



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Richard Savčinský

**Echo state siete a ich využitie na
predpovedanie časových radov**

Katedra softwaru a výuky informatiky

Vedoucí bakalářské práce: RNDr. František Mráz, CSc.

Studijní program: Informatika

Studijní obor: Programování a softwarové systémy

Praha 2018

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Rád by som poďakoval vedúcemu práce RNDr. Františkovi Mrázovi, CSc. za jeho trpezlivosť, za poskytovanie cenných rád a pripomienok a za ochotu s čímkoľvek pomôcť.

Název práce: Echo state siete a ich využitie na predpovedanie časových radov

Autor: Richard Savčinský

Katedra: Katedra softwaru a výuky informatiky

Vedoucí bakalářské práce: RNDr. František Mráz, CSc., Katedra softwaru a výuky informatiky

Abstrakt: Rekurentné neurónové siete (RNN) umožňujú modelovať dynamické systémy s premenlivou dĺžkou vstupu. Ich nevýhoda je v náročnom učení, teda ťažkom nastavovaní váh neurónov, ktoré sú v sieti spojené. Echo state siete (ESN) sú zvláštnym typom RNN, ktoré sa naopak dajú učiť relatívne jednoducho. Majú rezervoár neurónov, ktorých stav odráža históriu všetkých signálov v sieti, a preto je tento typ sietí vhodný na simuláciu a predikciu časových postupností. Dosiahnutie maximálneho výkonu ESN vyžaduje veľmi precízne nastavenie a experimentovanie. Preto sme v tejto práci vytvorili nástroj pre konštruovanie a testovanie takýchto sietí. Pre účely vyskúšania vytvoreného nástroja sme implementovali úlohu predpovedania vývoja časových radov. Konkrétne sme sa zamerali na predpovedanie vývoja cien akcií, čo predstavuje veľmi neistú a pre presnú predpoveď veľmi náročnú oblasť. Výsledky našich experimentov sme porovnali s výsledkami iných nástrojov, a zistili sme, že náš nástroj dáva porovnateľné výsledky.

Klíčová slova: neurónové siete, echo state siete, časové rady, finančné prognózy

Title: Echo state networks and their application in time series prediction

Author: Richard Savčinský

Department: Department of Software and Computer Science Education

Supervisor: RNDr. František Mráz, CSc., Department of Software and Computer Science Education

Abstract: Recurrent neural networks (RNN) enable to model dynamical systems with variable input length. Their disadvantage is in inherently difficult training which means adjusting weights of connections between neurons connected in the network. Echo state networks (ESN) are a special type of RNN which are by contrast trainable rather simply. They include a reservoir of neurons whose state reflect the history of all signals in the network and that is why this type of network is suitable for simulation and prediction of time series. To maximize the computational power of ESN, very precise adjusting and experimenting are required. Because of that, we have created a tool for building and testing such networks. We have implemented a time series forecasting task for the purpose of examination of our tool. We have focused on stock price prediction, which represents an uncertain and complicated area for achieving precise results in. We have compared our tool to other tools and it was comparably successful.

Keywords: neural networks, echo state networks, time series, financial forecasting

Obsah

Úvod	3
1 Neurónové siete	4
1.1 Úvod do neurónových sietí	4
1.1.1 Umelé neurónové siete	4
1.1.2 Rekurentné neurónové siete	4
1.2 Echo state siete	5
1.2.1 Základný princíp	5
1.2.2 Vstup, rezervoár, výstup	6
1.2.3 Echo state property	8
1.2.4 Porovnávanie sietí	8
1.3 Parametre	9
1.3.1 Veľkosť rezervoáru	9
1.3.2 Škálovanie vstupu	9
1.3.3 Spektrálny polomer	9
1.3.4 Miera úniku	10
1.3.5 Výber parametrov	10
1.4 Existujúce implementácie v oblasti ESN	11
2 Nástroj pre prácu s ESN	12
2.1 Dôvody pre vznik	12
2.2 Ciele a požiadavky	12
2.3 Výber technológií	13
2.4 Uživatelská dokumentácia	14
2.4.1 Požiadavky a inštalácia	14
2.4.2 Použitie programu	15
2.4.3 Základný popis prostredia	15
2.4.4 Hlavný formulár	15
2.4.5 Vytvorenie novej ESN	16
2.4.6 Vstupná vrstva	16
2.4.7 Rezervoár	17
2.4.8 Výstupná vrstva	18
2.4.9 I/O	19
2.4.10 Task Center	21
2.4.11 Grid Search	23
2.4.12 Map & Profit	24
2.4.13 Ovládacie prvky	26
2.4.14 Formát súboru	28
2.4.15 Príklad použitia aplikácie	28
2.4.16 Záver	30
3 Finančné prognózy a ESN	33
3.1 Základná myšlienka a kľúčové pojmy	33
3.2 Finančná prognóza a ESN	33
3.3 Existujúce práce	34

3.4	Porovnávanie kvality predpovede	36
3.5	Záver	38
4	Aplikácia nástroja pre ESN	39
4.1	Riešené úlohy	39
4.2	Vývojová dokumentácia	39
4.2.1	Iteration	40
4.2.2	Initialization	40
4.2.3	Training	40
4.2.4	Testing	40
4.2.5	CreateNewESNForm	41
4.2.6	CreateNewESNResultForm.cs	42
4.2.7	EsnMath	44
5	Experimenty	45
5.1	Vstupné dáta	45
5.2	Priebeh experimentov	46
5.2.1	Krátkodobá predpoveď	50
5.2.2	Dlhodobá predpoveď	51
5.2.3	Stocker	52
5.2.4	ARMA	53
5.3	Analýza výsledkov	54
	Záver	58
	Zoznam použitej literatúry	60
	Zoznam obrázkov	62
	Zoznam tabuliek	63
A	Prílohy	64
A.1	Prvá príloha	64

Úvod

Neurónové siete sú aktívnou oblasťou výskumu už mnoho rokov. V dnešnej dobe sú modely neurónových sietí založené na pevných matematických základoch. Sľubné výsledky a užitočné vlastnosti poskytuje skupina umelých neurónových sietí nazývaná *rekurentné neurónové siete*. Hlavným problémom rekurentných neurónových sietí je však náročný proces ich tréovania. Pri tréovaní siete na základe postupnosti vstupov a známej postupnosti výstupov sa snažíme nastaviť parametre siete tak, aby sa naučila závislosť výstupov na vstupoch. Natrénovaná sieť sa potom môže použiť na nové vstupy a generovať príslušné výstupy.

Jedným z modelov umelých neurónových sietí, ktoré sa pokúsili tento problém vyriešiť, je model *echo state siete* prvý krát predstavený Herbertom Jaegerom (2001) [8]. Echo state siete sú schopné riešiť široké spektrum úloh. Takou je aj úloha naučiť sa aproximovať vývoj časových radov.

Preto sme sa rozhodli vytvoriť nástroj, ktorý bude umožňovať s týmito sieťami pracovať. Vytvorený nástroj zároveň použijeme na vzorovú úlohu, ktorou je predpovedanie vývoja cien akcií na burze.

Keďže finančné trhy sú charakterizované neistotou, ktorá súvisí s budúcim vývojom ekonomiky a firiem spolu s ďalšími faktormi, schopnosť predpovedať aspoň do určitej miery budúci vývoj je veľmi náročnou úlohou. Cieľom tejto práce bude teda aj preskúmať možnosti použitia echo state sietí na aproximáciu budúceho vývoja cien akcií, a porovnanie výsledkov našej práce s existujúcimi výsledkami v tejto oblasti.

Práca je štrukturovaná nasledovne. V kapitole 1 si predstavíme bližšie echo state siete, z čoho sa sieť skladá, aké má parametre, vysvetlíme princíp výpočtu a matematickú podstatu fungovania týchto sietí. V tejto kapitole nahliadneme aj do problematiky nastavovania hodnôt parametrov siete. V kapitole 2 sa pozrieme na nami vytvorené prostredie pre prácu so sieťami, z čoho pozostáva a čo ponúka. V kapitole 3 uvádzame echo state siete v kontexte finančných prognóz a predpovedania vývoja cien akcií. Kapitola 4 obsahuje vývojovú dokumentáciu a popis fungovania našej aplikácie. Kapitola 5 patrí experimentom, to znamená, že sú v nej prezentované výsledky a analýzy predpovedí našej aplikácie. Dosiahnuté výsledky sú taktiež porovnané s výsledkami iných prác a nástrojov na predpoveď vývoja cien akcií. Nakoniec v závere sú zhrnuté výsledky práce a diskutované jej ďalšie možné rozšírenia. V prílohe sa nachádza vytvorený program.

1. Neurónové siete

V úvodnej kapitole celej práce si veľmi stručne predstavíme neurónové siete. Ďalej sa pozrieme na konkrétne typy neurónových sietí, ktoré sú pre našu prácu dôležité, pozrieme sa na ich charakteristiku a v čom sa líšia. Postupne sa prepracujeme až ku echo state sieťam, ktoré sú hlavným predmetom tejto práce.

1.1 Úvod do neurónových sietí

Ako napovedá názov, v tejto sekcii sa v krátkosti uvedieme do problematiky neurónových sietí. Predstavíme si umelé neurónové siete a ich rozdelenie podľa princípu fungovania. Zameriame sa potom bližšie na rekurentné neurónové siete.

1.1.1 Umelé neurónové siete

Umelé neurónové siete (Artificial neural networks (ANNs)) sú výpočetné systémy inšpirované biologickými neurónovými sieťami, ktoré tvoria mozog zvierat. Takéto systémy sa učia (progresívne sa zlepšujú) vykonávať úlohy podľa predložených vzorov, všeobecne bez špecifického programovania pre danú úlohu.

ANN je založená na skupine spojených jednotiek nazývaných umelé neuróny (analogicky k neurónom spojeným axónmi v biologickom mozgu). Každý spoj medzi dvomi neurónmi prenáša signál od prvého neurónu k druhému neurónu, pričom daný spoj prenášaný signál zosilňuje alebo zoslabuje tým, že ho násobí váhou spoja. Existujú dva hlavné typy neurónových sietí: feedforward (dopredné) a rekurentné.

V dopredných neurónových sieťach (Feedforward neural network (FNN)) sú signály propagované zo vstupných jednotiek do výstupných bez existencie cyklov medzi jednotkami. Matematicky, FNN implementujú statické vstupno-výstupné funkcie na rozdiel od rekurentných sietí (viz nižšie), ktoré implementujú dynamické systémy.

1.1.2 Rekurentné neurónové siete

Rekurentná neurónová sieť (Recurrent neural network (RNN)) je akákoľvek sieť, ktorej neuróny si môžu navzájom posielat signály ako spätnú väzbu, teda kde spojenia medzi neurónmi vytvárajú orientované cykly. Tým je umožnené dynamické správanie v čase. Na rozdiel od dopredných neurónových sietí, RNN môže používať svoju vnútornú pamäť na spracovanie ľubovoľne dlhej postupnosti vstupov. Vďaka tomu sú aplikovateľné na široké spektrum úloh. Poznáme dve základné triedy „učenia“ RNN s učiteľom a bez učiteľa.

Pri učení (tréňovaní) RNN s učiteľom začneme s tréningovými dátami, ktoré reprezentujú vstup do siete a zároveň požadovaný výstup siete. Tréningové dáta sú použité na nastavenie váh siete tak, aby na tréningový vstup sieť viac-menej presne reprodukovala tréningový výstup, dúfajúc že sa RNN následne zovšeobecní aj na nové vstupy.

1.2 Echo state siete

Echo state siete (angl. Echo state networks (ESNs)) sú špeciálnym typom rekurentných neurónových sietí, ktorý poskytuje architektúru a princíp učenia s učiteľom. Hlavná myšlienka je inicializovať náhodnú, veľkú, pevne zvolenú RNN vstupným signálom, teda vyvolať v každom neuróne tejto rezervoárovej siete (v originále reservoir network) nelineárny signál ako odpoveď, a potom spočítať požadovaný výstupný signál ako trénovateľnú lineárnu kombináciu všetkých týchto signálov.

Rezervoár má niekoľko funkcií. Slúži jednak ako nelineárne rozšírenie vstupného signálu, teda vstupné dáta, ktoré nie sú lineárne separovateľné v pôvodnom systéme, sa takými stanú často v rozšírenom systéme. Zároveň však rezervoár plní funkciu pamäte ponúkajúcej časovo obmedzený kontext. Práve to je kľúčový argument pre používanie RNN. V nasledujúcich oddieloch popíšeme fungovanie echo state sietí podrobnejšie.

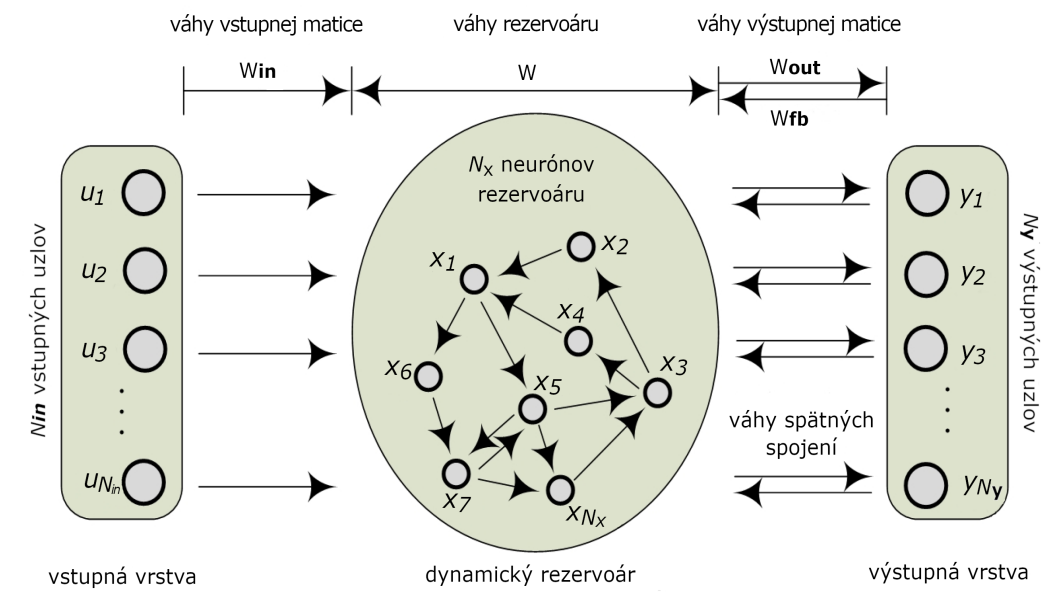
1.2.1 Základný princíp

Teraz popíšeme neformálne model ESN, ktorý je schématicky zobrazený na obr. 1.1. V nasledujúcich sekciách jeho časti popíšeme exaktne. Ešte si však dovoľme menšiu odbočku. Trend, ktorý začali ESN, sa nazýva Reservoir Computing (RC). RC je v súčasnosti plodnou oblasťou výskumu dávajúcou dôležitý pohľad do RNN. RC v dnešnej dobe pokrýva viacero metód a rozšírení pôvodnej myšlienky, avšak originály ESN prístup má stále čo ponúknuť pre jeho jednoduchosť a silu. Originálna metóda RC predstavená ESN bola:

Vstup: Postupnosť vstupných vektorov $\{\mathbf{u}(n)\}_{n=1}^m$, kde $\mathbf{u}(n) \in \mathbb{R}^{N_{in}}$, N_{in} je dĺžka vstupných vektorov, m je dĺžka vstupnej sekvencie.

Výstup: Postupnosť výstupných vektorov $\{\mathbf{y}^{target}(n)\}_{n=1}^m$, $\mathbf{y}^{target}(n) \in \mathbb{R}^{N_y}$, kde N_y je dĺžka výstupných vektorov, m je dĺžka výstupnej sekvencie.

1. Generovať veľký náhodný rezervoár RNN, váhy spojenia medzi neurónmi sú určené náhodne generovanou maticou $\mathbf{W} \in \mathbb{R}^{N_x \times N_x}$, kde N_x je počet neurónov v rezervoári.
2. Postupne vkladať vstupné vektory $\mathbf{u}(1), \mathbf{u}(2), \dots$ do rezervoáru a uložiť odpovedajúce aktivácie $\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(N_{in})$ na neurónoch po jednotlivých krokoch, teda stavy rezervoáru, do matice \mathbf{X} .
3. Spočítať výstupnú maticu $\mathbf{W}_{out} \in \mathbb{R}^{N_y \times (1+N_u+N_x)}$ váh spojov z rezervoáru na základe \mathbf{X} a \mathbf{y}^{target} s použitím lineárnej regresie. Cieľom je naučiť model s výstupom $\mathbf{y}(n) = \mathbf{W}_{out}\mathbf{x}(n)$ kde $\mathbf{y}(n) \in \mathbb{R}^{N_y}$ tak, aby $\{\mathbf{y}(n)\}_{n=1}^m$ čo najlepšie odpovedalo požadovanému výstupu $\{\mathbf{y}^{target}(n)\}_{n=1}^m$, teda minimalizovať rozdiel medzi \mathbf{y} a \mathbf{y}^{target} . Ešte dôležitejšie je však, aby sa model dobre zobecnil na nové dáta.
4. Použiť natrénovanú sieť na nový vstup $\mathbf{u}(n)$ počítajúc $\mathbf{y}(n)$ s použitím spočítanej výstupnej matice \mathbf{W}_{out} z aktivácií neurónov v rezervoári.



Obr. 1.1: ESN schéma [10].

1.2.2 Vstup, rezervoár, výstup

V tejto sekcii si podrobnejšie popíšeme jednotlivé časti výpočtu ESN modelu. Vstupná sekvencia slúži ako pohon pre rezervoár. Môže byť viacdimenzionálna. Vstup je pred podaním do rezervoáru vynásobený náhodne generovanou maticou $\mathbf{W}_{in} \in \mathbb{R}^{N_x \times (N_{in}+1)}$. Stav neurónov $\mathbf{x}(n)$ je aktualizovaný typicky podľa rovnice

$$\tilde{\mathbf{x}}(n) = \tanh(\mathbf{W}_{in}[1; \mathbf{u}(n)] + \mathbf{W}\mathbf{x}(n-1)), \quad (1.1)$$

$$\mathbf{x}(n) = (1 - \alpha)\mathbf{x}(n-1) + \alpha\tilde{\mathbf{x}}(n), \quad (1.2)$$

kde

- $\mathbf{x}(n) \in \mathbb{R}^{N_x}$ je vektor aktivácií na neurónoch v rezervoári,
- $\tilde{\mathbf{x}}(n) \in \mathbb{R}^{N_x}$ je vektor aktualizácií v kroku n ,
- $\tanh(\cdot)$ – prenosová funkcia; hyperbolický tangens je najbežnejšou voľbou pre prenosovú funkciu, i keď môže byť použitá aj iná sigmoidová funkcia,
- $[\cdot; \cdot]$ je vertikálne zretazenie matíc,
- $\mathbf{W}_{in} \in \mathbb{R}^{N_x \times (1+N_u)}$ je vstupná matica,
- $\mathbf{W} \in \mathbb{R}^{N_x \times N_x}$ je rekurentná váhová matica (váh spojov v rezervoári),
- $\alpha \in (0,1]$ je miera úniku.

Výstup je definovaný rovnicou

$$\mathbf{y}(n) = \mathbf{W}_{out}[1; \mathbf{u}(n); \mathbf{x}(n)], \quad (1.3)$$

kde $\mathbf{y}(n) \in \mathbb{R}^{N_y}$ je výstup siete v kroku n , $\mathbf{W}_{out} \in \mathbb{R}^{N_y \times (1+N_u+N_x)}$ je výstupná váhová matica, pre ktorú platí vzťah

$$\mathbf{Y}^{target} = \mathbf{W}_{out} \mathbf{X}^{-1}, \quad (1.4)$$

resp.

$$\mathbf{W}_{out} = \mathbf{Y}^{target} \mathbf{X}^T (\mathbf{X} \mathbf{X}^T + \beta \mathbf{I})^{-1}, \quad (1.5)$$

kde

- β je regularizačný koeficient,
- \mathbf{I} je jednotková matica.

Matica $\mathbf{Y}^{target} \in \mathbb{R}^{N_y \times m}$ je matica požadovaných výstupov, kde N_y je dĺžka výstupných vektorov a m je veľkosť tréningovej množiny. Definovať ju môžeme vzorcom:

$$\mathbf{Y}^{target} = \begin{pmatrix} \mathbf{y}_1(1) & \mathbf{y}_2(1) & \dots & \mathbf{y}_m(1) \\ \mathbf{y}_1(2) & \mathbf{y}_2(2) & \dots & \mathbf{y}_m(2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_1(N_y) & \mathbf{y}_2(N_y) & \dots & \mathbf{y}_m(N_y) \end{pmatrix}.$$

Matica \mathbf{X} predstavuje históriu aktivácií všetkých neurónov v každom kroku výpočtu. Nemusí však vždy existovať inverzná matica k matici \mathbf{X} , preto sa používa väčšinou (1.5), tzv. ridge regression. $\beta \mathbf{I}$ slúži ako regularizačná časť rovnice a používa sa jednak na zvýšenie numerickej stability pri inverzii \mathbf{X} , ale aj na zjemenie veľkých hodnôt \mathbf{W}_{out} . Maticu \mathbf{X} môžeme definovať nasledujúcim vzorcom:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1(1) & \mathbf{x}_2(1) & \dots & \mathbf{x}_m(1) \\ \mathbf{x}_1(2) & \mathbf{x}_2(2) & \dots & \mathbf{x}_m(2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_1(N_x) & \mathbf{x}_2(N_x) & \dots & \mathbf{x}_m(N_x) \end{pmatrix},$$

kde

- $\mathbf{x}_i(t)$ je aktivácia neurónu i v čase t ,
- N_x je počet neurónov v rezervoári,
- m je dĺžka vstupnej sekvencie.

Pri niektorých úlohách je vhodné do výpočtu pridať ďalšiu nelinearitu prenášaním výstupu naspäť do rezervoáru. V tomto prípade rovnicu (1.1) nahradíme rovnicou:

$$\tilde{\mathbf{x}}(n) = \tanh(\mathbf{W}_{in}[1; \mathbf{u}(n)] + \mathbf{W} \mathbf{x}(n-1) + \mathbf{W}_{fb} \mathbf{y}(n-1)), \quad (1.6)$$

kde $\mathbf{W}_{fb} \in \mathbb{R}^{N_x \times N_y}$ je matica spätných spojení (feedback connections). Pre generovanie \mathbf{W}_{fb} platia rovnaké princípy ako pre \mathbf{W}_{in} , bližšie popísané v sekcii 1.3.2.

1.2.3 Echo state property

Na to, aby mohol fungovať princíp ESN musí rezervoár spĺňať tzv. echo state property (ESP), ktorá spája asymptotické vlastnosti dynamiky rezervoáru a vstupný signál. Intuitívne, ESP znamená, že ak sa so sieťou pracuje dostatočne dlhý čas, tak aktuálny stav je jednoznačne daný históriou vstupov a výstupov, teda rezervoár sa zbavil akýchkoľvek informácií z počiatočného stavu. Inak povedané, pre každý vnútorný signál existuje echo funkcia, ktorá mapuje históriu vstupov/výstupov do aktuálneho stavu. Pre dosiahnutie ESP musí rezervoár spĺňať určité algebraické vlastnosti, dané parametrami siete. Najdôležitejším parametrom pre zaručenie ESP je spektrálny polomer váhovej matice \mathbf{W} (viz 1.3.3).

