Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-5212-72257

Oliver Moravčík

PREDIKCIA VÝROBY ELEKTRINY Z OBNOVITEĽNÝCH ZDROJOV SO ZOHĽADNENÍM EXTERNÝCH FAKTOROV

Bakalárska práca

Študijný program: Informatika

Študijný odbor: Informatika

Miesto vypracovania: Ústav informatiky, informačných systémov a softvérového inžinierstva, FIIT STU, Bratislava

Vedúci práce: doc. Ing. Viera Rozinajová, PhD.

máj 2016

Slovenská technická univerzita v Bratislave FIIT-5212-72257
Fakulta informatiky a informačných technológií

ZADANIE BAKALÁRSKEHO PROJEKTU

Meno študenta:	**Moravčík Oliver**
Študijný odbor:	Informatika
Študijný program:	Informatika
Názov projektu:	**Predikcia výroby elektriny z obnoviteľných zdrojov so zohľadnením externých faktorov**

Zadanie:
V súčasnosti sa venuje veľká pozornosť obnoviteľným zdrojom energie. Ich produkcia je závislá od mnohých parametrov, medzi ktoré patrí napríklad počasie. Z informatického pohľadu sú zaujímavé problémy predikcií, ktoré musia zohľadniť rôzne externé faktory. Ďalšia zaujímavá vlastnosť predikcií je sezónnosť, ktorá je v prípade napríklad fotovoltických elektrární založená najmä na dennej a ročnej báze.
Analyzujte metódy predikcie, ktoré budú vhodné pre zahrnutie externých faktorov. Navrhnite a implementujte predikčný model na predpoveď produkcie elektriny z obnoviteľných zdrojov energie. Riešenie overte na vybranej množine údajov.

Práca musí obsahovať:

Anotáciu v slovenskom a anglickom jazyku
Analýzu problému
Opis riešenia
Zhodnotenie
Technickú dokumentáciu
Zoznam použitej literatúry
Elektronické médium obsahujúce vytvorený produkt spolu s dokumentáciou

Miesto vypracovania: Ústav informatiky a softvérového inžinierstva, FIIT STU, Bratislava
Vedúci projektu: doc. Ing. Viera Rozinajová, PhD.

Termín odovzdania práce v zimnom semestri : 9. 12. 2015
Termín odovzdania práce v letnom semestri : 10. 5. 2016

Bratislava 21. 9. 2015

prof. Ing. Pavol Návrat, PhD.
riaditeľ ÚISI

Anotácia

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ
Študijný program: Informatika
Autor: Oliver Moravčík
Bakalárska práca: Predikcia výroby elektriny z obnoviteľných zdrojov so zohľadnením externých faktorov
Vedúci práce: doc. Ing. Viera Rozinajová, PhD.
Máj 2016

Cieľom tejto bakalárskej práce je navrhnúť a implementovať predikčný model, ktorý bude predpovedať množstvo vyrobenej elektrickej energie fotovoltaickou elektrárňou na deň dopredu. V teoretickej časti tejto práce sme analyzovali doménu predikcie výroby elektriny fotovoltaickými elektrárňami, externé faktory vplývajúce na výrobu elektriny a predikčné metódy používané na predikciu výroby elektriny. V praktickej časti sme opísali implementáciu predikčného modelu, ktorý sme použili na predikciu výroby elektriny a výsledky dosiahnuté použitím tohto predikčného modelu.

Pre predikciu sme použili predikčnú metódu náhodného lesa regresných stromov a dokázali sme, že táto metóda je použiteľná pre predikciu výroby elektriny fotovoltaickou elektrárňou. Dosiahnuté výsledky poukazujú na potenciál použitia takéhoto predikčného modelu v praxi.

Annotation

Slovak University of Technology Bratislava
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES
Degree Course: Informatics
Author: Oliver Moravčík
Bachelor thesis: Forecasting electricity production from renewable resources with external factors taken into account
Supervisor: doc. Ing. Viera Rozinajová, PhD.
May 2016

The aim of this bachelor thesis is to design and implement a forecasting model to predict electricity production of photovoltaic power plant for a day ahead. In theoretical part of the thesis we analyzed the domain of forecasting electricity production from photovoltaic power plants, external factors affecting the electricity production and forecasting methods used for forecasting electricity production. In practical part of the thesis we described implementation of forecasting model we used for forecasting electricity production and results achieved with this forecasting model.

We have used random forest of regression trees as a forecasting method and we have proven that this method can be successfully used for forecasting electricity production of photovoltaic power plant. The results show the potential of using forecasting model based on this method in practice.

Poďakovanie

Ďakujem doc. Ing. Viere Rozinajovej, PhD. za odborné vedenie, cenné rady, optimizmus a pozitívnu motiváciu.

Čestné prehlásenie

Čestne prehlasujem, že záverečnú prácu som vypracoval samostatne s použitím uvedenej literatúry a na základe svojich vedomostí a znalostí.

.....

Oliver Moravčík

Používané skratky

FVE	Fotovoltaické elektrárne
GHO	Globálne horizontálne ožiarenie
NPP	Numerická predpoveď počasia
UNS	Umelá neurónová sieť
SVM	Support vector machine
MOS	Model output statistics
NLRS	Náhodný les regresných stromov

Obsah

1 Úvod

Našou úlohou pri riešení bakalárskej práce je predpovedať výrobu elektrickej energie fotovoltaickou elektrárňou podľa predpovede počasia. Pre riešenie daného problému boli použité viaceré metódy. Tieto metódy máme za úlohu analyzovať a jednu metódu vybrať a použiť pre implementáciu vlastného riešenia tohto problému. Predikcia má predpovedať produkciu fotovoltaickej elektrárne na deň dopredu.

Na predikciu výroby elektrickej energie fotovoltaickou elektrárňou (FVE) sa používa viacero metód, ktoré sa bežne používajú na riešenie predikčných problémov. Sú nimi metódy založené na analýze časových radov ako regresné procesy ARMA a ARIMA, metódy strojového učenia ako napríklad umelá neurónová sieť, ale aj metódy navrhnuté na základe fyzikálnych faktorov a vzťahov vplývajúcich na produkciu FVE. Opisu a charakteristike týchto metód sa venujeme v kapitole 2.

Dosiahnuté výsledky predikcie je potrebné štatisticky spracovať. Na to slúžia viaceré metriky presnosti, ktorých vymenovanie a stručná charakteristika sú obsahom kapitoly 3. Vo štvrtej kapitole sme objasnili problém predikcie výroby elektrickej energie fotovoltaickými elektrárňami. Naše vlastné riešenie, implementácia predikčného modelu, experimenty a dosiahnuté výsledky sú opísané v kapitole 5. Zhodnotenie práce je obsahom kapitoly 6.

1.1 Motivácia

Slnečná energia je z pohľadu človeka nevyčerpateľný zdroj energie. Slnečná energia dopadá na zemský povrch neustále a na rozdiel od fosílnych palív je zadarmo a nikdy ju nevyčerpáme. Za jednu hodinu dopadne na zemský povrch viac energie, než zemská civilizácia spotrebuje za celý rok. Slnko poháňa aj vetry a vlny. Veterné elektrárne navyše na rozdiel od solárnych zaberajú veľmi málo zemského povrchu, a dokonca žiadny, ak sú postavené v mori, kde sú vetry najsilnejšie. Ak by sa ľudstvu podarilo zužitkovať iba miniatúrny zlomok prístupnej slnečnej a veternej energie, mohlo by produkovať elektrickú energiu pre svoje energetické potreby navždy a bez vypúšťania akéhokoľvek uhlíku do atmosféry.

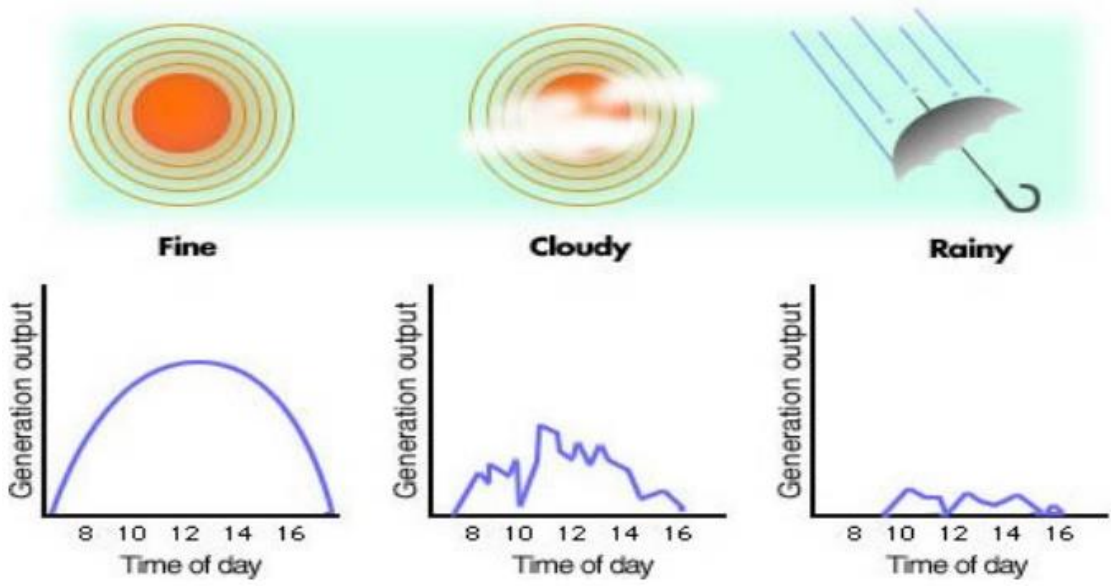
O nevyužitom potenciály slnečnej energie píše aj docentka Morvová vo svojej knihe [1]: „Každý rok dopadne zo Slnka na Zem asi 10 tisíckrát viac energie, ako ľudstvo za toto obdobie spotrebuje. Množstvo dopadajúcej slnečnej energie na územie Slovenska je asi 200-násobne väčšie, ako je súčasná spotreba primárnych energetických zdrojov u nás. Je to obrovský, do-

posiaľ takmer úplne nevyužitý potenciál. Využívanie slnečnej energie je dnes najčistejším spôsobom využívania energie vôbec a na rozdiel od iných zdrojov (aj obnoviteľných) sú dopady na okolité životné prostredie zanedbateľné.“

„V mnohých krajinách by stačilo pokryť menej ako 1 % územia (napr. strechy budov, nevyužité plochy) slnečnými technológiami, aby bol zabezpečený dostatok energie pre celú krajinu.“

„Podstatné je, že aj v našich klimatických podmienkach je potenciál slnečnej energie obrovský, veď len energia dopadajúca na strechu budovy vo väčšine prípadov presahuje spotrebu energie v nej. Intenzita slnečného žiarenia u nás predstavuje asi 1100 kWh/m^2 za rok, kým priemerná spotreba v obytných domoch je len asi 150 kWh/m^2 na vykurovanie a 25-50 kWh/m^2 na chod elektrospotrebičov a na varenie. Z uvedeného vyplýva, že množstvo dopadajúcej slnečnej energie je až 5-krát väčšie alebo vyjadrené inak je postačujúce na pokrytie spotreby až 5-poschodovej obytnej budovy (merané v hodnotách na m^2 horizontálneho povrchu).“

Kvôli miznúcim zásobám fosílnych palív sú solárne elektrárne stále žiadanejšie a buduje sa ich stále viac. Ich produkcia je ale veľmi závislá na počasí ako je vidieť na nasledujúcom obrázku z knihy docentky Morvovej [1] (Obrázok 1).



Obrázok 1: Závislosť produkcie elektriny fotovoltaikou od počasia. [1]

Pre zapojenie solárnych elektrární do elektrickej siete, pre efektívnu produkciu elektrickej energie a pre marketing na trhu s elektrickou energiou je potrebné predpovedať vo viacerých časových horizontoch produkciu solárnej elektrárne. Predikcia produkcie solárnych elektrární je potrebná aj z dôvodu, že elektrická energia sa nedá efektívne skladovať a solárne elektrárne na rozdiel od konvenčných elektrární nevedia prispôsobiť produkciu očakávanej spotrebe.

Efektívna predikcia pomáha operátorom elektrickej siete lepšie manažovať rovnováhu medzi množstvom požadovanej a produkovanej elektrickej energie.

1.2 Úvod do predikcie produkcie FVE

Predpoveď produkcie elektrickej energie solárnymi fotovoltaickými elektrárňami je blízko spojená s predpoveďou počasia. Predpoveď produkcie FVE sa v skutočnosti delí na dve časti. Prvou je predpoveď meteorologických premenných, ktoré majú vplyv na produkciu FVE, a druhou je predpoveď množstva vyrobenej elektrickej energie z FVE na základe predpovedaných meteorologických premenných a charakteristiky FVE. Podľa prečítaných zdrojov sa vždy ako hlavná meteorologická premenná používa *globálne horizontálne ožiarenie* (GHO). Je súčtom priameho kolmého ožiarenia a rozptýleného (difúzneho) horizontálneho ožiarenia [2]. Keďže fotovoltaika znamená priamu premenu slnečnej energie na elektrinu [1], tak pri fotovoltaickej technológii sú dôležité obe zložky. Pri solárnych termálnych elektrárňach, ktoré používajú zrkadlá na nasmerovanie slnečného žiarenia, ktoré ohrieva vodu a vytvára tak paru na výrobu elektriny v parnej turbíne, je dôležité len kolmé ožiarenie.

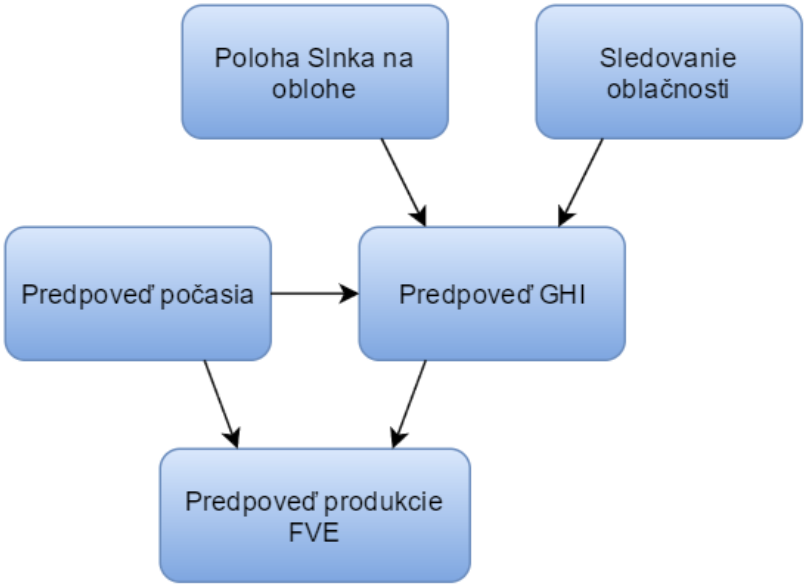
GHO môžeme brať ako numerický ukazovateľ množstva slnečnej energie dopadajúcej na jednotku plochy. GHO sa predpovedá vždy pre konkrétnu oblasť, pretože premenné vstupujúce do výpočtu sú relevantné vždy len pre jednu konkrétnu oblasť. Docentka Morvová vo svojej knihe píše o GHO [1]: „Množstvo dopadajúceho žiarenia na konkrétnom mieste však závisí od viacerých faktorov ako sú napr.:

- zemepisná poloha
- miestna klíma
- ročné obdobie
- sklon povrchu k dopadajúcemu žiareniu.“

Výpočet GHO závisí od slnečného svitu, a teda aj od dĺžky slnečného svitu a uhla, pod ktorým slnečné lúče dopadajú na povrch Zeme v danej zemepisnej šírke. Najväčšou premennou pri výpočte GHO je oblačnosť, ktorá blokuje slnečné lúče a jej premenlivosť je hlavnou príčinou chýb vo výpočte predpovede GHO.

Predpoveď GHO je pre predpovedanie výslednej produkcie FVE veľmi dôležitá, pretože presnosť výpočtu GHO má približne 90 % vplyv na presnosť predpovede produkcie FVE, vplyv teploty okolia je na úrovni 10 % a vplyv vetra asi 1 % [3].

Prepojenie predpovede GHO s ostatnými časťami predikcie produkcie FVE je znázornené na nasledujúcom obrázku (Obrázok 2).



Obrázok 2: Prepojenie častí predikcie produkcie FVE.

2 Metódy predikcie

Autori publikujúci v téme predikcie výroby elektriny z FVE, ako napríklad Diagne a kolektív [4], predikčné metódy rozdeľujú na štatistické a fyzikálne, kde štatistické sú založené na analýze a spracovaní historických dát a fyzikálne využívajú výpočty rovníc popisujúcich vzťahy platné medzi vstupnými premennými. Medzi štatistické metódy tak zaraďujú metódy založené na analýze časových radov aj metódy strojového učenia. Metódy strojového učenia však môžu byť použité aj na analýzu časového radu a predikciu ďalšej postupnosti časového radu, ako aj výpočet predikcie nezávisle od akejkoľvek postupnosti.

Letendre a kolektív [2] taktiež píšu, že predikčné metódy sú všeobecne charakterizované ako fyzikálne alebo štatistické, ale v praxi je hranica medzi týmito prístupmi nejednoznačná. Fyzikálne metódy explicitne modelujú fyzikálne atmosférické javy pri predikcii GHO použitím numerickej predpovede počasia (NPP) alebo snímok oblohy. Štatistické metódy predikujú GHO pomocou trénovania a štatisticky odvodených hodnôt. Ako príklad uvádzajú, že fyzikálny prístup k predpovedi môže použiť vektorovo založenú predikciu rozvoja oblakov použitím interpolácie nedávnych, po sebe nasledujúcich snímok oblohy a štatistický prístup môže použiť súčasné a historické výstupné hodnoty elektrárne k predikcii budúcich výstupných hodnôt.

2.1 Metódy založené na analýze časových radov

Do tejto kategórie patria hlavne autoregresné modely. Regresné metódy sa úspešne používajú v predikcii časových radov už dlhší čas. Použitím tohto prístupu zistíme vzťahy medzi prediktormi, premennými použitými na vstupe a premennými, ktoré máme predikovať [4].

