



Bakalářská práce

Predikce profilů spotřeby elektrické energie

Studijní program:

Studijní obor:

Autor práce:

Vedoucí práce:

B0613A140005 – Informační technologie

B0613A140005AI-80 – Aplikovaná informatika

Pavel Vácha

Ing. Jan Kraus, Ph.D.

Liberec 2024

Tento list nahrad'te
originálem zadání.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

11. 2. 2024

Pavel Vácha

PREDIKCE PROFILŮ SPOTŘEBY ELEKTRICKÉ ENERGIE

ABSTRAKT

Tato práce se zabývá problematikou krátkodobých a střednědobých predikcí spotřeby elektrické energie pomocí hlubokých neuronových sítí a dalších metod strojového učení. Finální modely jsou natrénovány na datových sadách poskytnuté společností Albistech a na veřejných datech společnosti UK Power Networks. Datové sady byly očištěny a byla na nich provedena explorační analýza pro dosažení lepších výsledků. Výsledky ukazují, že finální modely dokáží předpovídat až s 95% přesností v rámci jednoho dne a dokáží tak poskytnout spolehlivé výsledky pro budoucí implementaci v informačním systému NEO a poskytnout tak zákazníkům nástroj pro efektivnější hospodaření.

Klíčová slova: spotřeba energie, analýza časových řad, predikce, parametrické modelování, strojové učení

PREDICTION OF POWER CONSUMPTION PROFILES

ABSTRACT

This thesis explores the issue of short and medium term forecasting of electricity consumption using deep neural networks and other machine learning methods. The final models are trained on datasets provided by Albistech and UK Power Networks public data. The datasets were cleaned and exploratory data analysis was performed to obtain better results. The results show that the final models can forecast with up to 95% accuracy within one day, and can thus provide reliable results for the future implementation in the NEO system and provide customers with a tool for more efficient management.

Keywords: energy consumption, time series analysis, forecasting, parametric modeling, machine learning

PODĚKOVÁNÍ

Rád bych poděkoval všem, kteří přispěli ke vzniku tohoto díla. Zejména společnosti Albistech s.r.o za poskytnutá data a zázemí pro vypracování práce.

OBSAH

Seznam zkratek	9
1 Úvod	10
2 Časové řady	11
2.1 Komponenty časové řady	11
2.1.1 Trendová komponenta	11
2.1.2 Sezónní komponenta	11
2.1.3 Cyklická komponenta	12
2.1.4 Náhodná komponenta	12
2.2 Stacionariata	12
2.3 Dekompozice časové řady	13
2.3.1 Aditivní model	13
2.3.2 Multiplikativní model	14
2.3.3 Klouzavý průměr	14
2.3.4 Klasická dekompozice	15
3 Průzkum současných trendů v oblasti prediktivních metod	16
3.1 Regresní analýza	16
3.1.1 Regresní modely	17
3.2 ARIMA	17
3.2.1 Autoregresivní model	17
3.3 Rozhodovací stromy	19
3.3.1 Gradientní boostované stromy	19
3.4 Neuronové sítě	20
3.4.1 Perceptron	21
3.4.2 Vícevrstvé perceptrony	21
3.4.3 Rekurentní neuronové sítě	22
3.4.4 Long short-term memory networks	23
3.5 Konvoluční neuronové sítě	25
3.5.1 CNN-LSTM	25
4 Aplikace vybraných metod	26
4.1 Zdroj dat	26
4.1.1 IRIS Data Platform	28
4.2 Explorační analýza dat a jejich příprava	29

4.2.1	Předzpracování dat	29
4.2.2	Denní průměrné profily	29
4.2.3	Závislost počasí	29
4.2.4	Korelační matice	29
4.3	Implementace jednotlivých modelů	30
4.3.1	Hyper-parametry	30
4.4	Metodika vyhodnocení	30
5	Výsledky	31
6	Závěr	32
7	ChangeLog	33

SEZNAM ZKRATEK

LSTM	Long-short term memory, architektura rekurentní neuronové sítě
CNN	Convolutional Neural Network, konvoluční neuronová síť
MSE	Mean squared error, střední kvadratický chyba
MAE	Mean average error, průměrná absolutní odchylka
MAPE	Mean average error, průměrná procentuální absolutní odchylka
RMSE	Root mean squared error, směrodatný odchylka
ReLU	Rectified Linear Unit, aktivační funkce
GBT	Gradient boosted trees, gradientní boostované stromy
EDA	Exploratory data analysis, explorační datová analýza