1.2.4 Porovnávanie sietí

Pri použití ESN potrebujeme kontrolovať, či sa sieť naučila počítat správne výstupy, a tiež chceme porovnávať hotové siete podľa toho, ako dobre riešia zadanú úlohu. Musíme si teda určiť ohodnotenie výslednej siete, ktoré bude dobre reprezentovať kvalitu predpovede, a potom na základe tejto hodnoty môžeme siete porovnávať. Základnou a najbežnejšou metrikou pre porovnávanie výsledných modelov je stredná kvadratická odchýlka (MSE). MSE je miera kvality predpovede echo state siete. Nadobúda len nezáporné hodnoty a platí, že menšie hodnoty MSE predstavujú kvalitnejšiu predpoveď.

$$MSE(\mathbf{y}, \mathbf{y}^{target}) = \frac{1}{mN_y} \sum_{k=1}^{N_y} \sum_{n=1}^m (\mathbf{y}_k(n) - \mathbf{y}_k^{target}(n))^2 \quad (1.7)$$

Ďalšími alternatívami sú RMSE (root-mean-square error) alebo NRMSE (normalised-RMSE)

$$RMSE(\mathbf{y}, \mathbf{y}^{target}) = \sqrt{MSE(\mathbf{y}, \mathbf{y}^{target})}, \quad (1.8)$$

$$NRMSE(\mathbf{y}, \mathbf{y}^{target}) = \frac{RMSE(\mathbf{y}, \mathbf{y}^{target})}{(\mathbf{y}_{max}^{target} - \mathbf{y}_{min}^{target})} \quad (1.9)$$

kde

- $\mathbf{y}_{max}^{target}$ je maximálna požadovaná výstupná hodnota daná vzťahom

$$\mathbf{y}_{max}^{target} = \max_{k=1, \dots, N_y} \max_{n=1, \dots, m} \mathbf{y}_k^{target}(n), \quad (1.10)$$

- $\mathbf{y}_{min}^{target}$ je maximálna požadovaná výstupná hodnota daná vzťahom

$$\mathbf{y}_{min}^{target} = \min_{k=1, \dots, N_y} \min_{n=1, \dots, m} \mathbf{y}_k^{target}(n). \quad (1.11)$$

Pri niektorých úlohách môže byť pre vyhodnotenie dôležitá aj hodnota smerodajnej odchýlky MSE, ktorá je daná vzťahom,

$$\sigma = \sqrt{\frac{1}{mN_y} \sum_{n=1}^m \sum_{k=1}^{N_y} (\mathbf{y}_k(n) - \mathbf{y}_k^{target}(n) - MSE(\mathbf{y}, \mathbf{y}^{target}))^2} \quad (1.12)$$

kde nižšia hodnota znamená stabilnejšiu predpoveď, bez výraznejších alebo extrémnych výkyvov.

1.3 Parametre

Rezervoár siete a jeho dynamika je určený trojicou \mathbf{W}_{in} , \mathbf{W} , α , (prípadne \mathbf{W}_{fb}) kde vstupná matica \mathbf{W}_{in} a váhová matica rekurentných spojov \mathbf{W} sú generované náhodne na základe ďalších parametrov a hodnota miery úniku α je samostatným, nezávislým parametrom. Hodnoty týchto parametrov majú rozhodujúci vplyv na výslednú sieť. Aj malé zmeny hodnôt môžu mať veľký dopad, a preto vyžadujú veľmi precízne nastavenie pre optimalizáciu. Úlohou optimalizácie je dosiahnuť vlastnosti rezervoáru, ktoré čo najvernejšie simulujú systém, ktorý sa snažíme modelovať. Parametre často označujeme ako „globálne“, čo vyplýva z podstaty ich funkcie. Ďalej popíšeme princípy, ako nastavovať vybrané parametre modelu, aby čo najlepšie fungoval. Avšak popisované postupy je treba chápať ako heuristiky a typická aplikácia ESN vyžaduje hľadanie a skúšanie vhodných kombinácií parametrov [13].

1.3.1 Veľkosť rezervoáru

Jedným z kľúčových parametrov je veľkosť rezervoáru N_x , teda počet neurónov. Všeobecné pravidlo hovorí, že čím väčší rezervoár, tým lepšie správanie. To znamená, že pre komplikovanejšie úlohy je dôležité použiť najväčší rezervoár, aký je možný si výpočetne dovoliť. Veľkosť rezervoáru zásadne ovplyvňuje výpočtovú náročnosť modelu. Keďže tréning a výpočet ESN nie sú až tak výpočtovo náročné ako iné RNN prístupy, bývajú bežné veľkosti rezervoárov aj 10^4 jednotiek. Dolná hranica veľkosti rezervoáru je podľa [13] určená odhadom počtu nezávislých hodnôt vstupu, ktoré si musí rezervoár zapamätať pre vyriešenie úlohy.

1.3.2 Škálovanie vstupu

Škálovanie vstupnej matice \mathbf{W}_{in} je ďalším kľúčovým parametrom pre optimalizáciu. Škálovaním typicky rozumieme nastavenie intervalu $[-a, a]$, z ktorého sú vyberané hodnoty \mathbf{W}_{in} . Okrem škálovania je dôležité aj to, z akého prevdepodobnostného rozdelenia budú hodnoty váh. Typicky pripadajú do úvahy rovnomerné rozdelenie alebo normálne (Gaussovo) rozdelenie, ale opäť záleží od konkrétnej úlohy, ktorú riešime. Hodnoty \mathbf{W}_{in} potom určujú, ako nelineárne sa bude rezervoár správať. Pre veľmi lineárne úlohy by mali byť hodnoty \mathbf{W}_{in} malé. Pre väčšie hodnoty \mathbf{W}_{in} sa aktivácie neurónov často dostanú bližšie k svojim extrémom a výrazne preskakujú medzi malými a veľkými hodnotami, teda správajú sa viac nelineárne.

1.3.3 Spektrálny polomer

Veľmi dôležitým parametrom pre optimalizáciu je spektrálny polomer, t.j. maximálna absolútna hodnota vlastného čísla váhovej matice rezervoára \mathbf{W} . Postup výpočtu je nasledujúci:

1. Najprv vygenerujeme náhodnú váhovú maticu \mathbf{W} .
2. Spočítame spektrálny polomer $\rho(\mathbf{W})$.
3. \mathbf{W} vydělíme $\rho(\mathbf{W})$ a dostaneme maticu so spektrálnym polomerom rovným jednej.
4. Nakoniec \mathbf{W} vynásobíme koeficientom, ktorý odpovedá finálnemu spektrálnemu polomeru.

Spektrálny polomer zohráva rozhodujúcu úlohu pri zaistení ESP. Veľké hodnoty $\rho(\mathbf{W})$ môžu viesť k vzniku viacerých pevných bodov, ktoré narúšajú ESP a vedú k periodickému alebo až chaotickému správaniu rezervoáru. Vo väčšine situácií $\rho(\mathbf{W}) \leq 1$ zaručuje ESP [13]. Často je však ESP platná aj pri $\rho(\mathbf{W}) \geq 1$. Hodnota spektrálneho polomeru efektívne ovplyvňuje, ako rýchlo sa vplyv vstupnej aktivácie vytratí z rezervoáru a ako stabilné sú tieto aktivácie. Pre úlohy, kde je potrebná rozsiahlejšia história vstupov, by mala byť hodnota $\rho(\mathbf{W})$ väčšia, a naopak pri úlohách závislých viac na nedávnych vstupoch menšia.

1.3.4 Miera úniku

Miera úniku (angl. *leak rate*) α vyjadruje rýchlosť dynamiky zmien rezervoáru diskretizovanú v čase. Inak povedané, jedná sa o množstvo signálu, ktoré je zahodené v každom neuróne, po každom kroku. Hodnota α je z intervalu $(0,1]$ a mala by odpovedať rýchlosti dynamiky úlohy, ktorú sa rezervoár modeluje. Je ťažké odhadnúť ideálnu hodnotu, preto je často potrebné hodnotu α nastaviť metódou pokus-omyl. Na α je možné pozeráť sa aj ako na exponenciálne zjemňovanie signálu na každom neuróne. V niektorých prípadoch je možné citelne zväčšiť pamäť ESN nastavením malej hodnoty α , teda požadovať pomalú dynamiku zmien.

1.3.5 Výber parametrov

Tri hlavné parametre uvedené vyššie majú najväčší vplyv na výslednú sieť a preto majú najväčšiu prioritu optimalizácie. Jednou z hlavných výhod ESN je, že učenie výstupov je rýchle. To by malo byť využité pri vyhodnocovaní, ako kvalitná je výsledná ESN generovaná určitou množinou konkrétnych hodnôt parametrov. Preto, najpraktickejší spôsob vyhodnotenia nastavenia parametrov je natrénovať výstup a spočítať chybu na trénovacej množine. Výber parametrov býva manuálny ako aj automatizovaný. Do určitej miery je manuálny výber parametrov nevyhnutný, avšak preferované sú samozrejme automatizované prístupy. Najpriamočiarejšou technikou ladenia parametrov je grid search (GS). Grid search je metóda prehľadávania hrubou silou, ktorá pozostáva z väčšieho množstva výpočtov na určitom rozsahu hodnôt parametrov. Pre jeden alebo viacej parametrov sa určí nejaká konečná množina možných hodnôt a potom sa pre tento parameter vyskúšajú všetky možné hodnoty, alebo, keď bolo parametrov viacej, tak všetky možné kombinácie hodnôt pre tieto parametre. Keďže ESN nemá veľké množstvo parametrov potrebných ladiť dôkladnejšie, je aj tak výpočetne náročná metóda ako GS, prakticky použiteľná. Navyše nejedná sa o zložitú metódu na implementovanie. Inou metódou pre optimalizovanie parametrov je metóda najväčšieho spádu (gradient descent method) [3].

1.4 Existujúce implementácie v oblasti ESN

Existuje niekoľko voľne dostupných implementácií ESN:

- Najkomplexnejším súborom nástrojov pre Reservoir Computing je OrGanic Environment for Reservoir computing (OGER) [25]. OGER je toolbox napísaný v Pythone pre rýchle zostavovanie, tréning a vyhodnocovanie učenia modelov na veľkých súboroch dát. Je založený na knižnici Modular toolkit for Data Processing (MDP), takže modifikácia parametrov a celková manipulácia prebieha na úrovni kódu. V čase písania práce už originálna verzia nie je dostupná, avšak implementácia s malými zmenami je dostupná na [20].
- Ďalším nástrojom je Aureservoir [6] napísaný Georgom Holzmannom na Graz University. Jedná sa o veľmi efektívnu implementáciu knižnice pre prácu s Echo State Networks v jazyku C++. K dispozícii je niekoľko algoritmov pre tréning a simuláciu. Výhodou by okrem efektivity mala byť aj jednoduchá rozširiteľnosť.
- Za zmienku určite stojí aj ESNJava1.0.4 – Java implementácia ESN od skupiny Cognitive Modeling Group (Univerzita Tübingen) [12]. Ponúka pokročilé nástroje na experimentovanie a vizualizáciu, výhodou je detailná dokumentácia a užívateľsky príjemné GUI.

Existuje ešte niekoľko simulátorov, demo programov a iných nástrojov takmer výlučne v jazykoch Matlab a Python najmä kvôli dobrej podpore matematických knižníc a jednoduchej vizualizácii. Celkovo však nie je v oblasti ESN veľmi široká programová základňa.

2. Nástroj pre prácu s ESN

V predchádzajúcej kapitole sme sa pozreli na existujúce nástroje v oblasti echo state sietí. Cieľom tejto kapitoly bude najprv načrtnúť dôvody, prečo je zaujímavé vytvoriť si vlastný nástroj, potom sa podrobnejšie pozrieť, čo je cieľom takéhoto programu, čo od neho očakávať a čo by mal splňovať. Sučasťou bude aj analýza procesu výberu konkrétnych technológií, popis vytvorenej aplikácie a ako ju použiť aj v podobe užívateľskej dokumentácie.

2.1 Dôvody pre vznik

Nástroje z predchádzajúcej kapitoly a celkovo v oblasti ESN spadajú až na výnimky do kategórie knižníc a toolboxov. To znamená, že užívateľ, ktorý chce s takýmito sieťami pracovať, má k dispozícii všetky potrebné funkcie, avšak musí pracovať s týmito funkciami na úrovni kódu. Preto by mohlo byť užitočné, vytvoriť aplikáciu, ktorá užívateľa kompletne odtieňuje od kódu. Ten dostane do rúk aplikáciu, v ktorej si veľmi jednoducho a rýchlo môže nastaviť potrebné parametre, pustiť výpočet a vyhodnotiť výsledok. Nástroj, ktorý sa o to do určitej miery pokúsil, je ESNJava1.0.4 spomínaný v predchádzajúcej kapitole. Ešte väčší dôraz by mohol byť kladený na intuitívnejšie užívateľské rozhranie, zamerať sa viac na experimentovanie, teda opakované skúšanie rôznych nastavení parametrov.

2.2 Ciele a požiadavky

Nie je možné vytvoriť nástroj, ktorý bude optimálny pre každú úlohu. Každý program je zameraný na niečo iné a s tým súvisia výhody a prednosti daného nástroja. Práve preto jedným z cieľov tejto práce bude vytvorenie programu pre prácu s ESN. Dôraz bude kladený na jednoduché ovládanie, nastavovanie parametrov, tréning a budovanie modelov, to všetko v reálnom čase bez straty rýchlosti. To znamená vytvoriť prakticky použiteľné GUI, ktoré bude združovať všetky potrebné nastavenia na jednom mieste prehľadne a jednoducho. Prostredníctvom tohto rozhrania by mal užívateľ získať možnosť akokoľvek meniť vlastnosti modelu, a tým pádom získať absolútnu kontrolu potrebnú pre prácu a experimentovanie s ESN. Cieľom je zároveň dať užívateľovi nástroj na okamžité porovnanie kvality výsledných modelov aplikovaním, či už rôznych existujúcich metrík alebo vizualizáciou. Čiže používateľ by mal byť schopný pre konkrétny problém s týmto nástrojom nájsť jednoducho a rýchlo optimálnu ESN ako riešenie. Výsledok tejto práce by mal pokrývať všetky základné prostriedky nevyhnutné pre používanie ESN avšak nie je možné obsiahnuť úplne všetko, preto bude kladený dôraz aj na to, aby prípadné rozšírenia funkcionality boli jednoducho doprogramovateľné do už existujúceho softvéru. S tým súvisí aj kvalitná dokumentácia a možnosť nápovedy nielen pre manipuláciu s programom, ale aj pre jednoduchšie pochopenie problematiky ESN. Ako už som popisoval vyššie, hlavnou devízou tohto projektu by oproti uvedeným, už existujúcim prácam mala byť intuitívnosť a ľahká ovládateľnosť pre užívateľa. Užívateľ však napriek tomu nestratí funkcionality potrebnú

pre experimentovanie. Práve naopak pre bežné úlohy by aj s ohľadom na ob-
stojnú rýchlosť a dôrazom kladeným práve na opakované skúšanie, modifikovanie
a testovanie sietí, mal byť tento program voľbou číslo jedna.

Výsledný softvér by mal spĺňať nasledujúce požiadavky:

- Jednoduché vytváranie modelov ESN v grafickom užívateľskom rozhraní (okenná aplikácia, prehľadné menu, logické usporiadanie nastavení).
- Manuálne nastavenie hodnôt parametrov modelu a automatické hľadanie optimálnych hodnôt parametrov (GS).
- Užívateľ bude mať možnosť opakovného púšťania väčšieho počtu výpočtov.
- Porovnávanie a vyhodnotenie výsledných sietí za pomoci metrik (číselných aj grafických).
- Uložená história experimentov v samostatnom súbore (log).
- Možnosť ukladať a načítať vytvorené modely na disk a z disku.
- Vyššie uvedené prvky ovládať pomocou okennej aplikácie.
- Okrem ovládania pomocou okennej aplikácie aj možnosť načítať konfigu-
račný súbor.
- Dokumentácia a popis ako súčasť programu pre uľahčenie práce s aplikáciou
(tzv. našepkávanie).
- Dobrý objektový návrh pre udržovateľnosť a rozširovateľnosť existujúceho
softvéru.

2.3 Výber technológií

V tejto časti ukážeme, aké technológie sme sa rozhodli použiť na vývoj pro-
stredia. Prostredie je implementované v jazyku C# s použitím technológií .NET.
Hlavným dôvodom výberu je autorova známosť jazyka a príslušných technológií.
Ďalším veľmi dôležitým faktorom pre výber je dobrá podpora vývoja požadova-
ného typu softvéru danými technológiami. Keď hovoríme o danom type softvéru
máme na mysli desktopovú aplikáciu s GUI. Rozhodovali sme sa medzi dvomi
hlavnými frameworkami a to konkrétne Windows Forms (WinForms) a Windows
Presentation Foundation (WPF).

Windows Forms [16] poskytuje platformu pre vývoj funkčných, stabilných, ino-
vatívnych a graficky bohatých aplikácií bez straty jednoduchosti. Práve tieto
vlastnosti sú pre túto bakalársku prácu kľúčové. Jedná sa o framework použí-
vajúci event-driven model, prostredie generuje odpovede na užívateľove akcie.

Výhody:

- Jednoduchosť.
- Rýchla tvorba aplikácií.

- Staršia technológia, tým pádom viac vyskúšaná a testovaná.
- Dobrá podpora vo Visual Studiu.

Viac informácií nájdeme na stránkach Microsoftu [16].

WPF [15] platforma poskytuje širokú škálu nástrojov pre vývoj aplikácií. WPF používa Extensible Application Markup Language (XAML). V bežnej aplikácii je na implementáciu vzhľadu použitý XAML a logika je programovaná v manažovanom programovacom jazyku.

Výhody:

- Široké spektrum možností, flexibilnejšie.
- Novšie, modernejšie.
- Separácia vzhľadu a logiky.

Podrobnejší popis frameworku nájdeme na stránkach Microsoftu [15].

Rozhodli sme sa pre Windows Forms, najmä z dôvodu jednoduchosti a intuitívnosti. Práca s ESN nevyžaduje žiadne neobvyklé alebo zložité prvky, z toho dôvodu sú prostriedky poskytované nástrojom WinForms bohato dostačujúce.

Čo však rozhodne nespĺňa potreby a požiadavky pre komfortnú a efektívnu prácu s ESN je základná systémová matematická podpora jazyku C# a .NETu. Z tohto dôvodu je použitá jedna z dostupných knižníc poskytujúca nám efektívnu prácu s maticami a vektormi, ale takisto aj ďalšie matematické objekty a funkcie z oblasti lineárnej algebry. Zo širokej ponuky knižníc sme si vybrali Math.NET Numerics [18]. Hlavnou výhodou je fakt, že sa jedná o open source knižnicu, ktorá je široko používaná, teda má rozsiahlu komunitu, ktorá sa stará o stabilitu kódu a podporu. V porovnaní s inými open source knižnicami nezaostáva ani v efektivite a rýchlosti, čo je takisto veľmi dôležité, keďže pri výpočte bude dochádzať k veľmi veľkému počtu operácií na veľkých objektoch (matice s 10^6 prvkami atď).

2.4 Uživateľská dokumentácia

V tejto časti si popíšeme, ako sa prostredie používa. Postupne sa pozrieme na to, ako prostredie funguje a popíšeme si aj jednotlivé súčasti.

2.4.1 Požiadavky a inštalácia

Naša ESN aplikácia pracuje na platforme .NET Framework 4.6.1 a z toho vyplývajú požiadavky, ktoré sú minimálne:

- operačný systém – Windows 7 so Service Pack 1,
- procesor – 1 GHz,
- RAM – 512 MB,
- priestor na disku – 4.5 GB.

K programu nie je dodávaný inštalátor, pretože inštalácia programu nie je potrebná. Pre spustenie stačí skopírovať súbory `ESNWinFormsRP.exe` a `MathNet.Numerics.dll` do jedného adresára. V ňom stačí spustiť súbor `ESNWinForms.RP.exe`, a takisto je potrebné mať právo na vytváranie súborov a podadresárov. Tieto dva súbory sa nachádzajú v priečinku `ESN` v elektronickej prílohe.

2.4.2 Použitie programu

ESN aplikáciu je možné použiť na predpoveď vývoja ľubovolnej časovej postupnosti. V základnom prípade stačí spustiť aplikáciu, nastaviť všetky potrebné parametre ESN, to znamená aj si vybrať vstupný súbor z disku, ktorý obsahuje vstupné dáta, vybrať si výstupný súbor a spustiť výpočet. Aplikácia je použiteľná aj na úlohu predpovedania vývoja cien akcií, stačí ju spustiť so správnym súborom a nastavením. V programe sú implementované okrem hlavného výpočtu ESN aj rôzne pomocné funkcie na predspracovanie vstupov, výpočet rôznych metrík, ktoré sú k dispozícii pre ľahšie použitie. Výsledky sa okamžite po skončení výpočtu zobrazia a uložia do súborov a je možné s nimi ďalej pracovať. Aplikácia je ovládaná primárne prostredníctvom GUI. Všetky možné varianty výpočtu si nastavuje užívateľ v tomto prostredí z jedného miesta. Prostredie je navrhnuté a popísané tak, aby práca s ním bola intuitívna a jednoduchá. Program je stavaný na riešenie dvoch hlavných variant výpočtu ESN. Sú to: krátkodobá predpoveď s opakovaným tréningom v každom kroku, a dlhodobá predpoveď, pri ktorej dochádza k procesu tréningu len raz. Obom prístupom sa venujeme podrobnejšie v sekcii 4.1 .

2.4.3 Základný popis prostredia

Aplikácia je vytvorená vo frameworku Windows Forms, to znamená že sa jedná o súbor navzájom prepojených formulárov. Po spustení programu sa objaví formulár s hlavným menu. Z tohto formulára sa následne jednoducho dostaneme do formulára pre vytváranie nových ESN, kde sa nachádza väčšina funkcionality, keďže sa očakáva, že väčšinou bude chcieť užívateľ vytvárať nové echo state siete. Tu máme možnosť nastaviť si všetky potrebné parametre ESN a ďalšie nastavenia. Následne môžeme spustiť výpočet a dostaneme formulár so spočítanými výsledkami a môžeme sa rozhodnúť o svojom ďalšom postupe. ESN sa skladá z niekoľkých častí, čo sme si aj uvedomili v sekcii 1.2.1. Toto členenie sme využili aj pri návrhu našej aplikácie. Máme teda prístup k jednotlivým častiam prostredia, ktoré sú logicky oddelené. Návrh prostredia sme mierne prispôbili na riešenie úlohy predpovedania vývoja časových postupností. Prostredie taisto obsahuje aj špecifické nástroje pre predpovedanie vývoja cien akcií ako je výpočet zisku a iných metrík ktorým sa venujeme podrobnejšie v sekcii 3.4.

V ďalších častiach sekcie sa pozrieme bližšie na jednotlivé súčasti prostredia.

2.4.4 Hlavný formulár

Hlavný formulár je prvou časťou, s ktorou príde užívateľ do kontaktu pri spustení aplikácie. Pozostáva z menu obsahujúceho 3 časti:

- Tutorial – tutoriál pre prácu s nástrojom, ktorý obsahuje detailný popis prostredia a návod ako s týmto prostredím pracovať,
- Format – možnosť modifikovať dáta. Formulár má rovnaké rozhranie aj funkcionality ako časť Map&Profit nižšie v časti 2.4.12 a
- ESN options – možnosť vytvoriť (Create new ESN) novú sieť.

2.4.5 Vytvorenie novej ESN

Formulár pre vytvorenie novej ESN je rozdelený na 3 logické hlavné celky:

1. Ľavá časť, obsahuje menu s položkami, kde každá položka označuje logické zoskupenie parametrov týkajúcich sa konkrétnej časti vytvárania ESN. Po kliknutí na konkrétnu položku sa modifikuje pravá časť a zobrazí sa príslušná skupina parametrov. Zámer prostredia je teda, aby si užívateľ postupne prechádzal jednotlivé položky v ľavej časti a okamžite si nastavoval hodnoty parametrov patriacich logicky pod túto položku.
2. Pravá časť, obsahuje názvy parametrov a príslušné polia s hodnotami konkrétnej časti označenej v menu na ľavej strane formulára. Parametre, ktoré sa zobrazia, patria logicky k danej časti výpočtu ESN. Pre konkrétnu časť má sú k dispozícii na jednom mieste všetky príslušné parametre.
3. V dolnej časti sa nachádzajú globálne ovládacie prvky. Globálne z toho dôvodu, že sa nemenia so zmenami ostatných častí formulára. Zámerom je, aby boli ovládacie prvky k dispozícii po celý čas vytvárania ESN a zároveň viditeľne na jednom mieste.

Popíšeme si, čo sa zobrazí na pravej strane pri označení jednotlivých položiek v ľavom menu.

2.4.6 Vstupná vrstva

Prvá zastávka pri vytváraní ESN je nastavenie vstupnej vrstvy (input layer). Užívateľ si nastaví veľkosť vstupnej matice tak, aby veľkosť odpovedala rozmeru vektorov vstupnej sekvencie. Napríklad veľkosť 1 odpovedá sekvencii jednotlivých čísel (formálne sekvencia vektorov veľkosti 1). So vstupnou vrstvou ďalej súvisí aj nastavenie škálovania vstupnej matice, teda aby si užívateľ mohol prispôbiť dynamiku vstupu dynamike úlohy.

