

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2016

Daniel Herbrych



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY**

**A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

## **PREDIKCE SPOTŘEBY ELEKTŘINY V DISTRIBUČNÍ SÍTI**

PREDICTION OF ELECTRICITY CONSUMPTION IN THE DISTRIBUTION NETWORK

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Daniel Herbrych**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Václav Uher**

**BRNO 2016**

# Bakalářská práce

bakalářský studijní obor **Teleinformatika**  
Ústav telekomunikací

**Student:** Daniel Herbrych

**ID:** 164730

**Ročník:** 3

**Akademický rok:** 2015/16

**NÁZEV TÉMATU:**

## Predikce spotřeby elektřiny v distribuční síti

**POKYNY PRO VYPRACOVÁNÍ:**

Cílem práce je vytvořit nástroj pro predikci spotřeby elektrické energie z poskytnutých dat. Navrhněte a implementujte metodu, která dokáže předpovídat spotřebu elektřiny v krátkodobém horizontu. Dosažené výsledky zhodnoťte z pohledu přesnosti.

**DOPORUČENÁ LITERATURA:**

[1] LI, Kangji, et al. Building's electricity consumption prediction using optimized artificial neural networks and principal component analysis. *Energy and Buildings*, 2015.

[2] PENTIUC, R., et al. System for Short Term Forecast of Electric Power Consumption in a Distributed Company. *Elektronika ir Elektrotechnika*, 2015, 107.1: 75-78.

[3] ESCRIVÁ-ESCRIVÁ, Guillermo; ROLDÁN-BLAY, Carlos; ÁLVAREZ-BEL, Carlos. Electrical consumption forecast using actual data of building end-use decomposition. *Energy and Buildings*, 2014, 82: 73-81.

**Termín zadání:** 1.2.2016

**Termín odevzdání:** 1.6.2016

**Vedoucí práce:** Ing. Václav Uher

**Konzultant bakalářské práce:**

**doc. Ing. Jiří Mišurec, CSc., předseda oborové rady**

**UPOZORNĚNÍ:**

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Tato práce se zabývá predikcí spotřeby elektřiny. Pro předpovídání spotřeby elektřiny slouží spousta metod v závislosti na délce předpovídaného horizontu. Zde byla pro predikci použita lineární regrese a v závěru je pak zhodnocena přesnost uvedené metody s danými parametry.

## **KLÍČOVÁ SLOVA**

predikce, časové řady, neuronové sítě, spotřeba elektřiny, lineární regrese

## **ABSTRACT**

This thesis deals with electricity consumption forecasting. A lot of methods depending on the length of the predicted horizon is used forecasting electricity consumption for. Linear regression was used in this thesis. Accuracy is evaluated in conclusion.

## **KEYWORDS**

prediction, time series, neural networks, electricity consumption, linear regression

HERBRYCH, Daniel *Predikce spotřeby elektřiny v distribuční síti*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 33 s. Vedoucí práce byl Ing. Václav Uher,



## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Predikce spotřeby elektřiny v distribuční síti“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora(-ky)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Václavu Uhrovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

podpis autora(-ky)

# OBSAH

Úvod	8
<b>1 Teoretická část</b>	<b>9</b>
1.1 Časová řada	9
1.1.1 Problémy časových řad	9
1.1.2 Význam analýzy časových řad	9
1.1.3 Základní metody analýzy časových řad	9
1.1.4 Predikce	10
1.2 Krátkodobá předpověď spotřeby elektrické energie	10
1.2.1 Exponenciální vyrovnávání	10
1.2.2 Umělé neuronové sítě	12
1.3 Střednědobá předpověď spotřeby elektrické energie	14
1.3.1 Dekompozice časových řad	14
1.3.2 Neuronové sítě v kombinaci s evolučními algoritmy	16
1.3.3 Metoda založená na extrakci trendu	17
1.4 Dlouhodobá předpověď spotřeby elektrické energie	22
1.4.1 Genetické programování	22
<b>2 Praktická část</b>	<b>24</b>
2.1 Popis programu	25
2.1.1 Třída <code>Read</code>	25
2.1.2 Třída <code>Data</code>	25
2.1.3 Třída <code>Error</code>	26
2.1.4 Třída <code>Generator</code> a její potomci	26
2.1.5 Třída <code>SmileOLS</code>	26
2.1.6 Ostatní třídy	26
2.2 Data	26
2.3 Výsledky	27
<b>3 Závěr</b>	<b>29</b>
<b>Literatura</b>	<b>30</b>
<b>Seznam symbolů, veličin a zkratk</b>	<b>31</b>
<b>Seznam příloh</b>	<b>32</b>
<b>A Obsah přiloženého CD</b>	<b>33</b>

## SEZNAM OBRÁZKŮ

1.1	Model neuronu, převzato z [5] . . . . .	13
1.2	Porovnání chyb MAPE, převzato z [3] . . . . .	21
1.3	Příklad stromu v GP, převzato z [4] . . . . .	22
2.1	Reálná a teoretická spotřeba elektřiny za leden 2009 . . . . .	27

# ÚVOD

Tato práce se věnuje problematice předpovídání spotřeby elektřiny, především pak budou uvedeny různé metody a modely. Důležitý při predikci spotřeby elektřiny je horizont předpovědi, metody pro predikci jsou pak rozděleny do kategorií v závislosti na délce horizontu. Některé metody jsou používány například jak pro krátkodobé předpovídání, tak pro střednědobé předpovídání. O takové metodě bude pojednáno pouze jednou.

Predikce v různých odvětvích, ne pouze předpovídání spotřeby elektřiny, se neustále vyvíjí a jsou testovány nové metody, modifikace starších metod anebo jejich kombinace. Dokonalým příkladem jsou neuronové sítě, které jsou v poslední době při predikci hojně využívány, a bude také uvedeny. Z pochopitelných důvodů nemohou být uvedeny všechny metody předpovídání, je jich mnoho.

V praktické části pak bylo cílem otestovat přesnost metody prokládání polynommem. Tento polynom měl být třetího stupně, avšak otestovány byly pro porovnání i polynomy vyšších řádů. Program pro předpovídání byl psán v jazyce Java a ve vývojovém prostředí Eclipse. V závěru je pak zhodnoceno, zda je tato metoda vhodná či nevhodná nebo jak by se popřípadě dala tato metoda vylepšit.

# 1 TEORETICKÁ ČÁST

## 1.1 Časová řada

Při předpovídání hodnot předpovídáme budoucí hodnoty tzv. časové řady. Časová řada (někde též chronologická řada) je sekvence dat, která jsou pozorována v určitém časovém intervalu. Časové řady lze využít v různých oborech, např. v ekonomii pro předpovídání vývoje ceny akcií, ve fyzice pro předpovídání změny teploty, v medicíně apod. Definice časové řady může být  $y_t$ , kde  $t=1, \dots, t=n$  a kde  $n$  je počet pozorování. Tato řada je diskrétní. Spojité časové řady se vyskytují třeba ve fyzice. Takové spojité řady se pak pro další práci tzv. diskretizují, případně agregují či průměrují.

### 1.1.1 Problémy časových řad

Při práci s časovými řadami se vyskytují určité problémy, se kterými je potřeba počítat. Jedná se především o:

- problémy s volbou časových bodů
- problémy s kalendářem - různá délka kalendářních měsíců, Velikonoce, apod.
- problémy s délkou řad
- problémy s nesrovnalostmi měření

### 1.1.2 Význam analýzy časových řad

Za cíl si analýza časových řad většinou dává vykonstruování určitého modelu, abychom pochopili, jak hodnoty časové řady vznikají. Konstrukce modelu také umožňuje do určité míry řízení a optimalizaci činnosti příslušného systému vhodnou volbou vstupních parametrů a počátečních podmínek. Dalším cílem analýzy časových řad bývá předpovídání.