2.1.1 Perzistentný (stály) model

Perzistentný model nie je autoregresný ale je jednoduchý predikčný model, ktorý je nutné spomenúť. Tento model predpokladá, že predpovedaná hodnota v čase *t* je rovnaká ako v čase *t – 1* :

$$X_t = X_t - 1$$

Tento model je naivný, pretože predpokladá nemennosť predpovedanej hodnoty. V praxi sa však používa na predikciu globálneho horizontálneho ožiarenia s krátkym časovým horizontom, napríklad 15 minút a malou úpravou predikcie podľa pozície Slnka na oblohe [2]. Využívajú ho operátori FVE pri riadení procesu výroby elektrickej energie. Model je ale nepresný pre predpoveď na dlhší časoví horizont ako 1 hodina. Jeho presnosť sa výrazne znižuje, ak sa mení oblačnosť. Perzistentný model je ale užitočný ako základ porovnania predikčných model-

ov. Je užitočné porovnávať výsledky predikčného modelu s jednoduchým predikčným modelom, ako je tento. Implementovať komplexný predikčný nástroj je úplne zbytočné, pokiaľ nepreukáže jasne lepšie vlastnosti ako jednoduchý predikčný model [4].

2.1.2 AR proces

AR (autoregressive) model, alebo proces znamená autoregresný proces. Je to stacionárny stochastický proces. Stochastický proces je nekonečnou postupnosťou náhodných veličín usporiadaných v čase. Pre modelovanie stochastického procesu je potrebné porozumieť povahe jeho náhodnosti. Túto náhodnú zložku je ľahšie definovať pri radoch, ktorých rozdelenie pravdepodobnosti sa v čase nemení. Takéto rady nazývame stacionárnymi. Časové rady môžu byť silne stacionárne a slabo stacionárne. Podrobnejšie vysvetlenie týchto vlastností nebudeme potrebovať. Vlastnosť stacionarity opisuje závislosť rozdelenia bieleho šumu.

Pre pochopenie AR procesu definujeme najskôr autoregresný proces rádu 1. Autoregresný proces rádu 1 = AR(1) je definovaný nasledovne:

$$X_t = \phi_1 X_{t-1} + a_t$$

kde X_t je hodnota v čase t a a_t je takzvaný *biely šum* (náhodná zložka). Biely šum je postupnosť nezávislých náhodných premenných (hodnôt) z normálneho rozdelenia [5]. Φ_1 je váha (koeficient) zložky. Využitím operátora spätného posunu B, pre ktorý platí $BX_t = X_{t-1}$, môžeme vzťah definujúci AR(1) proces prepísať do tvaru:

$$(1 - \phi_1 B)X_t = a_t$$

Po pochopení jednoduchosti AR(1) môžeme ľahšie pochopiť autoregresný proces vyššieho rádu. Autoregresný proces rádu p = AR(p) je taký proces, kedy je aktuálna hodnota časového radu v čase t lineárnou kombináciou predchádzajúcich hodnôt tohto radu. Ak je len konečný počet váh π nenulový, $\pi_1 = \phi_1$, $\pi_2 = \phi_2$, …, $\pi_p = \phi_p$, a $\pi_k = 0$ pre $k > p$, tak výsledný proces je autoregresným procesom rádu p [5].

AR(p) je definovaný vzťahom:

$$X_t = \phi_1 X_{t-1} + \ldots + \phi_p X_{t-p} + a_t$$

alebo v tvare s operátorom spätného posunu:

$$\phi_p(B)X_t = a_t$$

pričom

$$\phi_p(B) = \left(1 - \phi_1 B - \cdots - \phi_p B^p\right).$$

Pretože $\sum_{i=1}^{\infty}|\pi_i| = \sum_{i=1}^{p}|\phi_i| < \infty$, AR proces je vždy invertovateľný. Aby bol AR proces stacionárny, korene $\phi_p(B) = 0$ musia ležať mimo jednotkového kruhu.

AR procesy sú užitočné pri opise situácií, v ktorých súčasná hodnota časového radu záleží na jeho predchádzajúcich hodnotách a bielom šume (náhodnej zložke) [5].

2.1.3 MA proces

Skratka MA znamená „moving average“ a je to proces kĺzavých priemerov. V tomto procese je opisovaná hodnota časového radu lineárnou kombináciou súčasných a minulých hodnôt bieleho šumu (náhodnej zložky). To znamená, že sa v časovom rade prejaví s rôznou intenzitou vplyv hodnôt z minulosti. MA je tiež stacionárnym procesom. Opäť pre pochopenie definujeme najskôr MA(1) – proces kĺzavých priemerov prvého rádu:

$$X_t = a_t - \theta_1 a_{t-1}$$

respektíve:

$$X_t = (1 - \theta_1 B) a_t$$

Ak je len konečný počet váh ψ nenulový, $\psi_1 = -\phi_1, \psi_2 = -\phi_2, \ldots, \psi_q = -\phi_q$ a $\psi_k = 0$ pre $k > q$, tak výsledný proces je procesom pohyblivých priemerov rádu q [5].

MA(p) je definovaný vzťahom:

$$X_t = a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q}$$

a v tvare s použitím operátora spätného posunu:

$$X_t = \theta_q(B) a_t$$

pričom

$$\theta_q(B) = \left(1 - \theta_1 B - \cdots - \theta_q B^q\right)$$

a θ_t je predstavuje doznievajúcu náhodnú hodnotu t krokov dozadu.

Pretože $1 + \theta_1^2 + \ldots + \theta_q^2 < \infty$, konečný MA proces je vždy stacionárny. MA proces je invertovateľný ak korene $\theta_q(B) = 0$ ležia mimo jednotkového kruhu. MA procesy sú užitočné pri opisovaní javu, v ktorom udalosti produkujú okamžitý efekt, ktorý trvá len krátke časové obdobie [5].

2.1.4 ARMA proces

Ako je evidentné už zo skratky názvu tohto procesu, ARMA proces (autoregressive moving average = autoregresný proces kĺzavých priemerov) je spojením predchádzajúcich dvoch procesov, teda ARMA proces má aj autoregresnú zložku aj zložku kĺzavých priemerov.

Stacionárne procesy môžu byť reprezentované v AR forme alebo MA forme. Problém v týchto reprezentáciách ale je, že obsahujú až priveľa parametrov ako pre konečný MA rad, tak aj pre konečný AR rad, pretože pre dobrú aproximáciu je potrebný model vysokého rádu.

Vo všeobecnosti vysoký počet parametrov znižuje efektivitu odhadu. Preto pri budovaní modelu môže byť nevyhnutné zahrnúť obe zložky, AR aj MA, čo vedie k ARMA modelu [5].

Takýto model je užitočný pri opise rôznych časových radov. Spojenie AR a MA procesu často presnejšie popisuje skutočné dáta. Je to zapríčinené nevyhovujúcimi podmienkami použitia AR alebo MA procesov.

Bolo dokázané, že model je vhodný na predikciu, ak je v časovom rade lineárna závislosť. Hlavnou požiadavkou na ARMA model je, že časový rad musí byť stacionárny [4], čo znamená, že stredná hodnota a rozptyl bieleho šumu sa nemenia s časom.

ARMA(1,1) obsahuje AR(1) aj MA(1):

$$X_t = \phi_1 X_{t-1} + a_t - \theta_1 a_{t-1}$$

alebo:

$$(1 - \phi_1 B) X_t = (1 - \theta_1 B) a_t$$

Analogicky s predchádzajúcimi modelmi môžeme definovať ARMA(p, q) ako:

$$X_t = \phi_1 X_{t-1} + \ldots + \phi_p X_{t-p} + a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q}$$

alebo:

$$\phi_p(B) X_t = \theta_q(B) a_t$$

kde

$$\phi_p(B) = \left(1 - \phi_1 B - \cdots - \phi_p B^p\right)$$

$$\theta_q(B) = \left(1 - \theta_1 B - \cdots - \theta_q B^q\right)$$

V ARMA(p, q) procese p a q reprezentujú rády príslušných AR a MA polynómov. Preto ARMA(p, 0) je ekvivalentný procesu AR(p) a opačne aj pre MA proces.

2.1.5 ARIMA proces

ARIMA je integrovaný autoregresný proces kĺzavých priemerov. Oproti ARMA procesu má ešte o jednu zložku naviac. ARIMA(p, d, q) slúži na spracovanie nestacionárnych časových radov a práve táto pridaná zložka d opisuje, koľko krát musel byť časový rad diferencovaný, kým z neho bol stacionárny rad, respektíve po diferencii koľkého rádu bol z pôvodného nestacionárneho časového radu rad stacionárny. Stacionárny proces, ktorý je výsledkom diferencie nestacionárneho radu nie je nevyhnutne biely šum. Všeobecne diferencovaný rad pripomína ARMA(p, q) proces. Preto môže byť ARIMA(p, d, q) zapísaný nasledovne:

$$\phi_p(B)(1 - B)^d X_t = \theta_0 + \theta_q(B) a_t$$

kde stacionárny AR operátor

$$\phi_p(B) = \left(1 - \phi_1 B - \cdots - \phi_p B^p\right)$$

a invertovateľný MA operátor

$$\theta_q(B) = \left(1 - \theta_1 B - \cdots - \theta_q B^q\right)$$

nezdieľajú žiadny spoločný faktor. Parameter θ_0 hrá veľmi rozdielnu rolu pre $d = 0$ a $d > 0$. Ak je d rovné nule, pôvodný proces je stacionárny a je viazaný na priemer radu. Ak je d väčšie ako nula, θ_0 je nazývané deterministickým trendom a je často vynechávané z modelu, pokiaľ nie je naozaj potrebné [5].

Daigne [4] píše, že Reikard aplikoval ARIMA metódu na predikciu GHO. Porovnal ARIMA s inými metódami na časovom horizonte 24 hodín, a tvrdí, že ARIMA model zachytáva presné prechody v ožiarení spojené s denným cyklom presnejšie než ostatné metódy.

2.2 Metódy strojového učenia

Metódy strojového učenia patria do oblasti umelej inteligencie. O výskum umelej inteligencie je stále väčší záujem. Pribúdajú nové techniky, známe techniky sa zlepšujú. Techniky umelej inteligencie sú použiteľné vo veľa oblastiach, nie len v predikcii, ale aj pre široké spektrum aplikácií, kompresii dát, optimalizácii, rozoznávaní vzorov a klasifikácii. Techniky umelej inteligencie boli brané aj pri predikcii za alternatívne riešenia, ale stali sa vďaka svojim výsledkom plnohodnotným prostriedkom na riešenie problému predikcie a sú častokrát po vhodnej implementácií presnejšie a spoľahlivejšie ako klasické riešenia. Je to kvôli tomu, že na rozdiel od klasických metód analýzy časových radov, kedy predikované hodnoty časového radu závisia od predchádzajúcich hodnôt, metódy strojového učenia môžu brať tieto hodnoty ako samostatné, neuvažujú pôvodnú štruktúru dát a dokážu tak riešiť aj nelineárne problémy, pri ktorých klasické metódy zlyhávajú alebo dodávajú neuspokojivé výsledky.

Tieto metódy väčšinou fungujú ako čierna skrinka, čo znamená, že nevieme ako model vypočítal výstupnú hodnotu. Pri metódach strojového učenia sa spolieha na ich schopnosť naučiť sa rozoznať vzory vo vstupných dátach. Metódy strojového učenia je možné natrénovať na rôznych dátach, čo z nich robí veľmi univerzálny nástroj.

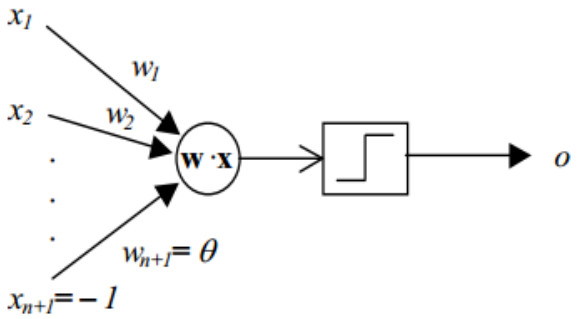
2.2.1 Umelá neurónová sieť

Umelá neurónová sieť (UNS) sa snaží napodobiť štruktúru a funkcionalitu prirodzených neurónových sietí, teda nervových systémov živých organizmov. Tak ako sa prirodzená neurónová sieť skladá z množstva neurónov, ktoré sú medzi sebou poprepájané synapsiami, UNS sa skladá z formálnych neurónov. Sinčák a Andrejková [6] definujú neurónovú sieť ako masívne paralelný procesor, ktorý má sklon k uchovávaniu experimentálnych znalostí a ich ďalšieho využívania. Napodobňuje ľudský mozog v dvoch aspektoch:

- poznatky sú zbierané v neurónovej sieti počas učenia,
- medzineurónové spojenia (synaptické váhy) sú využívané na ukladanie znalostí.

Jednou z veľmi významných vlastností neurónových sietí je, že svojim spôsobom je takzvaným univerzálnym aproximátorom funkcií. Môže sa nám stať, že máme systém, ktorého popis je mimoriadne náročný alebo je systém natoľko zložitý, že jeho popis je skoro nemožný. Máme však dáta, ktoré do systému vstupujú a k nim odpovedajúce výstupy. V takejto situácii, môžeme použiť vhodnú UNS a pokúsiť sa ju naučiť chovať sa ako sledovaný systém pomocou trénovacích údajov [6]. Presne tento prístup k modelovaniu výpočtu sa najčastejšie využíva pri predikcii výroby elektrickej energie z FVE.

Základným prvkom UNS je formálny neurón. Špeciálnym typom formálneho neurónu je perceptrón. Perceptrón má na vstupných prepojeniach váhy (Obrázok 3). Váha na danom prepojení patrí k hodnote, ktorú neurón prijíma daným prepojením. Vo váhach perceptrónu sa uchovávajú znalosti UNS.



Obrázok 3: Perceptrón. [7]

Beňušková definovala perceptrón nasledovne [7]: „Perceptrón je model neurónu, ktorý prijíma vstupné signály $\bar{x} = (x_1, x_2, \dots, x_{n+1})$ cez synaptické váhy tvoriace váhový vektor $\bar{w} = (w_1, w_2, \dots, w_{n+1})$. Vstupný vektor $\bar{x}$ sa nazýva *vzor* alebo obrazec (angl. *pattern*). Zložky vstupného vektora môžu nadobúdať reálne alebo binárne hodnoty. Zložky váhového vektora sú reálne čísla. Výstup perceptrónu o je daný vzťahom:

$$o = f(net) = f(\bar{w} \cdot \bar{x}) = f\left(\sum_{j=1}^{n+1} w_j x_j\right) = f\left(\sum_{j=1}^{n} w_j x_j - \theta\right)$$

kde premenná *net* označuje váhovanú sumu vstupov, t.j. skalárny (zložkový) súčin váhového a vstupného vektora. Funkcia *f* sa volá *aktivačná funkcia* perceptrónu. V tejto notácii predpokladáme, že perceptrón má *n+1* vstupov. Hodnota *(n+1)*-ého vstupu je vždy –1 a $w_{n+1} = \theta$, čo je hodnota prahu excitácie perceptrónu.“

Spôsobov opisu štruktúry neurónovej siete je viacero. Neuróny môžeme označiť za vrcholy, spojenia neurónov za hrany a celú neurónovú sieť môžeme opísať ako orientovaný graf. Analyzovať takto štruktúrovanú sieť je ťažké, preto bývajú častejšie neurónové siete organizované do jednoduchších pravidelných štruktúr ako je napríklad viacvrstvová štruktúra. Vo viacvrstvovej štruktúre rozlišujeme nasledujúce vrstvy:

- Vstupná vrstva – neuróny tejto vrstvy prijímajú vstupné informácie z prostredia mimo neurónovej siete. Informačný tok (výstup týchto neurónov) pokračuje k neurónom ďalšej vrstvy.
- Skrytá vrstva – neuróny tejto vrstvy prijímajú vstupné informácie zo vstupnej vrstvy, inej skrytej vrstvy, alebo aj z prostredia mimo neurónovej siete. Informačný tok pokračuje k neurónom ďalšej vrstvy.
- Výstupná vrstva – neuróny tejto vrstvy prijímajú vstupné informácie z predchádzajúcej vrstvy ale výstup neurónov tejto vrstvy vyúsťuje do prostredia mimo neurónovej siete.

Neurónová sieť má jednu vstupnú a jednu výstupnú vrstvu. Skrytých vrstiev môže byť niekoľko. Neuróny dvoch vrstiev sú prepojené spôsobom „každý s každým“, čiže ak je vrstva M s počtom neurónov m prepojená s vrstvou N s počtom neurónov n, počet prepojení medzi týmito dvomi vrstvami je $m \cdot n$.

Vo všeobecnosti sa štruktúra neurónových sietí rozdeľuje do dvoch kategórií [6]:

- Dopredné neurónové siete – pri týchto sieťach sa signál šíri po orientovaných synaptických prepojeniach len jedným smerom a to dopredu.
- Rekurentné neurónové siete - pri rekurentných sieťach je dosť ťažké rozdelenie vrstiev a neurónov na vstupné, resp. výstupné. Niekedy neuróny v rekurentných sieťach predstavujú vstupné ale aj výstupné typy neurónov a tým aj vrstiev. Špeciálnym prípadom sú tzv. čiastočne rekurentné UNS, v ktorých je stanovená určitá požiadavka na štruktúru a na prepojenia. Napríklad vrstvové čiastočne rekurentné siete pripúšťajú šírenie signálu oboma smermi.

UNS sa trénujú na trénovacej množine dát. V tejto množine je vzorka historických vstupných dát aj s požadovaným výstupom, ktorý má UNS predikovať. Existuje viacero trénovacích algoritmov, ktoré tu nie je nutné opisovať. Všeobecne sa pri trénovaní menia váhy perceptrónov tak, aby sa celkový výsledok priblížil k požadovanej hodnote. Pri trénovaní UNS hrozí takzvané *pretrénovanie*, kedy to vyzerá, že sa UNS naučí naspamäť predikovať hodnoty z trénovacej množiny, ale pri predikciách zo vstupných údajov mimo trénovacej množiny je veľmi nepresná.

Táto situácia nastáva ak je trénovacia množina príliš malá, alebo sa UNS trénovala na niekoľkých záznamoch opakovane.

Viacvrstvové UNS, ktoré sa trénujú metódou spätného šírenia chýb, sú schopné riešiť aj nelineárne problémy [7]. Preto sa využívajú pri predikcii tak, že sa natrénujú na historických dátach a potom vedia generovať výstup podľa vstupných dát, ktoré sú predikovanými vstupnými údajmi pre čas, pre ktorý má neurónová sieť predikovať výstupné údaje. Rekurentné UNS dokážu generovať časový rad podľa vzoru vstupnej vzorky časového radu. Vďaka svojej rekurentnosti si UNS uloží sebou generovaný člen časového radu, zahrnie ho do vstupnej vzorky a generuje nový člen časového radu.

Daigne [4] píše, že použitím trénovacích dát UNS zredukovali odmocninu zo strednej kvadratickej chyby (RMSE, vysvetlené v kapitole 3) priemerného denného GHO až o 15 % v porovnaní s 12 až 18 hodinovými predikciami NPP. Reikardove výsledky ukázali, že v rozlíšení 60, 30 a 15 minút dávajú ARIMA modely presnejšie výsledky. Podľa výsledkov Sfetsosa a Coonicka je najvhodnejšia UNS [4].