1 ÚVOD

Vzorce spotřeby a jejich dopady na naše životní prostředí jsou aktuálně mezi největšími výzvami naší doby. Díky pochopení jak a kdy lidé spotřebovávají energii a jak

se tyto vzorce mění, jsme schopni zajistit udržitelnou budoucnost. Tradiční modely spotřeby se obvykle opírají o národní nebo globální data, která nemusí zcela odrážet lokální vzorce spotřeby nebo jejich podmínky (např. klimatické). Naproti tomu modely lokální spotřeby mohou poskytnout přesnější a mnohem

relevantnější informace pro konkrétní region či oblast. V tomto projektu si kladu za cíl vyvinout model lokální spotřeby založený na historických datech spolu s environmentálními parametry, pokud jsou pro danou oblast dostupné. Tento model může být velmi cenný pro plánování a implementaci udržitelných opatření v oblasti energetiky a ochrany životního prostředí. Získání přesnějšího a místně relevantního pohledu na spotřebu energie nám umožní přijímat informovaná rozhodnutí a přizpůsobit naše strategie tak, aby byly co nejefektivnější a nejohleduplnější k životnímu prostředí. Pro vytvoření spolehlivého a vhodného modelu lokální spotřeby se část tohoto projektu bude zabývat rešerší různých technik strojového učení spolu s metodami ze statistické analýzy.

2 ČASOVÉ ŘADY

Spotřeba energie v domácnostech může být chápána jako posloupnost měření, kde každý jednotlivý záznam je v určitém časovém okamžiku. Před samotným průzkumem prediktivních metod je nutné si definovat několik pojmů z oblasti analýzy časových řad, které se budou v následujících kapitolách vyskytovat.

Časová řada je definována jako množina pozorování x_t , kde každé pozorování má záznam v čase t . ?? Příkladem typické časové řady může být vývoj ceny akcií na burze v čase nebo výše zmíněná spotřeba energie. Každá časová řada se dá rozložit (dekomponovat) na několik jednotlivých složek (komponent).

2.1 KOMPONENTY ČASOVÉ ŘADY

Dekompozice časové řady umožňuje získat jednotlivé komponenty, jenž odhalují určité vlastnosti časové řady. Tyto komponenty a informace co nesou, jsou poté užitečné při samotné analýze a následné predikci časové řady.

2.1.1 Trendová komponenta

Tato komponenta zachycuje celkový dlouhodobý směr zkoumaného jevu časové řady. Trend může být rostoucí, klesající a nebo kompletně bez trendu. V kontextu této práce může být tato složka ovlivněna různými faktory, jako jsou změny v ekonomice, demografii nebo technologický pokrok. Označuje se jako T_t .

2.1.2 Sezónní komponenta

Sezónní komponenta S_t určuje krátkodobý vzor, který se opakuje v pravidelných intervalech, avšak s frekvencí rok a méně. ?? V kontextu spotřeby energie v domácnostech může tato složka zahrnovat sezónní vzory spojené s ročními změnami, jako je zvýšená spotřeba v zimním období kvůli vytápění nebo v letním období kvůli klimatizaci. Tato složka bude klíčová pro identifikaci střednědobých cyklů spotřeby energie.

2.1.3 Cyklická komponenta

Analýza této cyklické komponenty C_t může být klíčová při zkoumání vlivu širších socioekonomických faktorů na dlouhodobou spotřebu energie v domácnostech. Tato složka totiž představuje dlouhodobější vzory, jejichž frekvence přesahuje minimálně jeden rok. ?? Zpravidla vyjadřuje kolísání okolo trendové komponenty.

2.1.4 Náhodná komponenta

Jedná se o náhodné výkyvy, které nemohou být předpovězeny pomocí předchozích komponent a mohou být způsobeny různými nepředvídatelnými událostmi, jako jsou chyby v měření nebo neočekávané výpadky dodavatele energie. Zpravidla se značí jako ϵ_t .

2.2 STACIONARITA

Pro analýzu časové řady je vhodné (pro většinu analýz podmínkou), aby řada byla takzvaně stacionární. Za stacionární časové řady se považují takové řady, které nemají trend, mají s měnícím se časem stejný rozptyl a stejný průběh autokorelační funkce. Tyto vlastnosti usnadňují predikci budoucích hodnot a proto je snaha převést řadu na stacionární.

Nejjednodušší transformace řady na stacionární je pomocí diferencování. Tato transformace přispívá ke snížení trendové složky. Například diferenci prvního řádu se zapisuje jako:

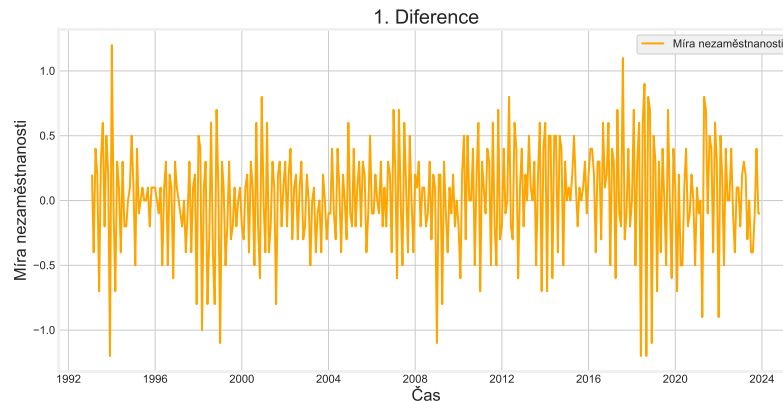
$$\Delta x_t = x_t - x_{t-1} \quad (2.1)$$

kde x_t je hodnota v čase t a x_{t-1} je vzorek předcházející. Při analýze následující řady níže, kde je vidět lineární rostoucí trend, je třeba provést diferenci, aby tento trend zmizel.