### 1.1.3 Základní metody analýzy časových řad

Obecně metod analýzy časových řad je několik a každá je vhodná pro něco jiného. V dalších částech této práce budou uvedeny už konkrétní metody pro predikci spotřeby elektrické energie.

- dekompozice časové řady
- Boxova-Jenkinsova metodologie
- dynamické modely
- spektrální analýza časových řad

Důležité je zmínit i výběr metody, který závisí na několika faktorech:

- na účelu analýzy

- na typu časové řady
- na zkušenostech daného člověka

### 1.1.4 Predikce

Jak již bylo uvedeno, analyzovat časovou řadu můžeme za účelem predikce. Tato práce je zaměřena konkrétně na predikci spotřeby elektrické energie. U predikce spotřeby elektrické energie je stěžejní předpovídaný interval, a proto se predikování spotřeby dělí dle [3] na:

- krátkodobou predikci - několik hodin až dní
- střednědobou predikci - řádově týdny, měsíce
- dlouhodobou predikci - roky

Pro každou položku budou uvedeny některé predikční metody.

## 1.2 Krátkodobá předpověď spotřeby elektrické energie

### 1.2.1 Exponenciální vyrovňování

Tuto metodu používají v National Grid [8], což je společnost zodpovědná za distribuci elektrické energie v Anglii a Walesu. Predikce je založena na půlhodinových datech. Interpolováním mezi půlhodinovými intervaly jsou schopni předpovědět spotřebu elektrické energie pro každou minutu. Dále National Grid používá hodinové předpovědi pro vyrovňování nabídky a poptávky elektrické energie.

#### a) Holt-Wintersova metoda

Standardní Holt-Wintersova metoda byla uvedena Wintersem a je vhodná pro řady s jednou sezónní složkou. Vztahy pro verzi s více sezónními složkami jsou následující:

$$S_t = \alpha(X_t/I_{t-s}) + (1 - \alpha)(S_{t-1} + T_{t-1}) \quad (1.1)$$

$$T_t = \gamma(S_t - S_{t-1}) + (1 - \gamma)T_{t-1} \quad (1.2)$$

$$I_t = \delta(X_t/S_t) + (1 - \delta)I_{t-s} \quad (1.3)$$

$$\hat{X}_t(k) = (S_t + kT_t)D_{t-s+1+k}W_{t-s+k}, \quad (1.4)$$

kde  $\alpha$ ,  $\gamma$ ,  $\delta$  a  $\omega$  jsou vyrovnávací parametry.  $\hat{X}_t(k)$  je  $k$ -tý krok předpovědi.  
 $S_t$  - level  
 $T_t$  - trend  
 $I_t$  - sezóna

## b) Dvousezónní Holt-Wintersova metoda

Ačkoli standardní Holt-Wintersova metoda je používána pro předpovídání sezónních časových řad, metoda je použitelná pouze pro jednu sezónu. Tato metoda je vhodná pro dvě sezónní složky v časové řadě. Vztahy:

$$S_t = \alpha(X_t/(D_{t-s_1}W_{t-s_2})) + (1 - \alpha)(S_{t-1} + T_{t-1}) \quad (1.5)$$

$$T_t = \gamma(S_{t-1} + T_{t-1}) + (1 - \gamma)T_{t-1} \quad (1.6)$$

$$D_t = \delta(X_t/(S_tW_{t-s_2})) + (1 - \delta)D_{t-s_1} \quad (1.7)$$

$$W_t = \omega(X_t/(S_tD_{t-s_1})) + (1 - \omega)W_{t-s_2} \quad (1.8)$$

$$\hat{X}_t(k) = (S_t + kT_t)D_{t-s_1+k}W_{t-s_2+k}, \quad (1.9)$$

kde  $\alpha$ ,  $\gamma$ ,  $\delta$  a  $\omega$  jsou vyrovnávací parametry.  $S_t$  je level,  $T_t$  je trend,  $D_t$  je sezóna 1 a  $W_t$  sezóna 2.

## c) Multiplikativní dvousezónní ARIMA modely

Tato metoda zůstala populární v průběhu let. ARIMA model (autoregressive integrated moving average model) s jednou sezónní složkou může být zapsán jako

$$\phi_p(L)\Phi_p(L^s)\nabla^d\nabla_s^D X_t = \theta_q(L)\Theta_Q(L^s)\varepsilon_t, \quad (1.10)$$

kde  $L$  je prodleva,  $s$  je počet period v sezónní složce,  $\nabla$  je diferenční operátor,  $\nabla_s$  je sezónní diferenční operátor,  $(1 - L^s)$ ,  $d$  a  $D$  jsou stupně difference,  $\varepsilon_t$  je bílý šum a  $\phi_p$ ,  $\Phi_p$ ,  $\theta_q$  a  $\Theta_Q$  jsou polynomiální funkce stupňů  $p$ ,  $P$ ,  $q$  a  $Q$ . Tento model je často vyjadřován jako  $ARIMA(p, d, q) \times (P, D, Q)_s$ . Multiplikativní je tato metoda v tom smyslu, že polynomiální funkce  $L$  a  $L^s$  jsou násobeny na každé straně rovnice s cílem dostat rozšířenou funkci se zpožďujícím operátorem. Tento model může být případně rozšířen o další sezónní složky. ARIMA model se dvěma sezónními složkami může být zapsán jako

$$\phi_p(L)\Phi_{P1}(L^{S1})\Phi_{P2}(L^{S2})\nabla^d\nabla_{s1}^{D1}\nabla_{s2}^{D2} X_t = \theta_q(L)\Theta_{Q1}(L^{S1})\psi_{Q2}(L^{S2})\varepsilon_t, \quad (1.11)$$



kde  $s_1$  a  $s_2$  jsou počty period v různých sezónních cyklech,  $\Phi_{P_2}$  a  $\psi_{Q_2}$  jsou polynomiální funkce stupňů  $P_2$  a  $Q_2$ . Tento model může být zapsán jako  $ARIMA(p, d, q) \times (P_1, D_1, Q_1)_{s_1} \times (P_2, D_2, Q_2)_{s_2}$ . V National Grid se používal model s půlhodinovými intervaly, bylo nastaveno  $s_1 = 48$  a  $s_2 = 336$  (168 půlhodin). Tento model se používal také v ČR, kde bylo  $s_1 = 24$  a  $s_2 = 168$ , tj. intervaly byly hodinové. ARIMA model může být jednoduše rozšířen o další sezónní složky.