2.2.2 Metóda podporných vektorov

Anglický názov tejto metódy je *support vector machine* (SVM). SVM je klasifikátor odvodený od strojového učenia, ktorý mapuje vektor prediktorov (vstupné parametre predikcie) do viacrozmerného priestoru cez lineárne alebo nelineárne jadrové (angl. *kernel*) funkcie. V probléme binárnej klasifikácie sú dve skupiny (-1 a +1) oddelené vo viacrozmernej nadrovine podľa princípu minimalizácie rizika. Zámerom je nájsť rozdeľujúcu nadrovinu skonštruovanú podľa vektora prediktorov namapovaného do viacrozmerného priestoru nelineárnou funkciou a vektor váh s počiatkom odchýlky, ktorá klasifikuje všetky hodnoty do jednej z dvoch skupín. Keďže v probléme binárnej klasifikácie je týchto rozdeľujúcich nadrovín nekonečný počet, cieľom je nájsť optimálnu rovinu, ktorá je od oboch skupín najviac vzdialená [8].

Zjednodušene by sme mohli povedať, že pri binárnej klasifikácii SVM rozdeľuje body v priestore do dvoch skupín (priestorov, nadrovín), ktoré môže oddeliť rovinou. Najlepším rozdelením je také, ktoré tieto dve skupiny od seba rozdeľuje s čo najväčšou medzerou medzi nimi.

Metóda SVM bola pôvodne používaná na rozoznávanie vzorov a binárnu klasifikáciu, ale princípy jej fungovania môžu byť ľahko rozšírené pre regresiu a predikciu časových radov. SVM dosahuje dobré výsledky pre nelineárne problémy, pričom nepotrebuje poznať štruktúru dát, čo je výhodou pri predikcii nelineárnych časových radov. SVM sa používa iba zriedka, hoci má veľa teoretických výhod pre klasifikačné aj regresné úlohy [9].

Thissen a kolektív [9] píšu, že pri zložitejšej trénovacej množine určenej na porovnanie výsledkov, SVM model dosiahol lepšie výsledky pri predikcii časových radov ako ARMA model a vo väčšine prípadov aj lepšie výsledky ako Elmanova neurónová sieť. Pri použití menšej trénovacej množiny, ktorá obsahovala len desatinu dát boli výsledky SVM a ARMA modelov rovnako dobré ale Elmanovu neurónovú sieť nebolo možné použiť pre predikciu časového radu.

2.2.3 Náhodné lesy klasifikačných alebo regresných stromov

Náhodné lesy sú kombináciou stromových prediktorov takých, že každý strom je založený na náhodnej vzorke z vektora hodnôt a s normálnym rozdelením pre všetky stromy v lese. Tento princíp môže byť aplikovaný aj na regresiu [10].

Teóriu náhodných lesov predstavil Breiman [10] aj s metódou *bagging*-u klasifikačných stromov, kedy stromy nezávisia od predchádzajúcich stromov a každý je nezávisle skonštruovaný zo vzorky z množiny dát. Breiman navrhol náhodné lesy, ktoré pridávajú ďalšiu vrstvu náhodnosti do metódy bagging-u. Okrem konštrukcie každého stromu z inej vzorky dát, náhodné lesy menia spôsob ako sú klasifikačné alebo regresné stromy konštruované. V klasických stromoch je každý uzol rozdelený najlepším možným rozdelením medzi všetkými premennými. V náhodných lesoch je pri regresii každý uzol rozdelený najlepším rozdelením spomedzi podmnožiny prediktorov, ktorá je náhodne vybraná pri danom uzle. Táto kontra-intuitívna stratégia dosahuje veľmi dobré výsledky v porovnaní s mnohými inými metódami klasifikácie ako SVM a UNS, a je odolná voči pretrénovaniu. Výsledná predikovaná hodnota je vybraná agregáciou predikcií jednotlivých stromov. Pri klasifikácii je vybraná väčšinová (najpočetnejšia) výstupná hodnota a pri regresii je vybraný priemer z hodnôt [11].

Pri konfigurácii náhodného lesa sú dôležité tri parametre:

1. Počet skonštruovaných stromov v lese.
2. Počet náhodne vybraných prediktorov pre vytvorenie uzla stromu.
3. Minimálna veľkosť koncových uzlov stromu.

Ideálny počet stromov v lese môže byť rovnaký pre klasifikáciu aj regresiu. Hodnoty ostatných dvoch parametrov sa však líšia pri klasifikácii a regresii.

V Španielsku použili kvantilové regresné lesy, ktoré sú založené na Breimanových náhodných lesoch. Kvantily dávajú viac informácií o rozložení výstupných hodnôt ako o funkcii prediktorov než o samotnom priemere. S použitím tejto metódy implementovali model, ktorý predikoval dennú produkciu FVE so strednou chybou menšou než 1,3 % a s celkovou priemernou absolútnou chybou menšou než 9,5 % [12].

2.2.4 Meranie presnosti metód strojového učenia

Pri meraní presnosti modelu, ktorý je založený na metóde strojového učenia, je potrebné vytvoriť dve množiny dát:

1. Trénovaciu množinu – obsahuje dáta, na ktorých sa model natrénuje (nastavia sa hodnoty váh perceptrónov UNS).
2. Testovaciu množinu – obsahuje dáta, na ktorých sa overí schopnosť modelu odhadnúť predikovanú hodnotu.

Pre zachovanie správnych podmienok pre testovanie musí platiť pravidlo, že prienikom týchto dvoch množín musí byť prázdna množina, čo znamená, že žiadna jednotka vstupných údajov z trénovacej množiny nesmie byť v testovacej množine. Zabezpečí sa tak nezávislosť testovacej množiny. Správny výber trénovacej a testovacej množiny je kritickým pre hodnotenie výkonu predikčného modelu.

2.3 Fyzikálne metódy

Pri fyzikálnych metódach predikčné modely predikujú výstupné hodnoty počítaním rovníc opisujúcich fyzikálne zákony a vzťahy medzi vstupnými parametrami. Fyzikálne predikčné modely na predikciu produkcie FVE sa implementujú na mieru pre konkrétnu FVE podľa charakteristiky fotovoltaických panelov elektrárne a vstupných hodnôt ako predpovedaná hodnota GHO a teplota vzduchu alebo teplota zadnej strany fotovoltaických panelov, respektíve teplota buniek fotovoltaických panelov pre daný časový interval, pre ktorý má model predikovať výslednú hodnotu.

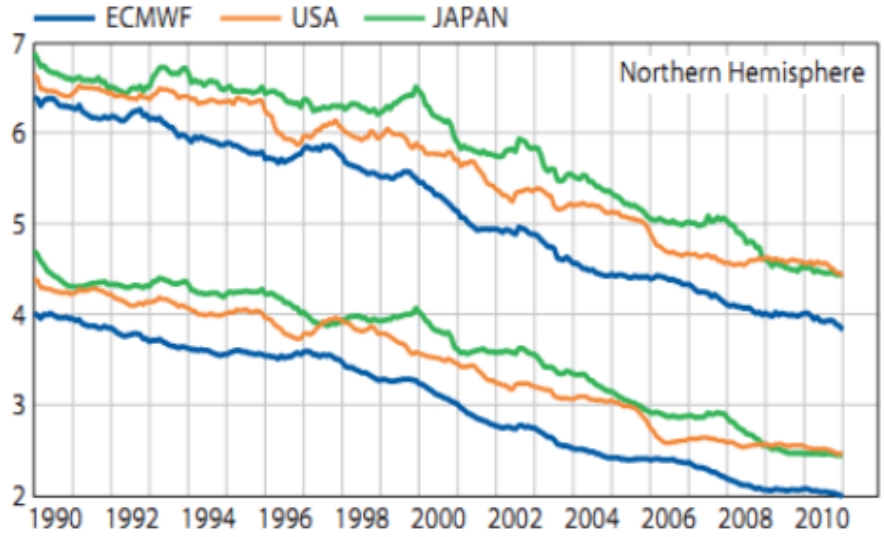
Fyzikálne modely sú veľmi presné po správnej implementácii na mieru danej FVE, ale sú náchylné na chyby v predikcii vstupných parametrov ako GHO. Z tohto dôvodu sa pri fyzikálnych modeloch často využíva dodatočné spracovanie výstupu nazývané *postprocessing*.

Fyzikálnym modelom pre predikciu celkového stavu atmosféry a teda aj potrebných vstupných údajov pre predikciu produkcie FVE ako teplota, oblačnosť alebo aj GHO, je model numerickej predpovede počasia (NPP).

2.3.1 Modely numerickej predpovede počasia

Modely NPP sa všeobecne používajú na predikciu stavu atmosféry až na 15 dní dopredu. Časový vývoj stavu atmosféry je modelovaný základnými diferenčnými rovnicami, ktoré popisujú fyzikálne zákony vládnuce počasiu. Začiatočné podmienky sú získavané z celosvetového pozorovania a v prvom kroku je budúci stav atmosféry vypočítaný globálnym modelom NPP.

Globálne modely NPP v súčasnosti fungujú v asi pätnástich spoločnostiach poskytujúcich predpoveď počasia. Vlastné globálne modely NPP majú napríklad US National Oceanic and Atmospheric Administration (NOAA) a European Centre for Medium-Range Weather Forecasts (ECMWF). Vlastný globálny model NPP majú aj v Japonsku. ECMWF v súčasnosti poskytuje najkvalitnejšie strednodobé a dlhodobé meteorologické predpovede na svete, ako je aj vidieť na nasledujúcom grafe (Obrázok 4) [3].



Obrázok 4: Porovnanie odmocniny zo strednej kvadratickej chyby (RMSE) pre trojdňovú (nižšie) a päťdňovú (vyššie) predpoveď. [3]

Od ECMWF má dáta aj Slovenský Hydrometeorologický ústav (SHMÚ), ktorý vytvoril vlastný model NPP s názvom Aladin, ktorý však nie je globálny, ale regionálny a poskytuje najpresnejšiu predpoveď počasia pre územie Slovenska [3].

Globálne modely majú zvyčajne hrubé rozlíšenie a nedovoľujú detailné mapovanie. Aj keď sa rozlíšenie za posledné roky zvýšilo, dnes je to stále v rozmedzí 16 až 50 kilometrov (horizontálne rozlíšenie). Regionálne modely pokrývajú menšiu časť Zeme, preto môžu operovať nad vyšším rozlíšením. ECMWF počíta strednodobé predpovede na 10 dní s horizontálnym rozlíšením 50 km, zatiaľ čo model Aladin má horizontálne rozlíšenie 9 km [13].

Na modely NPP sa môže aplikovať dodatočné spracovanie výstupu. Takýto postprocessing môže byť aplikovaný za účelom modelovania špecifických lokálnych efektov a vlastností konkrétnych oblastí ako nížiny a pohoria, kde je rozlíšenie na niekoľko kilometrov skresľujúce, alebo môže byť postprocessing aplikovaný za účelom zníženia chýb modelu NPP.

Výpočet predpovede počasia modelu NPP je výpočtovo náročný proces, preto sa výpočet vykonáva na superpočítačoch. V Japonsku používajú na výskum v oblasti predpovede počasia najvýkonnejší počítač na svete a najvýkonnejší počítač v Európe používa na predpoveď počasia ECMWF [14].

2.4 Hybridné metódy

Hybridné modely boli navrhnuté pre prekonanie nedostatkov individuálnych modelov ako sú napríklad fyzikálne modely. Hybridné metódy spájajú rozdielne metódy pre zvýšenie presnosti predpovede. Tieto modely dosahujú vyššiu presnosť predpovede kombináciou modelov s rôznym prístupom. Často spájajú metódy založené na analýze časových radov pre spracovanie lineárneho vstupu a využívajú strojové učenie na rozoznávanie nelineárnych vzorov. Veľmi často používanou kombináciou je predikcia meteorologických parametrov využitím metód založených na analýze časových radov alebo metódou strojového učenia a následné zadanie týchto predikovaných hodnôt ako vstupu do fyzikálneho modelu, ktorý vypočíta množstvo vyprodukovanej elektrickej energie za daných vstupných meteorologických podmienok.

Letendre [2] tvrdí, že predpovedať produkciu FVE použitím iba štatistických metód nie je vo všeobecnosti súčasťou moderných predikčných systémov solárnej energie, ale hybridný prístup využíva pokročilé štatistické techniky ku korekcii známych nedostatkov spojených s rozdielnymi predikčnými metódami, úpravami stredných chýb alebo metódami strojového učenia.

Pri fyzikálnych modeloch sa často využíva postprocessing so štatistickým prístupom. Pri takomto prístupe sa spracovávajú historické dáta pre korekciu predikovaných hodnôt. Metóda štatistickej korekcie výstupu sa nazýva *model output statistics* (MOS). MOS používa štatistické vzťahy medzi pozorovanými elementmi počasia a meteorologickými dátami, satelitnými údajmi alebo modelovanými parametrami k získaniu štatistickej úpravy výstupu [2]. Nevýhodou MOS je ako pri všetkých štatistických prístupoch potreba presných historických dát.

Hoci sa na MOS používajú prevažne metódy založené na analýze časových radov, akákoľvek štatistická metóda patrí do konceptu MOS. Daigne [4] píše o brazílskom modeli NPP, ktorého predpovede GHO boli veľmi nadhodnotené a aplikovaním UNS bolo dosiahnuté značné zlepšenie výsledkov.

3 Metriky presnosti predikcie

Existuje viacero metrík pre určenie presnosti predikcie výroby elektrickej energie z FVE. Rôzne metriky sa hodia pre rôzne účely. Operátori FVE potrebujú metriky, ktoré presne odrážajú ceny predikčných chýb (straty). Výskumníci vyžadujú indikátory na porovnanie výsledkov viacerých predikčných modelov alebo jedného modelu pri rozdielnych podmienkach [15].

Pri všetkých metrikách sa počíta s chybou e_i, ktorá je rozdielom predpovedanej hodnoty y_p a nameranej hodnoty $y_{n,}$, dolný index i označuje i-te poradie premenných v intervale od 1 po N, kde N je počet všetkých párov predpovedaná-nameraná hodnota v testovacej množine:

$$e_i = y_{i,p-}y_{i,n}$$

Štandardnými používanými metrikami presnosti predikcie sú nasledovné štatistické ukazovatele:

- Stredná chyba = mean bias error - MBE:

$$MBE = \frac{1}{N}\sum_{i=1}^{N} e_i$$

- Stredná kvadratická chyba = mean square error = MSE

$$MSE = \frac{1}{N}\sum_{i=1}^{N} {e_i}^2$$

- Odmocnina zo strednej kvadratickej chyby = root mean square error - RMSE:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} {e_i}^2}$$

- Priemerná absolútna chyba = mean absolute error = MAE:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |e_i|$$

- Maximálna absolútna chyba = maximal absolute error = MAXAE:

$$MAXAE = \max_{i=1,..N} |e_i|$$

- Štandardná odchýlka = standard deviation = SDE:

$$SDE = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N} (e_i - \overline{e_\iota})^2}$$

Charakteristika MBE (označuje sa aj „bias“) je priemernou predikčnou chybou a zapuzdruje systémovú tendenciu modelu k nadhodnoteniu alebo podhodnoteniu predpovedanej hodnoty (systematická chyba). Charakteristika MAE je priemerným rozsahom (rozptylom) predikčnej chyby. Charakteristika RMSE je MAE veľmi podobná, ale dáva viac váhy väčším chybám. Pellandová [15] spomína Madsena a kolektív, ktorý argumentuje, že väčšie chyby majú disproporčnú cenu, a preto RMSE lepšie odráža cenu predikčných chýb pre systémových operátorov než MAE. Preto je RMSE najpoužívanejšou metrikou presnosti predikcie výroby elektriny. Charakteristika MAXAE vyjadruje maximálnu chybu. Tento ukazovateľ je vhodný pre výskumníkov, ktorý môžu na základe tohto ukazovateľa porovnať maximálnu chybovosť porovnávaných modelov. Vzťahy medzi MSE, SDE a MBE sú nasledovné:

$$RMSE^2 = MSE = MBE^2 + SDE^2$$

Inými slovami, SDE zachytáva časť RSME, ktorá nie je spôsobená systematickou chybou a poskytuje náznak hodnoty RMSE, ktorá môže byť docielená elimináciou MBE [15].

Charakteristiky RMSE, MAE a MAXAE majú vlastnosť metriky, takže merajú vzdialenosť predpovede od nameraných údajov. Pre dokonalú predpoveď sú tieto charakteristiky rovné nule. Používajú sa aj relatívne respektíve percentuálne hodnoty rRMSE, rMAE a rMBE pričom normalizácia je vzhľadom na namerané hodnoty [3]. Takéto ukazovatele sú vhodné pre celkové hodnotenie presnosti predikcie, pretože sú normalizované. Odchýlka od očakávaného výsledku o 10 jednotiek je rovnaká aj pri očakávanej hodnote 100 jednotiek, aj pri očakávanej hodnote 1000 jednotiek. Po normalizácii vzhľadom na očakávané (namerané) hodnoty dostaneme odchýlku 10 % pri prvej hodnote a 1 % pri druhej hodnote, čo je desaťnásobný rozdiel. Preto je vhodné porovnávať výsledky v normalizovanom tvare.

4 Predikcia výroby elektriny fotovoltaickými elektrárňami

Pri predikovaní produkcie FVE sa používajú metódy opísané v kapitole 2. Predikčné metódy sme v tejto kapitole rozdelili do dvoch skupín: fyzikálne a štatistické. Fyzikálny prístup využíva vzťahy medzi premennými pre výpočet rovníc popisujúcich produkciu elektrickej energie. Štatistický prístup sa spolieha na historické dáta, podľa ktorých sa môže predikčný model trénovať. Fyzikálny model funguje ako biela skrinka, kedy vidíme ako model vypočíta výstupné hodnoty a štatistický model väčšinou funguje ako čierna skrinka, kde nevidíme ako model dospel k výsledkom.

Pre fyzikálny model sú potrebné presné meteorologické vstupné údaje ako GHO a teplota vzduchu alebo teplota zadnej steny panelov alebo teplota fotovoltaických článkov, a presné údaje o fotovoltaických paneloch, ktoré charakterizujú produkciu elektrickej energie. Preto sú fyzikálne modely navrhnuté presne na mieru pre danú špecifikáciu FVE a charakteristiku používaných fotovoltaických panelov.