Input Size

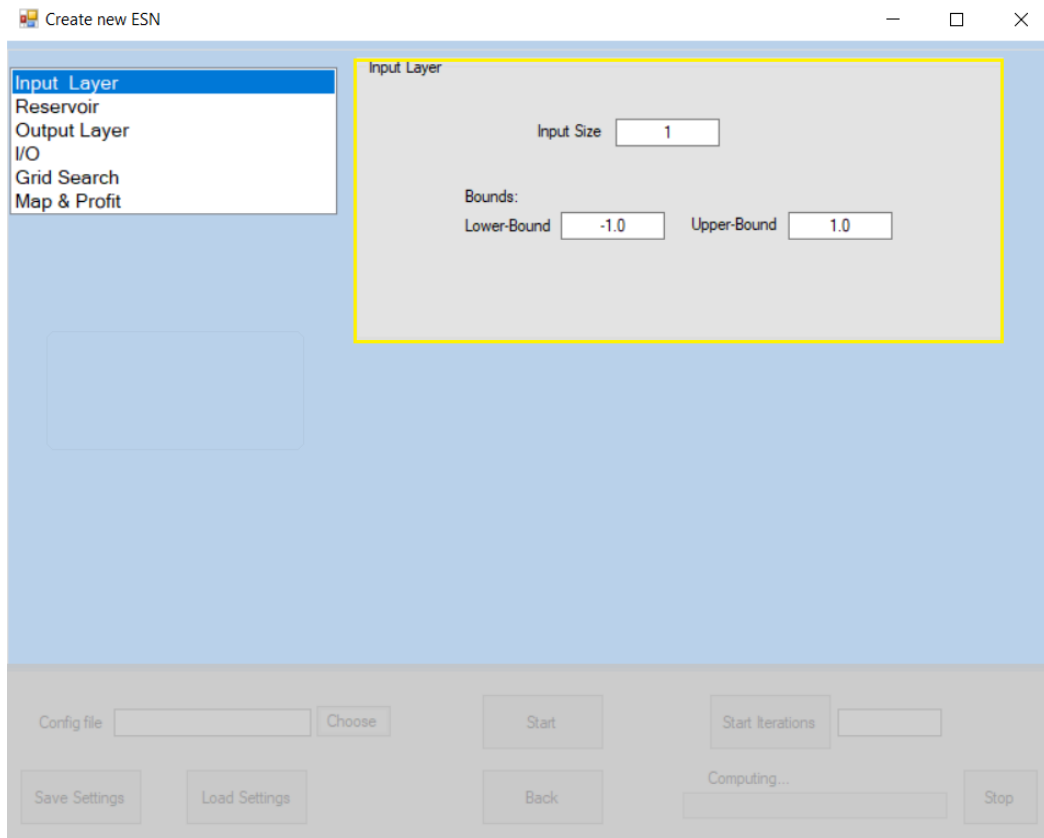
Veľkosť vstupnej vrstvy. Valídne sú len kladné celočíselné hodnoty.

Lower-Bound

Dolná medz intervalu pre výber hodnôt váh náhodne generovanej vstupnej váhovej matice \mathbf{W}_{in}

Upper-Bound

Horná medz intervalu pre výber hodnôt váh náhodne generovanej vstupnej váhovej matice \mathbf{W}_{in}



Obr. 2.1: Input layer – vstupná vrstva

2.4.7 Rezervoár

Väčšina kľúčových nastavení ESN sa nachádza práve v časti pre nastavenie rezervoáru. Pri nastavovaní je doporučené držať sa osvedčených postupov a heuristik zo sekcie 1.3. Určite je potrebné nastaveniu týchto parametrov venovať veľkú pozornosť.

Reservoir size

Veľkosť rezervoáru. Počet neurónov v sieti, viac informácií v sekcii 1.3.1,

Lower-Bound

Dolná medz intervalu pre výber hodnôt váh náhodne generovaného rezervoáru W ,

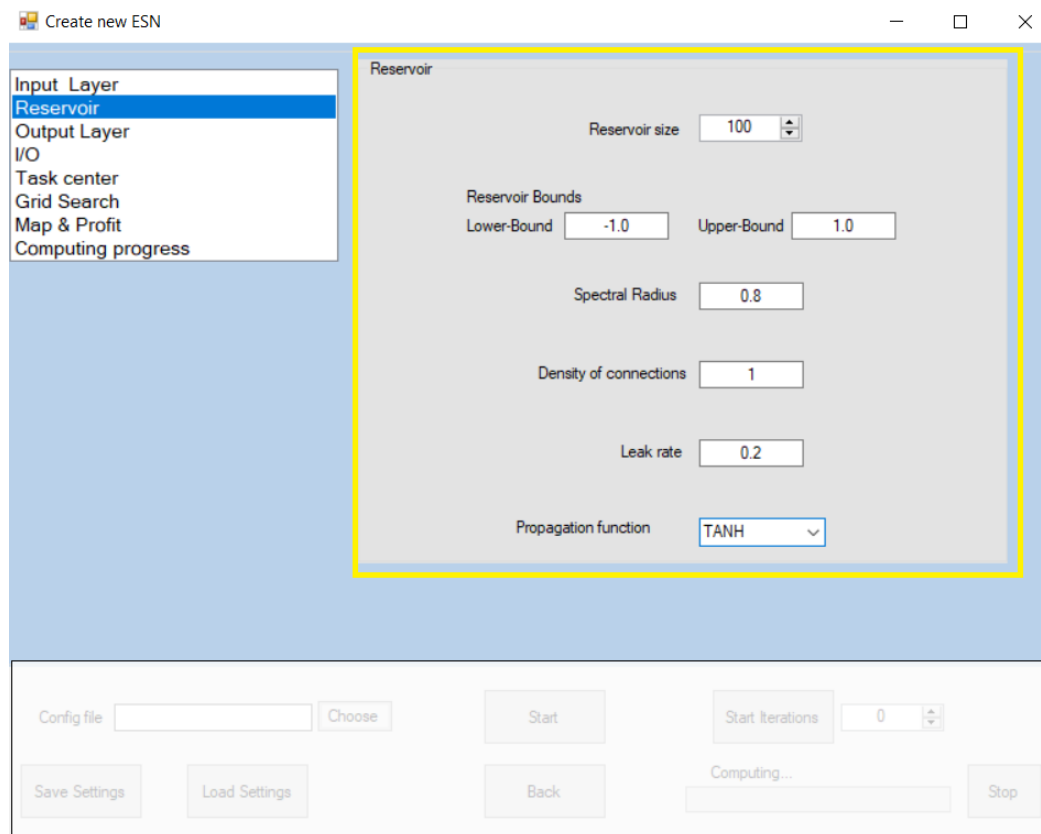
Upper-Bound

Horná medz intervalu pre výber hodnôt váh náhodne generovaného rezervoáru W ,

Spectral radius

Hodnota spektrálneho polomeru, ktorú bude mať vygenerovaný rezervoár, bližšie je vysvetlená v sekcii 1.3.3,

Density of connections



Obr. 2.2: Reservoir

Hustota spojení medzi neurónmi v rezervoári W . Možné hodnoty sú z intervalu $(0,1]$ kde 0 znamená žiadne spojenia medzi neurónmi a 1 znamená, že každý neurón je spojený so všetkými ostatnými neurónmi v rezervoári. Pri niektorých úlohách je výhodné mať redšie prepojenia medzi neurónmi v rezervoári W a tým zefektívniť výpočet. Podľa článku [13] však nastavenie hustoty spojení nemá zásadný vplyv na výkon siete, a preto nepatrí medzi kľúčové parametre pre optimalizovanie. Chýbajúce spojenie medzi neurónmi poznáme podľa toho, že váha príslušného spoja je 0.

Leak rate

Miera úniku je reálne číslo z intervalu $(0,1)$, bližšie vysvetlenie v sekcii 1.3.4.

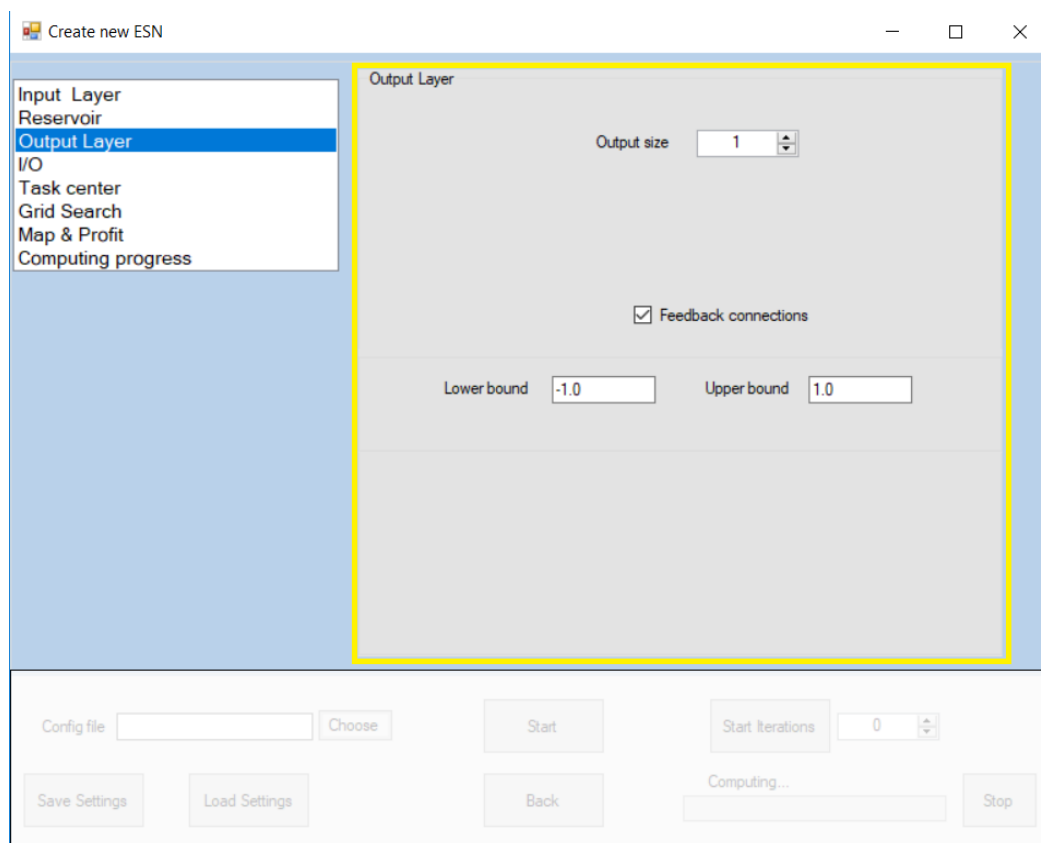
Propagation function

Prenosová funkcia v rezervoáre, na výber medzi funkciou tanh a identitou.

2.4.8 Výstupná vrstva

Nastavenie výstupnej vrstvy ponúka rovnako ako pri vstupnej vrstve nastavenie veľkosti. Často sa tieto veľkosti rovnajú. Užívateľ by mal tiež venovať pozornosť nastaveniu matice spätných spojení v prípade, že sa rozhodne tieto spojenia využiť pre účely úlohy. Výstupná vrstva predstavuje po vstupnej vrstve a rezervoári tretiu a zároveň poslednú časť siete.

Output size



Obr. 2.3: Output layer – výstupná vrstva

Veľkosť výstupnej vrstvy, počet výstupných prvkov.

Feedback connections

Spätné spojenia výstupu do rezervoáru. Ich použitie je voliteľné.

Lower Bound

Dolná medz intervalu pre výber hodnôt náhodne generovanej matice váh spätných spojení W_{fb} .

Upper Bound

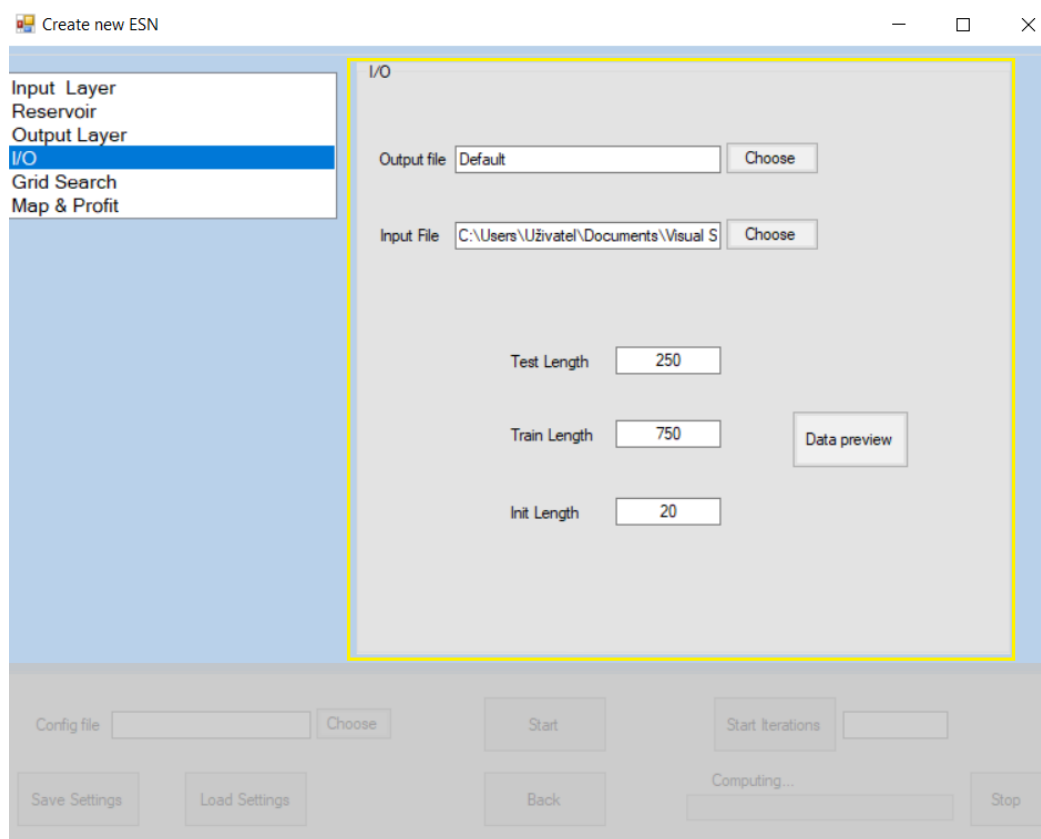
Horná medz intervalu pre výberu hodnôt náhodne generovanej matice váh spätných spojení W_{fb} .

2.4.9 I/O

Nastavenie vstupných a výstupných súborov uľahčuje prácu s aplikáciou. Užívateľ si môže jednoducho ukladať a triediť výsledky práce, rovnako tak aj načítať pohodlne z disku vstupné dáta. Do tejto časti sme sa rozhodli zaradiť aj rozdelenie vstupných dát na počiatočné, trénovacie a testovacie.

Output file

Výber súboru pre výstup výpočtu. Výber sa vykoná kliknutím príslušného tlačítka a následným výberom cesty k súboru. Výsledky výpočtov sa zapisujú do



Obr. 2.4: I/O nastavenia

dvoch súborov. Prvý pridáva k názvu súboru koncovku `_out.csv` a do súboru sú zapisované metriky ktoré sú zaškrtnuté v časti **Task Center**. Vždy je na jednom riadku uvedená skratka a po tabulátore priemerné hodnoty pre všetky iterácie na testovacej a na nasledujúcom riadku na tréningovej množine. Do súboru sú zapisované aj hodnoty náhodného semienka pre každú jednu iteráciu, ktoré nasledujú po nápisé „Random seeds“. Druhý súbor pridáva k názvu koncovku `_outdetail.csv` a do súboru sú zapisované skutočné hodnoty a vypočítané hodnoty na testovacej množine oddelené medzerou, teda na jednom riadku sa nachádzajú vždy dve hodnoty.

Input file

Výber súboru so vstupnou sekvenciou. Výber sa vykoná rovnako ako pri výbere výstupného súboru. Program očakáva reálne čísla s desatinnou bodkou oddelené medzerami. Jeden vstupný vektor sa nachádza na jednom riadku.

Test length

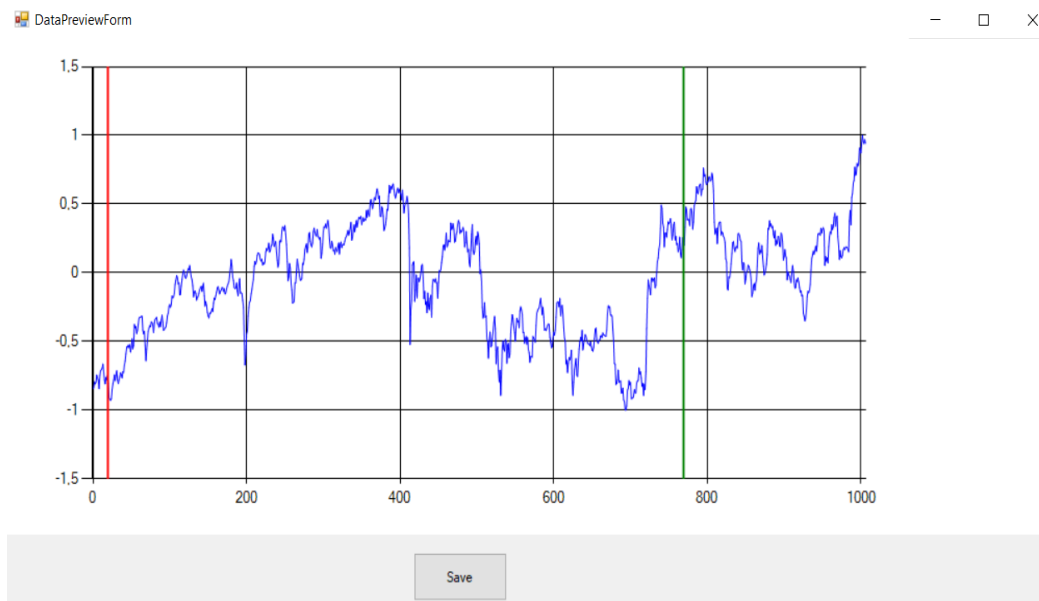
Dĺžka testovacej sekvencie. Počet hodnôt, na ktorých sa porovnáva výsledok výpočtu s očakávanými hodnotami.

Train length

Dĺžka tréningovej sekvencie. Počet hodnôt na ktorých sieť trénujeme.

Init length

Dĺžka počiatočnej sekvencie, ktorá nie je zahrnutá do tréningovania, ale používa sa len na tzv. zahriatie rezervoáru.



Obr. 2.5: Data preview formulár

Tlačítko `Data preview` slúži na manuálne doladenie testovacej, trénuvacej a počiatocnej dĺžky. Po kliknutí sa zobrazí graf vstupnej sekvencie a hodnoty vyplnené v poliach pri jednotlivých dĺžkach predstavujú intervaly medzi vertikálnymi čiarami viz obrázok 2.5. Červená čiara predstavuje hranicu medzi počiatocnou a trénuvajúcou sekvenciou a zelená čiara hranicu medzi trénuvajúcou a testovacou sekvenciou. Tieto čiary sú pohybovateľné a ich posunutím sa modifikujú hodnoty v poliach. Po kliknutí na tlačítko `Save` v dolnej časti sa modifikácie uložia a hodnoty v poliach sa podľa toho prenastavia. Táto funkcionálna je použiteľná len pre vstupné vektory dĺžky 1.

2.4.10 Task Center

Obsahuje nastavenia výpočtu podľa typu úlohy, ktorú chceme riešiť viz 2.6. V hornej časti sú na výber postupy výpočtu. Podľa zaškrtnutých polí v hornej časti sa v dolnej časti potom zobrazia rôzne metriky a je možné si zvoliť, ktoré z nich chceme zahrnúť do výpočtu.

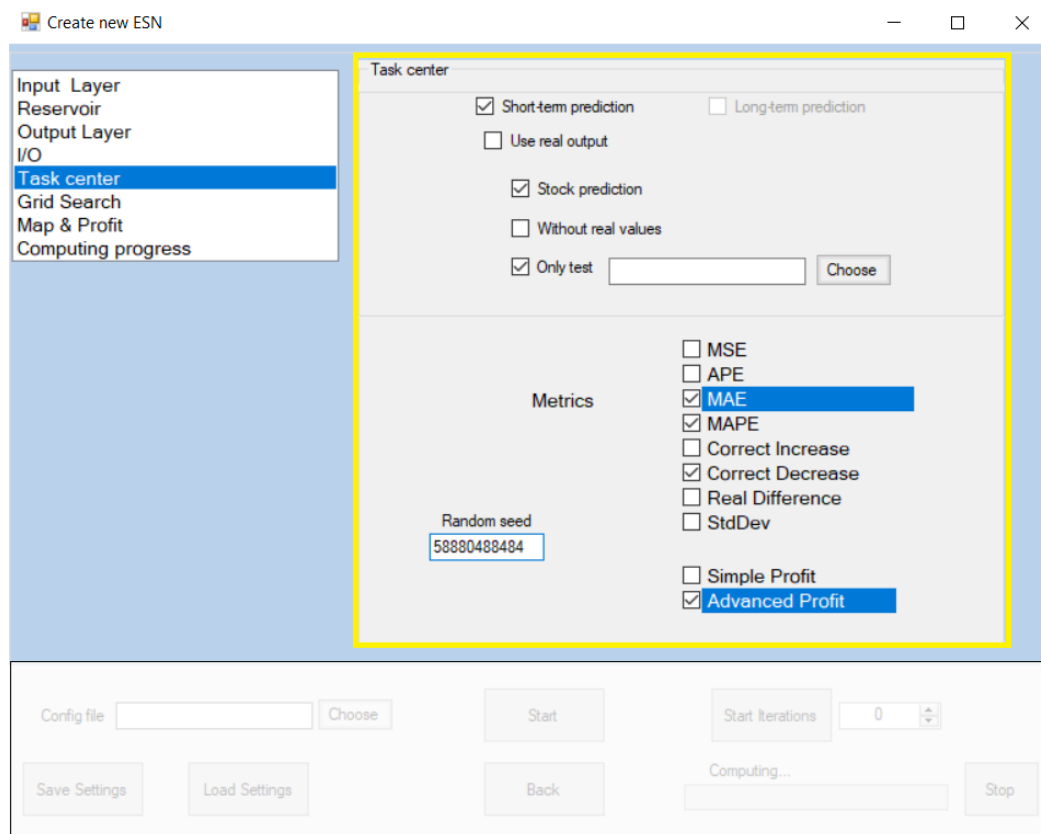
Short-term prediction

Slúži na určenie, či chceme počítať postupom krátkodobej predpovede s opakovaným tréningom. Pri označení tejto možnosti sa zobrazí súvisiaca možnosť `Use real output`.

Use real output Ak je políčko zaškrtnuté použije sa pri krátkodobej predpovedi skutočná hodnota prechádzajúceho kroku namiesto vypočítanej hodnoty sieťou v predchádzajúcom kroku a výpočet odpovedá predpovedi ktorú ďalej označujeme ako vylepšenú krátkodobú predpoveď viz sekcia 4.1.

Long-term prediction

Slúži na určenie, či chceme počítať postupom dlhodobej predpovede kde dochádza k tréningu len raz. Ide o duálne nastavenie k `Short-term prediction` a



Obr. 2.6: Formulár nastavenia typu úlohy

navzájom sa vylučujú.

Stock prediction

Zaškrtnutím dávame najavo úmysel počítať predpoveď vývoja cien akcií. Prostredie ponúka pri tejto možnosti špeciálne metriky: Simple Profit – jednoduchý výpočet zisku popísaný v sekcii 3.4, Advanced Profit – pokročilý výpočet zisku popísaný v sekcii 3.4.

Without real values

Táto položka označuje možnosť predpovedať vývoj bez porovnania so skutočnými hodnotami, teda bez použitia metrík. Možnosť predpovedať vývoj aj za predpokladu že skutočné dáta nemáme.

Only test

Možnosť ktorá umožňuje užívateľovi načítať zo súboru už naučenú sieť a aplikovať ju na nové vstupy. Konkrétne je potrebné zo súboru načítať vstupnú maticu \mathbf{W}_{in} , vektor aktivácií na neurónoch v rezervoári $\mathbf{x}(n)$, váhovú maticu \mathbf{W} a výstupnú maticu \mathbf{W}_{out} . Jednotlivé matice a vektory musia byť vo vstupnom súbore zapísané tak, že najprv sa na samostatnom riadku nachádzajú rozmery objektu v tvare [počet riadkov];[počet stĺpcov], a potom v ďalších riadkoch sú postupne riadky objektu, kde hodnoty v riadku sú oddelené medzerou alebo tabulátorom. Objekty sú od seba oddelené jedným prázdny riadkom.

Metrics

Označuje zoznam metrík, kde zaškrtnutie pri názve konkrétnej metriky znamená, že táto metrika bude zahrnutá do výpočtu a teda zapísaná do výsledného súboru. Časť metrík sme predstavili v sekcii 1.2.4, ďalšej časti sa venujeme v sekcii 3.4, metriky **Correct increase** a **Correct decrease** predstavujú počet období medzi dvomi po sebe idúcimi výstupmi keď sieť správne predpovedala rast hodnoty resp. pokles, vydelený počtom období keď skutočná hodnota naozaj rástla resp. klesala.