### 1.2.2 Umělé neuronové sítě

Neuronové sítě jsou inspirovány biologickými systémy, částečně zkoumáním lidského mozku. Jsou schopny učit se na příkladech, což je výhoda oproti ostatním metodám. V současné době se neuronové sítě používají v mnoha oborech, např. v průmyslu, medicíně, biologii atd. UNS zachycují jemné vztahy mezi daty, dokonce i když je vztahy velmi těžké popsat. UNS jsou vhodné zejména pro problémy, kde řešení vyžaduje znalosti, které jsou náročné na popis, ale zároveň kde je dostatek pozorovaných dat. Bylo ukázáno, že UNS mohou aproximovat jakoukoli spojitou funkci s jakoukoli požadovanou přesností. Předpovědicí modely obecně předpokládají, že existuje základní vztah mezi vstupy (minulé hodnoty časové řady nebo další proměnné). Tradiční statistické modely jsou většinou limitovány v odhadování těchto vztahů kvůli složitosti reálného systému. UNS jsou dobrou alternativou k nalezení těchto vztahů. Dále je nutno poznamenat, že UNS jsou nelineární, zatímco tradiční přístupy k predikci časových řad předpokládají, že zkoumané časové řady jsou generovány lineárními procesy. Lineární modely mají tu výhodu, že mohou být analyzovány do detailů, a je jednoduché je implementovat. Nicméně pokud je základní mechanismus nelineární, mohou být lineární modely velice nevhodné. UNS byly původně vyvinuty k napodobení základních neuronových systémů (lidský mozek) a jsou složeny z navzájem propojených prvků zvaných neurony. Každý neuron přijímá vstupní signál od ostatních neuronů nebo vnější podněty, zpracuje informaci určitými funkcemi a vytvoří nový výstupní signál pro další neurony, popř. externí výstup. Ačkoli jeden neuron je pomalý a nedokonalý, kolektivně neurony můžou efektivně zpracovávat překvapivě mnoho úkolů. Pro předpovídání jsou nejrozšířenější dopředné neuronové sítě s jednou skrytou vrstvou. Nevýhodou neuronových sítí je, že někdy nelze interpretovat některé parametry nebo identifikovat jednoduše závislost mezi proměnnými. Pokud je tedy potřeba zjistit, jakým způsobem se došlo k výsledku, pak neuronové sítě nejsou vhodnou volbou.

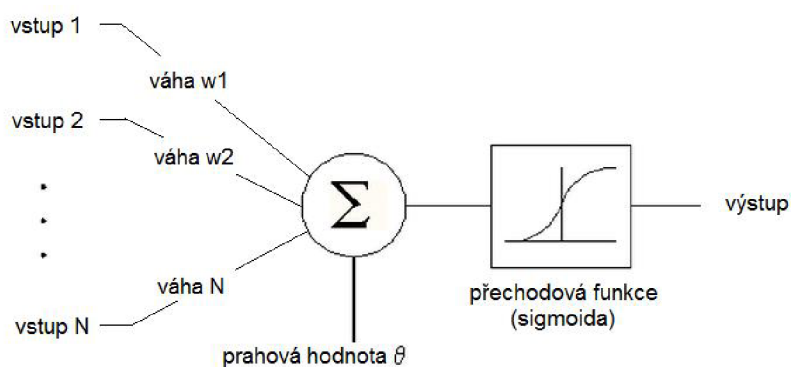
#### a) Model neuronu

Model neuronu má vstupní, výstupní a funkční část. Neuronu na vstupy přicházejí různé podněty od dalších neuronů (anebo podněty z vnějšího okolí) a aktivuje se

právě tehdy, když souhrn těchto podnětů překročí určitou prahovou hodnotu. Funkci neuronu lze popsat takto:

$$y = f\left(\sum_{i=1}^n x_i \cdot w_i - \theta\right),$$

kde  $x_i$  je hodnota vstupu,  $w_i$  váha vstupu,  $\theta$  prahová hodnota,  $n$  počet vstupů,  $f$  transformační funkce a  $y$  hodnota výstupu.



Obr. 1.1: Model neuronu, převzato z [5]

## b) Rozdělení dat

- Trénovací data - slouží pro učení sítě
- Testovací data - zastavení trénování, aby nedošlo k přeučení
- Validační data - zde se ověřuje, jak je model kvalitní

Toto je typické rozdělení v poměru 50:25:25, popř. 70:15:15.

## c) Přeučení sítě

Co je potřeba řešit v neuronových sítích, je počet neuronů ve skryté vrstvě. Pokud by byl počet neuronů příliš malý, pak schopnost dobře vystihnout závislosti v datech klesá. Pokud by síť měla naopak příliš mnoho neuronů, pak rostou nároky na proces učení. U příliš rozsáhlé sítě může docházet k tzv. přeučení (overfitting), tj. nemá schopnost generalizace, zaměřuje se na podrobnosti v trénovacích datech, které ale nejsou důležité. Bohužel obecné řešení problému, jak zvolit vhodný počet neuronů, neexistuje. K přeučení může dojít také při malém rozsahu trénovacích dat. Cílem je tedy kompromis mezi výkonem na trénovacích datech a schopností generalizace na nových datech.

## 1.3 Střednědobá předpověď spotřeby elektrické energie

Střednědobá predikce je predikce v horizontu týdnů až měsíců. Metody, které zde budou uvedeny, nejsou všechny, které se používají. Důvod je ten, že byly uvedeny už v jiné kapitole. Například pro střednědobou predikci se používají hojně ARIMA modely anebo též neuronové sítě, stejně jako při krátkodobých předpovědích.

### 1.3.1 Dekompozice časových řad

Zde se časová řada rozkládá na trendovou, cyklickou, sezónní a náhodnou složku. Dekompoziční metoda se zabývá především identifikací a modelováním trendové a sezónní složky.

**Trend** – vyjadřuje tendenci procesu dlouhodobé změny. Trend je rostoucí nebo klesající. Někdy řada může být i bez trendu. Trend se modeluje matematickými křivkami.

**Sezónní složka** – pravidelné změny v časové řadě, příčiny jsou různé. Například státní svátky, víkendy, změny ročního období apod.

**Cyklická složka** - charakterizuje kolísání okolo řady, kdy dochází k růstu nebo poklesu.

**Náhodná složka** – zbývá v řadě po odstranění ostatních složek. Jedná se o tzv. bílý šum s normálním rozdělením. Je to zbytek hodnot po vyloučení ostatních složek. Někdy se nazývá též nesystematická nebo zbytková složka.

Dekompozice časových řad se dělí na aditivní a multiplikativní. Aditivní má tvar:

$$y_t = T_{r_t} + C_t + S_{z_t} + \varepsilon_t, \quad t = 1, \dots, n.$$

Aditivní dekompozice se používá v případě, že hodnoty časové řady jdou vyčíslit součtem jednotlivých složek. Všechny složky jsou v absolutních hodnotách. Multiplikativní má tvar:

$$y_t = T_{r_t} C_t S_{z_t} \varepsilon_t.$$

U multiplikativní dekompozice je pouze trendová složka v absolutní hodnotě, ostatní hodnoty jsou relativní. Tento způsob dekompozice se používá, pokud se mění variabilita časové řady v čase. Pokud je potřeba, může se multiplikativní tvar pomocí logaritmování převést na aditivní tvar.

Zde je nutno vysvětlit určité pojmy.

Extrapolace – předpovídání hodnot mimo daný interval, čili mimo známé hodnoty.

Interpolace – výpočet přibližné hodnoty v nějakém intervalu, zatímco známy jsou i jiné body tohoto intervalu. Vyrovnávání časových řad znamená nahrazení naměřených dat teoretickými.

Klouzavé průměry – základ metody je rozdělení časové řady na menší úseky o lichém počtu hodnot, kde počítáme lokální polynomicke trendy určitého stupně.

#### a) Metody dekompozice

Metod dekompozice je více, zde ale bude uvedena ta nejběžnější a to **regresní analýza** založená na prokládání matematickou křivkou. Předpokladem pro prokládání matematickými křivkami je neměnnost trendu sledované časové řady. Pokud není tento předpoklad splněn, používá se tzv. adaptivní metoda. Bohužel předpovídání budoucích hodnot nemusí být dosti přesné už jen proto, že v praxi málokdy narazíme na neměnné prostředí, tj. neměnný trend.

Parametry dané křivky se většinou odhadují **metodou nejmenších čtverců**.