Pri štatistickom prístupe sú dôležité historické dáta ako predpovedané alebo namerané hodnoty GHO, ukazovatele počasia (teplota, rýchlosť vetra, oblačnosť), údaje sledovania oblačnosti, a k nim nameraná produkcia FVE za daných meteorologických podmienok. Takéto historické dáta sa použijú ako časový rad alebo ako trénovacia množina dát, ktorá sa použije na natrénovanie predikčného modelu ako napríklad umelá neurónová sieť. Pre štatistické metódy je lepšie, ak sa trénujú na historických dátach predpovedí, než na skutočných nameraných dátach, pretože predikcia sa vždy vypočítava na dátach predpovede meteorologických premenných a tie majú rovnakú chybu predikcie ako majú historické dáta predpovede meteorologických premenných. Rozdiel medzi predpovedanými a skutočnými hodnotami má v takomto prípade menší negatívny vplyv na presnosť výslednej predikcie výroby elektriny.

Presnosť predikcie výroby elektrickej energie závisí na presnosti predpovedi vstupných meteorologických premenných. Keďže modely využívajúce fyzikálny prístup berú do úvahy vstupné dáta pre daný jeden deň, ich presnosť býva nižšia v porovnaní so štatistickým prístupom, kvôli chybám v predpovedi hodnôt vstupných premenných. Hoci štatistický prístup nemodeluje priamo produkciu FVE, jeho výhodou je, že je menej náchylný na chyby v predpovedi vstupných údajov v jednotlivých dňoch, pretože trénovacia množina obsahuje viacero dní, podľa ktorých sa model učí predikovať produkciu FVE.

4.1 Predikcia globálneho horizontálneho ožiarenia

Štatistické metódy pre predikciu GHO sa používajú rovnaké ako štatistické metódy pre predikciu produkcie FVE. Rozdiel je vo vstupných a výstupných parametroch a štruktúre dát v trénovacích množinách. Fyzikálnymi modelmi pre predikciu GHO sú modely NPP (opísané v kapitole 2.3.1).

V štatistických modeloch sa používajú historické dáta, čo je problémom pre nové FVE v oblasti, z ktorej historické dáta zatiaľ neexistujú. To v praxi znamená, že FVE si musí dáta zbierať a až po čase sa presnosť predpovede spresní do prijateľnej miery. Väčšinou sa meracie prístroje v oblasti inštalujú dopredu, aby boli prístupné historické dáta už pri dokončení výstavby FVE.

Výber správnej metódy alebo modelu závisí od časového horizontu, pre ktorý chceme predikovať. Štatistické modely sú presnejšie pre krátkodobé predpovede a ich výpočet predikcie je na počítači veľmi rýchly, takže sa môžu často opakovať, čo ich priamo predurčuje pre krátkodobé predpovede. Modely NPP sú presnejšie pri predikcii na vzdialenejší časový horizont a aj ich výpočet trvá dlhšie.

Ako som spomínal vyššie, najväčší vplyv na hodnotu GHO a hlavne na zmenu tejto hodnoty má oblačnosť. Preto je *sledovanie oblačnosti* veľmi dôležitou súčasťou predikcie GHO. Niektoré predikčné modely sú založené z väčšej časti na sledovaní oblačnosti a tvoria tak samostatnú kategóriu. Dáta zo sledovania oblačnosti využívajú ale aj oba vyššie spomínané typy predikčných modelov.

4.1.1 Sledovanie oblačnosti

Najväčší vplyv na úroveň GHO má stav oblohy. Oblačnosť je veľmi premenlivá s časom aj priestorom, pretože sa oblaky v atmosfére hýbu a menia. Preto je sledovanie stavu oblačnosti základnou úlohou pre predpoveď GHO.

V súčasnosti sa pre predikciu GHO používajú snímky oblačnosti zo satelitov aj snímky zo zeme. Oba spôsoby sledovania oblačnosti poskytujú pomerne veľké časové aj priestorové rozlíšenie. Základom pre využitie týchto snímok je sledovanie štruktúry a pohybu oblakov v zaznamenaných časových krokoch. Pokročilejšie modely sledujú aj tiene oblakov. Na základe sledovaných snímok je možné predpovedať ako sa oblaky zmenia, kam smerujú a aké z toho vyplývajú následky pre úroveň GHO na povrchu v danej sledovanej oblasti.

Chyby v predikciách založených na sledovaní oblačnosti prudko vzrastajú pri nízkej výške Slnka na oblohe (nízkom uhle dopadu slnečných lúčov) a vysokých priestorových nepravidelnostiach povrchu (pohoria, doliny).

Spracovaním satelitných aj pozemných snímok, oblaky môžu byť detegované, charakterizované a využité k predpovedi GHO pomerne presne až na 6 hodín dopredu. Modely časových radov založené na dátach zo sledovania oblačnosti detegujú pohyb oblakových štruktúr používaním vektorových polí [4].

Modely spracovávajúce satelitné snímky dokážu zachytiť väčšiu oblasť a môžu preto sledovať pohyb oblačnosti na väčšom priestore po dlhší čas. Modely spracovávajúce snímky oblohy získané z povrchu Zeme nedokážu sledovať tak veľký priestor. Prakticky nevidia oblačnosť vzdialenú viac než 30 minút (závisí od rýchlosti pohybu oblačnosti). Majú ale vyššie priestorové a časové rozlíšenie a majú schopnosť zaznamenať náhle zmeny. Preto sú presnejšie pre predikcie do časového horizontu 30 minút. Satelitné snímky sú najpresnejšie pre predikcie pre časový horizont 30 minút až 6 hodín. Pre predpovede na obdobie vzdialenejšie ako 6 hodín sú už potom presnejšie modely NPP.

5 Vlastné riešenie

5.1 Špecifikácia

Našou úlohou je predikovať produkciu fotovoltaickej elektrárne na deň dopredu. Vstupnými prediktormi do predikcie sú meteorologické dáta opisujúce stav atmosféry, množstvo solárnej energie dopadajúcej na zemský povrch, slnečné koordináty popisujúce polohu slnka na oblohe a dĺžka slnečného svitu v daný deň. Výstupom predikcie je množstvo vyrobenej elektrickej energie za podmienok určených vstupnými prediktormi.

Naším cieľom je dosiahnuť presnosť predikcie dostatočne vysokú na to, aby použitie predikcie produkcie fotovoltaickej elektrárne bolo užitočné pri riadení produkcie elektrárne, pri jej integrácii do elektrickej siete a pri obchodovaní s elektrickou energiou.

K dispozícii máme historické záznamy produkcie troch fotovoltaických elektrární od 30.11.2014 do 31.10.2015 pre dve elektrárne a od 1.7.2014 do 31.10.2015 pre tretiu elektráreň. K týmto záznamom máme k dispozícii historické záznamy predpovede počasia. Dáta predpovede počasia máme od Slovenského hydrometeorologického ústavu (SHMÚ). Dáta od SHMÚ sú z modelu numerickej predpovede počasia (NPP) Aladin. Predikujeme produkciu FVE na nasledujúci deň, čo je časové obdobie za horizontom šiestich hodín, takže použitie dát z modelu NPP je najlepšou možnosťou, pretože v takomto časovom horizonte dosahujú modely NPP najpresnejšie výsledky v porovnaní s ostatnými predikčnými modelmi. Model Aladin navyše dosahuje najpresnejšie predpovede počasia na území Slovenska [3], čo tiež prispeje k presnosti našej predikcie. Dáta obsahujú potrebné meteorologické údaje na predpoveď výroby elektrickej energie ako sú globálne horizontálne ožiarenie (GHO), teplota vzduchu, rýchlosť vetra a iné. Pre lokalitu fotovoltaických elektrární sme vypočítali hodnoty slnečných koordinátov a dĺžku slnečného svitu.

Presnosť predikcie vyhodnocujeme pomocou štatistických metrík presnosti. Kľúčovými pre porovnávanie výsledkov sú normalizované hodnoty metrík odmocnina zo strednej kvadratickej chyby (rRMSE) a priemerná absolútna chyba (rMAE).

5.2 Návrh

Pre implementáciu vlastného riešenia sme si zvolili programovací jazyk R, ktorý je vhodný na dátovú analýzu. Pre predikciu sme plánovali použiť umelú neurónovú sieť (UNS) a keďže nás pri analýze predikčných metód zaujala metóda náhodného lesa regresných stromov (NLRS), rozhodli sme sa použiť ju pre porovnanie. Už pri prvých experimentoch dosahovala predikcia použitím NLRS o málo menšiu chybu predikcie a pri použití UNS sme narazili na

implementačné chyby v balíku *neuralnet*, ktorý implementuje UNS v jazyku R. Preto sme sa rozhodli ďalej pokračovať len s použitím NLRS implementovaného v balíku *randomForest*. Ako sa neskôr ukázalo, táto metóda je pre naše riešenie vhodnejšia, pretože pri nej môžeme použiť menšiu trénovaciu množinu obsahujúcu veľmi podobné záznamy, pričom pri použití UNS s takouto trénovacou množinou by nám hrozilo pretrénovanie, voči ktorému je metóda NLRS odolná.

Pri predikcii nepotrebujeme predikovať časový rad, keďže hodnota vyprodukovanej elektrickej energie nezávisí od postupnosti hodnôt produkcie predchádzajúcich záznamov, ale od meteorologických a iných podmienok. To nám umožňuje vybrať záznamy do trénovacej množiny nezávisle od času a dátumu vytvorenia záznamu. Pre zvýšenie presnosti predikcie vytvárame nový predikčný model pre každú samostatnú predikciu hodnoty vyprodukovanej energie a do trénovacej množiny vyberáme najpodobnejšie záznamy so záznamom, pre ktorý predikujeme, na rozdiel od klasických prístupov k predikcii, kedy sa natrénuje predikčný model a používa sa pre niekoľko predikcií, alebo kedy sa do trénovacej množiny vyberajú záznamy časovo predchádzajúce záznamu, pre ktorý predikčný model predikuje výslednú hodnotu. Pre výber najpodobnejších dát do trénovacej množiny počítame pre každý potenciálny záznam hodnotu jeho podobnosti so záznamom, pre ktorý predikčný model predikuje hodnotu vyprodukovanej energie.

Pre hodnotenie presnosti predikcie počítame štatistické metriky presnosti opísané v kapitole 3. Výsledky experimentov s nastavením predikčného modelu a výberu dát do trénovacej množiny porovnávame hlavne podľa hodnôt metrík rRMSE a rMAE, pretože hovoria o celkovej chybe predikcie a ich hodnoty sú normalizované, takže ich hodnoty môžeme porovnávať naprieč všetkými experimentami, na ktorých sme testovali presnosť predikcie.

Dáta historických záznamov produkcie FVE sme museli spracovať a všetky dáta, ktoré k predikcii potrebujeme, sme uložili do relačnej databázy pre ľahší prístup k dátam.

5.3 Dáta

Pre našu prácu máme k dispozícií historické záznamy meteorologických predpovedí z modelu Aladin lokalizované na umiestnenie fotovoltaických elektrární, z ktorých máme záznamy o ich produkcii za rovnaké obdobie. K týmto dátam sme doplnili hodnoty slnečných koordinátov a dĺžku slnečného svitu, ktoré sme vypočítali v jazyku R.

Model Aladin je model numerickej predpovede počasia (NPP) a dáta z tohto modelu sú najlepšími možnými dátami, aké sme mohli pre svoju prácu získať, pretože pre predpoveď počasia s časovým horizontom vzdialeným 24 hodín dopredu sú predpovede z modelov NPP

najpresnejšie. Model Aladin je regionálny model NPP a na území Slovenska dosahuje najpresnejšie výsledky predpovede, ako sme aj napísali už v kapitole 2.3.1.

Každý záznam predpovede počasia z modelu Aladin je predpoveď na 24 hodín dopredu, takže každý záznam má rovnakú chybu predpovede, čo nám dovoľuje pristupovať ku každému záznamu rovnako a každý náš výsledok predikcie závisí od rovnakej chyby predpovede modelu Aladin. Znamená to aj, že pri hypotetickom použití v reálnom procese produkcie elektrickej energie by boli tieto dáta dostupné 24 hodín dopredu, čo zabezpečuje dostatočné množstvo času na ich spracovanie a použitie pri predikcií výroby elektrickej energie.

Záznamy predpovede počasia sú bodové záznamy s hodinovým krokom, takže predpovedajú pre konkrétny časový okamih s periódou jednej hodiny. Časové údaje sú uložené v UTC (Coordinated Universal Time), teda v koordinovanom svetovom čase, ktorý sa často používa v technických kruhoch, pretože je akýmsi základom, na ktorý sa odkazujú všetky časové pásma a nepoužíva letný čas.

Predpoveď počasia obsahuje údaje o teplote vzduchu v dvoch metroch nad povrchom, rýchlosti a smere vetra v desiatich metroch nad povrchom, celkovej oblačnosti vyjadrenej percentuálne (100 % = úplne zamračené), relatívnej vlhkosti v dvoch metroch nad povrchom vyjadrenej percentuálne, atmosférickom tlaku redukovanom na hladinu mora a globálnom horizontálnom ožiarení (GHO). Z týchto údajov považujeme za relevantné pre predpoveď hodnoty GHO, teploty vzduchu, rýchlosti vetra, celkovej oblačnosti a relatívnej vlhkosti.

Dáta o produkcii FVE majú niekoľko rozdielností oproti dátam z modelu Aladin. Tieto dáta sme preto museli upraviť tak, aby sme mohli vytvoriť dvojicu dát: predpoveď počasia – produkcia FVE. Prvým rozdielom je, že záznamy z FVE sú uložené s časovým záznamom v CET (Central European Time) a v CEST (Central European Summer Time), teda v centrálnom európskom čase s posunom na letný čas. V praxi to znamená, že sme časové údaje museli posunúť o jednu, respektíve dve hodiny, aby boli v UTC, rovnako ako záznamy z modelu Aladin.

Druhým rozdielom je, že záznamy z FVE sú intervalové záznamy s pätnásť minútovou periódou, teda záznam z 12:00 vyjadruje produkciu FVE za časové obdobie od 11:45 do 12:00. Preto sme tieto záznamy zoskupili do hodinových intervalov a hodnoty o produkcii FVE sme sčítali. Pre najlepšie vytvorenie dvojíc predpoveď počasia – produkcia FVE sme spolu zoskupili záznamy z 11:45, 12:00, 12:15 a 12:30 do záznamu s časom 12:00. Takto môžeme zdvojiť predpoveď počasia s produkciou FVE, kde je predpoveď počasia platná pre okamih v strede časového intervalu produkcie FVE.

V záznamoch o produkcii FVE je veľa hodnôt, z ktorých sú pre nás zaujímavé len dve hodnoty: výkon a vyprodukovaná energia (práca). Ako nám upresnil Ing. Peter Janiga, PhD. z Fakulty elektrotechniky a informatiky, hodnoty výkonu nie sú merané hodnoty, ale sú vypočítané z hodnôt vyprodukovanej energie a vyjadrujú priemerný výkon za daný časový interval. Preto pracujeme priamo s nameranými hodnotami vyprodukovanej energie v kWh, podobne ako sa pri predikcii spotreby energie pracuje s hodnotami spotrebovanej energie. Hodnoty vyprodukovanej energie boli v záznamoch uložené aditívne (každým záznamom sa hodnota zvýšila o množstvo vyprodukovanej energie za daný časový interval), preto sme pri spracovávaní záznamov vypočítali rozdiel v hodnote vyprodukovanej energie oproti predchádzajúcemu záznamu.

Slnečnými koordinátmi nazývame údaje opisujúce polohu slnka na oblohe. Sú nimi azimut a elevácia. Azimut je hodnota veľkosti uhlu medzi priamkou smerujúcou k severnému magnetickému pólu a priemetom priamky slnečných lúčov do roviny zemského povrchu v danom mieste. Tento uhol je orientovaný v smere hodinových ručičiek. Elevácia závisí od výšky Slnka na oblohe a opisuje uhol medzi rovinou zemského povrchu a priamkou vodorovnou s dopadajúcimi slnečnými lúčmi na túto rovinu. Elevácia teda vyjadruje uhol dopadajúcich slnečných lúčov, čo je dôležitý faktor pri produkcii FVE. Pri východe a západe Slnka má elevácia nulovú hodnotu. Dĺžka slnečného svitu je počet hodín počas jedného dňa, kedy je Slnko nad obzorom.

Elevácia je priamo úmerná s hodnotou GHO počas jasného dňa a dĺžka slnečného svitu je priamo úmerná so vzdialenosťou dátumu dňa od zimného slnovratu a teda so zmenou maximálnej hodnoty elevácie v rámci dňa aj priemernou teplotou vzduchu. Preto sú obe tieto hodnoty relevantné pri predikcii produkcie FVE. Azimut ale nemá pre nás podobnú výhodu, čo sa prejavilo aj na výsledkoch experimentu, kedy sme ho pri predikcii použili. Z tohto dôvodu sme ho ďalej pri predikcii nepoužívali.

5.4 Implementácia

Naše riešenie sme sa rozhodli implementovať v jazyku R, ktorý je vhodný na dátovú analýzu, štatistiku a vizualizáciu dát, a bol za týmto účelom aj vytvorený. Jazyk R je skriptovací jazyk a veľkou výhodou pri jeho používaní je, že podporuje „open source“ vývoj balíkov. To znamená, že každý používateľ môže napísať balík s vlastnou implementáciou a zverejniť ho pre ostatných používateľov v repozitáre balíkov CRAN (The Comprehensive R Archive Network). Vďaka tomu sme nemuseli mnohú funkcionalitu sami implementovať ale použili sme funkcionalitu verejných balíkov.

Pre pripojenie na databázu sme použili balík *RPostgreSQL,* pre výpočet slnečných koordinátov sme využili funkcionalitu balíka *insol*, náhodný les regresných stromov, ktorý používame na predikciu, je implementovaný v balíku *randomForest*, pre manipuláciu dát sme použili balík *plyr* a pre výpočet štatistiky presnosti sme použili balík *sirad*. Pre predkompiláciu funkcií sme použili balík *compiler*, pre paralelizáciu výpočtov balík *snow* a pre porovnanie rýchlosti vykonávania zdrojového kódu balík *microbenchmark*.

5.4.1 Import dát do databázy

Rozhodli sme sa vložiť dáta do relačnej databázy pre uľahčenie výberu dát pre trénovaciu množinu. Konkrétnou databázou je PostgreSQL, ktorá je najpoužívanejšou „open source" databázou.