Random seed Možnosť nastavenia hodnoty náhodného semienka pre náhodné generovanie matíc. Ak políčko zostane nevyplnené použije sa náhodná hodnota.

2.4.11 Grid Search

Užívateľovi sme sa rozhodli ponúknuť aj možnosť optimalizácie parametrov siete metódou grid search (GS), viz 1.3.5. Na výber má z niekoľkých parametrov s tým, že si môže vybrať 2 a s ich hodnotami prehľadávať grid searchom. Pri výbere parametrov si užívateľ musí nastaviť aj rozlíšenie, ako podrobne chce daný parameter vyskúšať. To znamená, že užívateľ si môže spustiť GS a pomocou neho nastaviť parametre a následne si podľa toho upraví parametre v ostatných častiach formuláru. Užívateľ môže samozrejme použitie GS pri svojej činnosti vynechať, preto sme ho zaradili až za kľúčové nastavenia pre ESN. V hornej časti sa nachádza zaškrťavacie políčko, a po zaškrtnutí sa objaví rovnaká sada polí ako v ľavej časti s rovnakým významom, s tým že polia na ľavej strane patria prvému parametru a polia na pravej strane druhému parametru. Význam polí je vysvetlený nižšie.

Parameter

Výber názvu parametra, ktorý chceme modifikovať Grid searchom. Polia a hodnoty pod týmto polom potom patria k tomuto parametru.

Initial value

Počiatočná hodnota, výpočet začína s touto hodnotou parametru.

Step

Dĺžka kroku, hodnota o ktorú sa zmení aktuálna hodnota parametra po skončení iterácií s aktuálnymi hodnotami parametrov.

Number of steps

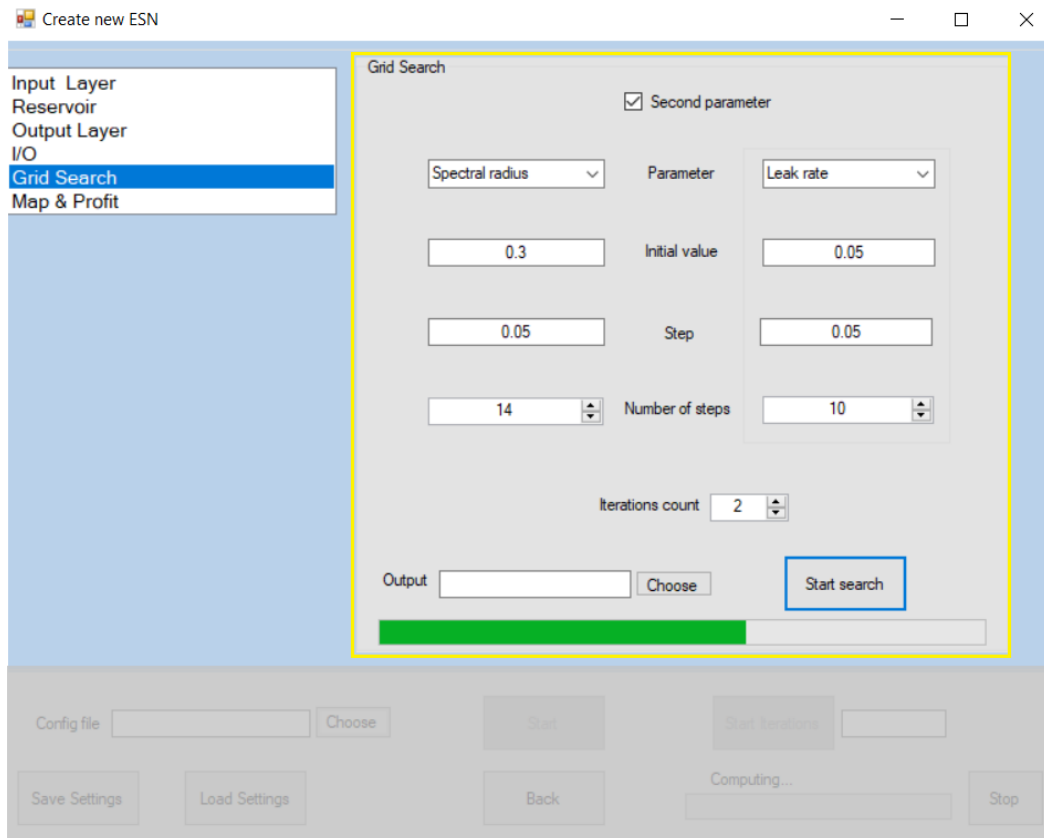
Určuje koľko krát daný parameter modifikujeme o hodnotu dĺžka kroku z počiatočnej hodnoty.

Iterations

Počet iterácií, to znamená počet kompletných výpočtov ESN s danými hodnotami parametrov. Tento parameter má význam, keď chceme fungovanie ESN vyhodnocovať štatisticky.

Output

Výber súboru pre výstup výpočtu. Výber sa vykoná kliknutím príslušného tlačítka a následným výberom cesty k súboru.



Obr. 2.7: Grid search nastavenia

Start

Spustenie výpočtu na základe hodnôt v poliach.

Nižšie sa nachádza progress bar, ktorý sa pri výpočte postupne naplňa, s tým, že vyplnená časť predstavuje dokončenú časť výpočtu.

2.4.12 Map & Profit

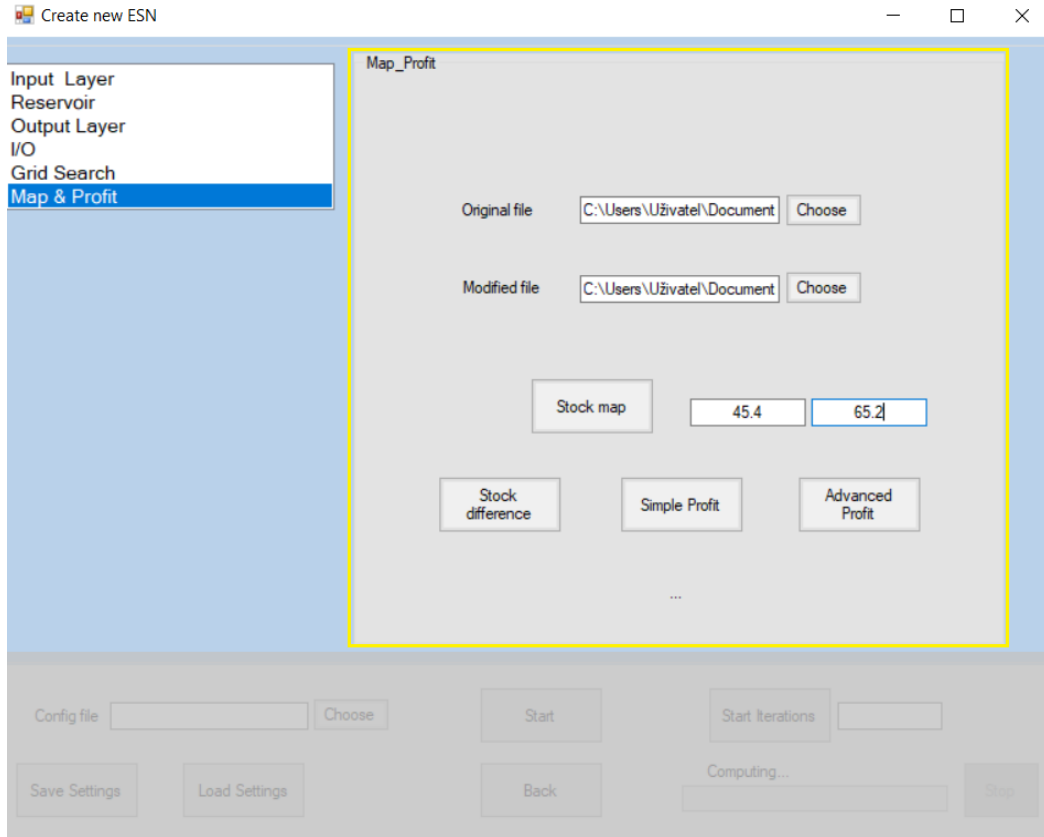
Poskytuje možnosť modifikovať hodnoty zo vstupného súboru do prijateľnejšej podoby pre výpočet ESN. Nesúvisí už priamo so samotným výpočtom ESN, ale skôr ponúka nástroj na spracovanie vstupných dát ale aj výsledkov. Opäť je výhodné pre užívateľa, aby mal takúto možnosť priam pri vytváraní siete.

Original file

Pôvodný súbor ktorého hodnoty chceme modifikovať. Výber súboru sa vykoná kliknutím príslušného tlačítka a následným výberom cesty k súboru. Požadovaný formát odpovedá formátu dát z Yahoo finance [2], teda hodnota ktorú normalizujeme sa nachádza v 5. stĺpci, alebo to môže byť súbor s jedinou hodnotou v riadku, a v tomto prípade sa normalizuje táto hodnota.

Modified file

Nový súbor s modifikovanými hodnotami. Výber súboru sa vykoná kliknutím príslušného tlačítka a následným výberom cesty k súboru. Modifikovaný súbor obsahuje vždy len jednu normalizovanú hodnotu v riadku.



Obr. 2.8: Možnosti Map&Profit

Stock Map

Skutočné vstupné hodnoty nemusia vyhovovať účelom ESN. Preto sa veľmi často hodnoty mapujú do vhodného intervalu. Sieť pracuje dobre s hodnotami okolo nuly. My sme sa rozhodli z tohto dôvodu mapovať hodnoty zo vstupného súboru do intervalu $[-1,1]$. Existuje opäť niekoľko možností ako to dosiahnuť. V našej práci používame rovnicu (2.1). Kliknutím tlačítka sa hodnoty z pôvodného súboru modifikujú podľa rovnice:

$$\mathbf{x}^{normalised} = 2 \times \frac{\mathbf{x}^{target} - \mathbf{x}_{min}^{target}}{\mathbf{x}_{max}^{target} - \mathbf{x}_{min}^{target}} - 1 \quad (2.1)$$

kde

- $\mathbf{x}^{normalised} \in [-1,1]$ je hodnota po modifikácii.
- \mathbf{x}^{target} je hodnota z pôvodného súboru pred modifikáciou.
- $\mathbf{x}_{max}^{target}$ je maximálna hodnota z pôvodného súboru pred modifikáciou.
 $\mathbf{x}_{max}^{target} = \max_{k=1,\dots,N_{in}} \max_{n=1,\dots,m} \mathbf{x}_k^{target}(n)$,
- $\mathbf{x}_{min}^{target}$ je minimálna hodnota z pôvodného súboru pred modifikáciou.
 $\mathbf{x}_{min}^{target} = \min_{k=1,\dots,N_{in}} \min_{n=1,\dots,m} \mathbf{x}_k^{target}(n)$.

Do ľavého poľa sa zobrazí hodnota minima $\mathbf{x}_{min}^{target}$ a do pravého hodnota maxima $\mathbf{x}_{max}^{target}$.

Stock difference

Ďalšou možnosťou úpravy vstupných dát je tzv. relatívny rozdiel. Nepočítame konkrétne hodnoty nejakej funkcie, ale rozdiel dvoch susedných hodnôt, čo využijeme s výhodou pri úlohe predpovede vývoja cien akcií, keď pre nás bude dôležitým údajom práve zmena počas dňa (rozdiel medzi hodnotami dvoch po sebe idúcich dní). Kliknutím tlačítka sa hodnoty z pôvodného súboru modifikujú podľa rovnice:

$$\mathbf{y}^{target} = (\mathbf{y}_2 - \mathbf{y}_1)/\mathbf{y}_1; (\mathbf{y}_3 - \mathbf{y}_2)/\mathbf{y}_2; \dots (\mathbf{y}_n - \mathbf{y}_{n-1})/\mathbf{y}_{n-1} \quad (2.2)$$

kde

- \mathbf{y}^{target} je hodnota po modifikácii.
- \mathbf{y}_n je hodnota vstupnej sekvencie na pozícii n .

Počítame teda v tomto prípade relatívny rozdiel. Sekvencia po modifikácii je ešte následne mapovaná do intervalu $[-1,1]$ podľa rovnice (2.1).

Simple profit & Advanced profit

Počítajú zisk pre úlohu s predpoveďou vývoja cien akcií. Na vstupe teda očakávajú súbor vo formáte ktorý ilustruje obrázok 2.14. V prípade, že sú vyplnené hodnoty textových polí pre minimálnu a maximálnu hodnotu je zisk počítaný na inverzne mapovaných dátach teda pôvodných dátach pred spracovaním ESN. Ak minimálna resp. maximálna hodnota chýba, alebo vo vstupnom súbore na ktorom počítame zisk sa nachádzajú hodnoty, ktoré nie je potrebné inverzne mapovať, je zisk počítaný na týchto dátach bez ďalšej úpravy. Bližšie vysvetlené v kapitole 5. Vo výstupnom súbore sa nachádza na prvom riadku spočítaný zisk a na ďalších riadkoch hodnoty správneho rastu (correct increase) a správneho poklesu (correct decrease) objasnené v sekcii 2.4.10.

2.4.13 Ovládacie prvky

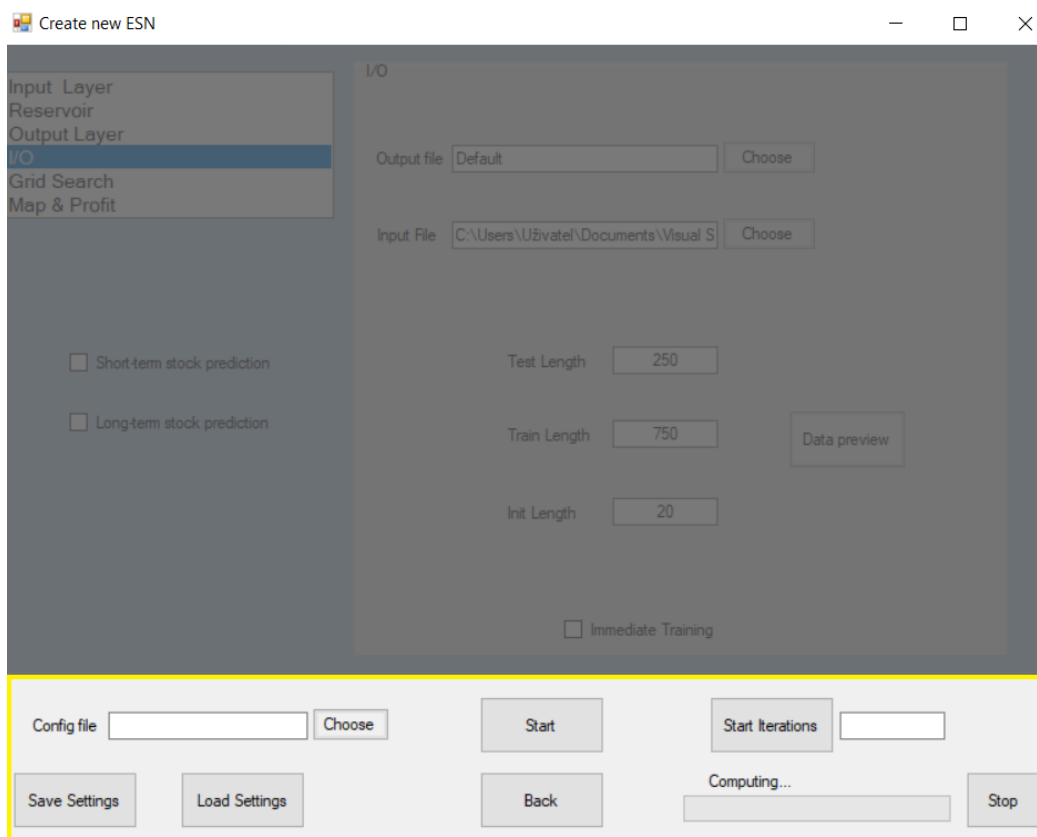
Ovládacie prvky sa nachádzajú v dolnej časti formuláru. Dávajú užívateľovi možnosť spúšťať opakovane ľubovoľný počet výpočtov, čo je veľmi dôležitou vlastnosťou aplikácie. Užívateľ vidí aj priebeh výpočtu, resp. množstvo dokončeného počítania, a môže kedykoľvek výpočet zastaviť, spustiť znovu s rovnakými parametrami alebo parametre pozmeniť. Pritom však celý čas zostáva v kontexte jedného formulára.

Save settings

Uloženie aktuálnych hodnôt všetkých parametrov do textového súboru, ktorý vyberieme z disku po kliknutí na toto tlačítko. Formát uloženia odpovedá formátu popísanému nižšie v sekcii 2.4.14

Load settings

Načítanie hodnôt všetkých parametrov z textového súboru, formát musí odpovedať formátu súboru, ktorý vzniká tlačítkom Save settings a je popísaný v sekcii 2.4.14. Hodnoty načítané zo súboru sa zobrazia v GUI v príslušných poliach a je možné ich ďalej upravovať.



Config file

Výber konfiguračného súboru obsahujúceho sekvencie príkazov pre nastavenie parametrov a spustenie výpočtov. Výber súboru sa vykoná kliknutím príslušného tlačítka a následným výberom cesty k súboru.

Start

Spustenie jednej iterácie výpočtu s parametrami určenými v príslušných poliach. Ak niektoré parametre nie sú nastavené, použijú sa buď implicitné hodnoty, alebo prostredie užívateľa upozorní na polia, kde nie je možné použiť implicitné hodnoty.

Back

Zatvorenie formulára a návrat do hlavného menu.

Start Iterations

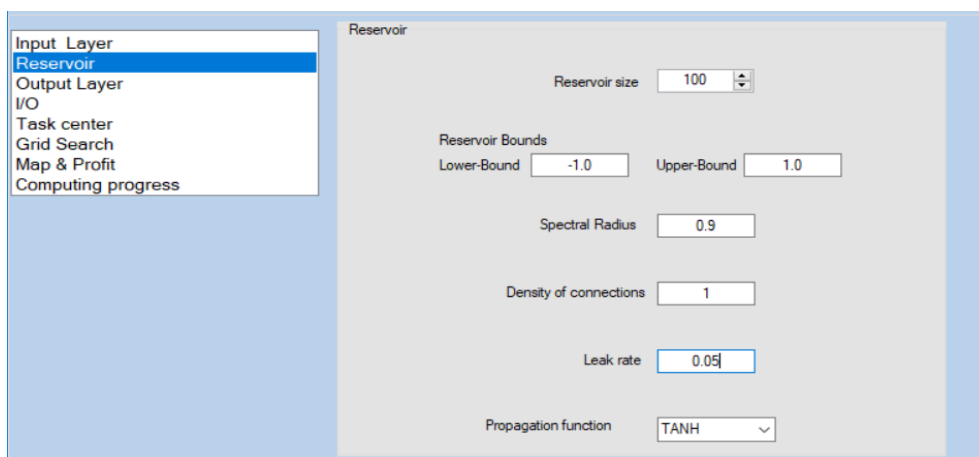
Spustenie väčšieho počtu iterácií s rovnakými parametrami. Počet iterácií je určený kladným celým číslom v príslušnom poli.

Computing...

Označuje progress bar, ktorý sa pri výpočte postupne naplňuje, s tým, že vyplnená časť predstavuje dokončenú časť výpočtu.

Stop

Tlačítka pre zastavenie a ukončenie výpočtu. Výpočet je potrebné začať úplne odznovu.



Obr. 2.9: Príklad nastavenia rezervoáru

2.4.14 Formát súboru

Súbor pozostáva z riadkov, kde na každom riadku sa nachádza jeden parameter v tvare [názov parametru]=[hodnota], kde medzi názvom a hodnotou nesmie byť medzera. Názov parametru odpovedá názvu parametru v GUI s tým, že názov je napísaný s veľkými písmenami bez medzier, pomlčiek a čiarok. Výnimkou sú hodnoty dolných a horných medzí, kde sa k názvu parametru pridávajú predpony: I- pre vstupnú vrstvu, R- pre rezervoár a FB- pre spätné spojenia. Príklad časti konfiguračného súboru vidíme v tabuľke 2.1. Vzorové konfiguračné súbory sa nachádzajú v elektronickej prílohe v priečinku `ESNdata`.

2.4.15 Príklad použitia aplikácie

Táto sekcia obsahuje príklad použitia našej aplikácie. Aby sme využili čo najviac nástrojov, použijeme aplikáciu na predpovedania vývoja ceny akcie Google. Postup bude nasledovný:

1. V hlavnom menu zvolíme položku `ESN options` a klikneme na možnosť `Create new ESN`, čím sa dostaneme do formuláru pre vytváranie sietí.
2. Vo formulári postupne nastavíme všetky parametre. Začneme vstupnou vrstvou. V tomto prípade ponecháme pôvodné hodnoty v poliach, pretože nám hodnoty vyhovujú.
3. Ďalej nastavíme rezervoár. Veľkosť nastavíme na 100, medze intervalu pre výber váh v tomto prípade ponecháme, spektrálny polomer nastavíme na 0.9, hodnotu hustoty nezmeníme, mieru úniku nastavíme na 0.05 a ako prenosovú funkciu zvolíme `Tanh`.
4. Nastavíme výstupnú vrstvu. Veľkosť vstupu nastavíme na 1, spätné spojenia nepoužijeme.
5. V nastaveniach I/O si zvolíme vstupný súbor a výstupný súbor. Nastavíme testovaciu dĺžku na 250, tréningovú na 750 a počiatočnú 20.

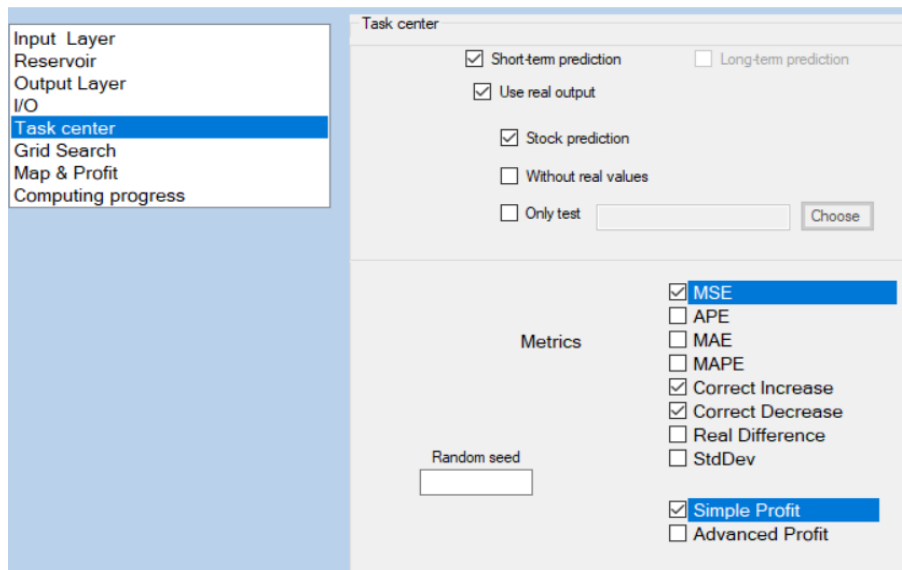
6. V Task Centre zaškrtneme krátkodobú predpoveď s použitím skutočných výstupov. Ďalej zaškrtneme možnosť Stock prediction. Ako metriky zvolíme MSE, Correct Increase, Correct Decrease a Simple Profit.
7. V tomto ilustračnom príklade predpokladáme, že máme k dispozícii dáta z Yahoo finance. Pre výpočet ich však potrebujeme normalizovať. Použijeme

```

INPUTSIZE=1
ILOWERBOUND=-1.0
IUPPERBOUND=1.0
RESERVOIRSIZE=250
RLOWERBOUND=-1.0
RUPPERBOUND=1.0
SPECTRALRADIUS=1.2
LEAKRATE=0.01
OUTPUTSIZE=1
FUNCTION=TANH
TESTLEN=250
TRAINLEN=750
INITLEN=20
ITERATIONS=5
MAPINPUTFILE=AMZN_bear.csv
MAPOUTPUTFILE=AMZN_bear_m
OUTFILE=AMZN_bear_short
INFILE=AMZN_bear_m
MAP
SHORT
REAL
STOCK
MSE
APE
CORRECTDEC
START
SPECTRALRADIUS=0.9
LEAKRATE=0.05
ITERATIONS=20
MAPINPUTFILE=AMZN_bull.csv
MAPOUTPUTFILE=AMZN_bull_m
OUTFILE=AMZN_bull_long
INFILE=AMZN_bull_m
LONG
STOCK
REALDIFF
CORRECTINC
MAP
START

```

Tabuľka 2.1: Ukážka časti konfiguračného súboru



Obr. 2.10: Nastavenie Task center v príklade

nástroje vo formulári Map&Profit, vyberieme cestu k pôvodnému vstupnému súboru s dátami z Yahoo finance, vyberieme cestu k výstupnému súboru s normalizovanými dátami. Ukážka vstupného súboru je na obrázku 2.11. Klikneme na tlačítko `Stock map` a okamžite sa nám v dvoch textových poliach objavia hodnoty maxima a minima z pôvodného súboru a zároveň prebehol zápis modifikovaných dát do výstupného súboru. Tento súbor použijeme ako vstupný súbor v nastavení I/O.