#### b) Použití

Dekompozice časových řad byla použita v [6]. V této práci je prezentován postup predikce pro 1 až 31 dní, konkrétně si dala za úkol předpovědět spotřebu elektrické energie ve východním Slovensku na jeden měsíc dopředu v lednu roku 1999. Historická data spotřeb byla k dispozici pro roky 1997-1998 a bylo pracováno též s průměrnými denními teplotami z let 1995-1998. Jiná meteorologická data k dispozici nebyla. Protože předpovídán byl leden, byly z dat použity pouze zimní měsíce. Dále byly vyloučeny speciální případy jako např. Vánoce. Model byl validován na dvaceti dnech. Co se týče počítání průměrných denních teplot, tak bylo použito exponenciální vyrovnávání.

Tato metoda byla dále vylepšena a to přidáním ARMA filtru pro predikování nesystematické složky. Chyba MAPE bez uvedeného filtru byla 1,62%, kdežto s filtrem 1,58%. V tomto vylepšeném modelu dále byla využita teplota jako vstupní proměnná. Teploty pro leden 1999 bylo nutno předpovědět, pro to byly jednoduše použity klouzavé průměry počítané dle let 1997 a 1998.

Nevýhody této metody jsou dvě. První je ta, že na Slovensku v té době byla nestabilní ekonomie a to by pro delší časové řady mohl být problém. Při volbě této metody je tedy potřeba vzít v úvahu faktor jako je ekonomie dané země. Druhou nevýhodou je předpovídání počasí, které je spolehlivé v horizontu několika dní dopředu. Tato metoda by mohla být vhodná i pro krátkodobé předpovědi, ovšem jak již bylo naznačeno, kvalita predikce je silně závislá na kvalitě předpovědi průměrných denních teplot.

### 1.3.2 Neuronové sítě v kombinaci s evolučními algoritmy

V článku [1] je použita kombinace neuronových sítí a evolučních algoritmů. Narozdíl od předchozího modelu zde není vůbec uvažován ekonomický faktor, který by byl důležitý pro dlouhodobější předpověď. Zde bude predikovaný horizont až 31 dní. Navíc zde nejsou uvažovány průměrné denní teploty, relativní vlhkost a další meteorologické jevy, protože předpovídání počasí na měsíc dopředu s vyhovující přesností je velký problém. Když zde bude řeč o spotřebě, je tím myšleno denní maximální zatížení.

Tento model je převážně založen na spotřebě posledního dne, jakožto na vstupní proměnné, nicméně mohou být vybrány další vhodnější proměnné. Dále tento model pracuje s kalendářními indikátory, tj. je rozlišováno, zda je víkend, svátek apod. V tomto článku byla použita dvoukroková korelační analýza. V prvním kroku se vypočítá, kteří kandidáti ze vstupních proměnných jsou nejvhodnější a ve druhém kroku se křížovou korelační analýzou z těchto kandidátů vyberou už konkrétní vstupní proměnné. Byly testovány i jiné techniky selekce vstupních proměnných, konkrétně se jedná o regresní analýzu, Akaikovo informační kritérium a autokorelační funkce. Nicméně pro střednědobou predikci se ukázalo z hlediska přesnosti, že je vhodnější volbou dvoukroková korelační analýza.

Při konstrukci této předpovědací metody byla použita vícevrstvá perceptronová síť, protože je flexibilní a jednoduchá na implementaci. Pro učení neuronové sítě byl použit tzv. Levenberg–Marquardtův algoritmus. Přestože algoritmů učení bylo vyzkoušeno více, tento měl pro střednědobou predikci nejlepší výsledky. Tento algoritmus je podobný Newtonově metodě a počítá přibližnou Hessovu matici.

Výsledky trénovací fáze se předávají evolučnímu algoritmu. Vztahy tohoto evolučního algoritmu jsou

$$\Delta W_{n+1} = m \cdot \Delta W_n + (1 - m) \cdot g \cdot W_n \quad (1.12)$$

$$W_{n+1} = W_n + \Delta W_{n+1}, \quad (1.13)$$

kde  $W$  je volitelný parametr neuronové sítě a  $\Delta W$  značí změnu.  $n$  a  $n+1$  reprezentují dvě postupné generace evolučního algoritmu (rodič a potomek),  $g$  je malé náhodné číslo generované pro každý volitelný parametr a  $m$  je konstanta momenta. Zde byly voleny parametry  $m = 0,5$  a  $g$  bylo z intervalu  $\langle 0; 0,1 \rangle$ . Na začátku evolučního algoritmu je  $W$  hodnota obdržená z algoritmu učení pro parametr  $W$  a  $\Delta W_0 = 0$ . V každém cyklu evolučního algoritmu se opakují uvedené rovnice, dokud nejsou získány následující generace všech parametrů. Následně je vyhodnocena chybová funkce neuronové sítě. Pokud má potomek menší chybu jako rodič, pak je rodič

potomkem nahrazen. Pokud ne, pak rodič je obnoven a další cyklus algoritmu je přerušeno. Na konci algoritmu je mezi generacemi vybráno nejlepší řešení.

Shrnutí by vypadalo tedy následovně.

1. Zvolení hodnot pro volitelné parametry korelační analýzy a počtu skrytých prvků v neuronové síti.
2. Z dostatečně velkého souboru denních spotřeb se vyberou ty vstupní funkce, které nejvíce korelují s výstupní proměnnou, a které zároveň mají nejnižší stupeň lineární závislosti. Tyto vstupní funkce v kombinaci s kalendářními indikátory jsou použity pro naučení první neuronové sítě - předpředpovídací blok, zatímco pro další dvě neuronové sítě je považován za vstup predikce předchozí neuronové sítě. Po přiřazení všech vstupních funkcí lze přistoupit k trénování neuronových sítí na základě historických dat.
3. Odstranění dat, která budou potřeba pro validaci.
4. Trénování neuronových sítí Levenberg–Marquardtovým algoritmem a evolučního algoritmu.
5. Pokud již byly přezkoumány všechny kombinace volitelných parametrů, vybere se nejlepší vzhledem k validační chybě.
6. Vrácení validačních dat zpět do tréninkových dat a provedení konečné tréninkové fáze. Predikování dat na dalších 31 dní.

Tato metoda byla testována na dvou místech. Za zmínku stojí dvě techniky, a to bez kalendářních indikátorů a s jedním kalendářním indikátorem. Při prvním testování byla chyba MAPE 1.23 u verze bez kalendářních indikátorů 1,65% a s jedním kalendářním indikátorem 1,60%. Při druhém testování byly chyby MAPE v tomtéž pořadí 1,85% a 1,76%.

### 1.3.3 Metoda založená na extrakci trendu

V článku [3] je prezentována metoda založená na rozdělení časové řady na dvě složky - trend a fluktuaci okolo trendu. V této práci byly použity neuronové sítě k predikci jednotlivých, už uvedených složek. Trénovací množina pro neuronové sítě byla staršího data, a to od ledna 1975 do prosince 2002. Faktory, jako je počasí nebo ekonomická data, byla vynechána, protože takové faktory se musí prvně předpovědět. Navíc takové faktory jsou už, dle názoru autorů, zahrnuty v samotném chování časové řady.

#### a) Extrakce trendu

Cílem tohoto procesu je získat vyrovnané časové řady, které budou co nejpřesnější jako časové řady reprezentující vývoj spotřeby elektřiny. Pro výpočet byly použity

dvě rovnice. První měří přesnost a je definována jako

$$R^2 = 1 - \frac{\sum_{i=1}^N (C[i] - T[i])^2}{\sum_{i=1}^N (C[i] - \hat{C})^2}, \quad (1.14)$$

kde  $C[i]$  jsou řady spotřeb,  $T[i]$  jsou vyrovnané časové řady a  $\hat{C}$  průměrná spotřeba. Druhá rovnice vyjadřuje hladkost trendu časové řady a je definovaný jako

$$S = \sqrt{\frac{1}{N} \sum_{i=1}^N (\Delta^2 T[i])^2}, \quad (1.15)$$

kde  $\Delta^2 T[i]^2$  reprezentuje druhou derivaci trendu.