Pri spracovaní dát sme narazili na niekoľko problémov v dátach z FVE, ktoré sme museli vyriešiť. Prvým problémom bolo formátovanie *csv* súborov, v ktorých sú dáta uložené. Súbory mali za hlavičkou dva nadbytočné riadky a na oddelenie desatinných miest je použitá čiarka, pričom pre kompatibilitu s databázou preferujeme použitie bodky. V týchto súboroch je aj veľký počet stĺpcov dát, ktoré nepotrebujeme. Pre tieto dôvody sme sa rozhodli dáta najskôr predspracovať v programovacom jazyku R a nie priamou funkcionalitou databázy ako v prípade spracovania dát predpovede počasia z modelu Aladin.

Druhým problémom boli rozdielnosti v dátach opísané vyššie, teda časové záznamy v CET a CEST na rozdiel od UTC, intervalové záznamy na rozdiel od bodových a 15-minútová perióda záznamov na rozdiel od hodinovej periódy.

Tretím problémom bola absencia niektorých záznamov, respektíve absencia dát v záznamoch. Niektoré záznamy chýbali úplne a vytvorili v dátach „dieru". Chýba v nich niekoľko dní, resp. hodín. Tento problém sme sa rozhodli vyriešiť odstránením dát za celé tieto dni, pretože neexistuje spôsob, ako by sme mohli chýbajúce dáta nahradiť a ich odstránením prídeme len o malé množstvo dát.

K dátam sme pridali údaj o počte dní od, respektíve do najbližšieho zimného slnovratu, ktorým sme substituovali uhol dopadajúceho slnečného žiarenia, ktorý sa mení v rámci roka. Neskôr sme ale túto hodnotu nahradili skutočnými hodnotami slnečných koordinátov a dĺžkou slnečného svitu, ktoré sme opísali vyššie. Tieto hodnoty sme vypočítali v R použitím funkcionality balíka *insol*.

Z databázy sme následne odstránili aj dáta, ktoré sme neodstránili v dôsledku problémov v dátach, ale za účelom zmenšenia celkového objemu dát o záznamy, ktoré nepotrebujeme. Boli

nimi záznamy, ktoré nemali v dátach svoju dvojicu predpoveď počasia – produkcia FVE a záznamy, kde bolo v tejto dvojici predpovedané GHO aj výsledný výkon FVE, resp. výsledná vyprodukovaná energia, nulové, teda dáta z času, kedy na fotovoltaické panely nedopadalo slnečné žiarenie a panely neprodukovali žiadnu elektrickú energiu. Po doplnení dát slnečných koordinátov sme toto odstránenie dát nahradili odstránením záznamov, kde bola hodnota elevácie menšia ako nula, čo sú záznamy z času, kedy slnko nebolo nad obzorom a teda žiadne slnečné žiarenie na fotovoltaické panely nedopadalo.

V databáze sme pripravili aj tabuľku na ukladanie dát o experimentoch, kde môžeme uložiť údaje o nastavení predikčného modelu, spôsobu výberu trénovacej množiny a výsledkoch predikcií vo forme štatistických metrík presnosti predikcie za daných podmienok.

5.4.2 Výber trénovacej množiny

Pre zvýšenie presnosti predikcie sme v našom riešení navrhli výber záznamov do trénovacej množiny podľa ich podobnosti so záznamom, pre ktorý predikujeme produkciu elektrickej energie. Produkcia FVE vo veľkej miere závisí od meteorologických podmienok a hodnoty vyprodukovanej energie sa výrazne líšia aj pri malých rozdieloch v hodnotách premenných opisujúcich meteorologické podmienky. Preto veríme, že trénovanie predikčných modelov na záznamoch, ktoré opisujú výrazne odlišné meteorologické podmienky má viac negatívny vplyv na presnosť predikcie ako pozitívny. Tak ako je evidentný rozdiel v produkcii FVE pri jasnom a zamračenom stave oblačnosti, pri nízkej (ráno alebo večer) a pri vysokej (poludnie) elevácii Slnka na oblohe, alebo pri porovnaní produkcie v lete a v zime, tak výrazná musí byť aj chyba predikcie pri trénovaní predikčného modelu na takto odlišných záznamoch.

Naopak, ak natrénujeme predikčný model na záznamoch podobných so záznamom, pre ktorý má model predikovať, zúžime tak rozsah intervalu možných výstupných hodnôt predikcie a pri viac homogénnej trénovacej množine môže trénovanie predikčného modelu byť cielenejšie a „detailnejšie“ alebo „jemnejšie“.

Predpokladáme, že takáto trénovacia množina umožní vytvoriť regresné stromy, ktorých priemerná výstupná hodnota bude bližšia skutočnej hodnote, ktorú sa snažíme predikovať. Predpokladáme aj že rozdiel medzi skutočnou a predikovanou hodnotou nebude nikdy tak veľký, ako môže byť veľký bez použitia výberu najpodobnejších záznamov do trénovacej množiny.

Klasickým postupom práce pri predikcii metódami strojového učenia je rozdelenie dát na trénovaciu a testovaciu množinu, ako sme vysvetlili v kapitole 2.2.4. Na záznamoch v trénovacej množine sa natrénuje predikčný model a jeho presnosť sa testuje na záznamoch v testovacej

množine. Keďže chceme predikovať pre každý záznam čo najpresnejšie, v našom riešení vyberieme novú trénovaciu množinu a natrénujeme nový predikčný model pre každý záznam, pre ktorý predikujeme. Rozdeľujeme tak dáta na množinu, ktorá obsahuje záznamy z dňa, kedy bol vytvorený záznam, pre ktorý predikujeme, a na množinu obsahujúcu všetky ostatné záznamy, ktoré nie sú v prvej množine. Z tejto druhej množiny záznamov potom vyberieme potrebný počet záznamov do trénovacej množiny.

Produkcia FVE nezávisí od produkcie v predchádzajúcich hodinách alebo dňoch, ale závisí od meteorologických podmienok, ktoré samotné majú túto závislosť. Hodnoty meteorologických premenných však pre nás predpovedal model Aladin a my môžeme obmedziť predikciu iba na závislosť od aktuálnych hodnôt meteorologických premenných. Keďže v našom riešení nezávisí predikcia od predchádzajúcich hodnôt a postupnosti záznamov, nepracujeme s časovým radom, ako to býva pri iných predikciách (predikcia spotreby elektrickej energie). Pretože nepracujeme s časovým radom, stávajú sa v našom riešení dátum a čas záznamov irelevantné až na jedinú podmienku, že nemôžeme trénovať predikčný model na záznamoch z toho istého dňa, pre ktorý predikujeme hodnotu vyprodukovanej energie, pretože tieto záznamy by ani v skutočných podmienkach neboli prístupné. V trénovacej množine sa ale môžu vyskytnúť historické záznamy z dní, ktorých dátum je neskorší než dátum záznamu, pre ktorý predikujeme. Uvedomujeme si, že tieto záznamy by v skutočných podmienkach tak isto neboli prístupné, no rozhodli sme sa ich používať z dôvodu nedostatku historických záznamov.

Máme historické záznamy pre tri fotovoltaické elektrárne a po odstránení chybných záznamov máme dostupných 336 dní pre prvú, 334 pre druhú a 449 pre tretiu elektráreň. Keby sme dodržiavali časovú postupnosť dátumov a netrénovali predikčný model aj na záznamoch z dní neskorších než deň, pre ktorý predikujeme, nemali by sme dostatok dát pre výber dostatočne podobnej trénovacej množiny aj pre meranie štatistiky presnosti predikcie s dostatočnou výpovednou hodnotou. Keďže ale predikcia produkcie FVE v našom riešení nezávisí od časovej postupnosti záznamov, môžeme s historickými záznamami pracovať len ako so záznamami opisujúcimi meteorologické podmienky, ktoré mohli nastať v ktorýkoľvek deň alebo rok aj v minulosti a kompenzujeme tým nedostatok historických záznamov. Berieme na vedomie, že toto riešenie nedostatku dát nie je teoreticky správne, pretože v skutočnosti neexistujú záznamy „z budúcnosti“, ale v praxi majú prevádzkovatelia fotovoltaických elektrární k dispozícii viac historických záznamov než 334 alebo 449 dní, ak samozrejme nie je samotná FVE funkčná kratší čas. Na naše výsledky sa môžeme pozerať ako na presnosť predikcie FVE po tom, čo má FVE k dispozícii aspoň 334, respektíve 449 dní historických záznamov.

5.4.3 Výber záznamov podľa podobnosti

Ako sme už viac krát napísali, do trénovacej množiny vyberáme najpodobnejšie záznamy so záznamom, pre ktorý chceme predikovať produkciu FVE. Vytvorili sme jednoduchý spôsob ohodnotenia podobnosti záznamov. Pre každý záznam v množine záznamov, z ktorých vyberáme záznamy do trénovacej množiny, vypočítame jeho hodnotu podobnosti, respektíve rozdielnosti, pretože číselná hodnota je väčšia pre viac rozdielne záznamy a najpodobnejšie záznamy majú hodnotu blízku nule. Podľa tejto hodnoty podobnosti všetky ohodnotené záznamy usporiadame vzostupne a vyberieme prvých *n* záznamov do trénovacej množiny.

Naše prvé experimenty s výberom najpodobnejších dát hodnotili záznamy podľa podobnosti v hodnote GHO, ktorá je najpodstatnejšou pri produkcii FVE. Náš výpočet hodnoty podobnosti vysvetlíme v nasledujúcich odsekoch.

Máme množinu potenciálnych záznamov (množina P o veľkosti m) na výber záznamov do trénovacej množiny. Máme záznam Z, pre ktorý predikujeme produkciu FVE za meteorologických podmienok, ktoré opisuje. Máme škálu GHO (S_{GHO}), ktorá je rozdielom maximálnej a minimálnej hodnoty GHO v množine P.

$$S_{GHI} = \max(P_{GHO}) - \min(P_{GHO})$$

Hodnotu rozdielnosti R_i každého záznamu P_i z množiny P, kde $i \in \langle 1, m \rangle,\ i\ \in \mathbb{Z}$, ohodnotíme nasledovne. Vypočítame absolútnu hodnotu rozdielu v hodnote GHO medzi Z a P_i. Túto hodnotu normalizujeme na škálu GHO (S_{GHO}).

$$R_i = \frac{\left|Z_{GHO} - P_{i,GHO}\right|}{S_{GHO} \times 100}$$

Teraz vieme usporiadať záznamy v množine P podľa podobnosti so záznamom Z. Hodnotiť podobnosť záznamov iba podľa hodnoty GHO je nepostačujúce. Ak chceme hodnotiť podobnosť záznamov podľa viacerých hodnôt a chceme aby každá hodnota mala na podobnosť rôzny vplyv, tak ako aj produkcia FVE závisí rozdielne od týchto hodnôt, musíme vyjadriť tento vplyv a zakomponovať ho do výpočtov. Pre tento účel sme vytvorili *faktory podobnosti*. Faktor podobnosti GHO vyjadruje silu vplyvu podobnosti záznamov v hodnote GHO na celkovú podobnosť záznamov.

Pre ďalší experiment sme sa rozhodli hodnotiť podobnosť záznamov podľa GHO, teploty vzduchu a rýchlosti vetra s faktormi podobnosti 90 pre GHO, 10 pre teplotu vzduchu a 1 pre rýchlosť vetra. Tieto hodnoty sme zobrali z kapitoly 1.2, kde píšeme, že presnosť predikcie produkcie FVE závisí od presnosti výpočtu GHO na úrovni 90 %, od teploty vzduchu na úrovni 10 % a od rýchlosti vetra na úrovni 1 %. Pri výpočte hodnoty celkovej podobnosti jednotlivé

podobnosti v hodnotách GHO, teploty vzduchu (T) a rýchlosti vetra (RV) vynásobíme ich faktorom podobnosti F a tieto hodnoty sčítame.

$$R_i = \left(\frac{|Z_{GHO} - P_{i,GHO}|}{S_{GHO} \times 100} \times F_{GHI}\right) + \left(\frac{|Z_T - P_{i,T}|}{S_T \times 100} \times F_T\right) + \left(\frac{|Z_{RV} - P_{i,RV}|}{S_{RV} \times 100} \times F_{RV}\right)$$

Rovnako ako škálu GHO vypočítame aj škálu teploty vzduchu a rýchlosti vetra.

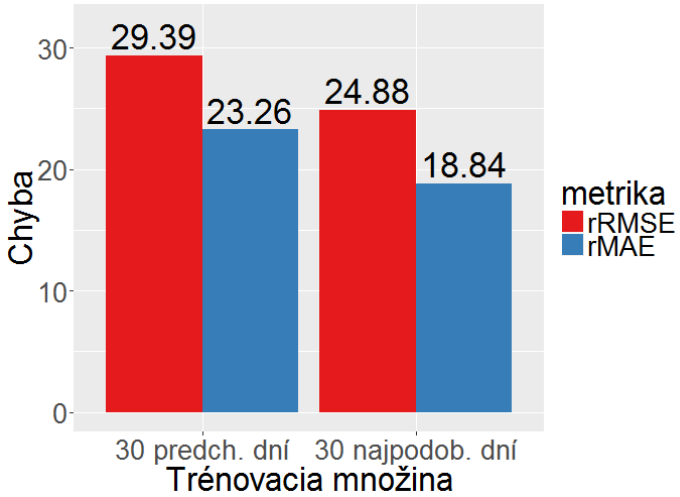
$$S_{GHO} = \max(P_{GHO}) - \min(P_{GHO})$$

$$S_T = \max(P_T) - \min(P_T)$$

$$S_{RV} = \max(P_{RV}) - \min(P_{RV})$$

Musíme ešte zmieniť, že hodnota faktorov podobnosti nemá výpovednú hodnotu sama o sebe, ale dôležitý je pomer faktorov podobnosti, a že rovnako vieme ohodnotiť podobnosť záznamov aj podľa všetkých prediktorov.

Už výberom najpodobnejších dát do trénovacej množiny podľa vyššie uvedeného príkladu s GHO, teplotou vzduchu a rýchlosťou vetra sa nám podarilo dosiahnuť nižšiu chybu predikcie oproti klasickému spôsoby výberu niekoľkých predchádzajúcich záznamov. Pri tomto experimente sme predikovali produkciu FVE za celý deň (hodinové záznamy boli sčítané). Pri klasickom spôsobe sme do trénovacej množiny vybrali 30 predchádzajúcich dní a pri výbere trénovacej množiny podľa podobnosti sme vybrali 30 najpodobnejších dní. Výsledky sú zobrazené na grafe (Obrázok 5), kde môžeme porovnať chybu predikcie pomocou metrík *rRMSE* a *rMAE*, ktoré sme vysvetlili v kapitole 3.



Obrázok 5: Porovnanie chyby predikcie poľa spôsobu výberu trénovacej množiny.

5.4.4 Zrýchlenie výpočtov v R

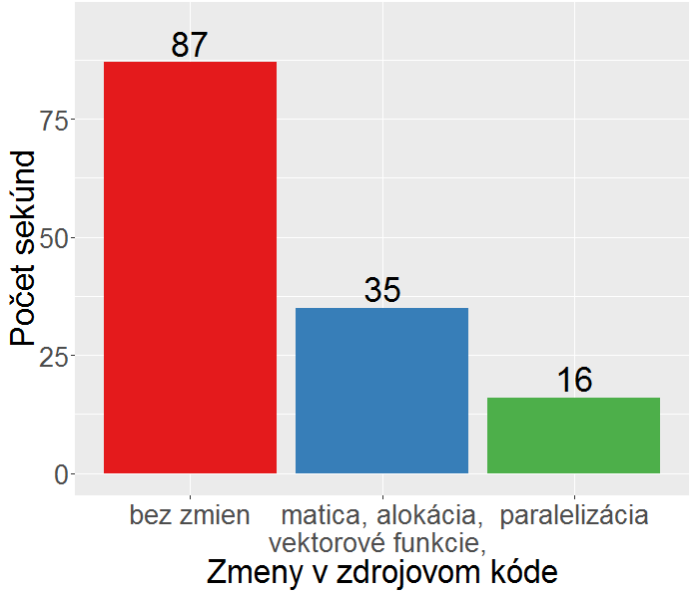
Jazyk R je vhodný a pohodlný pre implementáciu nášho riešenia, ale vykonávanie skriptov v jazyku R je v porovnaní s inými jazykmi pomalé. Skripty na testovanie presnosti predikcie použitím rôznych kombinácií nastavení predikčného modelu a rôznych kombinácií faktorov

podobnosti pri výbere záznamov do trénovacej množiny vyžadovali veľa procesorového času. Hľadali sme preto spôsob ako vykonávanie týchto skriptov zrýchliť.

Jazyk R kompiluje funkcie, ktoré sme definovali v externom súbore počas vykonávania zdrojového kódu. Preto sme použili balík *compiler* na predkompiláciu funkcií. Použitím balíka *microbenchmark* sme vedeli porovnať rýchlosť vykonávania skriptu pred a po predkompilácii a rovnako aj rôzne spôsoby implementácie rovnakej funkcionality.

Christenson a Morris [16] merali efektívnosť zrýchlenia vykonávania jazyka R na strojoch s operačným systémom Windows. Podľa ich meraní na 64-bitovom operačnom systéme Windows XP použitím matíc namiesto tabuliek[1] dosiahli v priemere zrýchlenie o hodnote 97,76 %, predbežnou alokáciou pamäte po blokoch (na rozdiel od postupného alokovania pamäti) 89,73 % a paralelizáciou na štyroch jadrách 49,12 %.

Rozhodli sme sa preto tiež aplikovať tieto zmeny na náš zdrojový kód aj s použitím vektorových funkcií namiesto cyklov. Podarilo sa nám zrýchliť beh skriptov viac ako päť násobne. Na nasledujúcom grafe (Obrázok 6) je vidieť priemerný čas behov skriptov pred aplikovaním zmien zdrojového kódu, po použití matice namiesto tabuľky, predbežnej alokácií pamäte a použití vektorových funkcií namiesto cyklov a po následnej paralelizácii. Pre paralelizáciu výpočtov sme použili, rovnako ako Christenson a Morris, balík *snow*, ktorý umožňuje paralelizáciu na operačnom systéme Windows. Rozdiely po aplikácií prvej sady zmien boli pochopiteľne väčšie pri skriptoch, ktoré pracovali s väčším obsahom dát. Graf znázorňuje výsledky zrýchlenia na kratšom testovacom skripte.



Obrázok 6: Porovnanie časov vykonávania skriptu.