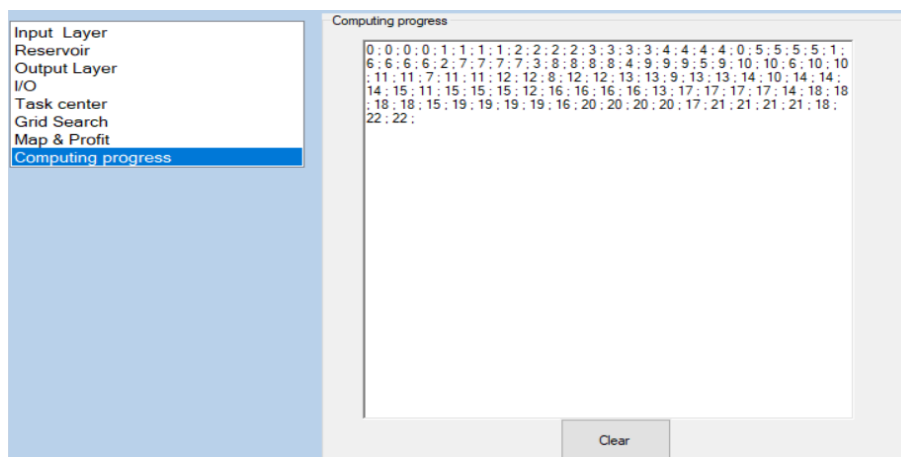
8. Už môžeme našu pozornosť presunúť do dolnej časti s ovládacími nastaveniami. Počet iterácií nastavíme na 5 a môžeme spustiť výpočet tlačítkom `Start Iterations`.
9. Priebeh výpoču môžeme zastaviť tlačítkom `Stop` a postup výpoču môžeme sledovať na `progress bare` pod nápisom „Computing...“. Na `progress bare` však vidíme len množstvo celých iterácií ktoré je dokončené. Pri krátkodobej predpovedi nás zaujíma koľko výstupov v rámci iterácie je spočítaných, keďže výpočet jednej iterácie môže trvať dlho. Na to slúži v ľavej časti položka `Computing Progress`, kde môžeme v textovom poli sledovať číselne vyjadrené poradie spočítaného výstupu.
10. Zobrazí sa nám výsledkový formulár a výsledky sa zapíšu do výstupných súborov. Príklady výstupných súborov môžeme vidieť na obrázkoch 2.13 pre `_out.csv` a 2.14 pre `_outdetail.csv`.

2.4.16 Záver

V tejto sekcii sme sa pozreli na návrh našej aplikácie z pohľadu logického členenia formulárov a nastavení všetkých častí ESN. Videli sme, že jednotlivé časti sú kompaktné a pokrývajú funkcionality pre plnohodnotnú prácu s ESN. Naším cieľom je dať užívateľovi do rúk nástroj, s ktorým si „z jedného miesta“

Date	Open	High	Low	Close	Adj Close	Volume
2013-12-31	556,676697	561,061035	553,683655	560,915894	560,915894	2713000
2014-01-02	558,288269	559,434448	554,684692	557,117126	557,117126	3639100
2014-01-03	558,058044	559,024048	553,018005	553,05304	553,05304	3330000
2014-01-06	557,062073	559,98999	553,773804	559,219238	559,219238	3535000
2014-01-07	563,063049	570,415405	561,141113	570,450012	570,809998	5100000
2014-01-08	573,573547	574,234253	567,212219	571,186157	571,186157	4480500
2014-01-09	572,292297	572,682678	563,343323	565,685669	565,685669	4164800
2014-01-10	570,110107	570,110107	561,686707	565,65564	565,65564	4282700
2014-01-13	563,798828	574,029053	559,144165	562,052063	562,052063	4832900
2014-01-14	569,544556	576,07605	564,609619	575,275269	575,275269	4960200
2014-01-15	577,072083	578,078064	572,467468	574,884888	574,884888	3896400
2014-01-16	575,125122	579,544556	574,574585	578,68866	578,68866	3364400
2014-01-17	579,004028	580,895874	572,672668	575,84082	575,84082	5397100
2014-01-21	581,046021	582,582581	576,226196	582,432434	582,432434	3955800
2014-01-22	583,888916	584,529541	580,01001	583,093079	583,093079	3139600
2014-01-23	580,580566	581,826843	577,762756	580,630615	580,630615	3894300
2014-01-24	576,081055	577,352356	562,062073	562,477478	562,477478	7777200
2014-01-27	563,613586	563,813843	541,676697	551,166138	551,166138	8698400
2014-01-28	555,715698	563,438416	555,525513	562,067078	562,067078	4415900

Obr. 2.11: Příklad formátu vstupného súboru



Obr. 2.12: Computing progress v príklade

MSE	0,00115363328304871	
TrainMSE	0,000164439470997337	
simpleProfit	287969,504212001	
Correct	increase	0,66803840877915
Correct	decrease	0,451550387596899
Random	seeds	
1122627734		
269548474		
361709742		
302596119		
1434747710		

Obr. 2.13: Ukážka súboru s príponov _out.csv v príklade 2.4.15

Real output	ESN output
0,0840787681552564	0,0708155129106771
0,0809136244416235	0,0746885702161268
0,0832873790205759	0,071181214784982
0,0717283247081792	0,0747256026422551
0,056935470429738	0,0617609039619076
0,0276248203488425	0,0456024730757765
0,0526352090921918	0,0125130285220774
0,0774734153715866	0,0432120977204509
0,083700186907935	0,0742691342840516
0,102896612002295	0,0832560856642353
0,137711510590023	0,106950504035005
0,142940697180661	0,147994489841116
0,1371953966645	0,154205172472174

Obr. 2.14: Ukážka časti súboru s príponou `_outdetail.csv` v príklade 2.4.15

nastaví všetky potrebné parametre podľa vlastného uváženia, spustí aplikáciu a môže ju vyhodnotiť. To, že sa nám to podarilo dosiahnuť sme ukázali v tejto kapitole aj na ilustračnom príklade v záverečnej časti kapitoly. Detailný popis implementácie sa nachádza v kapitole 4.2.

3. Finančné prognózy a ESN

V tejto kapitole si najprv vysvetlíme pojmy nevyhnutné pre predpoveď vývoja cien akcií v sekcii 3.1. Ďalej sa v sekcii 3.2 budeme venovať charakteristike, vlastnostiam takéhoto predpovedania vývoja a zdôvodníme, prečo sa jedná o veľmi náročnú úlohu. V tej istej sekcii sa pozrieme aj na vzťah finančnej prognózy a ESN. Následne uvedieme existujúce práce a nástroje na predpovedanie vývoja cien akcií v sekcii 3.3. Posledná sekcia 3.4 patrí možnostiam, ako porovnávať výsledné predpovede.

3.1 Základná myšlienka a kľúčové pojmy

Hlavný princíp akciového trhu (ďalej v texte budeme voľne nahradzovať pojem akciový trh pojmom burza a naopak) je že obchodníci a investori obchodujú s akciami určitej firmy. Obchodovanie má zmysel, pretože hodnota firmy sa časom mení. Investori tým pádom môžu zarobiť alebo prísť o peniaze na základe toho, ako dobre odhadli stav „trhu“. Trh je obrovská množina investorov a obchodníkov, ktorí nakupujú a predávajú akcie, zdvíhajú ceny akcií, alebo ich naopak tlačia dole. Pre každú transakciu s akciami musí existovať predávajúci a kupec, ktorí sa na cene predaja (resp. kúpy) musia dohodnúť. Cieľom obchodovania s akciami je teda zarobiť kupovaním akcií firiem, ktorým sa bude v budúcnosti dariť a teda ich hodnota bude rásť.

Okrem ceny akcií jednotlivých firiem sú dôležité aj tzv. burzové indexy. Burzové indexy reprezentujú agregovanú cenu viacerých akcií, teda keď sa niečo označuje ako burza často tým myslia jeden z hlavných indexov, ako sú Dow Jones Industrial Average (DJIA) alebo Standard & Poor's 500 (S&P500).

DJIA je index založený na cene 30 najväčších amerických korporácií. Obsahuje relatívne málo firiem, preto nie je až takým dobrým indikátorom trhu.

Naopak S&P pozostáva až z 500 veľkých firiem v Spojených štátoch amerických, a tým pádom je oveľa lepším odrazom vývoja trhu.

Každý deň obchodovania na burze predstavuje boj medzi optimizmom a pesimizmom, ktoré rozdeľujú vnímanie trhu na dva prípady, ktoré výrazne súvisia aj s ekonomickou situáciou:

- **bull market** je charakterizovaný optimizmom, že ceny budú rásť a očakávaniami, že tieto výsledky budú pokračovať mesiace až roky.
- **bear market** predstavuje pesimistické vnímanie trhu a ceny akcií klesajúce.

Z dlhodobého pohľadu majú akcie ako celok tendenciu stúpať.

3.2 Finančná prognóza a ESN

V minulej sekcii sme si uvedomili, že akciový trh je založený na obchodovaní s akciami jednotlivých firiem. Z toho vyplýva, že zmeny v cene jednej firmy ovplyvňujú celý trh. Cena akcií firmy je však ovplyvnená nie len burzou, ale aj tým, ako dobre firma funguje, aký zisk produkuje aktuálne, alebo bude produkovať v budúcnosti, ako sa vyvíja odvetvie, v ktorom firma podniká, ako sa mení

hospodárska situácia, kde firma funguje, a takto by sme mohli pokračovať veľmi dlho. Dostávame teda skutočne obrovské množstvo faktorov, ktoré majú dopad na vývoj cien akcií. Tým sme sa chceli priblížiť k jednej z hlavných myšlienok tejto kapitoly a zároveň k tomu, čo bolo cieľom predchádzajúcej sekcie, a síce poukázať na dôležitý avšak intuitívne zrejmý fakt, že predpovedať vývoj cien akcií je veľmi zložitá úloha.

V tejto oblasti bolo aplikované veľké množstvo rôznych kvalitatívnych a kvantitatívnych techník [19] [17] [24] [14]. Pri takýchto analýzach a výskume ľudia stále narážali na nedostatok použiteľných matematických a štatistických techník. Jednou zo schopností neurónových sietí ako takých je práve schopnosť naučiť sa aproximovať aj komplexné funkcie. Keď ešte vezmeme do úvahy obrovské množstvo dát k dispozícii, tak sa neurónové siete stávajú dôležitým nástrojom na detekovanie a modelovanie takýchto zložitých procesov a vzťahov. V roku 2007 uspeli ESN v konkurencii iných algoritmov na tréningovanie neuronových sietí [7]. Umiestnili sa na prvom mieste v predpovedaní časových radov, kde cieľom bolo predpovedať 111 finančných časových radov na 18 mesačnú dobu. Týmito sľubnými výsledkami dali priestor pre ďalší výskum a testovanie ESN v oblasti predpovedania finančných časových radov, konkrétne napríklad predpovedania vývoja cien akcií.

3.3 Existujúce práce

V tejto sekcii si predstavíme existujúce práce, ktoré sa zaoberali otázkou predpovede vývoja cien akcií pomocou ESN.

Jednou z týchto prác je Reservoir Computing in Forecasting Financial Markets od Jenny Su [21]. Autorka sa v tejto diplomovej práci venuje aplikovaniu základného ESN prístupu pre finančný predpoveď. Ako vstupnú sekvenciu používa hodnotu indexu S&P 500 upravenú rovnicou (2.2), teda snaží sa predpovedať zmeny cien medzi jednotlivými dňami. Ide o krátkodobú predpoveď, teda systém je natrénovaný na 1000 vstupoch, dostaneme jednu hodnotu predpovede a následne je celý rezervoár opäť preučovaný na 1000 vstupoch posunutých o 1. Tento postup je opakovaný 500 krát. Výsledky potom autorka porovnáva s náhodnými výbermi z Cauchyho rozdelenia a Gaussovho rozdelenia, na základe priemernej kvadratickej chyby a jej smerodajnej odchýlky.

Ďalšou prácou je článok od X. Lina, Z. Yanga a Y. Songa [11]. V tejto práci sa autori pokúsili o porovnanie ESN prístupu a iných NN prístupov. Navyše ešte analyzujú aj zlepšenie výkonu ESN pomocou Principal Component Analysis (PCA). Používajú bohatšiu vstupnú sekvenciu (vektor veľkosti 6) a vstup normalizujú podľa rovnice (2.1). Sieť trénujú na 1000 vstupných vektoroch a testujú len na 100 výstupoch. Výsledky neporovnávajú MSE ale metrikou absolútnej percentuálnej chyby (APE) – je definovaná nižšie rovnicou (3.7).

Existujú samozrejme ešte viaceré iné práce venujúce sa využitiu ESN na predpovedanie vývoja cien akcií, avšak tie nie sú pre účel tejto práce až také dôležité. Väčšinou sú postavené na nejakej modifikácii ESN prístupu, prípadne na inom ANN princípe, a často skúšajú nejakým spôsobom tento prístup vylepšiť (napr. [27], [5]).

Predstaviť by sme chceli ešte aj nástroj Prophet [23] [22]. Prophet je open source softvér vydaný Facebook Core Data Science tímom. Jedná sa o nástroj na predpoveď časových radov. Prophet produkuje všetky analýzy a predpovede na základe additívneho regresného modelu. To znamená, že kombinuje celkový vývoj časovej postupnosti s rôznymi krátkodobými vzormi a trendami. Prophet je používaný v mnohých aplikáciách po celom Facebooku a produkuje spoľahlivé predpovede na plánovanie a nastavovanie cieľov. Podľa dokumentácie najlepšie funguje na časových radoch s veľkým vplyvom aktuálneho obdobia a s väčším množstvom dát z predchádzajúceho obdobia. Táto charakteristika dobre odpovedá aj nášmu zámeru použiť Prophet na úlohu predpovede vývoja cien akcií a porovnať ho s našou aplikáciou ESN pre predpoveď. Prophet sa stal open source softvérom len nedávno (2017) avšak možnosť použiť Prophet na predpoveď vývoja cien akcií si všimol William Koehrsen ktorý vytvoril nad týmto softvérom nástroj Stocker napísaný v Pythone [9].

Ďalší nástroj vhodný pre porovnanie s našou aplikáciou je ARMA (Autoregressive Moving Average Model) model [4]. Model pozostáva z dvoch častí:

- AR(p) – autoregresivná časť, ktorá zjednodušene predstavuje závislosť aktuálnej hodnoty postupnosti na predchádzajúcich hodnotách, a parameter p je rád autoregresnej časti, teda počet predchádzajúcich hodnôt, ktoré sa použijú na výpočet budúcej hodnoty.
- MA(q) – časť klzavého priemeru rádu q , ktorá predstavuje závislosť aktuálnej hodnoty na predchádzajúcich zmenách hodnôt postupnosti.

Matematicky môžeme ARMA model vyjadriť ako:

$$\mathbf{y}(n) = \phi_0 + \phi_1\mathbf{y}(n-1) + \phi_2\mathbf{y}(n-2) + \dots + \phi_p\mathbf{y}(n-p) + \varepsilon_n + \omega_1 + \varepsilon_{n-1} + \omega_2\varepsilon_{n-2} + \dots + \omega_q\varepsilon_{n-q} \quad (3.1)$$

kde:

- ϕ_0, \dots, ϕ_p sú regresné koeficienty pre hodnoty z predchádzajúcich krokov,
- $\varepsilon_n, \dots, \varepsilon_{n-q}$ sú predchádzajúce zmeny,
- $\omega_1, \dots, \omega_q$ sú zvyškové hodnoty v predchádzajúcich krokoch,
- $\mathbf{y}(n)$ je ďalší výstup a $\mathbf{y}(n-1), \dots, \mathbf{y}(n-p)$ sú predchádzajúce výstupy,

V literatúre často nájdeme ARMA model zapísaný pomocou tzv. *lag operátora* L ktorý vyjadruje:

$$L^k\mathbf{y}(n) = \mathbf{y}(n-k) \quad (3.2)$$

potom:

$$\omega(L)\varepsilon_n = \phi(L)\mathbf{y}(n) \quad (3.3)$$

kde

- $\phi(L)$ sú regresné koeficienty pre hodnoty z predchádzajúcich krokov zapísané pomocou *lag operátora* L , vyjadriť ich môžeme rovnicou:

$$\phi(L) = 1 - \sum_{i=1}^p \phi^i L^i. \quad (3.4)$$

- $\omega(L)$ sú zvyškové hodnoty v predchádzajúcich krokoch zapísané pomocou lag operátora L , vyjadriť ich môžeme rovnicou:

$$\omega(L) = 1 - \sum_{i=1}^q \omega^i L^i \quad (3.5)$$

a

$$\varepsilon_i = \frac{\phi(L)}{\omega(L)} \mathbf{y}(i). \quad (3.6)$$

Každá z vyššie uvedených prác v tejto sekcii ponúka trochu iný pohľad na prognózu cien akcií. Okrem iného pohľadu, však aj rôzne metódy vyhodnocovania výsledkov, a tým pádom aj rôzne metriky kvality predpovede. Ponúka sa teda otázka, ktoré metriky budú pre našu prácu dôležité a naopak, hodnota ktorých ukazateľov nebude pre náš výsledok až tak kľúčová. Preto sa v ďalšej sekcii pozrieme na doposiaľ spomenuté i nespomenuté metriky, ktoré by nás mohli zaujímať.

3.4 Porovnávanie kvality predpovede

V tejto sekcii ukážem hlavné možnosti porovnávania kvality predpovedí vývoja cien akcií. Spomedzi nich si potom vyberieme tie, ktoré sú pre nás dôležité.

1. **MSE a RMSE** (priemerná kvadratická odchýlka a odmocnená priemerná kvadratická odchýlka)

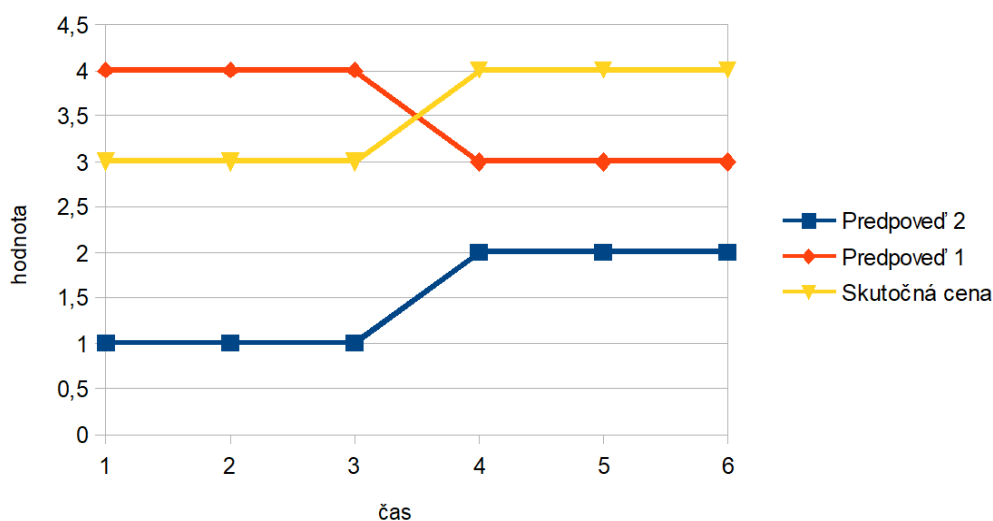
Na MSE sme sa pozreli v sekcii 1.7. Predstavuje samozrejme základnú metriku pre porovnávanie kvality na základe toho, ako veľmi sa predpovedané hodnoty blížia ku skutočným hodnotám. Pre účely vyhodnotenia kvality predpovede vývoja cien akcií je to veľmi dobrá metrika. V praxi pre účely obchodovania s akciami to však nemusí byť najlepšia varianta. Na obrázku 3.1 môžeme vidieť dve rôzne predpovede vývoja hodnoty a skutočný vývoj hodnoty nejakej ceny.

$$MSE(\text{predpoveď 1}) = 1, \quad MSE(\text{predpoveď 2}) = 4$$

Teda druhá predpoveď je horšie podľa MSE. Ak však predpovede používame tak, že akcie nakupujeme iba vtedy, keď model predpovedá rast ceny, tak zisky dopadnú opačne (presnejší algoritmus obchodovania je popísaný v bode 4 nižšie).

$$Zisk(\text{predpoveď 1}) = 0, \quad Zisk(\text{predpoveď 2}) = 33,33\%$$

kde zisk počítame podľa 3.4, ale len v percentách.



Obr. 3.1: Príklad vzťahu MSE a zisku.

Ako sme si rozmysleli v úvode tejto kapitoly, hlavným cieľom obchodovania je maximalizovať zisk. Aj model, ktorý dosiahne malú priemernú kvadratickú chybu nemusí znamenať veľký zisk. Do určitej miery však existuje súvis medzi MSE a ziskom. Je jasné že model, ktorý opakovane dosahuje dobrý odhad s malou hodnotou MSE, je použiteľným nástrojom.

2. **std** Smerodajná odchýlka priemernej kvadratickej chyby viz 1.12. Táto hodnota by nás mohla zaujímať, keď budeme mať dve metódy dosahujúce podobné hodnoty MSE. Tá z nich, ktorá bude mať menšiu hodnotu std, bude stabilnejšia.

3. **APE** absolútna percentuálna chyba

APE je kritérium porovnávania dané rovnicou:

$$APE = \frac{1}{mN_y} \sum_{k=1}^{N_y} \sum_{n=1}^m \left| \frac{\mathbf{y}_k(n) - \mathbf{y}_k^{target}(n)}{\mathbf{y}_k^{target}} \right| \times 100\%, \quad (3.7)$$

4. **MAE** Mean absolute error – priemerná absolútna chyba

$$MAE = \frac{1}{mN_y} \sum_{k=1}^{N_y} \sum_{n=1}^m \left| \mathbf{y}_k(n) - \mathbf{y}_k^{target}(n) \right|, \quad (3.8)$$

5. **MAPE (Mean APE)** priemerná absolútna percentuálna chyba

$$MAPE = \frac{1}{mN_y} \sum_{k=1}^{N_y} \sum_{n=1}^m \left| \frac{\mathbf{y}_k(n) - \mathbf{y}_k^{target}(n)}{\mathbf{y}_k^{target}} \right|. \quad (3.9)$$

Výhody MSE ako metriky pre porovnanie a vyhodnotenie nástrojov a techník pre predpovedanie vývoja cien akcií sme načrtli vyššie. Rovnako podnetnými

údajmi pre vyhodnotenie kvality predpovede sú aj ostatné vyššie uvedené metriky. Čo nás zaujíma najviac, je kvalita predpovede v podobe MSE, avšak do výsledkov experimentov podľa očakávania zahrnieme aj zisk modelu. MSE a ostatné vyššie uvedené metriky nie sú síce priamym ukazateľom zisku modelu, ale aj tak sú veľmi cennou smerodajnou informáciou. Ziskom rozumieme samozrejme hypotetický zisk modelu keďže model testujeme na historických dátach. Výpočet zisku si však zjednodušíme oproti reálnej situácii (reálnu situáciu predstavujú komplexné transakcie medzi nakupujúcim a predávajúcim). Zisk budeme počítat dvomi spôsobmi:

- Jednoduchší spôsob (budeme pre účely práce označovať ako jednoduchý zisk), kde pravidlá sú:
 1. Každý deň, keď model predpovedá že cena bude rásť, kúpime akcie na začiatku dňa a predáme ich na konci dňa.
 2. Ak model predpovedá, že cena bude klesať akcie nekupujeme.
 3. Ak sme kúpili akcie a cena akcií počas dňa stúpila, „zarobili“ sme hodnotu, o ktorú cena stúpila, vynásobenú počtom akcií, ktoré sme kúpili (počet akcií si určíme pri konkrétnej firme, nie je to pre výpočet dôležité).
 4. Ak sme kúpili akcie a cena akcií počas dňa klesla, prišli sme o hodnotu, o ktorú cena klesla, vynásobenú počtom akcií, ktoré sme kúpili.
- Komplikovanejší prístup (budeme pre účely práce označovať ako pokročilý zisk), ktorého pravidlá sú:
 1. Predstavme si, že máme nejaké počiatočné množstvo peňazí.
 2. Pri každej transakcii obchodujeme so všetkými peniazmi a platíme poplatok za transakciu vo výške 0.5%.
 3. Stratégia je opäť podobná, začneme s počiatočným množstvom, ak model predpovedá že cena bude počas dňa rásť kúpime akcie. Ak nasledujúci deň, bude cena opäť rásť, akcie držíme a nevykonávame žiadnu operáciu. V prípade že máme akcie kúpené a model predpovedá že cena bude klesať, akcie predávame a zisk za predaj bude tvoriť naše nové aktuálne množstvo peňazí.