Dalším nástrojem pro studování procesu vyhlazování je analýza výkonového spektra flukтуаční časové řady, který dokáže detekovat zbytkové složky ve vyhlazované časové řadě. Flukтуаční časovou řadu lze získat tak, že se odečte trendová složka od původní časové řady spotřeb.

V citovaném článku jsou uvedeny dvě techniky vyrovnávání - klouzavé průměry a splajny. Byly testovány i jiné techniky jako třeba digitální filtry s dolní propustí, avšak neměly příliš dobré výsledky.

**a.1) Klouzavý průměr** Testovány byly čtyři typy klouzavých průměrů, kde jednotlivé elementy měly různou váhu.

1. Konstantní váha (neboli jednoduchý klouzavý průměr):  $i$ -tý prvek je získán ze vztahu

$$T[i] = \frac{1}{n} (C[i] + C[i-1] + \dots + C[i-(n-1)]), \quad (1.16)$$

kde  $n$  je počet prvků.

2. Exponenciálně klesající váha (neboli exponenciální klouzavý průměr):

$$T[i] = \frac{1}{r} C[i] + \left(1 - \frac{1}{r}\right) T[i-1], \quad r = \frac{2}{n+1} \quad (1.17)$$

3. Lineárně klesající váha:

$$T[i] = \frac{2}{n+1} \sum_{j=1}^n \frac{j}{n} C[i-n+j] \quad (1.18)$$

4. Modifikovaný klouzavý průměr:

$$T[i] = T[i-1] - \left[ \frac{C[i] - T[i-1]}{n} \right] \quad (1.19)$$

Jako nejlepší se překvapivě ukázal jednoduchý klouzavý průměr s  $n = 12$ , který poskytoval dobré vyrovnávání.

**a.2) Vyrovnávací splajny** Kubický splajn je kubická polynomiální funkce definovaná na každém subintervalu  $\langle x_i, x_{i+1} \rangle$ , kde  $x_i \in \langle a, b \rangle$ ,  $i = 0, \dots, n$ .

$$p \sum_{i=1}^N (y_i - S_p(x_i))^2 + (1-p) \int_a^b (D^2 S_p(t))^2 dt. \quad (1.20)$$

V tomto vztahu  $S_p$  reprezentuje splajn,  $D^2 S_p$  je druhá derivace a  $p$  je tzv. vyrovnávací parametr. Tento parametr byl testován v intervalu  $\langle 0,00001; 0,002 \rangle$  a bylo ukázáno, že nejlepší volbou je  $p = 0,00005$ .

## b) Měsíční predikce spotřeby elektřiny

**b.1) Neuronové sítě pro předpovídání časových řad** Po získání trendu a vypočítání fluktuace okolo trendu se měsíční predikce bude tedy skládat ze dvou hodnot. Jak již bylo uvedeno, samotné predikování mají na starost neuronové sítě. Použity byly běžné dva modely: MLP (multilayer perceptron) a RBF (radial basis function), široce používané pro předpověď časových řad.

1. MLP: Neboli vícevrstvá perceptronová síť. Tato síť je tvořena ze vstupní vrstvy, výstupní vrstvy a jedné nebo více skrytých vrstev. Každá vrstva je tvořena určitým počtem neuronů, které počítají vždy vážené součty svých vstupů. Výsledek je tvaru

$$y_k = \Psi \left( \sum_{j=0}^n \omega_{jk} x_j + \theta_k \right), \quad (1.21)$$

kde  $x_j$  reprezentuje  $j$ -tý vstup  $k$ -tého neuronu,  $\omega_{jk} x_j$  je síla propojení mezi neuronem a předchozí vrstvou,  $y_k$  je výstupní neuron a  $\theta_k$  je bias. Funkce  $\Psi$  je většinou logická funkce nebo hyperbolický tangens. K naučení MLP byla použita metoda zpětného šíření.

2. RBF: Neboli radiální bázové funkce. Struktura této neuronové sítě je podobná struktuře MLP, ale s tím rozdílem, že RBF má pouze jednu skrytou vrstvu a výstupem je Gaussova funkce. Výstup neuronu je definován jako

$$y_k = e^{-\|x - c_k\|/\sigma}, \quad (1.22)$$

kde  $\|x - c_k\|$  je euklidovská vzdálenost,  $c_k$  je vektorový parametr,  $x$  je vstupní vektor a  $\sigma$  je parametr definovaný jako  $\sigma = disp \sqrt{-\ln 0,5}$ , kde  $disp$  je disperzní parametr.

Algoritmus učení je zde jiný, než který byl použit u MLP. Tzv. metoda OLS (orthogonal least squares) je obvykle používána k přizpůsobení vah u jednotlivých neuronů.



3. Aplikace neuronové sítě: Před trénováním neuronové sítě je třeba definovat strukturu - počet vrstev sítě a počet neuronů ve vrstvách. Správná volba struktury je zásadní pro fungování neuronové sítě, protože příliš málo neuronů by nebylo schopno předpovídat s uspokojivou přesností. Naopak příliš mnoho neuronů způsobí, že bude výpočetní čas příliš dlouhý, vytížení paměti vysoké a v neposlední řadě sít nebude schopna zobecňování získaných znalostí. Nalezení odpovídající velikosti sítě není jednoduchý úkol, avšak existují algoritmy, které mohou pomoci v hledání. Používají se tzv. metody prořezávání (pruning algorithms), které jsou ale bohužel náročné na čas a ne vždy poskytnou nejmenší sít.

Jednodušší a flexibilnější metodou byla metoda pokus-omyl (trial and error), kde se navrhne několik sítí s různými velikostmi, následně se otestují a vybrána je ta nejlepší z nich. Když už je sít definována, lze přistoupit k trénování. Data se rozdělí na trénovací a validační část. V uvedené práci bylo 276 údajů pro trénování a 60 pro validaci.

Přesnost předpovídání ve validační fázi je měřena chybou MAPE (mean absolute percentage error) danou vztahem

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \left| \frac{C[i] - \text{Pred}[i]}{C[i]} \right| 100, \quad (1.23)$$

kde  $C[i]$  je aktuální měsíční spotřeba,  $\text{Pred}[i]$  je predikce a  $N$  je počet dat ve validační množině.

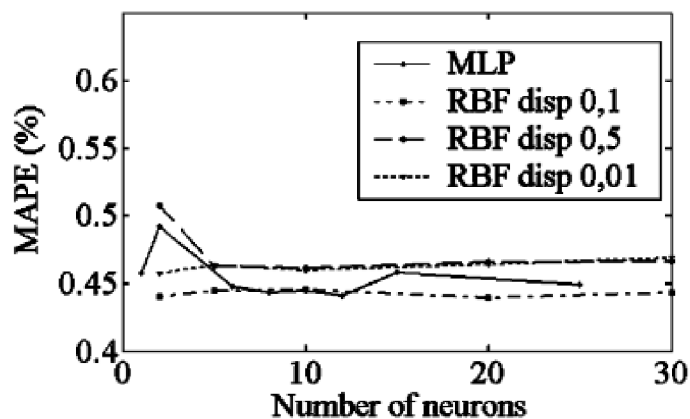
**b.2) Predikce trendu** Jak trend prezentuje chování časové řady, tak je nezbytné normalizovat její data kvůli řádnému zpracování hodnot neuronovou sítí. K tomu slouží klasická metoda - diferenciace. První částí diferenciace se získá nová řada s více či méně stacionárním chováním, jejíž prvky musí být rozděleny určitou konstantou (mají od sebe stejnou vzdálenost). Tato konstanta byla testována metodou pokus-omyl a její hodnota byla zvolena 1000.

$$T_n[i] = \frac{T[i] - T[i - 1]}{1000}. \quad (1.24)$$

Když byl trend extrahován, jeho hodnoty normalizovány a struktura neuronové sítě definována, může se přistoupit k trénování. Neuronová sít je trénována vstupními vektory se 12 hodnotami normalizované řady  $T_n[i]$  a očekávaným výstupem je vždy následující hodnota. Po vytrénování může být neuronová sít už použita k predikci dat a jejich porovnáním s validačními daty se zjistí chyba předpovídání.