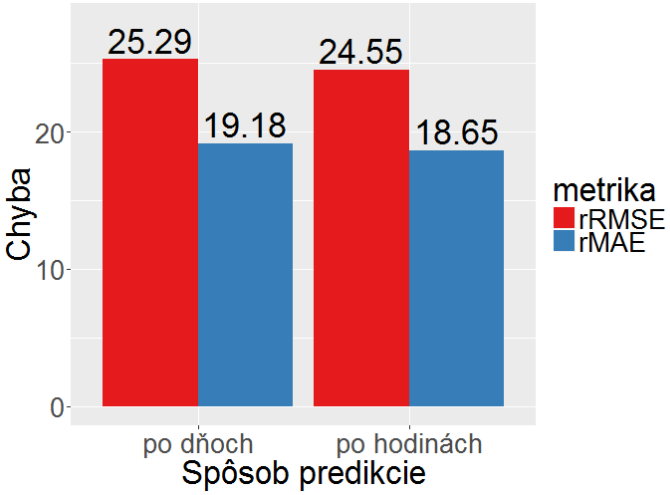
[1] V jazyku R implementované ako data.frame

Naše výsledky sa samozrejme líšia od výsledkov Christensona a Morrisa a sú individuálne pre každý stroj s inou špecifikáciou. My sme tieto výsledky dosiahli na operačnom systéme Windows 10 (64-bit) a procesore Intel Core i5-2450M CPU s frekvenciou 2.50 GHz s dvomi jadrami a štyrmi vláknami (angl. *thread*).

5.5 Experimenty a testy presnosti predikcie

Väčšina našich experimentov spočívala v testovaní rôznych kombinácií nastavení predikčného modelu. Testovaním a porovnávaním presnosti sme hľadali kombináciu nastavení, pri ktorých predikcia dosahovala najmenšiu chybu podľa metrík rRMSE a rMAE. Kombinovali sme rôzne kombinácie premenných vstupujúcich do predikcie s kombináciami nastavení náhodného lesa regresných stromov aj s kombináciami rôznych hodnôt faktorov podobnosti pri výbere záznamov do trénovacej množiny. Skripty na testovanie všetkých kombinácií vyžadovali veľa procesorového času, takže sme dosahovali pokrok veľmi pomaly a postupne sme obmedzovali kombinácie podľa výsledkov presnosti predikcie.

Porovnávali sme aj predikciu po celých dňoch a po jednotlivých hodinách. Rozdiely v presnosti predikcie boli len minimálne, ale predikcia po hodinách bola aj podľa našich očakávaní presnejšia, hoci sme predpokladali, že rozdiel bude väčší. Rozdiel v hodnotách metrík rRMSE a rMAE sa pohyboval len okolo 0.5 % až 1 % a napríklad pri hodnotách faktorov podobnosti 90 pre GHO, 10 pre teplotu vzduchu a 1 pre rýchlosť vetra bola chyba predikcie po dňoch menšia[2]. V nasledujúcom grafe (Obrázok 7) je vidieť porovnanie chyby predikcie po dňoch a po hodinách.

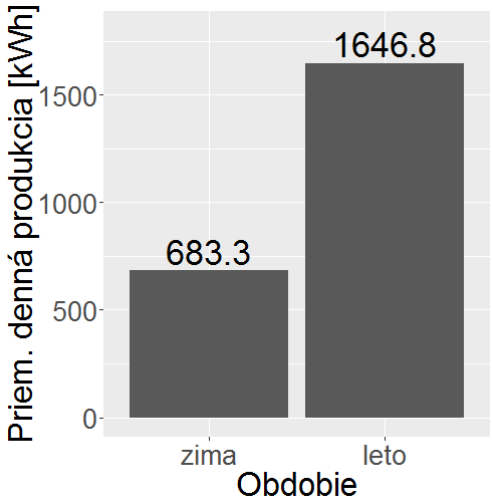


Obrázok 7: Porovnanie chyby predikcie po dňoch a po hodinách.

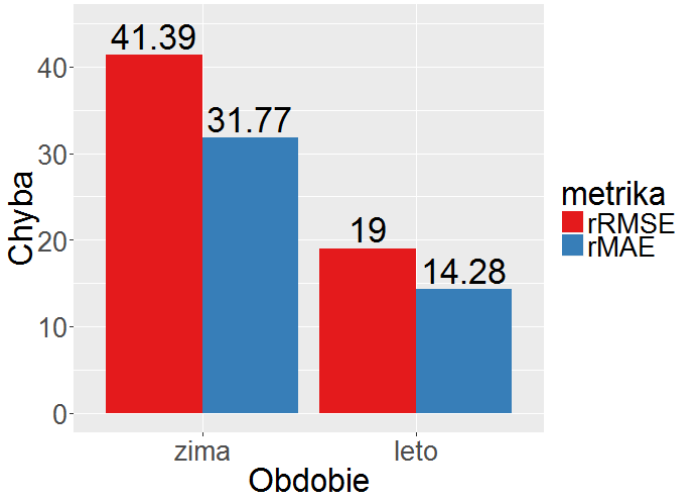
[2] Tieto malé rozdiely platia pri veľmi dobrých nastaveniach predikčného modelu a pri použití výberu najpodobnejších záznamov do trénovacej množiny. Pri klasickom prístupe výberu postupnosti predchádzajúcich záznamov je rozdiel veľký v neprospech predikcie po hodinách.

Pri spracovaní a importe dát do databázy sme si všimli, že zmeny v GHO a produkcii FVE sú na pohľad veľmi podobné v letnom období, čo odzrkadľuje fakt, že produkcia FVE závisí najmä od GHO. V zimnom období si ale tieto zmeny v hodnotách GHO a produkcie FVE v rovnakom čase na pohľad neodpovedali tak ako v lete. Vykonali sme preto experiment, kedy sme rozdelili množinu dát na dve časti. Letná časť dát obsahovala záznamy z obdobia od jarnej do jesennej rovnodennosti (21. marec až 23. september) a zimná časť obsahovala záznamy z obdobia od jesennej do jarnej rovnodennosti (24. september až 20. marec). Pre obe podmnožiny sme testovali presnosť predikcie zvlášť.

Na grafe vľavo (Obrázok 8) je porovnanie priemernej produkcie energie za jeden deň v oboch obdobiach[3]. Produkcia v letnom období je približne 2,4-krát väčšia oproti produkcii v zimnom období. Na grafe vpravo (Obrázok 9) je porovnanie chyby predikcie v oboch obdobiach. Chyba v letnom období je približne 2,2-krát menšia.



Obrázok 8: Porovnanie priemernej dennej produkcie v oboch obdobiach.



Obrázok 9: Porovnanie chyby predikcie v oboch obdobiach.

Výsledky tohto experimentu nás podnietili k analýze dát, jednotlivých záznamov a výsledkov predikcie pre tieto záznamy. Zistili sme, že výrazné chyby predikcie nastali pri záznamoch s nízkymi hodnotami GHO a nízkymi hodnotami vyprodukovanej elektrickej energie. Naplánovali sme preto ďalší experiment, kde sme obmedzili množstvo takýchto záznamov v dátovej množine.

SHMÚ v elektronickej komunikácii tvrdí, že za slnečný svit sa dá označiť stav, kedy je hodnota globálneho horizontálneho ožiarenia (GHO) väčšia ako 120 W/m^2. Preto sme vytvorili podmnožinu záznamov, ktorá obsahuje všetky záznamy s hodnotou GHO väčšou ako 120.

[3] Tieto hodnoty sa vzťahujú len na našu množinu dát.

V nasledujúcich výsledkoch experimentov je vidieť nižšia chyba predikcie pre záznamy z tejto podmnožiny oproti predikcii pre všetky záznamy.

Vytvorili sme ešte druhú podmnožinu záznamov, ktorá je podmnožinou vyššie opísanej podmnožiny a obsahuje záznamy, ktoré nie sú prvým ani posledným záznamom daného dňa. Týmto sme chceli vymedziť podmnožinu záznamov, v ktorej nie sú záznamy z času východu a západu Slnka, kedy je uhol dopadajúcich slnečných lúčov malý, produkcia FVE je zanedbateľná a kedy je presnosť predikcie produkcie FVE nízka, ako sme zistili pre analýze. Toto vymedzenie podmnožiny dát však nie je úplne korektné, pretože sme vylúčili aj niektoré záznamy, ktoré nie sú z času východu a západu Slnka, najmä v zimnom období, kedy sú hodnoty GHO nízke. Celkovo sme ale touto podmnožinou vymedzili podmnožinu záznamov, pre ktorú experimenty dosahujú ešte nižšiu chybu predikcie.

Pre vytvorenie korektnej podmnožiny sme vymedzili tretiu podmnožinu, kde sme najskôr vybrali záznamy, ktoré nie sú prvým alebo posledným záznamom z daného dňa a následne sme z týchto záznamov vybrali tie, ktorých predpovedaná hodnota GHO je väčšia ako 120. Pre účely porovnania sme vymedzili aj štvrtú podmnožinu, ktorá obsahuje záznamy, ktoré nie sú prvým alebo posledným záznamom daného dňa, bez obmedzenia hodnoty GHO.

Pre jednoduchosť označenia týchto dátových množín v grafoch tieto množiny pomenujeme písmenami abecedy:

A - Množina všetkých záznamov, s ktorou sme pracovali v predchádzajúcich experimentoch.
B - Podmnožina záznamov množiny A, ktorých hodnota GHO je väčšia ako 120.
C - Podmnožina záznamov množiny B, pre ktorú platí, že záznam v množine C nie je prvým alebo posledným záznamom daného dňa v množine B.
D - Podmnožina záznamov množiny A, pre ktorú platí, že záznam v množine D nie je prvým alebo posledným záznamom daného dňa v množine A.
E - Podmnožina záznamov množiny D, ktorých hodnota GHO je väčšia ako 120.

V nasledujúcej tabuľke (Tabuľka 1) je porovnanie obsahu dátových množín.

Tabuľka 1: Porovnanie obsahu dátových množín.

množina	počet záznamov	p. podiel záznamov	p. podiel produkcie
A	14 117	100,00%	100,00%
B	9 421	66,74%	93,34%
C	7 265	51,46%	83,65%
D	11 823	83,75%	98,85%
E	9 368	66,36%	93,23%

V tabuľke je vidieť, že aj po obmedzení značného množstva záznamov je podiel celkovej produkcie v týchto množinách znížený len minimálne a preto zvýšenie presnosti predikcie na záznamoch v týchto dátových množinách je významné. Najpresnejšie výsledky predikcie sa nám podarilo dosiahnuť kombináciou nastavení opísanou v nasledujúcich odsekoch.

Pri kombináciách premenných vstupujúcich do predikcie sme dosiahli najpresnejšie výsledky s použitím všetkých premenných, ktoré sme považovali za relevantné alebo vhodné pre predikciu. Sú nimi GHO, teplota vzduchu, rýchlosť vetra, celková oblačnosť, relatívna vlhkosť, dĺžka slnečného svitu v daný deň a elevácia Slnka na oblohe. Tento výsledok sme aj dopredu predpokladali, pretože každý z týchto prediktorov má vplyv na produkciu FVE.

Najlepšou kombináciou nastavení predikčného modelu s použitím náhodného lesa regresných stromov (NLRS) je kombinácia týchto hodnôt:

- Veľkosť trénovacej množiny = 30
- Počet stromov v lese = 500
- Počet náhodne vybraných prediktorov pre vytvorenie uzla stromu = 2
- Minimálna veľkosť koncových uzlov stromu = 3

Počet stromov v lese a počet náhodne vybraných prediktorov pre vytvorenie uzla stromu odpovedajú predvoleným (angl. *default*) hodnotám týchto parametrov pre implementáciu náhodného lesa v balíku *randomForest*. Pre regresiu[4] je počet náhodne vybraných prediktorov pre vytvorenie uzla stromu predvolene $p/3$, kde p je počet prediktorov ale minimálna veľkosť koncových uzlov stromu je predvolene 5 [17].

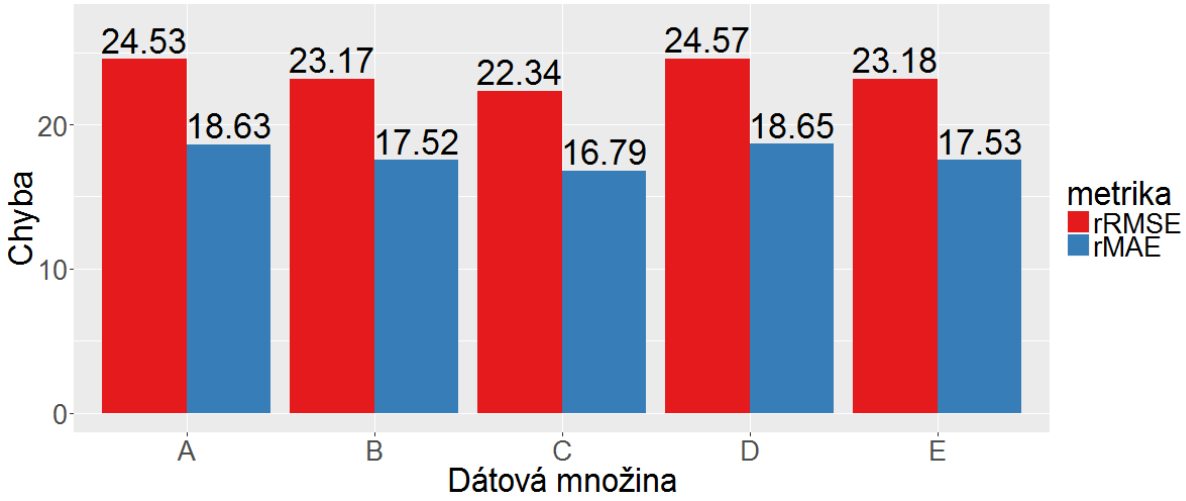
Najlepšou kombináciou nastavení faktorov podobnosti je kombinácia týchto hodnôt faktorov podobnosti:

- GHO = 190
- Celková oblačnosť = 100
- Teplota vzduchu = 30
- Rýchlosť vetra = 5,5
- Relatívna vlhkosť = 1,5
- Dĺžka slnečného svitu = 53
- Elevácia Slnka na oblohe = 1270

[4] Pre klasifikáciu sú predvolené hodnoty $\sqrt{p}$ pre počet náhodne vybraných prediktorov pre vytvorenie uzla stromu a 1 pre minimálnu veľkosť koncových uzlov stromu.

Z týchto hodnôt môžeme usudzovať, že podobne závisí od týchto hodnôt aj celková produkcia FVE. Faktor podobnosti elevácie má veľmi vysokú hodnotu oproti ostatným hodnotám, takže už malý rozdiel v tejto hodnote môže zapríčiniť, že záznam inak veľmi podobný v ostatných hodnotách, nebude vybraný do trénovacej množiny. Preto si myslíme, že by bolo vhodné použiť inú metódu na prvotný výber záznamov podľa elevácie a druhotne vyberať záznamy podľa výpočtu podobnosti použitím faktorov podobnosti.

Na nasledujúcom grafe (Obrázok 10) sú vyjadrené chyby predikcie, ktoré sme dosiahli s použitím práve opísanej kombinácie nastavení. Na grafe sú zobrazené aj chyby predikcie pre jednotlivé dátové množiny definované v predchádzajúcej kapitole ako množiny A, B, C, D a E.



Obrázok 10: Porovnanie chyby predikcie pre dátové množiny.

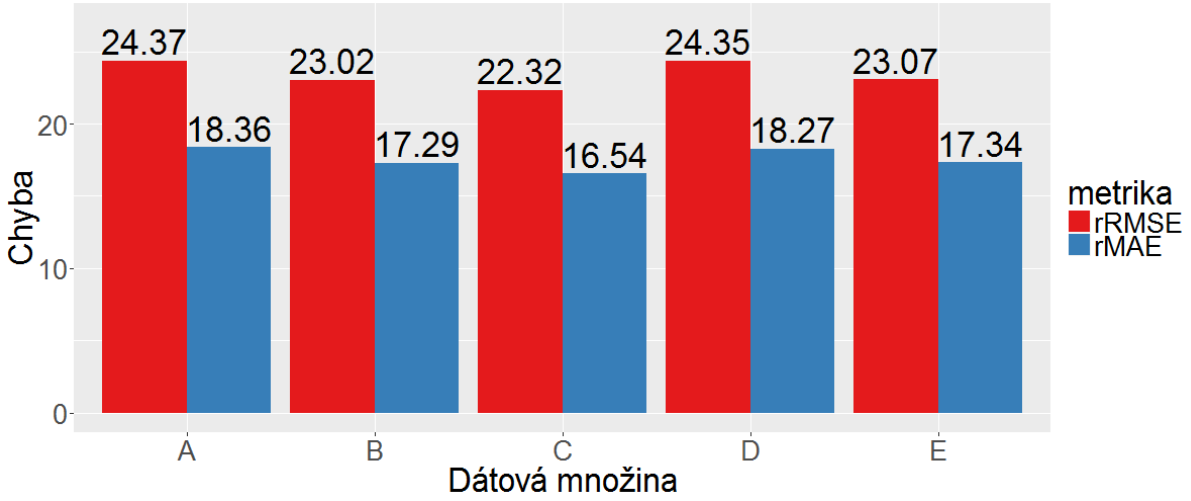
Ako ďalší experiment sme skúsili prvotne ohraničiť množinu potenciálnych záznamov na výber do trénovacej množiny podľa hodnoty elevácie. Najskôr z množiny všetkých dostupných záznamov do trénovacej množiny vyberieme len tie, ktorých hodnota elevácie nie je rozdielna o viac ako q od hodnoty elevácie záznamu, pre ktorý predikujeme produkciu FVE. Následne z vyhovujúcich záznamov vyberáme záznamy podľa podobnosti, ale s rozdielnymi faktormi podobnosti ako v predchádzajúcom experimente. Pre tento spôsob sme hľadali nové nastavenie hodnôt faktorov podobnosti a hodnotu q, pri ktorých predikcia dosahuje najmenšiu chybu predikcie.