3.5 Záver

V tejto kapitole sme si predstavili existujúce práce popisujúce postupy predpovedania vývoja cien akcií, ktoré budeme porovnávať s našou aplikáciou. Pre účely porovnania sme sa zamerali na bežne používané druhy metrík, a zároveň sme načrtli aj ich nevýhody pre reálne použitie. Takisto sme sa rozhodli, že naším hlavným cieľom bude okrem predpovedania ceny akcií aj maximalizovanie zisku, a ponúkli sme dva oproti reálnej situácii zjednodušené prístupy výpočtu zisku. V ďalšej kapitole sa konečne pozrieme na aplikáciu nášho nástroja pre výpočet vývoja cien akcií.

4. Aplikácia nástroja pre ESN

V tejto kapitole sa budeme venovať našej aplikácii pre predpovedanie vývoja cien akcií. V oddiele 4.1 popíšeme úlohy, ktoré budeme riešiť, a akým spôsobom ich budeme riešiť. Potom v oddiele 4.2 predstavíme vnútornú štruktúru aplikácie.

4.1 Riešené úlohy

Náš nástroj je dostatočne obecný na to, aby mohol slúžiť na predpovedanie krátkodobého a dlhodobého vývoja ľubovolnej postupnosti (s využitím MSE, APE a ďalších spomínaných metrik). V našej práci sa zameriavame na riešenie dvoch typov úloh, ktorými sú predpoveď ceny na nasledujúci deň (takúto predpoveď budeme ďalej označovať pre účely tejto práce ako krátkodobú) a predpoveď cien na niekoľko nasledujúcich dní (tento typ predpovede budeme označovať ako dlhodobú). Pre účely predpovedania cien akcií vezmeme do úvahy zisk ako metriku.

Krátkodobá predpoveď sa vyznačuje tým, že proces tréningu opakujeme pre každý výstup. To znamená, že sieť trénujeme na počte vstupov určených parametrom *Train Length*, pomocou natrénovanej siete vykonáme predpoveď na ďalší deň. Tento postup následne opakujeme na dátach posunutých o jedna. Teda, ak prvý výstup dostaneme tréňovaním na dátach $\mathbf{u}(1), \mathbf{u}(2), \dots, \mathbf{u}(\text{TrainLength})$, tento výstup označíme $\mathbf{u}(\text{TrainLength} + 1)$, ďalší výstup dostaneme tréňovaním na vstupnej sekvencii $\mathbf{u}(2), \mathbf{u}(3), \dots, \mathbf{u}(\text{TrainLength} + 1)$. Takto postupujeme pre každý požadovaný výstup určený parametrom *Test Length* s tým, že spočítaný výstup v jednom kroku používame ako vstup v nasledujúcom kroku. Tento postup je použitý aj v práci [21], ktorú sme spomínali v predchádzajúcej kapitole. Tréning ESN je oproti ostatným častiam výpočtu časovo náročný. Pri opakovanom tréningu sa tým pádom celý výpočet výrazne predlžuje za cenu kvalitnejších výsledkov.

Miernou modifikáciou predchádzajúceho prístupu je výpočet, pri ktorom nepoužívame spočítaný výstup v jednom kroku ako vstup v nasledujúcom kroku, ale namiesto toho použijeme skutočnú hodnotu predchádzajúceho kroku. Tento rozdiel môže mať výrazný vplyv na kvalitu predpovede výslednej siete a preto je použitý v ďalšej kapitole okrem krátkodobej predpovede aj tento prístup. Ak ďalej v práci označíme predpoveď ako vylepšenú krátkodobú, myslíme tým práve zavedenú modifikáciu krátkodobej predpovede.

Pri dlhodobej predpovedi rezervoár prejde procesom tréningu len raz, a následne už len počítame výstupy. Takýto postup je bežnejší a viac používaný, a keďže sa vyhýba opakovanému tréningu, je výrazne rýchlejší ako krátkodobá predpoveď.

4.2 Vývojová dokumentácia

V tejto sekcii sa nachádza vývojová dokumentácia. V jednotlivých častiach tejto sekcie sú predstavené hlavné formulárové a obyčajné triedy a takisto kľúčové

metódy. V elektronickej prílohe v priečinku *Projects* sa nachádza Visual studio 2015 C# solution *ESNWinFormRP.sln*, ktoré obsahuje projekt našej aplikácie.

4.2.1 Iteration

Trieda **Iteration** implementuje hlavnú triedu predstavujúcu jeden kompletný výpočet úlohy. Časté alebo typické použitie programu bude opakovaný výpočet úlohy. Preto sa nám veľmi hodí, mať informácie o konkrétnych hodnotách, s ktorými počítame, a výsledky týchto výpočtov na jednom mieste, v jednej triede. Inštancie tejto triedy teda môžeme pri opakovanom počítaní úlohy jednoducho združovať, a tým si držíme všetky potrebné výsledky na jednom mieste bez straty informácie o jednotlivých iteráciách výpočtu. Jedna inštancia výpočtu je však relatívne veľká úloha, a z princípu fungovania ESN prístupu je možné a vhodné ju rozdeliť na menšie časti. Kompletný výpočet je reprezentovaný v našej implementácii tromi triedami (logickými podčasťami výpočtu): **Initialization**, **Training**, **Testing**. Trieda **Iteration** je potomkom abstraktnej triedy **BasicIteration**, ktorá obsahuje všetky dáta potrebné pre výpočet, to znamená parametre siete, vstupné dáta, ale aj hodnoty rezervoáru a takisto všetky výsledky výpočtu. Trieda **Iteration** potom k svojmu predkovi pridáva špecifická pre konkrétnu úlohu, ktorú model počíta.

4.2.2 Initialization

Trieda pre inicializáciu rezervoáru. Ide o abstraktnú triedu s jedinou metódou **Initialize**. Konkrétnym potomkom tejto triedy je **StockInitialization**. Úlohou je vygenerovať a náhodne inicializovať vstupnú maticu \mathbf{W}_{in} , rezervoár \mathbf{W} a takisto maticu spätných spojení \mathbf{W}_{fb} . Tieto dáta sú potom zabalené do triedy **InitResults**, ktorá ešte obsahuje aj hodnotu náhodného semienka, a celá trieda je návratovou hodnotou spomínanej metódy **Initialize**.

4.2.3 Training

Trieda pokrývajúca tréningovú časť výpočtu ESN. Použitý je rovnaký návrh ako pre inicializáciu, teda existuje jedna trieda **Training** s abstraktnou metódou **Train**. Konkrétne tréningové výpočty sú potom vykonávané v triedach, ktoré dedia od tejto triedy. Sú to konkrétne triedy **LongTermTraining** (pre úlohu počítajúcu našu dlhodobú predpoveď) a **ShortTermTraining** (pre úlohu počítajúcu našu krátkodobú predpoveď). V oboch prípadoch jednotlivé metódy **Train** vracajú inštanciu triedy **TrainResults** obsahujúcu okrem iného výsledky tréningovania. Pri **LongTermTraining** spočítame výstupnú maticu \mathbf{W}_{out} iba raz (tréning prebieha len jedenkrát) a následne proces pokračuje triedou **Testing** (viz. ďalší odstavec). Pri **ShortTermTraining** je ťažko implementovateľné tréning a počítanie výstupu oddeliť, a tak metóda **Train** v tejto triede počíta obe úlohy.

4.2.4 Testing

Aj v tomto prípade využívame abstraktnú triedu **Testing** a jej potomkov. Ako sme si uvedomili v predchádzajúcej sekcii, túto triedu potrebujeme len pri

dlhodobej predpovedi, preto potomkom je iba trieda `LongTermTesting`. Výsledok výpočtu je zabalený v triede `TestResults`.

4.2.5 CreateNewESNForm

Toto je formulárová trieda obsahujúca prostredie výberu a nastavovania hodnôt parametrov. Obsahuje obsluhu udalostí všetkých Windows Forms komponent, ktoré sa nachádzajú v tomto formulári. V tejto triede sa nachádzajú všetky hlavné funkcie ako načítanie vstupných dát zo súboru, vytváranie nových inštancií triedy `Iteration`, zbieranie a zobrazovanie výsledkov. Okrem toho sú v tejto triede implementované aj všetky pomocné a vedľajšie funkcie pre výpočet ESN. Ďalej v tejto sekcii nahliadneme postupne bližšie na dôležitejšie z týchto funkcií.

StartIterations

Metóda pre samotné spustenie výpočtu, ktorá sa volá stlačením tlačítka `Start Iterations`.

V tejto metóde sa najprv pripraví a skontrolujú všetky polia vyplnené užívateľom, či sú hodnoty v nich validné. Na základe vyplnených hodnôt potom načítame dáta zo vstupného súboru vo funkcii `ProcessInput` pomocou inštancie triedy `StreamReader` (všetky procesy načítania dát zo súboru vo všetkých častiach aplikácie vykonávame pomocou tejto triedy), ktorá je potomkom triedy `TextReader`, do dátovej štruktúry `data` typu `List<double[]>`

(`System.Collections.Generic.List`). Dáta postupne pridávame na koniec v poradí, ako ich čítame, vďaka vlastnosti triedy `List`, v konštantnom čase. Čo sa týka štruktúry, sú prvky nášho zoznamu typu pole dvojblov (64-bitových desatinných čísel), keďže vieme, že vstupné dáta môžu byť viac-dimenzionálne vektory.

Po úspešnom spracovaní vstupných dát a vstupov od užívateľa je volaná funkcia `Perform`. Funkciu `Perform` voláme asynchrónne pomocou triedy `Task` z menšieho priestoru `System.Threading.Tasks`, konkrétne pomocou metódy `Task.Run`. Tým dosahujeme viacvláknový výpočet a zároveň nestrácame kontrolu nad GUI, ktoré naďalej obsluhuje udalosti komponent. V metóde `Perform` ako takej vytvárame opäť pomocou triedy `Task` asynchrónne nové inštancie triedy `Iteration`, ktoré potom ukladáme do dátovej štruktúry `List<BasicIteration>`. Pomocou metódy `Task.WaitAll` počkáme na dobehnutie výpočtov všetkých `Taskov`, a ako návratovú hodnotu vrátime všetky inštancie triedy `Iteration` ako spomínaný `List<BasicIteration>`.

Nakoniec si pomocou funkcie `Task.ContinueWith` zaistíme, že až po skončení funkcie `Perform` vyhodnotíme výpočet vytvorením inštancie formulárovej triedy `CreateNewESNResultForm`.

StartGridSearch

Metóda pre spustenie postupu nazývaného grid search. Postup volania aj implementácia volaných funkcií je veľmi podobná ako pri spúšťaní obyčajných iterácií výpočtu metódou `StartIterations`, ktorá je popísaná vyššie. Výkonná metóda sa v tomto prípade sa volá `PerformGS` a má navyše niekoľko parametrov, ktoré si užívateľ nastavuje pri menu položke `GridSearch`. V ostatných častiach sa metóda veľmi nelíši od metódy `Perform`, pracuje sa s triedou `Task` a metódami tejto triedy, rovnako tak vytvárame inštancie triedy `Iteration` a vraciame `List<BasicIteration>`. Po skončení tejto metódy nevytvárame inštanciu

triedy `CreateNewESNResultForm`, ale výsledky zapisujeme do užívateľom určitého súboru. Na zápis v tomto prípade, ale môžeme pokojne povedať, že vo všetkých prípadoch zápisu do súboru v našej aplikácii, používame potomka triedy `TextWriter`, ktorým je trieda `StreamWriter`.

Ďalšími funkciami v tejto triede, ktoré si zaslúžia pozornosť sú:

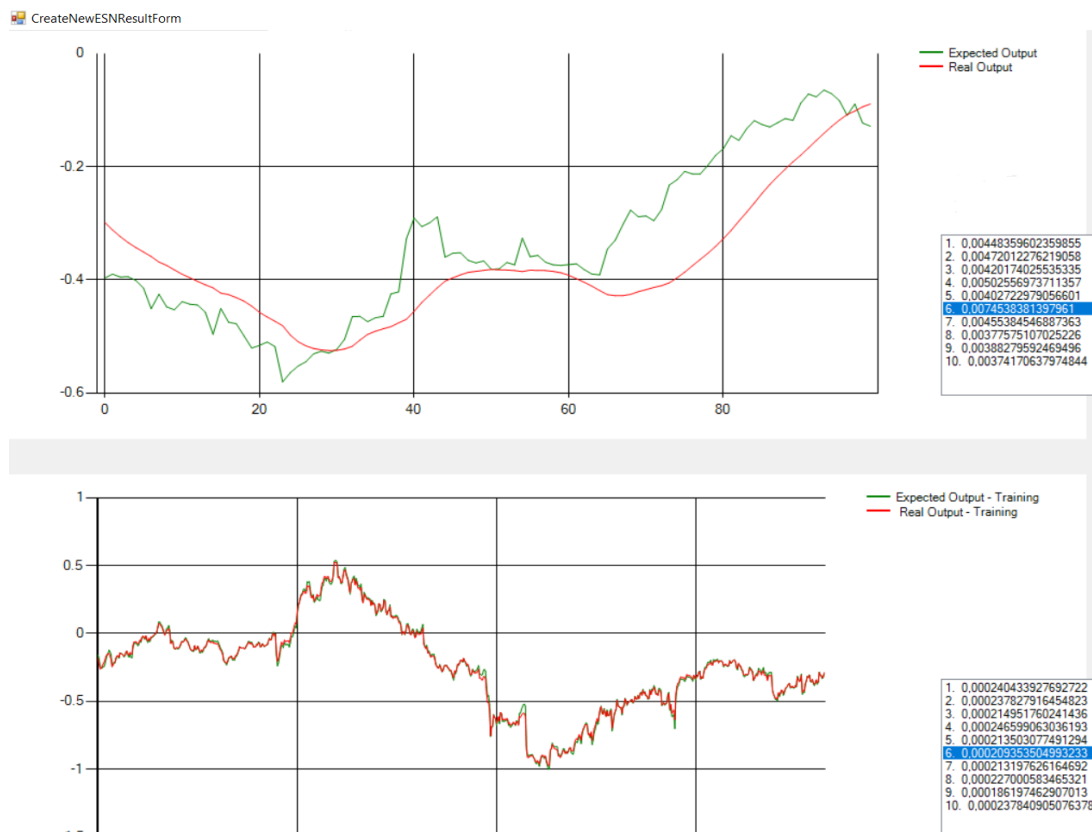
- `Map` – funkcia, ktorá mapuje hodnoty z datovej štruktúry `List<double>` do intervalu $[-1,1]$ pomocou rovnice (2.1).
- `StockDiffMap` – funkcia, ktorá upravuje hodnoty podľa rovnice (2.2) a následne ešte volá funkciu `Map` na upravených dátach.
- `StockMap` – funkcia, ktorá sa stará o načítanie dát zo súboru a na týchto dátach volá funkciu `Map`.
- `CountSimpleProfit` – funkcia počítajúca jednoduchý zisk podľa postupu popísaného v časti 3.4. Funkcia ďalej výsledok výpočtu zapíše do súboru.
- `CountAdvProfit` – funkcia, ktorá počíta a vypisuje do súboru pokročilý zisk postupom z časti 3.4.
- `ExecuteConfigFile` – funkcia poskytujúca užívateľovi možnosť pripraviť si konfiguračný súbor s príkazmi na modifikovanie parametrov a spúšťanie iterácií výpočtu ESN. Funkcia postupne číta konfiguračný súbor a vykonáva požadované úkony. Formát príkazov odpovedá tvaru `[názov parametru][hodnota]` a ide o formát popísaný v sekcii 2.4.14. Zoznam všetkých príkazov je súčasťou tutoriálu.
- `CountMSE` – ako napovedá názov, funkcia dostáva inštanciu triedy `Iteration`, to znamená jeden dokončený výpočet ESN, a počíta priemernú kvadratickú chybu podľa rovnice (1.7).

Okrem funkcie `ExecuteConfigFile`, ktorá je implementovaná priamo v tejto triede, sú funkcie z predchádzajúceho zoznamu len volania funkcií implementovaných vo verejnej triede `MapAndProfit`. To z dôvodu, aby mal tieto funkcie užívateľ k dispozícii počas celého priebehu výpočtu. Viaceré z nich sú užitočné a použiteľné na už spočítaných dátach, preto sa v hlavnom menu nachádza možnosť, ktorá vytvára formulárovú triedu `MapProfitForm`, ktorá umožňuje použiť tieto funkcie na už vopred spočítaných výsledkoch. Inak povedané, užívateľ môže pri výpočte vypočítať len hodnoty postupnosti, a potrebné mapovania a výpočty potencionalného zisku vykonať nezávisle.

Zvyšné funkcie triedy `CreateNewESNForm` už vykonávajú len pomocné výpočty pre vyššie uvedené metódy, resp. starajú sa o plynulý chod GUI, prípadne sa jedná o triviálne ošetrenia udalostí na komponentách GUI a ich popis v zdrojovom kóde vo forme komentárov je pre ich úplne pochopenie dostačujúci.

4.2.6 `CreateNewESNResultForm.cs`

Formulárová trieda pre prostredie vyhodnotenia výsledkov ESN. Prostredie je logicky rozdelené na 2 časti. V hornej časti sa nachádza graf porovnania výsledkov



Obr. 4.1: Vzhľad formulárovej triedy CreateNewESNForm

siete na testovacej množine dát a skutočných dát viz obrázok 4.1. Napravo od grafu sa nachádza komponenta `listBox`. V nej sa nachádzajú hodnoty priemernej kvadratickej chyby spočítané rovnicou (2.1) jednotlivých iterácií, a po kliknutí na konkrétnu hodnotu v listboxe sa do grafu vykreslia výsledky na testovacej množine danej iterácie. Rovnaký princíp platí aj pre dolnú časť, kde sa nachádza graf porovnania výsledkov tréningu siete, čiže výsledkov na tréningovej množine dát.

Z pohľadu implementácie táto trieda obsahuje dátové položky:

`List<BasicIteration>` čiže zoznam všetkých inštancií triedy `Iteration`, kde ich počet je určený parametrom pri vytváraní siete v hlavnom formulári, hodnoty testovacej a tréningovej dĺžky pre vykresľovanie grafov, a takisto sú tam aj hodnoty pre inverzné mapovanie výsledných hodnôt z intervalu $[-1,1]$ do pôvodného intervalu.

Ďalej sa pozrieme na hlavné funkcie z tejto triedy.

DrawResults & DrawResultsTraining

Metóda `DrawResults` vykresľuje hodnoty skutočných dát testovacej množiny do grafu a zároveň pridáva všetky spočítané iterácie do spomínaného listboxu, to znamená, že je počítaná aj hodnota MSE podľa rovnice (2.1). Metóda `DrawResultsTraining` plní rovnakú úlohu ako `DrawResults` avšak na tréningovej množine dát.

PrintResults

Funkcia ktorá vypisuje výsledky výpočtov do súborov. Je volaná okamžite po

fukciách `DrawResults` a `DrawResultsTraining` pri vytváraní výsledných grafov. Zapisuje výsledky do súboru, kde cestu k tomuto súboru si užívateľ výberá pri vytváraní siete (pole `Output file`) a k názvu tohto súboru je pridaná koncovka `output.csv`. Do súboru je zapisované v tvare MSE, APE, smerodajná odchýlka MSE, `simpleProfit` – jednoduchý zisk, `advProfit` – pokročilý zisk, MAE, MAPE, `RealDiff` – rozdiel spočítanej a skutočnej ceny akcie v skutočnej peňažnej hodnote. Všetky tieto zapisované hodnoty sú priemerné hodnoty všetkých iterácií. Vo výslednom súbore sa potom nachádzajú výsledky získané na testovacej množine a takisto aj výsledky na tréningovej množine.

SelectedItem

Táto metóda je volaná pri kliknutí na hodnotu v listBoxe. Do grafu sa pridajú hodnoty kliknutím označenej iterácie. Ďalej spočítame absolútnu percentuálnu chybu podľa rovnice (3.7), MSE, a takisto použijeme inverzné mapovanie do pôvodných hodnôt a spočítame absolútnu hodnotu rozdielu skutočnej hodnoty a hodnoty spočítanej sieťou. Všetky tieto informácie potom zapíšeme do niekoľkých súborov v takomto poradí: skutočná hodnota, spočítaná hodnota, APE, MSE, pôvodná skutočná hodnota (pôvodná znamená inverzne mapovaná), pôvodná spočítaná hodnota, absolútna hodnota rozdielu pôvodných hodnôt. Hodnoty v tomto poradí zapisujeme do užívateľom zadaného súboru rozšíreného o koncovku `_test.csv` pre vyhodnocovanie testovacej množiny, a `_train.csv` pre tréningovú množinu.

4.2.7 EsnMath

Trieda obsahujúca funkcie počítajúce metriky spomenuté v texte 1.2.4.

5. Experimenty

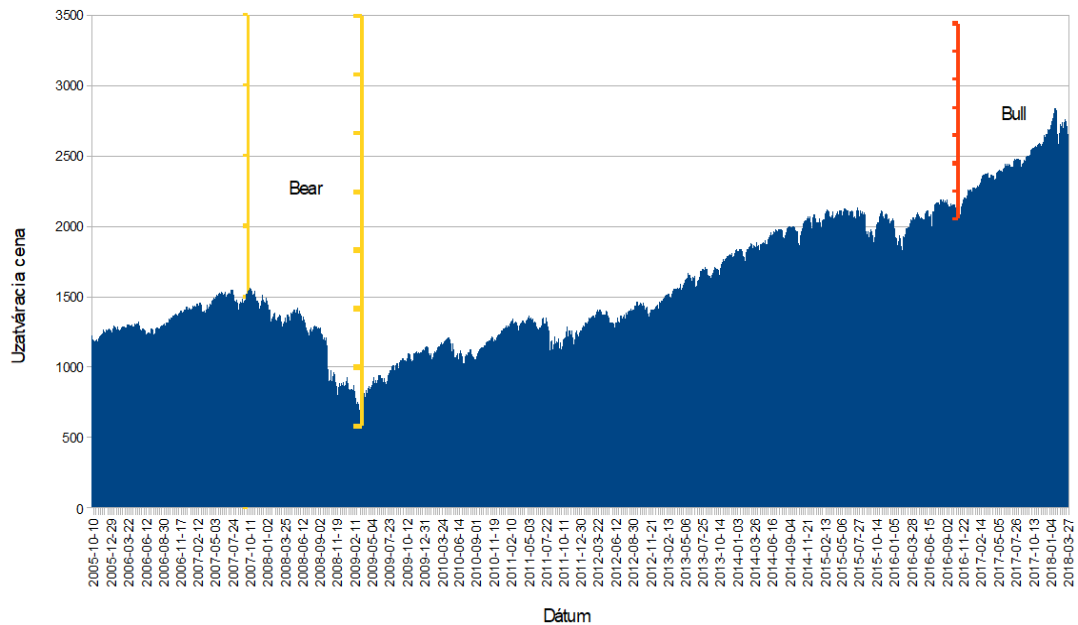
V tejto kapitole sa venujeme experimentom s našou aplikáciou, konkrétne riešime úlohu predpovedania vývoja cien akcií. Súčasťou je podrobná analýza výsledkov našej práce v podobe grafov a tabuliek. Dosiahnuté výsledky takisto porovnáваме v kontexte iných existujúcich prác v oblasti predpovedania vývoja cien akcií spomínaných v sekcii 3.3, konkrétne sú to nástroj Stocker, ARMA model a Buy And Hold stratégia. Ak sa na to pozrieme z pohľadu sekcií tak v sekcii 5.1 sa nachádza popis vstupných dát pre naše experimenty. V sekcii 5.2 je k dispozícii popis priebehu experimentov, ktorý pozostáva najmä z popisu nastavovania parametrov pre konkrétny typ predpovede. Sekcia 5.3 obsahuje analýzu výsledkov našej aplikácie, to znamená porovnanie našej aplikácie s inými nástrojmi, porovnanie rozdielnych typov predpovedí, a porovnanie vplyvu bull a bear marketu na kvalitu predpovede.