Tento proces byl aplikován na trend získaný klouzavými průměry a splajny. V obou případech byly testovány různé počty neuronů pro jednu a dvě skryté vrstvy (pro MLP) a různé hodnoty  $disp$  (pro RBF) k nalezení nejlepší konfigurace pro co

nejlepší predikci. Jak lze vidět na obrázku 1.2, nejlepších výsledků se dosáhlo s RBF sítí, která měla 20 skrytých neuronů a  $disp = 0,1$ .



Obr. 1.2: Porovnání chyb MAPE, převzato z [3]

Pro vyhlazování trendu se ukázaly lepší klouzavé průměry jak splajny. Klouzavé průměry ale bohužel nejsou schopny správně extrahovat prvky s nízkou hodnotou a to může být problém při predikci fluktuace.

**b.3) Predikce fluktuace** Jak již bylo řečeno, fluktuaci získáme vyloučením trendu z původní časové řady. Stejně jako u extrakce trendu i zde bylo testováno několik konfigurací a velikostí MLP a RBF sítí. Zde bylo dosaženo opačného výsledku, splajny měly lepší výsledky jak klouzavé průměry. Konkrétně u klouzavých průměrů byla chyba MAPE 1,84%, u splajnů potom 1,76%. Co se sítí týče, nejlepších výsledků bylo dosaženo MLP sítí se 3 skrytými vrstvami (8, 4 a 8 neuronů).

### c) Shrnutí

Tato citovaná práce přináší novou metodu vhodnou pro střednědobou predikci. Výsledky byly porovnány s jinými metodami, jako je například ARIMA, regresní analýza nebo expertní systémy. Nejlépe z těchto tří vyšel model ARIMA s chybou MAPE 3,8%.

Oddělením trendu a fluktuace se dosáhlo lepších výsledků než pouze s jednou neuronovou sítí. Navíc tento způsob vykazoval větší přesnost než uvedený ARIMA model.

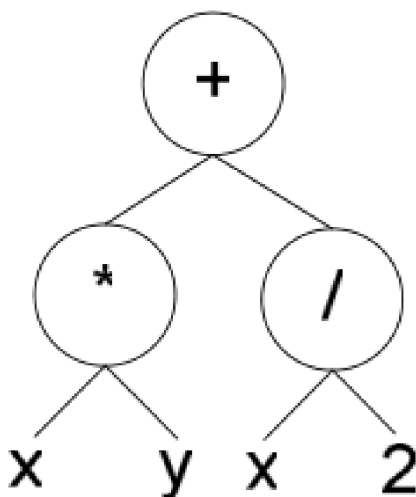
## 1.4 Dlouhodobá předpověď spotřeby elektrické energie

Pro dlouhodobou předpověď spotřeby elektrické energie se, stejně jako u krátkodobé a střednědobé predikce, hojně využívají taktéž neuronové sítě. Mohl by tu být uveden nějaký model, který UNS využívá, ale protože o UNS už něco napsáno bylo, bude ukázána jiná možnost předpovědi spotřeby elektřiny v rádech let.

### 1.4.1 Genetické programování

V článku [4] byla představena metoda, která využívá genetického programování (GP) a pomocí které byla předpovídána spotřeba elektřiny ve městě na jihovýchodě Turecka. GP bylo vyvinuto v 90. letech J. Kozou v USA a je tak trochu konkurent neuronových sítí. Použita byla tzv. symbolická regrese, což je častá aplikace GP.

GP pracuje se skupinou možných řešení, tato skupina se nazývá populace. Populace se skládá z jedinců, což jsou programy nebo funkce. Cílem GP je vybrat nejlepší řešení neboli nejlepšího jedince z populace. V genetickém programování jsou jedinci reprezentováni stromy. Elementy stromů jsou funkce a proměnné, na které jsou tyto funkce aplikovány. Pro příklad, obrázek 1.4.1 ukazuje jednoduchý výraz  $xy + \frac{x}{2}$ .



Obr. 1.3: Příklad stromu v GP, převzato z [4]

Před samotným spuštěním algoritmu je potřeba udělat určité kroky. Konkrétně je třeba předdefinovat soubor proměnných, soubor primitivních funkcí a další parametry (např. kontrola velikosti populace za běhu programu). Nyní k samotnému algoritmu, zjednodušeně by se dal popsat následovně:

- Vygeneruje se prvotní populace náhodného složení funkcí a proměnných.
- Je zkontrolováno, zda se nemá program ukončit a vrátit nejlepší výsledek. Pokud program končit nemá, jsou iterovány následující kroky.

1. Každému jedinci v populaci je přiřazena hodnota (zdatnost) vypočítaná tzv. fitness funkcí, která udává, jak je jedinec schopný řešit zadanou úlohu. Čím vyšší tato hodnota je, tím je jedinec lepší. Cílem programu je tedy najít jedince s nejvyšší zdatností.
2. Je vytvořena nová populace na základě následujících operací. Operace jsou aplikovány na jedince, kteří jsou vybráni z populace s určitou pravděpodobností, která je založená na zdatnosti.
  - (a) Reprodukce existujícího jedince - zkopírování jedince do nové populace.
  - (b) Tzv. crossover - vytvoření dvou nových jedinců náhodnou záměnou dvou jejich částí.
  - (c) Mutace - určitá část jedince je náhodně pozměněna, tj. vznikne nový zmutovaný jedinec.
3. Jedinec, identifikovaný jako výsledek, je vrácen a může být řešením, anebo alespoň přibližným řešením.

Model vygenerovaný pomocí GP měl 51 elementů a 17 úrovní. Pro zajímavost, výpočet trval cca 1 hodinu. V tomto článku byla porovnávána přesnost modelu vytvořeného pomocí GP s dalšími dvěma regresními modely. První regresní model byl založen na polynomu čtvrtého stupně, tj.

$$f(x) = p_1x^4 + p_2x^3 + p_3x^2 + p_4x + p_5. \quad (1.25)$$

Druhý model byl založen na rovnici

$$f(x) = ax^b + c. \quad (1.26)$$

Vstupní data (spotřeby) byla použita z let 1994 až 2005.

Nejmenší chybu vykazoval model GP a to 0,00038000. První regresní model pak 0,00608797 a druhý regresní model 0,02194693. Uvedené chyby jsou chyby SSE (sum of squared errors) a platí vztah

$$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (1.27)$$

kde  $y_i$  je  $i$ -tá hodnota, která má být předpovězena a  $\hat{y}_i$  je předpovězená hodnota.

Závěrem tedy lze říci, že model GP z tohoto porovnání vyšel nejlépe a určitě je vhodný pro dlouhodobou predikci spotřeby elektrické energie.

## 2 PRAKTICKÁ ČÁST

V praktické části práce bylo úkolem vytvořit program, který by předpovídal spotřebu elektřiny v distribuční síti. Naprogramován byl v jazyce Java ve vývojovém prostředí Eclipse.