Najlepšie výsledky sme dosiahli s týmito hodnotami:

- q = 6
- GHO = 210
- Celková oblačnosť = 90
- Teplota vzduchu = 7,5
- Rýchlosť vetra = 5,5

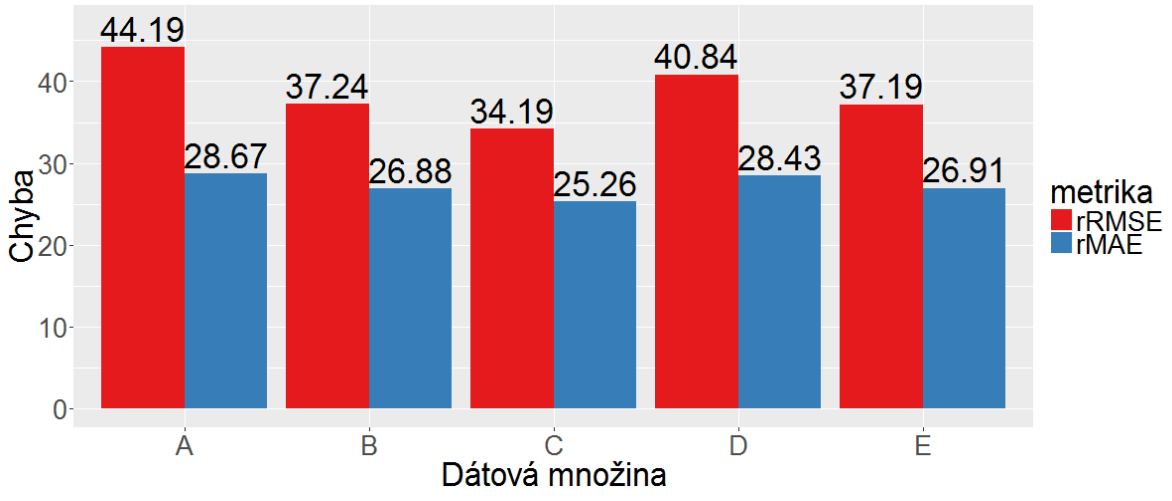
- Relatívna vlhkosť = 3,5
- Dĺžka slnečného svitu = 59
- Elevácia Slnka na oblohe = 41

Ak usudzujeme, že podobne závisí od týchto hodnôt aj celková produkcia FVE, na hodnotách faktorov podobnosti vidíme, že sa nám podarilo znížiť faktor podobnosti elevácie, a že najväčšiu hodnotu má faktor podobnosti GHO, čo odpovedá znalostiam o produkcii FVE nadobudnutých pri analýze. Chyba predikcie použitím takéhoto nastavenia výberu záznamov do predikčnej množiny je zobrazená na nasledujúcom grafe (Obrázok 11). Na hodnotách oboch metrík je vidieť, že chyba predikcie klesla, ale len o desatiny percenta.



Obrázok 11: Porovnanie chyby predikcie so zúženým výberom podľa elevácie.

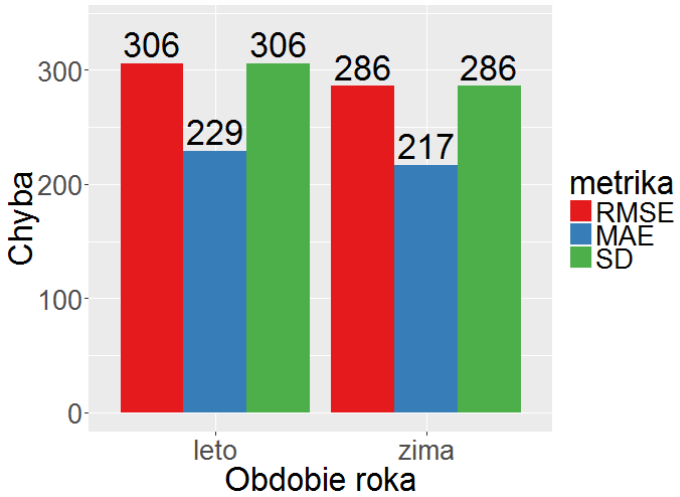
Doteraz zobrazené chyby predikcie na grafoch boli počítané ako denná chyba predikcie, kedy do výpočtu vstupovali hodnoty sčítané za celý deň. Na nasledujúcom grafe (Obrázok 12) sú zobrazené hodinové chyby predikcie počítané po jednotlivých hodinách. Obe chyby predikcie (denná aj hodinová) sú ale výsledkom predikcie po hodinových záznamoch.



Obrázok 12: Porovnanie hodinovej chyby predikcie pre dátové množiny.

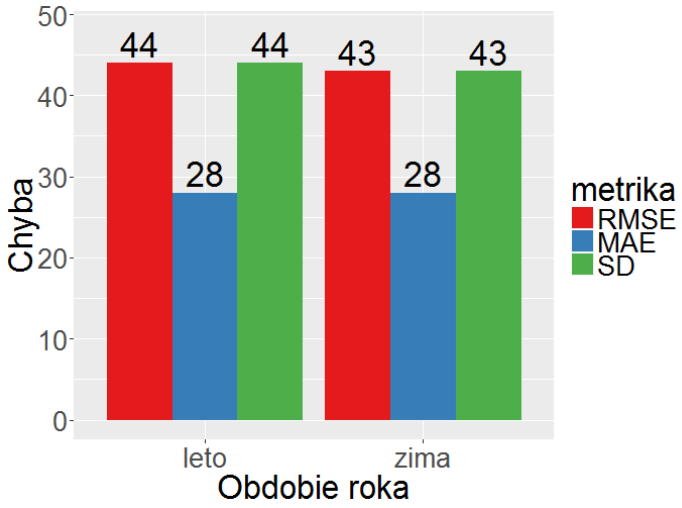
S rovnakým nastavením predikčného modelu a výberu dát do trénovacej množiny ako v predchádzajúcom experiemnte sme znovu vykonali experiment s rozdelením dátovej množiny na letnú a zimnú časť roka. Pre letnú časť roka sme dosiahli dennú chybu predikcie pod 14 % rMAE a pod 18 % rRMSE na všetkých dátových množinách a rozdiely v chybe predikcie na týchto množinách boli minimálne. Pre zimnú časť sa hodnoty na množinách líšia výraznejšie. Najväčšie hodnoty chyby predikcie sú na množine A, kde predikcia mala dennú chybu 31,6 % rMAE a 41,7 % rRMSE. Najnižšiu chybu predikcia mala na množine C, kde denná chyba dosiahla hodnoty 27 % rMAE a 36,8 % rRMSE.

Rozdiel medzi letným a zimným obdobím je výrazný, pretože metriky rMAE a rRMSE sú normalizované a v letnom období je produkcia 2,4-krát vyššia. Keď sme porovnali hodnoty štandardnej odchýlky a metrík MAE a RMSE, ktoré nie sú normalizované, rozdiel nebol tak výrazný. Hodnoty dennej štandardnej odchýlky (SD, angl. *standard deviation*), MAE a RMSE sú pre predikciu v zimnom období dokonca nižšie na všetkých dátových množinách. Na grafe (Obrázok 13) sú zobrazené denné hodnoty metrík RMSE, MAE a SD na dátovej množine A pre letné a zimné obdobie. Uvedené hodnoty odpovedajú chybe predikcie v jednotke kWh.



Obrázok 13: Porovnanie denných nenormalizovaných metrík presnosti pre predikciu v letnom a zimnom období.

Hodinové hodnoty týchto metrík hovoria v neprospech predikcie pre zimné obdobie na všetkých dátových množinách okrem množiny A. Na tejto množine predikcia dosahuje identické výsledky podľa týchto metrík. Konkrétne hodnoty je vidieť na grafe (Obrázok 14). To znamená, že predikcia sa mýli v oboch obdobiach rovnako na množine všetkých záznamov, ale relatívne k množstvu vyprodukovanej energie je táto chyba v zimnom období väčšia. Preto sú hodnoty rRMSE a rMAE väčšie v zimnom období.



Obrázok 14: Porovnanie hodinových hodnôt nenormalizovaných metrík presnosti pre predikciu v letnom a zimnom období.

5.6 Zhodnotenie výsledkov experimentov

Výsledky experimentov dokazujú, že nami navrhnutý a implementovaný spôsob výberu trénovacej množiny podľa podobnosti významne prispel k zvýšeniu presnosti predikcie pri použití náhodného lesa regresných stromov pre predikciu produkcie FVE. Na tej istej dátovej množine sme oproti klasickému prístupu znížili chybu predikcie približne o 5 %.

Z výsledkov experimentov vyplýva, že na záznamoch s väčšou produkciou elektrickej energie dosahujeme vyššiu relatívnu presnosť predikcie. Rozdiel je vidieť na predikciách na záznamoch z dátových množín, z ktorých sme vylúčili záznamy s minimálnou a zanedbateľnou produkciou. Môžeme z toho usúdiť, že na dôležitejších záznamoch sa nám podarilo dosiahnuť menšiu chybu predikcie.

Z výsledkov taktiež vyplýva, že pri predikcii pre letné obdobie sme dosiahli vyššiu relatívnu presnosť predikcie oproti predikcii pre zimné obdobie. Metriky chyby predikcie, ktoré nie sú normalizované (relativizované) vzhľadom na množstvo vyprodukovanej elektrickej energie, hovoria, že chyba predikcie v oboch obdobiach je takmer rovnaká.

Výsledky experimentov taktiež naznačujú, že spôsob výberu dát do trénovacej množiny je možné zlepšiť. My sme dokázali prvotným výberom záznamov podľa hodnoty elevácie znížiť chybu predikcie len o desatiny percenta, ale komplexnejší spôsob výberu dát mohol celkovo značnejšie znížiť chybu predikcie. Navrhujeme zvážiť a vyskúšať pokročilejšie metódy ako zhlukovanie alebo klasifikáciu potenciálnych záznamov na výber do trénovacej množiny.

Naše najpresnejšie predikcie dosahujú dennú chybu predikcie 18,3 % a hodinovú chybu predikcie 28,7 % na množine všetkých záznamov a nižšiu na množinách bez záznamov so zanedbateľnou produkciou. Výrazne nižšiu chybu predikcie dosahujeme pri predikcii pre letné obdobie, kedy elektráreň produkuje viacej elektrickej energie.

5.7 Možnosti rozšírenia práce

Naša práca by mohla pokračovať tromi spôsobmi s cieľom zvýšiť presnosť predikcie:

1. Chyba predikcie by mohla byť nižšia použitím inej predikčnej metódy, ako je náhodný les regresných stromov (napríklad metóda podporných vektorov). Pre vyskúšanie iných predikčných metód a nájdenie najlepších nastavení predikčných modelov by sme ale potrebovali veľa procesorového času a predpokladáme len malý potenciál pre zlepšenie presnosti výslednej predikcie oproti NLRS. Pri analýze metód strojového učenia používaných pri predikcii sme sa dozvedeli, že NLRS dosahuje veľmi dobré výsledky v porovnaní s inými metódami ako umelá neurónová sieť a metóda podporných vektorov. Pri analýze používaných predikčných metód sme sa dokonca dočítali, že pri porovnaní rôznych metód strojového učenia pre predikciu, dosiahla metóda NLRS prvé aj druhé najlepšie výsledky s dvomi odlišnými implementáciami. Žiaľ, na zdroj tejto informácie sa nevieme odkázať.
2. Výber najpodobnejších záznamov do trénovacej množiny zvyšuje presnosť predikcie, no nami implementovaný výber týchto záznamov je pomerne jednoduchý, kedy číselne ohodnocujeme záznamy podľa ich podobnosti, respektíve rozdielnosti so záznamom, pre ktorý predikujeme výslednú hodnotu vyprodukovanej elektrickej energie. Spôsob výberu najpodobnejších záznamov do trénovacej množiny by ale mohol byť sofistikovanejší napríklad použitím zhľukovania (angl. *clustering*), klasifikácie alebo iných metód používaných na dátovú analýzu. Lepším výberom dát do trénovacej množiny by chyba predikcie mohla klesnúť, ale predpokladáme len mierne zníženie celkovej chyby predikcie.
3. Najväčší potenciál vidíme v aplikovaní štatistického spracovania hodnôt predikcie (postprocessing). Úprava výsledkov predikcie podľa štatistiky presnosti by mohla výrazne znížiť celkovú chybu predikcie, no na implementáciu postprocessingu by sme potrebovali veľa človekohodín a samotná implementácia by bola veľmi náročná. V kombinácií s predchádzajúcim návrhom na možné pokračovanie práce by bolo možné dosiahnuť veľmi hodnotnú presnosť predikcie.

Naša práca by mohla pokračovať aj iným smerom. Nami vytvorený predikčný model vždy trénujeme na dátach z jednej elektrárne a predikcia je platná len pre túto jednu elektráreň. Do predikcie nevstupujú údaje, ktoré charakterizujú samotnú elektráreň ako napríklad inštalovaný výkon elektrárne. Predikcia je úplne nezávislá od špecifikácie elektrárne a takáto predikcia je nepoužiteľná pre iné elektrárne. Zaujal nás preto nápad, že by do predikcie vstupovali aj parametre špecifikujúce schopnosť produkcie elektrárne ako napríklad inštalovaný výkon, efektivita produkcie elektrickej energie a maximálna produkcia za hodinu alebo deň. Takto by bolo možné predikovať pre rôzne elektrárne a trénovať predikčný model aj na dátach z iných elektrární. Pre možnosť takejto predikcie by sme ale potrebovali dáta z viacerých fotovoltaických elektrární s rôznou špecifikáciou, ktoré by nám poskytli dostatočné spektrum záznamov na vytvorenie takéhoto predikčného systému.

Nevieme odhadnúť presnosť takéhoto predikčného systému. Vstupné parametre charakterizujúce FVE a možnosť trénovať predikčný model na dátach z iných elektrární by mohli mať pozitívny aj negatívny vplyv na zmenu hodnoty chyby predikcie. Za predpokladu, že by bola chyba predikcie na prijateľnej úrovni by však univerzálnosť takéhoto predikčného systému bola veľmi prínosnou. Predikovať by sme tak mohli aj pre elektrárne s malou alebo žiadnou bázou vlastných historických záznamov. Zapojením viacerých FVE do takéhoto systému by sa rozšírovala báza dostupných historických záznamov, čo by prispievalo ku kvalite univerzálnosti predikcie. V prípade úspešnej implementácie takéhoto predikčného systému s malou chybou predikcie, by tento predikčný systém bol globálne veľmi užitočný a mal by veľký komerčný potenciál.

6 Zhodnotenie

V našej práci sa nám podarilo úspešne navrhnúť, implementovať a otestovať predikciu produkcie fotovoltaických elektrární použitím predikčnej metódy náhodného lesa regresných stromov a výberom záznamov do trénovacej množiny podľa podobnosti so záznamom, pre ktorý predikčný model predikuje množstvo vyrobenej elektrickej energie. Preukázali sme, že tento predikčný model je vhodný na riešenie tohto problému, a že správny výber dát do trénovacej množiny je dôležitá súčasť predikcie. Rovnako sme aj preukázali, že nami implementovaný výber záznamov do trénovacej množiny významne prispel k zvýšeniu presnosti predikcie. Navrhli sme aj niekoľko spôsobov zvýšenia celkovej presnosti predikcie.

Dosiahli sme výsledky s presnosťou predikcie na úrovni 82 %. Predikcia s takouto presnosťou je použiteľná v praxi a mohla by prispieť k efektivite využívania fotovoltaických elektrární a riadenia ich prevádzky. Kvalitná a dostupná predikcia nasadená v praxi by mohla zvýšiť dopyt po tejto technológii a znížiť tak závislosť na fosílnych palivách v oblasti energetiky.

Použitá literatúra

[1] M. Morvová, Princípy metód a využitie obnoviteľných zdrojov energie, Bratislava: Knižničné a edičné centrum FMFI UK, 2008.

[2] S. Letendre, M. Makhyoun and M. Taylor, *Predicting solar power production: irradiance forecasting models, applications and future prospects,* SEPA - solar electric power association, 2014.

[3] Fakulta elektrotechniky a informatiky, Slovenská technická univerzita v Bratislave, „Analýza možnosti predpovedania výroby elektrickej energie z fotovoltických elektrárni,“ Bratislava.

[4] M. Diagne, M. David, P. Lauret, J. Boland and N. Schmutz, "Review of solar irradiance forecasting methods and a proposition for small-scale insular grids," *Renewable and Sustainable Energy Reviews,* no. November 2013, pp. 65-76, 2013.

[5] W. W. S. Wei, Time series analysis: univariate and uultivariate methods, Pearson, 2006.

[6] P. Sinčák a G. Andrejková, Neurónové siete: inžiniersky prístup (1. diel), Košice: ELFA-press, 1996.

[7] Ľ. Beňušková, „Umelé neurónové siete,“ rev. *Umelá inteligencia*, Bratislava, Vydavateľstvo STU, 2002.

[8] J. Maroco, D. Silva, A. Rodrigues, M. Guerreiro, I. Santana, A. d. Mondença and A. de Mendonça, "Data mining methods in the prediction of Dementia: A real-data comparison of the accuracy, sensitivity and specificity of linear discriminant analysis, logistic regression, neural networks, support vector machines, classification trees and random forests," *BMC Research Notes,* no. 4:299, 2011.

[9] U. Thissen, R. van Brakel, A. de Weijer, W. Melssen and L. Buydens, "Using support vector machines for time series prediction," Chemometrics and Intelligent Laboratory Systems, 2003.

[10] L. Breiman, "Random Forests," *Machine Learning,* no. 1, pp. 5-32, 2001.

[11] A. Liaw and M. Wiener, "Classification and Regression by randomForest," *R News,* no. 3, pp. 18-22, 2002.

[12] M. P. Almeida, O. Perpiñán and L. Narvarte, "PV Power Forecast Using a Nonparametric PV Model," *Solar Energy,* no. 115, pp. 354-368, 2015.

[13] Slovenský hydrometeorologický ústav, „Model ECMWF - popis,“ [Online]. Available: http://www.shmu.sk/sk/?page=1164.

[14] Slovenský hydrometeorologický ústav, „Model Aladin - popis,“ [Online]. Available: http://www.shmu.sk/sk/?page=1016.

[15] S. Pelland, J. Remund, J. Kleissl, T. Oozeki and K. De Brabandere, "Photovoltaic and Solar Forecasting: State of the Art," International energy agency, 2013.

[16] D. P. Christenson a J. A. Morris, „A Note on Speeding Up R for Windows,“ *The Political Methodologist,* zv. 17, %1. vyd.1, pp. 4-8, 2009.

[17] A. Liaw and M. Wiener, "randomForest: Breiman and Cutler's Random Forests for Classification and Regression," 06 10 2015. [Online]. Available: https://cran.r-project.org/web/packages/randomForest/randomForest.pdf.

Príloha A: Obsah elektronického média

K práci je priložený kompaktný disk, na ktorom sa nachádzajú nasledovné súbory:

- Súbor *bakalarska_praca.pdf*, ktorý je elektronickou formou tohto dokumentu.
- Adresár *SQL*, ktorý obsahuje textové súbory s SQL príkazmi SQL/PL funkciami použitými na správu databázy a import dát do databázy.
- Adresár *R_source*, ktorý obsahuje skripty v jazyku R používané na predikciu, experimenty a testovanie predikcie.
- Adresár *ostatne*, ktorý obsahuje textové súbory s SQL príkazmi a skripty v jazyku R, ktoré nie sú súčasťou finálnej verzie, ale slúžili ako pomocné SQL príkazy alebo pomocné skripty pri práci.
- Súbor *popis.txt*, ktorý obsahuje popis adresárov a súborov, ktoré sa nachádzajú na elektronickom médiu.

