LaTeX. Dôležité je taktiež poznamenať, že inštalácia balíčku tensorflow, nie je zrovna jednoduchá a v prípade, že sa budete snažiť o trénovanie modelov za použitia grafickej karty, tak je možné, že budete potrebovať iné verzie balíčkov cudnn a cudatoolkit.

V prípade, že vaše prostredie splňuje všetky vyššie uvedené prerekvizity, tak v prípade, že chcete trénovať modely spomínané v experimente 1 až 5, tak stačí spustiť súbor patričného experimentu (napr. pre experiment 3.: `python3 experiment-3.py`), pričom natrénované modely budú uložené v zložke `models/` v podadresári patričného experimentu vo forme `.h5` súboru a výsledky týchto experimentov budú uložené v zložke `experiment-results/` a patričnom podadresári daného experimentu vo formáte `.csv` či obrázku. V prípade, že by ste chceli meniť parametre trénovania, tak môžete zmeniť predvolené parametre v súbore `src/constants.py`, ale je taktiež potrebné overiť, že konkrétny experiment používa predvolené parametre, keďže niektoré experimenty majú špecifické parametre, ktoré sú definované vždy na vrchu súboru daného experimentu.

V prípade, že nechcete trénovať modely, tak môžete z odovzdaných `.csv` súborov vygenerovať grafy prezentované v tejto práci nasledovne: `python3 vizualizer.py`. Tento príkaz vám vygeneruje grafy všetkých experimentov v tejto práci, pričom ak chcete generovať grafy len niektorých experimentov, tak je nutné zakomentovať volania patričných metód experimentov pomenovaných podľa rovnakej konvencie na konci tohto súboru. Vygenerované obrázky môžete následne nájsť v zložke `experiment-results/` a štatisticky potrebné k vytvoreniu grafov prezentovaných v tejto práci budú vytlačené na štandardný výstup programu.