Naměřená data spotřeby elektřiny jsou uložena v souboru. Tato data jsou načtena a po dobu běhu programu si program tato data drží. Data jsou rozdělena na trénovací a testovací množinu. Následuje trénování modelu, čili zjištění parametrů vícenásobné lineární regrese, která je pro předpovídání spotřeb elektrické energie použita. Potom se za pomoci testování určí přesnost, s jakou je model schopen spotřebu elektřiny předpovídat.

Program sestává z několika částí - tříd, kde každá má na starost určitou činnost. Jednotlivé třídy obstarávají načtení dat ze souboru, uchovávání dat, samotnou predikci, vykreslení do grafu a počítání chyby. Třídami jsou též reprezentovány parametry, které mají být zohledněny při trénování modelu.

Trénování znamená, že se vezme určitá část dat a tato data jsou následně použita, v našem případě, pro lineární regresi. V této práci je trénovací část 90% z celkového počtu dat. 10% dat potom slouží k testování modelu. V tomto procesu je zhodnoceno, jak je testovaný model úspěšný v předpovídání - to je určeno chybou.

Ve třídě určující chybu je počítána chyba MAPE (mean absolute percentage error). Vztah pro chybu MAPE již byl uveden dříve. V dalším textu bude uvedena vždy jen „chyba“. Konkrétními parametry, které byly zohledňovány při trénování, jsou hodina, den v týdnu, měsíc, jedna nebo více předchozích spotřeb a také zda je pracovní den.

V semestrální práci byla použita polynomiální regrese s neuspokojivými výsledky, chyba byla přes 50%. Bylo to patrně důsledkem toho, že byl použit pouze jeden parametr, a to čas v milisekundách od roku 1970. Java má totiž metodu `getTime()`, která vrací zmíněné číslo.

Byla proto použita vícenásobná lineární regrese se vztahem

$$Y = X \cdot b + u, \quad (2.1)$$

kde  $Y$  je vektor o velikosti  $n$ ,  $X$  je matice o velikosti  $n \times k$ ,  $b$  je vektor velikosti  $k$  a  $u$  je vektor o velikosti  $n$ . Vektor  $Y$  je takzvaný regresand (závislá proměnná), sloupce v matici  $X$  jsou regresory (nezávislé proměnné), ve vektoru  $b$  jsou regresní parametry a vektor  $u$  reprezentuje chyby. Regresand je v našem případě spotřeba elektřiny, regresory jsou parametry, které jsou zohledněny při trénování.

V programu byla prvně použita knihovna Apache Commons a následně knihovna Smile, která je komplexnější a kromě vícenásobné lineární regrese obsahuje i další algoritmy.

## 2.1 Popis programu

Zde je popsána struktura programu. Kromě níže uvedených tříd byly použity některé knihovny - konkrétně Apache Commons Math v3.6, dále knihovna JFreeChart v1.0.19 a knihovna Smile v1.0.4. Knihovna Commons Math byla použita v semestrální práci pro predikci pomocí polynomu a potom i v této práci pro vícenásobnou lineární regresi. Nakonec ale byla, též pro vícenásobnou lineární regresi, použita knihovna Smile. Díky knihovně Smile například odpadlo testování, zda neobsahuje duplicitní řádky. Pro import knihovny Smile bylo potřeba Java projekt převést na Maven projekt. Pro používání nástroje Maven je třeba ho do Eclipse nainstalovat, a to pomocí pluginu m2eclipse.

### 2.1.1 Třída Read

Ve třídě `Read` jsou deklarovány dvě pole, pro spotřebu a pro čas. V cyklu je procházen excelovský soubor a tyto hodnoty jsou ukládány do polí. Tato třída obsahuje taktéž metodu (tak se v Javě nazývá funkce), která načte stejným způsobem státní svátky, které jsou uloženy v dalším souboru. Je zde také kontrola duplicitních řádků, které jsou následně přidány do pole pouze jednou. V souboru s daty jsou totiž některá měření v tentýž čas dvakrát.

V této třídě jsou navíc metody, které umožňují předání polí dále.

### 2.1.2 Třída Data

Tato třída slouží k uchovávání dat. Jsou zde tyto důležité proměnné - poměr trénovací části ku celé množině, pole spotřeb a k nim odpovídající čas. Nejdůležitější metodou v této třídě je ta, která vytvoří matice z dat, která jsou načtena třídou `Read`. Vytvoření probíhá tak, že při cyklování skrze načtená data se vždy devět prvků uloží do prvního pole a desátý prvek je uložen do druhého pole. Původní pole spotřeb je tak rozděleno v daném poměru - v tomto případě 90% dat bude v prvním poli, které je určeno pro trénování, a v druhém poli 10% dat určených pro testování.

Dále z času, který vždy odpovídá dané spotřebě, jsou určeny jednotlivé parametry. Tyto parametry již byly uvedeny a každý z nich je reprezentován samostatnou třídou. Z těchto parametrů jsou vytvořeny dvě matice v daném poměru trénovacích dat, každý řádek má odpovídající spotřebu v poli spotřeb.

Příklad: Pokud by celá množina měla tisíc spotřeb a k nim uvedený čas, tak budou vytvořeny prvně dvě pole pro spotřeby o velikosti devět set a sto. Posléze se vytvoří na základě času dvě matice, první o devíti stech řádcích a několika parametrech, druhá o stu řádcích a stejným počtem parametrů. Matice a pole o devíti stech řádcích poslouží pro natrénování, ty o stu řádcích pak budou pro testování.

### 2.1.3 Třída Error

Tato třída má dvě pole, která jsou předána jako parametr při vytváření instance třídy. Tato pole mají velikost 10% celé množiny dat a jsou určena pro testování. V prvním poli jsou reálné spotřeby, v druhém poli jsou pak spotřeby předpovězené čili teoretické. Dle vzorce je pak vypočítána chyba.

### 2.1.4 Třída Generator a její potomci

Třída Generator je abstraktní, drží referenci na načtená data. Dále jsou definovány dvě abstraktní metody pro výpočet určitého parametru. Z této třídy dědí třídy reprezentující jednotlivé parametry, které již byly uvedeny (hodina, den v týdnu, atd.). Tyto třídy pak mají vlastní implementaci metod pro výpočet parametru.

### 2.1.5 Třída SmileOLS

Tato třída převezme data vrácená pomocí metod ze třídy Data a na základě těchto dat natrénuje model. Stejná metoda, která trénuje, hned po natrénování modelu vytvoří pole, do kterého uloží teoretické hodnoty. Toto pole je předáváno třídě Error.

Další metoda je určena k předpovídání. Prvně předpoví spotřebu pro celou množinu dat a potom předpovídá na X hodin dopředu, kde X je zadáno manuálně. Vrací pole předpovězených spotřeb společně s odpovídajícím časem.

### 2.1.6 Ostatní třídy

Dalšími třídami jsou Main a DeviationRendererDemo2, což je třída pro vykreslování grafu. Tato třída převezme načtená data a teoretická data, která následně vykreslí do grafu.