Poznámka: Dáta, ktoré sme pre predikciu použili nie sú obsahom elektronického média, pretože dáta so záznamami o produkcii fotovoltaických elektrární nemáme dovolené ďalej sprostredkovať alebo zverejňovať. Tým pádom je zbytočné prikladať dáta so záznamami predpovede počasia, pretože bez dát o produkcii FVE nie je možné použiť zdrojové kódy našej implementácie.

Príloha B: Technická dokumentácia

Naše riešenie predikcie produkcie fotovoltaickej elektrárne sme implementovali v jazyku R vo verzii 3.2.3 vydanej 10.12.2015. Požili sme niekoľko balíkov z verejného repozitára CRAN, ktoré nie sú súčasťou verzie 3.2.3. Použité balíky: randomForest, RPostgreSQL, insol, sirad, plyr, snow, compiler, microbenchmark.

Neimplementovali sme stabilnú aplikáciu alebo systém, ale skripty vykonávajúce predikciu a overenie presnosti predikcie. Dáta, s ktorými pracujeme pri predikcii, sme uložili do relačnej databázy PostgreSQL vo verzii 9.5.

Nasledujúci zdrojový kód v jazyku R je poslednou verziou skriptu, ktorým vykonávame predikciu a overujeme jej presnosť. V postupnosť krokov v zdrojovom kóde:

1. načítanie používaných balíkov a funkcii z externého súboru,
2. pripravenie paralelizácie na štyroch jadrách a exportovanie funkcie *format.time* a balíkov *randomForest* a *plyr* na všetky jadrá,
3. inicializácia premenných používaných na nastavenie parametrov predikčného modelu a výberu dát do trénovacej množiny,
4. vytvorenie karteziánskeho súčinu všetkých hodnôt nastavení,
5. výpis času začatia vykonávania skriptu,
6. vytvorenie spojenia s databázou,
7. inicializácia a výpočet hodnôt premenných používaných na výpis stavu predikcie,
8. cyklus pre každý riadok z tabuľky karteziánskeho súčinu nastavení,
9. cyklus pre každú z elektrární,
10. vytiahnutie záznamov z databázy, výpočet škály pre hodnotenie podobnosti záznamov,
11. exportovanie premenných na všetky jadrá procesora, paralelný výpočet predikcie pre každý záznam zvlášť, vráti vektor predikovaných hodnôt pre jednu elektráreň,
12. vymedzenie potenciálnych záznamov do trénovacej množiny,
13. ohodnotenie potenciálnych záznamov podľa podobnosti,
14. výber najpodobnejších záznamov do trénovacej množiny,
15. vytvorenie náhodného lesa regresných stromov,
16. predikcia produkcie pre vybraný záznam,
17. pridanie skutočných a predikovaných hodnôt do vektorov spoločných pre všetky elektrárne,
18. výpis stavu predikcie,
19. výpočet hodinovej štatistiky presnosti a zápis výsledkov do databázy,

20. úprava výsledkov pre výpočet dennej štatistiky presnosti,
21. výpočet dennej štatistiky presnosti a zápis výsledkov do databázy,
22. konečný výpis,
23. ukončenie možnosti paralelizácie a zatvorenie spojenia s databázou.

Zdrojový kód 1: Skript pre predikciu a výpočet štatistiky presnosti.

```
# 1. načítanie používaných balíkov a funkcii z externého súboru
library(RPostgreSQL)
library(plyr)
library(randomForest)
library(snow)
source('~/GitHub/baka/R source/new_way/functions.R')
# 2. pripravenie paralelizácie na štyroch jadrách a exportovanie funkcie
# format.time a balí-kov randomForest a plyr na všetky jadrá
cl <- makeCluster(4, type='SOCK')
clusterEvalQ(cl, format.time <- function(x) UseMethod("format.time"))
clusterEvalQ(cl, { library(plyr); library(randomForest) })
# 3. inicializácia premenných používaných na nastavenie parametrov
# predikčného modelu a výberu dát do trénovacej množiny
write_results <- TRUE
dataset <- c("v_data_all", "v_data_120", "v_data", "v_data_vz",
"v_data_vz_120")
fve <- c(1, 2, 3)
tm_velkost <- c(30)
f_ntree <- c(500)
f_mtry <- c(2)
f_nodesize <- c(3)
pod_gho <- c(210)
pod_obl <- c(90)
pod_tep <- c(7.5)
pod_vie <- c(5.5)
pod_vlh <- c(3.5)
pod_dlz <- c(59)
pod_ele <- c(41)
ele_res <- c(6)
select <- " SELECT datum, cas, praca, gho, oblacnost,
            teplota, vietor, vlhkost, dlzkadna, elev
            FROM %s WHERE fve = %d ORDER BY cas"
# 4. vytvorenie karteziánskeho súčinu všetkých hodnôt nastavení
settings <- expand.grid(dataset = dataset, velkost = tm_velkost,
    ntree = f_ntree, mtry = f_mtry, nodesize = f_nodesize,
    pod_gho = pod_gho, pod_obl = pod_obl, pod_tep = pod_tep,
    pod_vie = pod_vie, pod_vlh = pod_vlh, pod_dlz = pod_dlz,
    pod_ele = pod_ele, ele_res = ele_res)
# 5. výpis času začatia vykonávania skriptu
time.start <- Sys.time()
print(sprintf("Start: %s ", time.start), quote = F)
# 6. vytvorenie spojenia s databázou
db.drv <- dbDriver("PostgreSQL")
if (exists("db.con")) dbDisconnect(db.con)
db.con <- getConnection(db.drv)
# 7. inicializácia a výpočet hodnôt premenných používaných na výpis
prog.diff <- 0
prog.printed_all <- -10000
prog.printed_ops <- -10000
prog.print_perc_all <- 0
prog.baseAll <- 0
```

```
prog.opsAll <- 0
prog.i <- 0
prog.op <- 0
prog.opsAll <- nrow(settings)
prog.baseAll <- 0
for (i.dataset in dataset) {
  b_select <- "select count(*) as ccc from (select distinct * from
                    (select cas, fve from %s where fve IN (%s)) s1) s2"
  prog.baseAll <- prog.baseAll + dbGetQuery(db.con,
           sprintf(b_select, i.dataset, toString(fve)))$ccc
}
prog.baseAll <- prog.baseAll * prog.opsAll / length(dataset)
hours_done <- 0
ops_done <- 0
# 8. cyklus pre každý riadok z tabuľky karteziánskeho súčinu nastavení
for (i.sett in 1:prog.opsAll) {
  setting <- settings[i.sett,]
  ops_done <- i.sett
  actual <- c()
  output <- c()
  # 9. cyklus pre každú z elektrární
  for (i.fve in fve) {
    # 10.   vytiahnutie záznamov z databázy,
    # výpočet škály pre hodnotenie podobnosti záznamov
    all_hours <- dbGetQuery(db.con, sprintf(select,
                                          setting$dataset, i.fve))
    ad_ncol <- ncol(all_hours)
    maxims <- apply(all_hours[,4:ad_ncol], 2, max)
    minims <- apply(all_hours[,4:ad_ncol], 2, min)
    scale <- abs(maxims - minims)
    all_hours <- data.matrix(all_hours)
    chosen_hours <- all_hours
    # 11. exportovanie premenných na všetky jadrá procesora, paralelný
      výpočet predikcie
    # pre každý záznam zvlášť, vráti vektor predikovaných hodnôt pre jednu
      elektráreň
    clusterExport(cl, list("chosen_hours", "all_hours", "scale",
                                                          "setting"))
    fve_output <- parSapply(cl, 1:nrow(chosen_hours), function(y) {
      # 12. vymedzenie potenciálnych záznamov do trénovacej množiny
      hourh <- chosen_hours[y,]
      potencial <- all_hours[all_hours[,'datum'] != hourh[['datum']],]
      potencial <- potencial[abs(potencial[,'elev'] - hourh[['elev']]) <=
                                                      setting$ele_res,]
      # 13. ohodnotenie potenciálnych záznamov podľa podobnosti
      diff <- vector(mode = "numeric", length = nrow(potencial))
      diff <- sapply(1:length(diff), function(x) {
        ret <- abs(hourh[['gho']] - potencial[[x,'gho']])
                              * 100 / scale[['gho']] * setting$pod_gho
        ret <- ret + abs(hourh[['oblacnost']] - potencial[[x,'oblacnost']])
                              * 100 / scale[['oblacnost']] * setting$pod_obl
        ret <- ret + abs(hourh[['teplota']] - potencial[[x,'teplota']])
                              * 100 / scale[['teplota']] * setting$pod_tep
        ret <- ret + abs(hourh[['vietor']] - potencial[[x,'vietor']])
                              * 100 / scale[['vietor']] * setting$pod_vie
        ret <- ret + abs(hourh[['vlhkost']] - potencial[[x,'vlhkost']])
                              * 100 / scale[['vlhkost']] * setting$pod_vlh
        ret <- ret + abs(hourh[['dlzkadna']] - potencial[[x,'dlzkadna']])
                              * 100 / scale[['dlzkadna']] * setting$pod_dlz
        ret <- ret + abs(hourh[['elev']] - potencial[[x,'elev']])
                              * 100 / scale[['elev']] * setting$pod_ele
```

```
            return(ret)})
      # 14. výber najpodobnejších záznamov do trénovacej množiny
      train_set <- arrange(as.data.frame(potencial), diff)
      [1:setting$velkost,]
      # 15. vytvorenie náhodného lesa regresných stromov
      forest <- randomForest(
            praca~gho+oblacnost+teplota+vietor+vlhkost+dlzkadna+elev,
            data=train_set, ntree = setting$ntree, mtry = setting$mtry,
            nodesize = setting$nodesize)
      # 16. predikcia produkcie pre vybraný záznam
      predic <-predict(forest, data.frame(gho = hourh[['gho']],
                                          oblacnost = hourh[['oblacnost']],
                                          teplota = hourh[['teplota']],
                                          vietor = hourh[['vietor']],
                                          vlhkost = hourh[['vlhkost']],
                                          dlzkadna = hourh[['dlzkadna']],
                                          elev = hourh[['elev']]),
                       type="response", norm.votes=TRUE)
      return(predic)
    }) # koniec paralelizovaného kódu
    # 17. pridanie skutočných a predikovaných hodnôt do vektorov spoločných
      # pre všetky elektrárne
    fve_actual <- chosen_hours[,'praca']
    actual <- append(actual, fve_actual)
    output <- append(output, fve_output)
    # 18. výpis stavu predikcie
    hours_done <- hours_done + nrow(all_hours)
    prog.i <- hours_done
    prog.print_perc_all <- (prog.i * 100 / prog.baseAll)
    prog.op <- ops_done
    prog.print_perc_ops <- (prog.op * 100 / prog.opsAll)
    if (prog.print_perc_all > prog.printed_all + prog.diff) {
      prog.actual_time <- as.numeric(difftime(Sys.time(),
                                              time.start, units = "sec"))
      prog.estimated_time <- prog.actual_time * 100 / prog.print_perc_all
      print(sprintf("Forest perc: %6.2f%s, ops: %7.d/%d, day: %9.d/%d,
                    Estimated time: %s, Actual: %s", #
                    prog.print_perc_all, "%", prog.op, prog.opsAll, prog.i,
                    prog.baseAll, format.time(prog.estimated_time),
                    format.time(prog.actual_time)),
            quote=F)
      prog.printed_all <- prog.print_perc_all
    }
  } # koniec cyklu pre každú elektráreň
  # 19.  výpočet štatistiky presnosti
  stats <- all_statistics(actual, output)
  if (write_results) {
    for (name in names(stats)) {
      if (is.infinite(stats[[name]]) | !is.numeric(stats[[name]]) |
                                                  is.nan(stats[[name]]))
        stats[[name]] <- 999.999
    }
    insert <- sprintf("INSERT INTO t_experiment (cas_behu, metoda, param1,
                      param2, param3, param4, param5, N, MBE, RMBE, RMSE,
                      RRMSE, MAE, RMAE, MPE, MAXAE, SD, tm_velkost,
                      tm_opis, tm_select, fve, den_hod, pod_gho,
                      pod_oblacnost, pod_teplota, pod_vietor,
                      pod_vlhkost, pod_tlak, pod_dlzkadna, pod_azim,
                      pod_elev, in_gho, in_oblacnost, in_teplota,
                      in_vietor, in_vlhkost, in_tlak, in_dlzkadna,
```

```
                        in_azim, in_elev) VALUES ('%s', '%s', '%s', '%s',
                        '%s', '%s', '%s', %d, %f, %f, %f, %f, %f, %f, %f,
                        %f, %f, %d, '%s', '%s', '%s', '%s', %f, %f, %f, %f,
                    %f, %f, %f, %f, %f, %s, %s, %s, %s, %s, %s, %s, %s, %s);",
                        time.start, "stats_hod", setting$dataset, "ntree "
                        %s% as.character(setting$ntree) , "mtry " %s%
                        as.character(setting$mtry), "nodesize " %s%
                        as.character(setting$nodesize), " ", stats$N,
                        stats$MBE, stats$RMBE, stats$RMSE, stats$RRMSE,
                        stats$MAE, stats$RMAE, stats$MPE, stats$MAXAE,
                        stats$SD, setting$velkost, "30 najpodob hodin",
                        select, toString(fve), "hod", setting$pod_gho,
                        setting$pod_obl, setting$pod_tep, setting$pod_vie,
                        setting$pod_vlh, 0, setting$pod_dlz, 0,
                        setting$pod_ele, TRUE, TRUE, TRUE, TRUE, TRUE,
                        FALSE, TRUE, FALSE, TRUE)
    db.result <- dbGetQuery(db.con, insert)
  }
# 20.   úprava výsledkov pre výpočet dennej štatistiky presnosti
  s_select <- "SELECT fve, datum, cas, gho, oblacnost,
                      teplota, vietor, vlhkost, dlzkadna, elev, praca
                      FROM %s WHERE fve IN (%s) ORDER BY fve, cas"
  all_data <-  dbGetQuery(db.con, sprintf(s_select, setting$dataset,
                                                    toString(fve)))
  all_data <- cbind(all_data, output)
  to_see <- cbind(all_data, dif = abs(output - actual) * 100 / actual)
  to_see <- arrange(to_see, to_see$dif)
  groupped <- ddply(all_data,~datum+fve,summarise, gho=sum(gho),
                      oblacnost=sum(oblacnost), teplota=sum(teplota),
                      vietor=sum(vietor), vlhkost=sum(vlhkost),
                      dlzkadna=max(dlzkadna), praca=sum(praca),
                      output=sum(output))
  # 21. výpočet dennej štatistiky presnosti a zápis výsledkov do databázy
  stats_day <- all_statistics(groupped$praca, groupped$output)
  if (write_results) {
    for (name in names(stats_day)) {
      if (is.infinite(stats_day[[name]]) | !is.numeric(stats_day[[name]]) |
                                            is.nan(stats_day[[name]]))
          stats_day[[name]] <- 999.999
    }
    insert <- sprintf("INSERT INTO t_experiment (cas_behu, metoda, param1,
                                  param2, param3, param4, param5,
                                   N, MBE, RMBE, RMSE, RRMSE, MAE, RMAE,
                                  MPE, MAXAE, SD, tm_velkost, tm_opis,
                                  tm_select, fve, den_hod, pod_gho,
                                  pod_oblacnost, pod_teplota, pod_vietor,
                                  pod_vlhkost, pod_tlak, pod_dlzkadna,
                                  pod_azim, pod_elev, in_gho,
                                  in_oblacnost, in_teplota, in_vietor,
                                  in_vlhkost, in_tlak, in_dlzkadna,
                                  in_azim, in_elev) VALUES ('%s', '%s',
                                  '%s', '%s', '%s', '%s', '%s', %d, %f,
                                  %f, %f, %f, %f, %f, %f, %f, %f, %d,
                                  '%s', '%s', '%s', '%s', %f, %f, %f, %f,
                                  %f, %f, %f, %f, %f, %s, %s, %s, %s, %s,
                                  %s, %s, %s, %s);", time.start,
                                  "stats_den", setting$dataset, "ntree "
                                  %s% as.character(setting$ntree) , "mtry
                                  " %s%, as.character(setting$mtry),
```

```
                        "nodesize " %s% as.character(setting$nodesize),
                        " ", stats_day$N, stats_day$MBE, stats_day$RMBE,
                        stats_day$RMSE, stats_day$RRMSE, stats_day$MAE,
                        stats_day$RMAE, stats_day$MPE, stats_day$MAXAE,
                        stats_day$SD, setting$velkost, "30 najpodob hodin",
                        select, toString(fve), "hod", setting$pod_gho,
                        setting$pod_obl, setting$pod_tep, setting$pod_vie,
                        setting$pod_vlh, 0, setting$pod_dlz, 0,
                        setting$pod_ele, TRUE, TRUE, TRUE, TRUE, TRUE,
                        FALSE, TRUE, TRUE, FALSE)
    db.result <- dbGetQuery(db.con, insert)
  }
} # koniec cyklu pre každý riadok z tabuľky karteziánskeho súčinu nastavení
# 22. konečný výpis,
time.end <- Sys.time()
print(sprintf("Start: %s, End: %s, Duration: %s",
              time.start, time.end,
              format.time(difftime(time.end, time.start, units = "sec"))),
      quote = F)
# 23. ukončenie možnosti paralelizácie a zatvorenie spojenia s databázou.
stopCluster(cl)
if (exists("db.con")) dbDisconnect(db.con)
```

V nasledujúcom zdrojovom kóde implementácia vlastných funkcií použitých v predchádzajúcom skripte.

Zdrojový kód 2: Implementácia vlastných funkcií v externom súbore.

```
# vráti spojenie s databázou
getConnection <- function(drv) {
  con <- dbConnect(drv, dbname = "test_db", host = "localhost",
                   port = 5432, user = "postgres", password = "password")
  return(con)
}
# výpočet štatistiky presnosti
library(Metrics)
library(sirad)
all_statistics <- function(actual, predicted) {
  mdval <- modeval(predicted, actual,stat = c("N", "MBE", "RMBE", "RMSE",
                  "RRMSE", "MAE", "RMAE", "MPE", "SD"), minlength = 2)
  mdval$MAXAE <- max(ae(actual, predicted))
  return(mdval)
}
# formátovanie času
format.time <- function(x) UseMethod("format.time")
format.time.difftime <- function(x) {
  units(x) <- "secs"
  x <- unclass(x)
  NextMethod()
}
format.time.default <- function(x) {
  y <- abs(x)
  sprintf("%s%02dh:%02dm:%02ds", ifelse(x < 0, "-", ""),
      y %% 86400 %/% 3600, y %% 3600 %/% 60, y %% 60 %/% 1)
}
# vlastný operátor na spájanie reťazcov
`%s%` <- function(s1, s2) paste0(s1, s2)
```