5.1 Vstupné dáta

Ako vstup pre našu úlohu používame skutočné historické dáta firm, kde zdrojom dát je Yahoo finance [2]. Tieto dáta sú v rozmedzí dní a pozostávajú z niekoľkých hodnôt. Sú to: dátum (jednoznačný identifikátor konkrétneho dňa), otváracia cena (opening price), najvyššia cena počas daného dňa (high), najnižšia cena počas daného dňa (low), uzatváracia cena pre daný deň (closing price), upravená uzatváracia cena (adjusted closing price), objem (volume), ktorý reprezentuje počet transakcií, ktoré prebehli v danom dni. Pre účely našej práce počítame a snažíme sa predpovedať len jednu z týchto hodnôt a tou je tzv. closing price, teda cena akcií s akou sa uzavrelo obchodovanie v daný deň. Ako vstup pre našu sieť používame takisto len hodnoty closing price pre jednotlivé dni. Predpovedanie vývoja cien v období, keď sa nedejú na trhu výrazné zmeny na trhu je jednoduchšie, než v období výrazných zmien. Aby sme mohli dôkladnejšie porovnať výsledky predpovedí, tak budeme pre každú firmu predpovedať 2 obdobia: obdobie bear marketu a obdobie bull marketu definované v sekcii 3.1. Vývoj cien akcií konkrétnej firmy v jednom období sa môže výrazne líšiť od vývoja v druhom období, na druhej strane rovnaké obdobie aj pri rozdielnych firmách môže mať spoločné charakteristiky vývoja ceny akcií. Obdobím bear marketu, teda ceny akcií klesajú, pre účely tejto práce rozumieme obdobie od októbra 2007 do marca 2009. Ako príklad bull marketu používame obdobie od septembra 2016 do apríla 2018. Tieto obdobia sme zvolili kvôli tomu, že majú výrazné črty bear resp. bull marketu čomu odpovedá graf 5.1.

Čo sa týka rozdelenia vstupných dát, prvých 20 vstupov tvorí počiatočnú množinu vstupných dát, nasleduje 750 tréningových vstupov a 250 vstupov používame na testovanie. Na tréningovanie používame ceny akcií počas obdobia 3 rokov pred testovacím obdobím. Konkrétne teda pre bear market je to obdobie od januára 2005 do januára 2008, a následne testujeme na období od 1.1.2008 do 1.1.2009. Tréningovanie pre bull market prebieha od januára 2014 do januára 2017 a testovanie od 1.1.2017 do 1.1.2018.

Počítame 2 typy predpovedí: dlhodobú a krátkodobú - viz oddiel 4.1. Ďalej rozlišujeme aj 2 prístupy k úprave pôvodných dát do vhodnejšej podoby pre ESN.



Obr. 5.1: Vývoj uzatváraciej ceny indexu S&P 500.

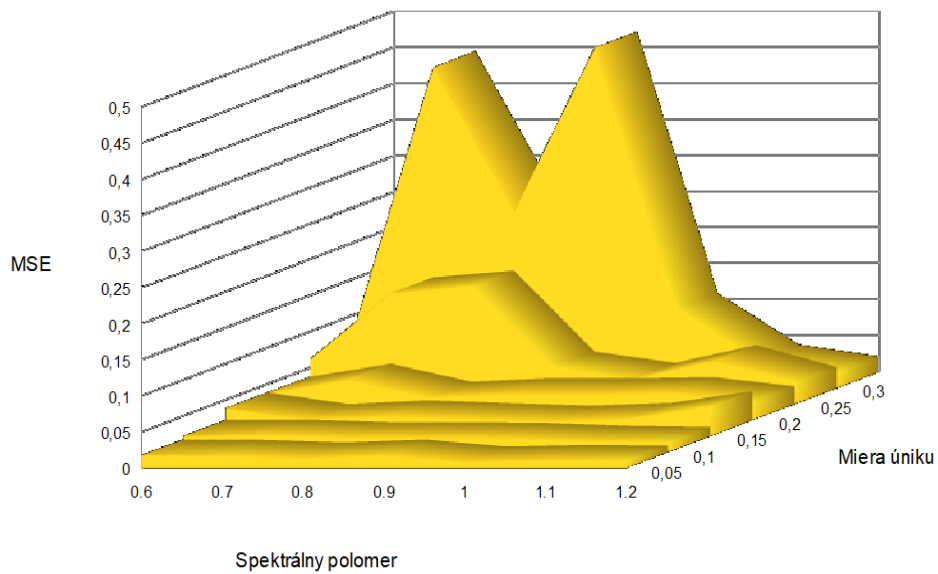
Oba prístupy sme v priebehu tejto práce už videli. Sú to lineárne mapovanie do intervalu $[-1,1]$ podľa rovnice (2.1) a lineárne mapovanie percentuálnej zmeny hodnoty tiež do intervalu $[-1,1]$ použitím vzorcov (2.2) a (2.1).

5.2 Priebeh experimentov

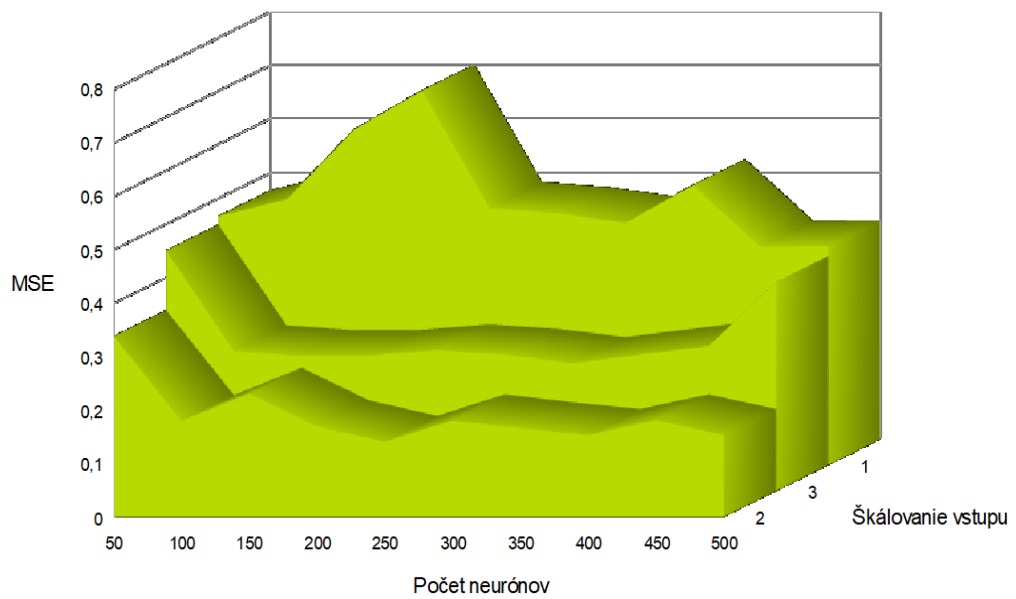
Aby výsledky našej práce boli spoľahlivejšie je potrebné vykonať niekoľko iterácií výpočtu ESN a vyhodnotiť spočítané hodnoty štatisticky. Z tohto dôvodu počítame pre každú zo zvolených firiem 20 iterácií kompletného výpočtu s rôznymi hodnotami náhodného semienka, a výsledok dostávame ako aritmetický priemer všetkých iterácií.

Pre dosiahnutie čo najlepších výsledkov je samozrejme kľúčové nastaviť, pokiaľ je to možné, ideálne hodnoty parametrov. Ako sme si uvedomili v sekcii 1.3, je ladenie parametrov komplikovaná úloha a navyše sa potrebné hodnoty líšia v závislosti od konkrétnej úlohy, pre rôzne akcie sú potrebné rôzne nastavenia, kde rôznymi akciami myslíme aj akcie jednej firmy v rôznom období (bull alebo bear market).

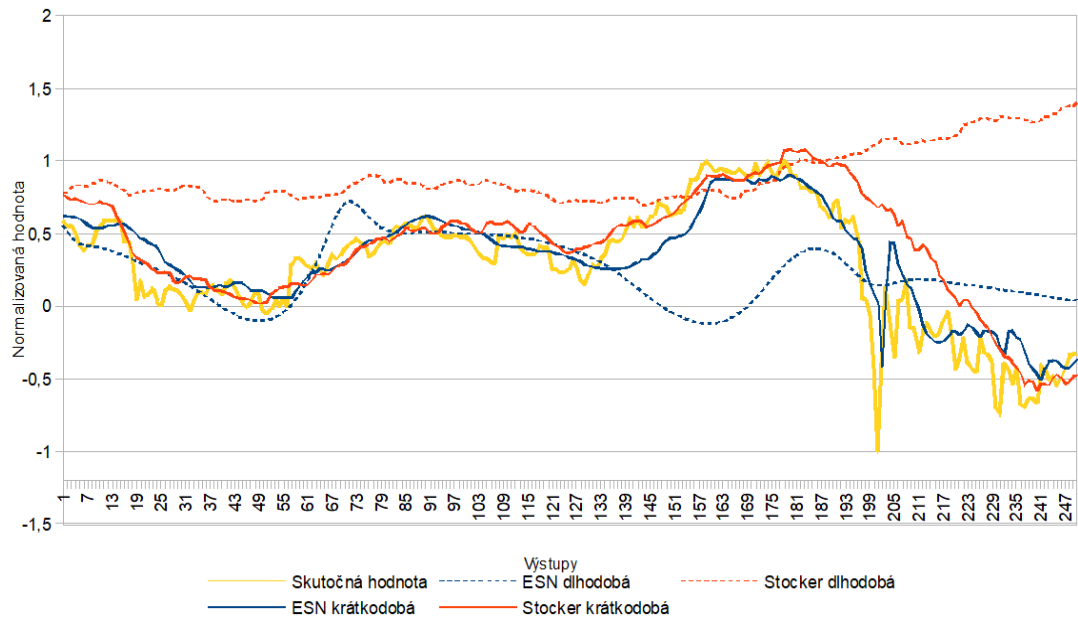
Experimenty sme vykonali na 10 najväčších firmách na Newyorskej burze cených papierov (NYSE), ktorých dáta sú k dispozícii vo vybraných obdobiach. Pre každú firmu sme predpovedali vývoj ceny v období bull a bear marketu. Zisk sme počítali jednoduchším spôsobom viz. 3.4, a rovnakým spôsobom počítal zisk aj Stocker a ARMA model. Pre každé vstupné dáta sme optimalizovali nastavenie parametrov pre dosiahnutie minimálnej hodnoty priemernej kvadratickej chyby. Proces optimalizácie prebiehal v súlade s doporučenými postupmi ako kombinácia manuálneho nastavovania a použitia grid searchu.



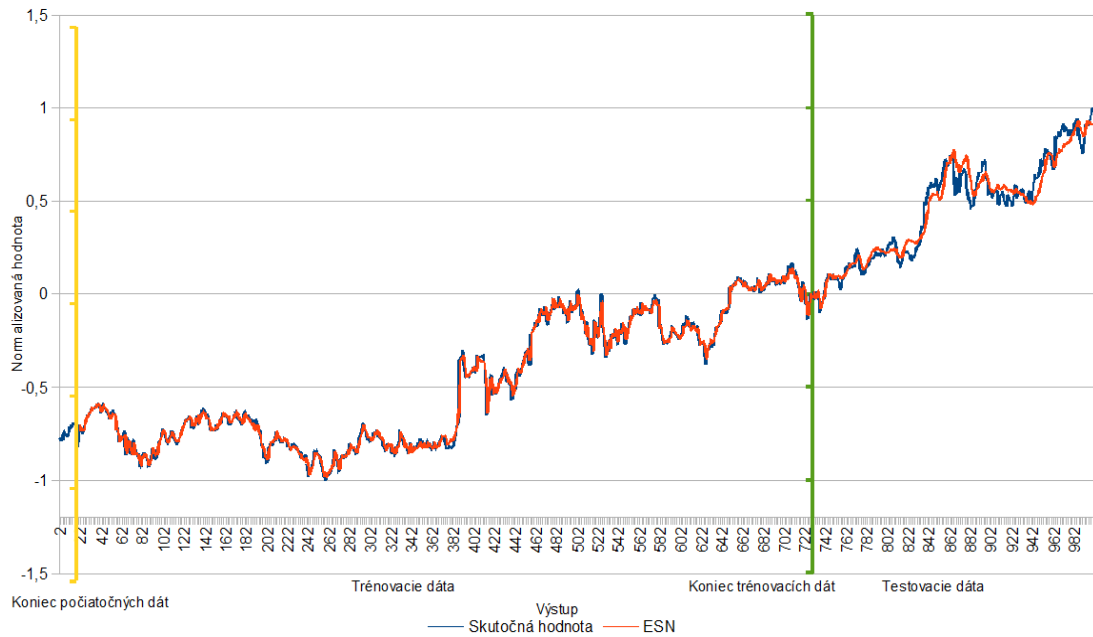
Obr. 5.2: Grid search pre krátkodobú predpoveď na akciách AMZN (bear market).



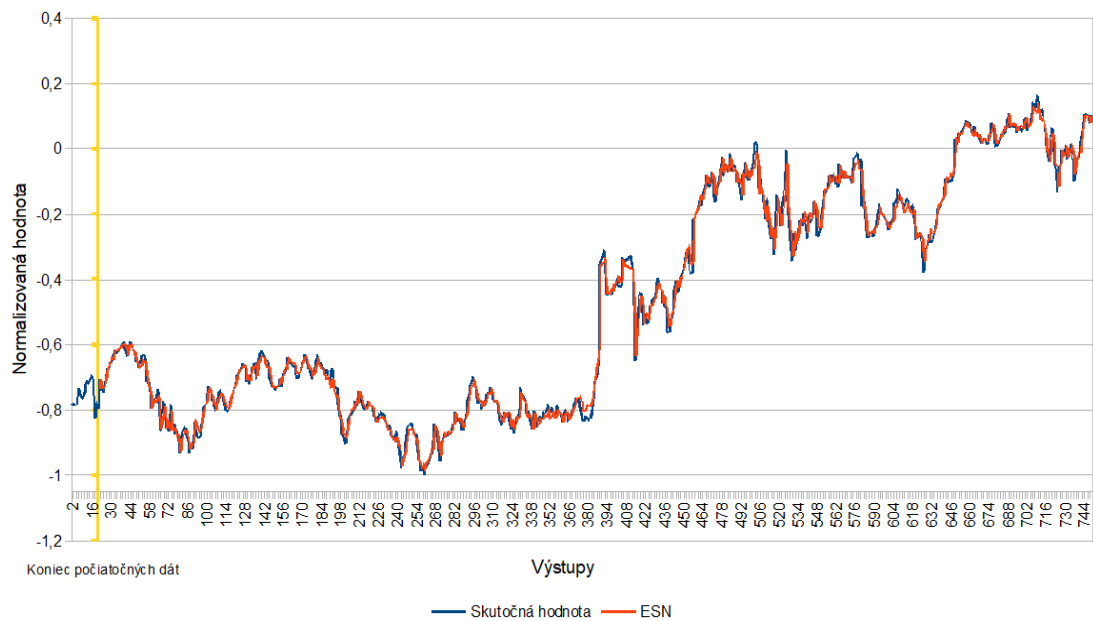
Obr. 5.3: Grid search pre dlhodobú predpoveď na akciách WMT (bull market).



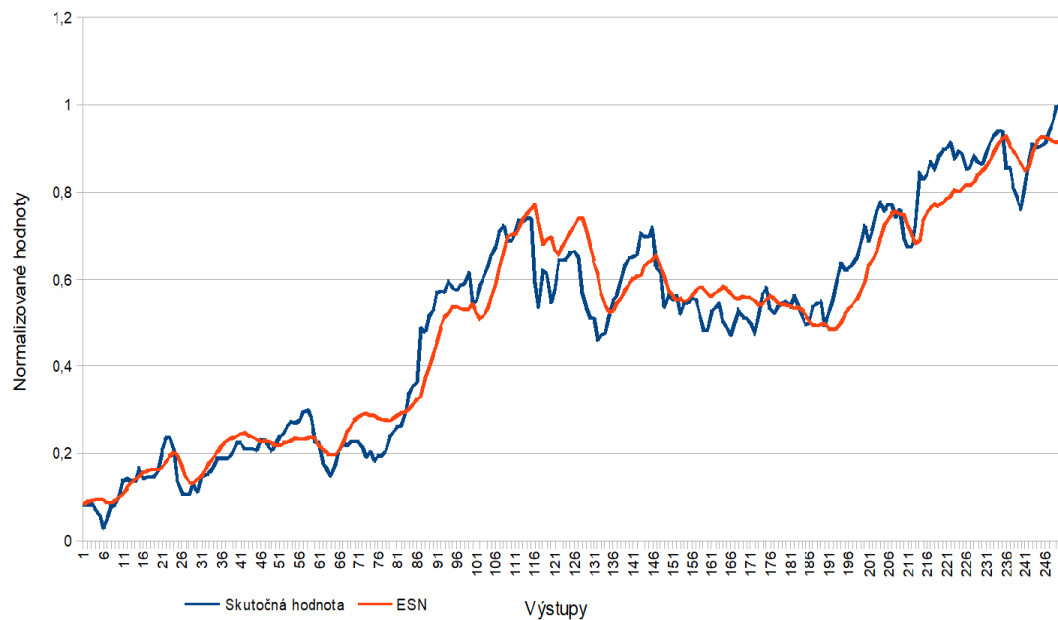
Obr. 5.4: Predpoveď ESN v porovnaní s predpoveďou Stockeru pre JNJ (bear market).



Obr. 5.5: Vstupné dáta a krátkodobá predpoveď ESN pre GOOGL (bull market).



Obr. 5.6: Vstupné dáta a krátkodobá predpoveď ESN na tréningových dátach pre GOOGL (bull market).



Obr. 5.7: Vstupné dáta a krátkodobá predpoveď ESN na testovacích dátach pre GOOGL (bull market).

5.2.1 Krátkodobá predpoveď

Pri krátkodobej predpovedi sa sieť správa stabilnejšie, a preto si môžeme dovoliť aj vyššie hodnoty spektrálneho polomeru a zároveň menšie hodnoty miery úniku, čo vedie k presnejším predpovediam. Výsledky grid searchu na týchto dvoch parametroch pre krátkodobú predpoveď vývoja cien akcií Amazon (AMZN) môžeme vidieť na grafe 5.2. Všetky iterácie počas GS sú počítané s rezervoárom veľkosti 100. Vidíme, že najlepšie výsledky sieť dosahuje so spektrálnym polomerom 1.2 čím potvrdzujeme naše tvrdenie zo sekcie 1.3.3, že vlastnosť echo state môže byť platná aj pri hodnotách spektrálneho polomeru väčších ako 1. Čo sa týka miery úniku, ideálne hodnoty sa pohybujú blízko nuly. Podobné výsledky sme dosiahli aj pri ladení parametrov na iných akciách. Na základe týchto výsledkov sme krátkodobú predpoveď našej aplikácie pre všetky akcie počítali s nastavením parametrov, ktoré môžeme vidieť v tabuľke 5.7. Používame teda výrazne väčší

Firma	Zisk [%]	Chyba test ¹ [%]	Chyba tréning ² [%]	Správne rast ³ [%]	Správne pokles ⁴ [%]
Bull market					
AMZN	41,29	32,47	0,91	58,43	37,23
BAC	20,20	9,96	0,93	51,64	44,9
BRK	13,09	8,67	0,53	56,14	41,92
GOOGL	25,99	17,61	0,77	63,13	39,88
JNJ	22,03	9,00	0,55	60,55	46,63
JPM	22,68	10,61	0,85	54,31	46,15
MSFT	40,54	6,51	0,81	58,23	—
WFC	6,22	7,99	0,76	52,87	45,46
WMT	17,10	9,64	0,60	58,45	39,02
XOM	-4,51	4,45	0,71	53,27	50,46
Bear market					
AMZN	-28,70	33,56	1,16	47,38	51,86
BAC	-44,35	29,42	0,68	36,07	61,62
BRK	-20,15	17,41	0,43	44,80	52,93
GOOGL	-66,58	49,13	1,05	41,97	58,68
JNJ	4,19	4,94	0,43	50,13	54,46
JPM	-3,34	19,00	0,76	42,21	59,62
MSFT	-19,77	9,99	0,65	45,65	57,74
WFC	-18,99	9,37	0,68	39,8	55,86
WMT	3,87	14,05	0,66	48,5	47,22
XOM	-14,49	12,82	0,74	50,38	51,57

*Pozn:*¹ Priemerná chyba na testovacích dátach.

² Priemerná chyba na tréningových dátach.

³ Počet dní keď model správne predpovedal, že cena bude rásť.

⁴ Počet dní keď model správne predpovedal, že cena bude klesať.

Tabuľka 5.1: Výsledky dlhodobých predpovedí našej ESN aplikácie.

rezervoár ako pri GS, avšak počas práce s testovaním siete a z výsledkov experimentov sme vypožorovali, že väčší rezervoár neovplyvňuje zásadne nastavenie ostatných parametrov. Výsledky sú zhrnuté v tabuľke 5.3.

5.2.2 Dlhodobá predpoveď

Pri dlhodobej predpovedi bolo nájdenie optimálnych hodnôt parametrov výrazne komplikovanejšie, keďže tieto hodnoty sa výrazne líšili medzi akciami. Pri niektorých akciách bola sieť stabilná len pri veľmi úzkom spektre hodnôt parametrov. Opäť najväčší dôraz pri ladení vyžadovali hodnoty spektrálneho polomeru a miery úniku. Priemerná hodnota optimálneho spektrálneho polomeru pre všetky akcie bola 0.77 a priemerná hodnota miery úniku 0.11. Zaujímavé je však, že veľkosť rezervoáru nemala až taký vplyv na výslednú sieť, ako by sme očakávali. Tento fakt môžeme vidieť na obrázku 5.3, pričom sme po vyhodnotení pre túto

Firma	Zisk [%]	Chyba test ¹ [%]	Chyba tréning ² [%]	Správne rast ³ [%]	Správne pokles ⁴ [%]
Bull market					
AMZN	48,53	30,53	2,57	56,72	44,74
BAC	32,80	11,50	2,57	47,34	53,75
BRK	13,57	3,46	1,22	50,63	38,46
GOOGL	7,46	20,51	2,01	55,56	43,90
JNJ	17,71	8,91	1,13	54,48	53,91
JPM	22,68	5,49	1,57	48,47	41,51
MSFT	26,70	13,64	1,92	54,30	42,86
WFC	12,41	4,66	1,79	47,33	53,54
WMT	35,73	7,81	1,17	62,26	46,67
XOM	-5,08	13,51	1,43	50,35	46,23
Bear market					
AMZN	-14,03	75,03	4,35	46,39	62,35
BAC	-26,48	11,97	1,41	41,58	55,33
BRK	-43,25	30,83	1,04	43,11	58,33
GOOGL	-58,73	80,40	2,86	40,13	56,38
JNJ	-0,23	6,73	0,81	45,86	52,13
JPM	34,78	7,64	1,59	51,75	62,04
MSFT	-35,40	32,22	1,72	38,61	53,76
WFC	44,12	9,62	1,08	49,38	55,29
WMT	4,09	16,07	1,64	52,34	46,34
XOM	-16,72	18,29	1,69	46,81	50,91

*Pozn.*¹ Priemerná chyba na testovacích dátach.

² Priemerná chyba na tréningových dátach.

³ Počet dní keď model správne predpovedal, že cena bude rásť.

⁴ Počet dní keď model správne predpovedal, že cena bude klesať.

Tabuľka 5.2: Výsledky dlhodobých predpovedí Stockeru

konkrétnu akciu použili sieť s nastavením zobrazeným v tabuľke 5.8.

Matica spätných spojov nebola použitá ani v jednom prípade. V oboch prípadoch totiž používame spočítaný výstup v jednom kroku ako vstup v ďalšom kroku. Tento vstup prechádza maticou \mathbf{W}_{in} takže použitie \mathbf{W}_{fb} v tomto prípade nedáva zmysel.

5.2.3 Stocker

Na identických dátach sme spustili aj nástroj Stocker [9] a výsledky sú k dispozícii v tabuľke 5.2 pre dlhodobú predpoveď a 5.4 pre krátkodobú predpoveď. Nástroj Stocker dáva zakaždým rovnaké výsledky takže sme počítali len 1 iteráciu. Stocker počíta zisk jednoduchým spôsobom, ktorý sme videli v sekcii 3.4.