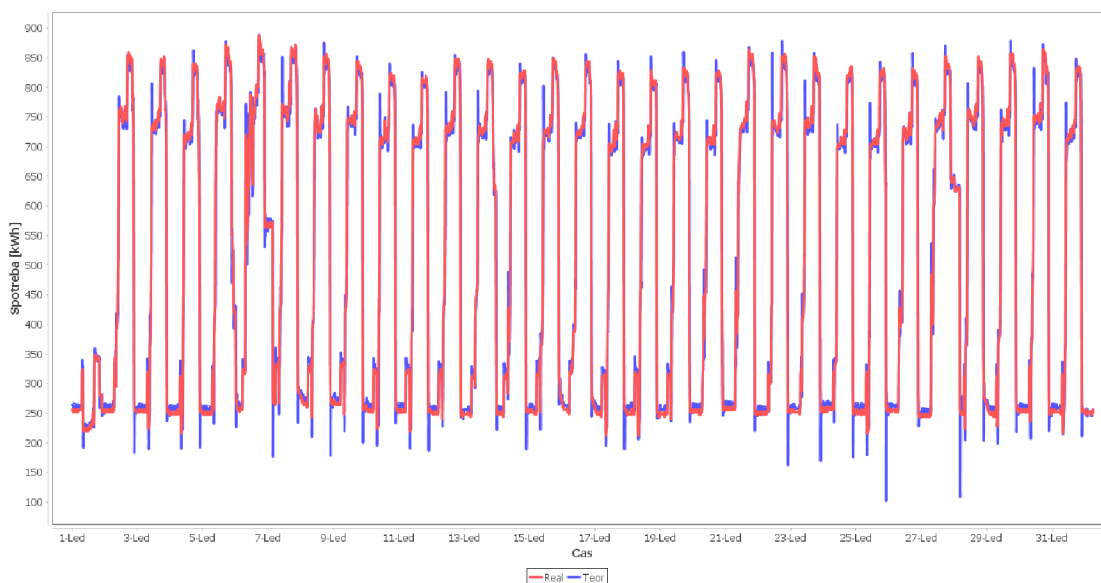
## 2.2 Data

Data ve vstupním souboru jsou určena spotřebou elektřiny v určitém čase. Ta má jednotku **kWh** - kilowatthodina. Watthodina (ani kilowatthodina) není v soustavě SI. V soustavě SI je jednotkou energie totiž joule, což je jedna wattsekunda. Zmíněná kilowatthodina tedy odpovídá 3,6 megajoulům.

Dat bylo k dispozici konkrétně šestnáct listů v Excelu po cca 35000 měřeních. Intervaly mezi jednotlivými měřeními byly 15 minut. Pokud bychom chtěli odhadovat spotřebu například v horizontu několika let dopředu, intervaly by stačily samozřejmě mnohem větší.

## 2.3 Výsledky

Na grafu níže je ukázána reálná i teoretická spotřeby elektřiny pro jeden měsíc, přesněji pro leden 2009. Předpovídán byl i celý rok, ale graf je velice podobný měsíčnímu a není toho na něm příliš vidět. Predikční horizont je 15 minut.



Obr. 2.1: Reálná a teoretická spotřeba elektřiny za leden 2009

Konkrétně pro tato data, leden 2009, bylo dosaženo nejnižší chyby predikce 4,95%. Pro celý rok, 2009, byla chyba 6,1%. U obou modelů byly při trénování zohledněny parametry uvedené výše, tj. hodina, den v týdnu, zda je pracovní den, měsíc a dvě předchozí spotřeby. Otestováno bylo například zohlednění více předchozích spotřeb při trénování, ale chyby se lišily už jen v řádu tisícín procenta a navíc se znatelně prodlužovala doba běhu programu.

Zásadními parametry byly předchozí spotřeba a hodina během dne, ostatní parametry měly už menší vliv. Například pracovní den jakožto parametr neměl takový vliv pravděpodobně z toho důvodu, že státních svátků je málo a víkendy jsou už jednou obsaženy v jiném parametru - den v týdnu.

Pokusem bylo také zohlednění více předchozích spotřeb, které byly průměrovány. Konkrétně tento postup se ukázal jako velice neefektivní a chyba se zvýšila až na 12,7%.

V porovnání s uvedenými pracemi v teoretické části je výsledek horší o několik procent. Například v porovnání s metodou využívající kombinaci evolučních algoritmů a neuronových sítí je výsledek horší o 3,35%, v porovnání s ARIMA modelem je pak výsledek této práce horší o 1,15%.



Další zpřesnění by teoreticky mělo přinést přidání jednoho nebo více parametrů, a to například teploty, která bývá často využívána. Jelikož se ale teplota, celkově počasí, špatně předpovídá v dlouhodobějším horizontu, výhodný by tento parametr mohl být hlavně při předpovídání v horizontu krátkodobém. Teplota ale není jediným parametrem, přestože je z meteorologických jevů pro předpovídání nejdůležitější. Použity by mohly být i jiné parametry, konkrétně relativní vlhkost, srážky a oblačnost.

Další možností, která by mohla dále ovlivnit přesnost, je využití ekonomických faktorů.

### 3 ZÁVĚR

Tato práce pojednává o predikci spotřeby elektrické energie, v teoretické části byly popsány některé její metody.

V praktické části bylo primárním cílem vytvoření nástroje pro predikci a nalezení vhodného algoritmu a parametrů, aby byla chyba predikce co možná nejmenší. Byl vytvořen program v jazyce Java, který dokáže po natrénování sám předpovídat spotřeby elektrické energie. S jeho pomocí lze vytvořit i model pro jiná data.

Jako algoritmus pro predikci byla zvolena vícenásobná lineární regrese, která se ukázala být vhodným algoritmem. Největší podíl na vylepšování přesnosti měl parametr hodina společně s dvěma předchozími spotřebami. Dále byly ukázány jednotlivé chyby MAPE daných předpovídacích modelů, které byly oproti modelům uvedeným v teoretické části horší v řádech %.

## LITERATURA

- [1] AMJADY, Nima; KEYNIA, Farshid. *Mid-term load forecasting of power systems by a new prediction method*. Energy Conversion and Management, 2008, 49.10: 2678-2687.
- [2] Escrivá-Escrivá, Guillermo, Carlos Roldán-Blay, and Carlos Álvarez-Bel. *Electrical consumption forecast using actual data of building end-use decomposition*. Energy and Buildings 82 (2014): 73-81
- [3] Gonzalez-Romera, Eva, Miguel A. Jaramillo-Moran, and Diego Carmona-Fernandez. *Monthly electric energy demand forecasting based on trend extraction*. Power Systems, IEEE Transactions on 21.4 (2006): 1946-1953.
- [4] KARABULUT, Korhan; ALKAN, Ahmet; YILMAZ, Ahmet S. *Long term energy consumption forecasting using genetic programming*. Mathematical and Computational Applications, 2008, 13.2: 71.
- [5] Mendelova univerzita v Brně. *Neuronové sítě* [online]. Dostupné z URL: <[https://is.mendelu.cz/eknihovna/opory/zobraz\\_cast.pl?cast=21471](https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=21471)>.
- [6] Pelikán, Emil. *Middle-term electric load forecasting by time series decomposition*. Electricity Load Forecast using Intelligent Technologies (2002): 167-176.
- [7] StatSoft. *Úvod do neuronových sítí* [online]. Dostupné z URL: <[http://www.statsoft.cz/file1/PDF/newsletter/2013\\_02\\_05\\_StatSoft\\_Neuronove\\_site\\_linky.pdf](http://www.statsoft.cz/file1/PDF/newsletter/2013_02_05_StatSoft_Neuronove_site_linky.pdf)>.
- [8] TAYLOR, James W. *Short-term electricity demand forecasting using double seasonal exponential smoothing*. Journal of the Operational Research Society, 2003, 54.8: 799-805.
- [9] WINTERS, Peter R. *Forecasting sales by exponentially weighted moving averages*. Management Science, 1960, 6.3: 324-342.

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

*ARIMA* autoregressive integrated moving average model

*GP* genetické programování

*MAPE* mean absolute percentage error

*MLP* multilayer perceptron

*RBF* radial basis function

*SSE* sum of squared errors

*UNS* umělé neuronové sítě

# SEZNAM PŘÍLOH

A Obsah přiloženého CD

33

## **A OBSAH PŘILOŽENÉHO CD**

Na přiloženém CD se nachází vytvořený program, který byl exportován do archivu.  
Program byl vytvořen ve vývojovém prostředí Eclipse verze 4.5.0.