Firma	Zisk [%]	Chyba test ¹ [%]	Chyba tréning ² [%]	Správne rast ³ [%]	Správne pokles ⁴ [%]
Bull market					
AMZN	44,10	3,47	0,91	62,72	44,36
BAC	25,52	3,39	0,93	52,00	47,47
BRK	22,58	1,28	0,53	59,18	46,07
GOOGL	28,88	2,00	0,76	63,13	39,88
JNJ	25,02	1,33	0,55	64,08	53,27
JPM	15,51	1,94	0,85	51,72	41,34
MSFT	27,35	1,96	0,81	56,79	39,08
WFC	15,94	3,15	0,76	55,88	49,55
WMT	19,88	1,95	0,60	58,27	39,79
XOM	-1,27	1,53	0,71	52,94	50,76
Bear market					
AMZN	-24,15	6,96	1,16	44,16	49,71
BAC	-17,16	5,12	0,68	32,57	59,42
BRK	-8,53	3,50	0,43	45,71	52,77
GOOGL	-34,61	4,67	1,05	45,74	60,45
JNJ	-1,98	1,93	0,43	50,48	54,79
JPM	-4,34	5,94	0,76	42,85	60,77
MSFT	-41,81	3,66	0,65	40,00	54,08
WFC	17,75	4,95	0,68	46,42	62,04
WMT	-4,21	2,68	0,66	52,11	50,46
XOM	-43,40	3,27	0,74	46,15	48,48

*Pozn.*¹ Priemerná chyba na testovacích dátach.

² Priemerná chyba na tréningových dátach.

³ Počet dní keď model správne predpovedal, že cena bude rásť.

⁴ Počet dní keď model správne predpovedal, že cena bude klesať.

Tabuľka 5.3: Výsledky krátkodobých predpovedí našej ESN aplikácie

5.2.4 ARMA

ARMA model nie je vhodný na dlhodobú predpoveď. Predpovedáme teda cenu akcií pre nasledujúci deň. To odpovedá našej krátkodobej predpovedi. V kapitole 4 popisujeme dva prístupy krátkodobej predpovede. Pre porovnanie s ARMA modelom používame krátkodobú predpoveď, ktorá využíva ako vstup pri testovaní skutočné hodnoty predchádzajúceho dňa. Vidíme tak zároveň rozdiely v oboch prístupoch pre krátkodobú predpoveď. Výsledky ARMA modelu sú k dispozícii v tabuľke 5.5, a výsledky našej vylepšenej krátkodobej predpovede sú k dispozícii v tabuľke 5.6. Pre ARMA model sa konkrétna voľba hodnoty parametrov p a q mohla líšiť pre rôzne akcie a rôzne stavy trhu, ale hodnoty vždy patrili do intervalov $1 \leq p \leq 5$ a $1 \leq q \leq 3$.

Firma	Zisk [%]	Chyba test ¹ [%]	Chyba tréning ² [%]	Správne rast ³ [%]	Správne pokles ⁴ [%]
Bull market					
AMZN	32,68	3,91	2,57	58,06	47,31
BAC	24,64	5,09	2,57	45,45	51,04
BRK	20,61	1,90	1,22	55,94	47,16
GOOGL	12,04	4,28	2,01	55,84	44,21
JNJ	17,32	1,86	1,13	55,26	56,7
JPM	11,79	3,10	1,57	47,62	45,09
MSFT	26,54	2,03	1,92	52,94	40,50
WFC	2,90	3,27	1,79	47,36	54,31
WMT	25,71	3,00	1,17	58,71	40,42
XOM	2,27	1,84	1,43	52,54	48,85
Bear market					
AMZN	-33,63	5,53	4,35	42,39	55,62
BAC	-35,63	5,53	1,41	40,00	53,50
BRK	-30,74	5,36	1,04	43,29	57,41
GOOGL	-20,59	12,56	2,86	41,51	58,21
JNJ	2,68	2,11	0,81	47,86	54,07
JPM	-34,95	4,79	1,59	43,47	54,74
MSFT	-18,30	4,28	1,72	38,77	55,84
WFC	-60,85	4,60	1,08	38,26	47,05
WMT	-14,64	2,46	1,64	51,29	43,87
XOM	-28,41	3,08	1,69	43,51	50,34

Pozn.¹ Priemerná chyba na testovacích dátach.

² Priemerná chyba na tréningových dátach.

³ Počet dní keď model správne predpovedal, že cena bude rásť.

⁴ Počet dní keď model správne predpovedal, že cena bude klesať.

Tabuľka 5.4: Výsledky krátkodobých predpovedí Stockeru

5.3 Analýza výsledkov

Ako môžeme vidieť v tabuľkách s výsledkami, naša aplikácia dosiahla výsledky porovnateľné s výsledkami nástroja Stocker. Pri dlhodobej predpovedi dosahuje naša aplikácia mierne menšie hodnoty priemernej chyby, teda naša aplikácia lepšie aproximuje skutočný vývoj ceny akcií a získavame presnejšiu predpoveď. Avšak čo sa týka zisku, dosiahla naša aplikácie mierne horšie výsledky teda nižší zisk. Podstatu tohto problému sme ukázali na ilustračnom príklade 3.1, teda predpoveď Stockeru vystihuje lepšie fluktáciu ceny konkrétnej akcie, ale horšie aproximuje celkový vývoj ceny. Rozdiely v dosahovaných výsledkoch Stockeru a našej aplikácie pri úlohe dlhodobej predpovede nie sú výrazné. Naopak citeľnejšie rozdiely môžeme vidieť pri úlohe krátkodobej predpovede, kde naša aplikácia dosahuje pravidelne menšie hodnoty priemernej chyby v porovnaní so Stockerom.

Čo sa týka porovnanie krátkodobej a dlhodobej predpovede nášho nástroja, vidíme, že pri krátkodobej predpovedi dosahujeme výrazne menšiu priemernú

Firma	Zisk [%]	Chyba test ¹ [%]	Správne rast ² [%]	Správne pokles ³ [%]
Bull market				
AMZN	41,60	1,31	59,28	47,74
BAC	19,27	1,33	42,62	47,28
BRK	22,55	0,63	56,11	47,23
GOOGL	15,90	0,92	58,15	46,36
JNJ	17,42	0,61	54,26	53,27
JPM	1,76	0,98	46,72	47,28
MSFT	11,63	0,87	50,00	38,05
WFC	7,34	0,89	49,48	56,15
WMT	21,97	0,74	57,93	41,5
XOM	-8,26	0,52	46,72	44,18
Bear market				
AMZN	-64,14	2,09	37,5	51,7
BAC	-59,85	2,70	39,09	51,77
BRK	16,08	1,84	50,90	63,83
GOOGL	-95,27	2,57	35,51	54,16
JNJ	-17,21	1,09	43,9	49,21
JPM	-78,81	3,53	40,00	52,20
MSFT	-58,18	2,00	38,73	55,71
WFC	-84,11	3,26	42,98	49,63
WMT	-11,94	1,65	52,27	45,37
XOM	-61,15	2,19	40,98	44,96

*Pozn.*¹ Priemerná chyba na testovacích dátach.

² Počet dní keď model správne predpovedal, že cena bude rásť.

³ Počet dní keď model správne predpovedal, že cena bude klesať.

Tabuľka 5.5: Výsledky krátkodobých predpovedí ARMA modelu

chybu, čo samozrejme súvisí s princípom výpočtu, teda opakované tréningovanie je síce časovo veľmi náročné, ale výsledkom je výrazne lepšia predpoveď z pohľadu

Firma	Zisk [%]	Chyba test¹ [%]	Správne rast² [%]	Správne pokles³ [%]
Bull market				
AMZN	74,44	1,36	67,13	48,11
BAC	5,17	1,47	54,61	50,42
BRK	18,28	0,63	55,88	41,59
GOOGL	43,95	0,93	66,89	45,54
JNJ	25,81	0,60	62,93	51,88
JPM	17,68	1,08	54,41	45,13
MSFT	39,72	0,87	60,54	44,11
WFC	14,37	0,97	52,59	45,61
WMT	20,97	0,81	56,84	36,89
XOM	4,56	0,49	55,11	52,45
Bear market				
AMZN	-1,85	3,70	48,73	53,07
BAC	25,70	2,44	48,48	69,33
BRK	22,49	2,10	49,56	58,2
GOOGL	-18,34	2,50	45,28	60,83
JNJ	-0,91	1,05	53,50	57,77
JPM	30,31	2,87	46,15	62,75
MSFT	-14,94	1,73	46,42	58,39
WFC	26,59	2,95	49,53	64,08
WMT	-4,33	1,50	49,60	48,36
XOM	-8,76	1,98	49,58	51,56

*Pozn.*¹ Priemerná chyba na testovacích dátach.

² Počet dní keď model správne predpovedal, že cena bude rásť.

³ Počet dní keď model správne predpovedal, že cena bude klesať.

Tabuľka 5.6: Výsledky vylepšenej verzie krátkodobých predpovedí ESN

Názov parametra	Hodnota
spektrálny polomer	1,25
miera úniku	0,01
počet neurónov	500
dolná medz (W_{in} a W)	-1,0
horná medz (W_{in} a W)	1,0
počiatočná dĺžka	20
trénovacia dĺžka	750
testovacia dĺžka	250

Tabuľka 5.7: Nastavenie parametrov pre krátkodobú predpoveď s našim ESN nástrojom

minimalizovania priemernej chyby. Priemerná chyba pri dlhodobej predpovedi je približne 15%, pri krátkodobej predpovedi len 3%. Zaujímavým pozorovaním je však fakt, že krátkodobá predpoveď nedosahuje zásadne lepšie výsledky očakávaného zisku ako dlhodobá predpoveď. Celkovo dosiahla naša aplikácia lepší zisk ako Stocker aj ako stratégia Buy And Hold (B&H) viz tabuľka 5.9. Zjednodušene je B&H pasívna stratégia, pri ktorej investor nakúpi akcie na dlhé obdobie pričom ho nezaujímajú fluktácie na trhu, a pre vyhodnotenie stratégie sa používa rozdiel cien akcie na konci a na začiatku obdobia, viac [1].

Vylepšená verzia krátkodobej predpovede dosahuje podľa očakávania kvalitnejšie predpovede než obyčajná krátkodobá predpoveď vo všetkých ohľadoch. To samozrejme vyplýva z princípu implementácie, keďže dostáva presnejšiu informáciu na vstupe. Takisto pozorujeme, že táto verzia predpovede nevyžaduje až také úzke spektrum možných nastavení parametrov. Pri porovnávaní s ARMA modelom sme sieť spúšťali so spektrálnym polomerom 1, mierou úniku 0.01 a veľkosťou siete 300 neurónov.

Pri porovnaní ARMA modelu a vylepšenej verzie krátkodobej predpovede vidíme, že dosahujú takmer rovnaké hodnoty priemernej chyby. To znamená že ARMA model veľmi dobre aproximuje skutočné hodnoty vývoja cien avšak nezachycuje až tak dobre denné zmeny cien akcií, čo sa odzrkadľuje pri menších percentách správne predpovedaného rastu resp. poklesu, a najväčšie rozdiely pozorujeme pri predpokladanom zisku.

Z hľadiska rozdielov medzi bull a bear marketom vidíme, že nástroj dosahuje menšiu chybu pri predpovedi bull marketu. Záleží samozrejme od konkrétneho obdobia, z ktorého čerpáme vstupné dáta. Rozdiel vo výsledkoch našej práce medzi týmito obdobiami je však rozhodne citelný a pravdepodobne nie je výsledkom náhody (12% ku 6%).

Názov parametra	Hodnota
spektrálny polomer	0,95
miera úniku	0,05
počet neurónov	100
dolná medz (\mathbf{W}_{in})	-2,0
horná medz (\mathbf{W}_{in})	2,0
počiatočná dĺžka	20
trénovacia dĺžka	750
testovacia dĺžka	250

Tabuľka 5.8: Nastavenie parametrov pre dlhodobú predpoveď akcie WMT bull market s našim ESN nástrojom

Postup	Zisk Bear¹ [%]	Zisk Bull² [%]	Zisk Avg.³ [%]
Krátkodobá predpoveď			
ESN	-16,24	22,35	3,05
Stocker	-27,53	17,65	-5,03
B&H	-33,95	31,37	-1,29
ARMA	-51,46	15,12	-18,17
Dlhodobá predpoveď			
ESN	-20,83	20,73	-0,05
Stocker	-11,18	21,25	-4,93
B&H	-33,95	31,37	-1,29
Vylepšená krátkodobá predpoveď			
ESN	5,59	26,5	16,04

*Pozn.*¹ Priemerný zisk počas bear marketu.

² Priemerný zisk počas bull marketu.

³ Priemerný zisk za celé obdobie (bear aj bull).

Tabuľka 5.9: Zisk jednotlivých nástrojov a postupov

Záver

V tejto práci sme vytvorili nástroj pre rýchle vytváranie, trénovanie a testovanie umelých neurónových sietí nazývaných echo state siete. Vytvorený nástroj sme následne použili na predpovedanie časových radov. Konkrétne sme sa zaoberali predpovedou vývoja cien akcií na burze. Zároveň sme sa snažili splniť všetky ciele, ktoré sme si stanovili v úvodných kapitolách.

Konkrétne sme:

- Predstavili echo state siete ako výpočtový model schopný naučiť sa aproximovať funkcie.
- Popísali ESN prístup a detailne sa pozreli na princíp fungovania a na rôzne upravené varianty pôvodného ESN prístupu.
- Porovnali a popísali možnosti ako výsledné siete porovnávať.
- Vysvetlili a poukázali na problematiku hľadania optimálneho nastavenia parametrov siete, ukázali sme parametre, ktoré majú na výslednú kvalitu siete rozhodujúci vplyv, a ponúkli princípy a heuristiky ako pri ladení týchto parametrov postupovať.
- Popísali existujúce nástroje a implementácie v oblasti echo state sietí.
- Implementovali vlastný nástroj pre prácu s ESN a zdôvodnili výber konkrétne použitých technológií pri vývoji nástroja.
- Vytvorili grafické užívateľské rozhranie ako súčasť nášho nástroja, pre pohodlné, prehľadné a rýchle vytváranie a experimentovanie s vytvorenými modelmi s pohľadu užívateľa.
- Ponúkli užívateľovi možnosť použiť tento nástroj na predpovedanie vývoja ľubovolnej postupnosti.
- Predstavili sme predpoveď vývoja cien akcií na burze ako jednu z možných aplikácií nášho nástroja a ESN sietí všeobecne. Pri tom sme sa mierne ponorili do problematiky komplexnosti a zložitosti, ako takúto predpoveď realizovať, a do dôvodov, prečo sú ESN na túto úlohu použiteľné.
- Spomedzi existujúcich prác v oblasti predpovedania vývoja cien akcií sme vybrali tie, ktoré sme mohli porovnať s našou prácou. Nakoniec to boli nástroje Stocker, ARMA model a jednoduchá stratégia Buy&Hold. Museli sme teda vyriešiť problém, ako tieto nástroje porovnávať.
- Aplikovali a upravili sme náš nástroj na riešenie úlohy predpovede vývoja cien akcií s použitím reálnych historických dát skutočných firiem.
- Výsledné predpovede sme analyzovali, a poukázali sme na zaujímavé zistenia a výsledky.

Na záver by sme ešte chceli zhrnúť podstatné poznatky našej práce. Naš nástroj pri experimentovaní s predpoveďou vývoja cien akcií obstál v konkurencii iných nástrojov, dosiahol porovnateľné výsledky s inými nástrojmi. Naša aplikácia je dostatočne obecná na predpovedania vývoja akejkoľvek časovej rady. Je vhodná na opakované skúšanie, testovanie a experimentovanie, keďže ponúka dostatočnú rýchlosť výpočtu. Navyše má užívateľ k dispozícii intuitívne GUI, prostredníctvom ktorého má k dispozícii na jednom mieste, pohodlne všetky potrebné nastavenia pri vytváraní siete. Aplikácia je takisto postavená tak, aby sa v budúcnosti dali implementovať rôzne ďalšie varianty ESN napr.:

- Využitie PCA (principal component analysis) pri predspracovaní vstupov, čo odfiltruje menej dôležité informácie viz [11].
- Model MI-ESN, ktorý sa snaží optimalizovať štruktúru siete pomocou tzv. mutual information, to znamená na základe korelácií medzi vstupmi a výstupmi siete [26].
- Špeciálny typ deterministických ESN z [5], ktoré sa snažia odstrániť potrebu opakovane skúšať a testovať náhodne generované ESN.

Zoznam použitej literatúry

- [1] Buy and hold, investopedia. <https://www.investopedia.com/terms/b/buyandhold.asp>. [Online] [Citácia 5.7.2018].
- [2] Yahoo! finance. <https://finance.yahoo.com>. [Online] [Citácia 3.7.2018].
- [3] ARFKEN, G. (1985). The method of steepest descents. *Mathematical methods for physicists*, **3**, 428–436.
- [4] BOX, G. E., JENKINS, G. M., REINSEL, G. C. a LJUNG, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.
- [5] DAN, J., GUO, W., SHI, W., FANG, B. a ZHANG, T. (2014). Deterministic echo state networks based stock price forecasting. In *Abstract and Applied Analysis*, volume 2014. Hindawi.
- [6] HOLZMANN, G. *aureservoir, Efficient C++ library for analog reservoir computing neural networks (Echo State Networks)*. [Online] [Citácia 21.6.2018]. Dostupné z <http://aureservoir.sourceforge.net>.
- [7] ILIES, I., JAEGER, H., KOSUCHINAS, O., RINCON, M., SAKENAS, V. a VASKEVICIUS, N. (2007). Stepping forward through echoes of the past: forecasting with echo state networks. *Short report on the winning entry to the NN3 financial forecasting competition, Dostupné online z http://www.neural-forecasting-competition.com/downloads/NN3/methods/27-NN3_Herbert_Jaeger_report.pdf*, **53**, 76.
- [8] JAEGER, H. (2001). The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, **148**(34), 13.
- [9] KOEHRSEN, W. (2018). Stock prediction in python. *Towards Data Science, Dostupné online z <https://towardsdatascience.com/stock-prediction-in-python-b66555171a2>*.
- [10] LI, G., LI, B.-J., YU, X.-G. a CHENG, C.-T. (2015). Echo state network with bayesian regularization for forecasting short-term power production of small hydropower plants. *Energies*, **8**(10), 12228–12241. ISSN 1996-1073. doi: 10.3390/en81012228. URL <http://www.mdpi.com/1996-1073/8/10/12228>.
- [11] LIN, X., YANG, Z. a SONG, Y. (2009). Short-term stock price prediction based on echo state networks. *Expert systems with applications*, **36**(3), 7313–7317.
- [12] LOHMANN, J. a V.BUTZ, M. *Echo-State Network In Java: Introduction, Manual, and Evaluations*. [Online] [Citácia 21.6.2018]. Dostupné z http://www.uni-tuebingen.de/index.php?eID=tx_securedownloads&p=26136&u=0&g=0&t=1529688969&hash=3d067a32bb65c54f191c1f7160b9b8b7aa178a8e&file=/fileadmin/Uni_

Tuebingen/Fakultaeten/InfoKogni/WSI/KogModel/ext00209/user_upload/Publications/2010/ESNJava1.0Report.pdf.

- [13] LUKOŠEVIČIUS, M. (2012). A practical guide to applying echo state networks. In *Neural networks: Tricks of the trade*, pages 659–686. Springer.
- [14] MANDELBROT, B. B. (1997). The variation of certain speculative prices. In *Fractals and scaling in finance*, pages 371–418. Springer.
- [15] MICROSOFT. *Windows Presentation Foundation*. [Online] [Citácia 11.6.2018]. Dostupné z <https://docs.microsoft.com/en-us/dotnet/framework/wpf/>.
- [16] MICROSOFT. *Windows Forms*. [Online] [Citácia 11.6.2018]. Dostupné z <https://docs.microsoft.com/en-us/dotnet/framework/winforms/>.
- [17] MULIERI, P. A. (1989). *Stock market price behavior: Random walks and nonlinear dynamics*. PhD thesis, State University of New York at Stony Brook.
- [18] OPENSOURCE. *Math.NET Numerics*. [Online] [Citácia 11.6.2018]. Dostupné z <https://numerics.mathdotnet.com>.
- [19] OSBORNE, M. F. (1959). Brownian motion in the stock market. *Operations research*, **7**(2), 145–173.
- [20] SCHAETTI, N. *Reservoir Computing 2, An introduction to the Python framework Oger*. [Online] [Citácia 21.6.2018]. Dostupné z <http://www.nilsschaetti.ch/2018/01/30/introduction-reservoir-computing-2-oger-toolbox/>.
- [21] SU, J. (2015). Reservoir computing in forecasting financial markets. Master's thesis, Duke University.
- [22] TAYLOR, S. J. a LETHAM, B. (2018). Forecasting at scale. *The American Statistician*, **72**(1), 37–45.
- [23] TEAM, F. C. D. S. *Prophet, Forecasting at scale*. [Online] [Citácia 11.6.2018]. Dostupné z <https://facebook.github.io/prophet>.
- [24] VAN EYDEN, R. J. (1996). The application of neural networks in the forecasting of share prices.
- [25] VERSTRAETEN, D., SCHRAUWEN, B., DIELEMAN, S., BRAKEL, P., BUTENEERS, P. a PECEVSKI, D. (2012). Oger: modular learning architectures for large-scale sequential processing. *Journal of Machine Learning Research*, **13**(Oct), 2995–2998.
- [26] WANG, H., NI, C. a YAN, X. (2017). Optimizing the echo state network based on mutual information for modeling fed-batch bioprocesses. *Neuro-computing*, **225**, 111–118.
- [27] ZHANG, H., LIANG, J. a CHAI, Z. (2013). Stock prediction based on phase space reconstruction and echo state networks. *Journal of Algorithms & Computational Technology*, **7**(1), 87–100.

Zoznam obrázkov

1.1	ESN schéma [10].	6
2.1	Input layer – vstupná vrstva	17
2.2	Reservoir	18
2.3	Output layer – výstupná vrstva	19
2.4	I/O nastavenia	20
2.5	Data preview formulár	21
2.6	Formulár nastavenia typu úlohy	22
2.7	Grid search nastavenia	24
2.8	Možnosti Map&Profit	25
2.9	Príklad nastavenia rezervoáru	28
2.10	Nastavenie Task center v príklade	30
2.11	Príklad formátu vstupného súboru	31
2.12	Computing progress v príklade	31
2.13	Ukážka súboru s príponou <code>_out.csv</code> v príklade 2.4.15	31
2.14	Ukážka časti súboru s príponou <code>_outdetail.csv</code> v príklade 2.4.15	32
3.1	Príklad vzťahu MSE a zisku.	37
4.1	Vzhľad formulárovej triedy <code>CreateNewESNForm</code>	43
5.1	Vývoj uzatváracej ceny indexu S&P 500.	46
5.2	Grid search pre krátkodobú predpoveď na akciách AMZN (bear market).	47
5.3	Grid search pre dlhodobú predpoveď na akciách WMT (bull market).	47
5.4	Predpoveď ESN v porovnaní s predpoveďou Stockeru pre JNJ (bear market).	48
5.5	Vstupné dáta a krátkodobá predpoveď ESN pre GOOGL (bull market).	48
5.6	Vstupné dáta a krátkodobá predpoveď ESN na tréningových dátach pre GOOGL (bull market).	49
5.7	Vstupné dáta a krátkodobá predpoveď ESN na testovacích dátach pre GOOGL (bull market).	49

Zoznam tabuliek

2.1	Ukážka časti konfiguračného súboru	29
5.1	Výsledky dlhodobých predpovedí našej ESN aplikácie.	50
5.2	Výsledky dlhodobých predpovedí Stockeru	51
5.3	Výsledky krátkodobých predpovedí našej ESN aplikácie	52
5.4	Výsledky krátkodobých predpovedí Stockeru	53
5.5	Výsledky krátkodobých predpovedí ARMA modelu	54
5.6	Výsledky vylepšenej verzie krátkodobých predpovedí ESN	55
5.7	Nastavenie parametrov pre krátkodobú predpoveď s našim ESN nástrojom	55
5.8	Nastavenie parametrov pre dlhodobú predpoveď akcie WMT bull market s našim ESN nástrojom	56
5.9	Zisk jednotlivých nástrojov a postupov	57

A. Prílohy

A.1 Prvá príloha

Súčasťou tejto bakalárskej práce je aj elektronická príloha ktorá obsahuje celý projekt a všetky programy. Máme ju rozdelenú do troch priečinkov:

- **ESN** Adresár so spustiteľným programom `ESNWinFormsRP.exe` a knižnicou `MathNet.Numerics.dll`.
- **ESNdata** Obsahuje všetky datové súbory. Vzorové formáty vstupných súborov, konfiguračný súbor, výsledky experimentov.
- **Project** Nachádza sa tu priečinok `ESNWinFormsRP`, ktorý obsahuje Visual Studio 2015 solution obsahujúci celý projekt vrátane dokumentácie.

Súčasťou tejto prílohy je aj text práce.