Obrázek 2.1: Míra registrované nezaměstnanosti v ČR od roku 1993

Po aplikaci difference prvního řádu lze vidět, že řada již nemá zjevný rostoucí trend a bylo by možné pokračovat v další analýze.



Obrázek 2.2: Míra registrované nezaměstnanosti po spočtení první difference

Pro otestování zdali je řada stacionární existuje několik testů ze statistické analýzy. V implementační části této práce bude využito Dickey-Fullerova testu pro ověření stacionarity dat, nad kterými bude autor práce tvořit jednotlivé modely.

2.3 DEKOMPOZICE ČASOVÉ ŘADY

Jak bylo zmíněno výše, dekompozice umožňuje rozložit časovou řadu na jednotlivé komponenty. Z těch poté lze lépe pochopit, jakým způsobem se v časové řadě projevuje trend a sezónní variabilita. To je užitečné pro interpretaci dat a odhalení skrytých trendů nebo sezónních vzorů, což může být klíčové pro predikci a optimalizaci energetické efektivity v domácnosti.

Někdy se samotné komponenty po dekompozici využívají k predikci, obzvláště pokud je trend jednoduchý. Lze ho totiž lépe predikovat, a stejně tak i sezónní vzory. ??

Pro dekompozici se zpravidla používají dva základní modely, a to aditivní a multiplikativní. Volba modelu závisí na charakteru sezónnosti a trendu v časové řadě.

2.3.1 Aditivní model

Pro tento model se předpokládá, že řadu lze rozložit jako součet jednotlivých komponent.

$$x_t = T_t + S_t + C_t + \epsilon_t \quad (2.2)$$

kde x_t je samotná hodnota časové řady v čase t .

2.3.2 Multiplikativní model

Analogicky tento model naopak předpokládá, že řadu lze rozložit jako součin jednotlivých komponent.

$$x_t = T_t * S_t * C_t * \epsilon_t \quad (2.3)$$

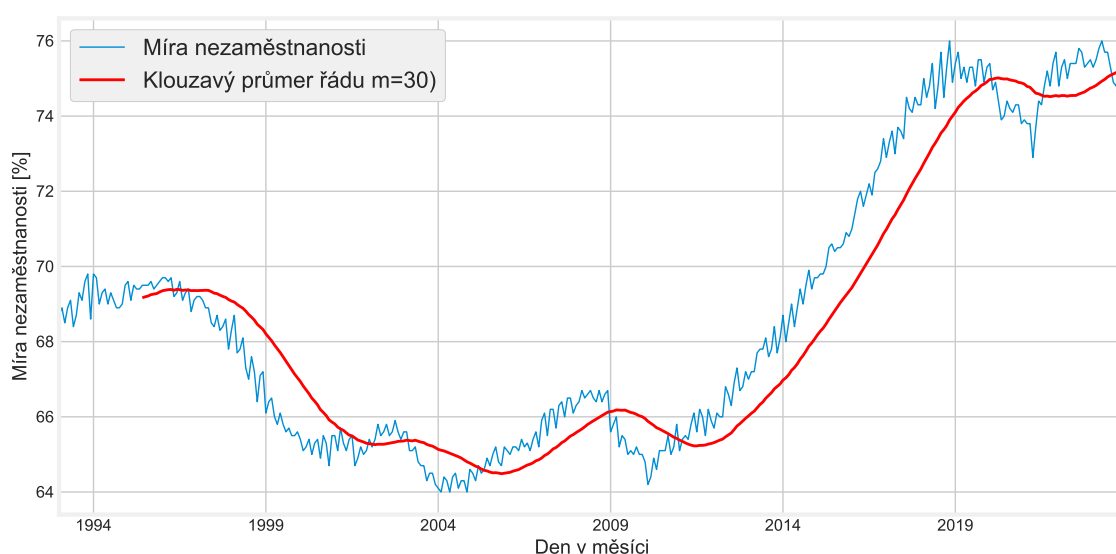
Tento model se využívá, pokud se výkyvy sezonní složky vůči trendu mění v čase.

2.3.3 Klouzavý průměr

Tento jednoduchý výpočet, který je schopen zvýraznit trend-cyklickou složkou, je jednou z nejjednodušších metod dekompozice. Mimo jiné dokáže vyhladit časovou řadu od krátkodobých výkyvů. Spočívá v jednoduchém průměrování různých pozorování a nebo celé řady. Klouzavý průměr řádu m lze zapsat jako:

$$SMA_m = \frac{1}{m} \sum_{i=-k}^k x_{t+i} \quad (2.4)$$

kde k se používá k určení, kolik hodnot před a po aktuálním čase t se zahrne do výpočtu klouzavého průměru. Platí, že $m = 2k + 1$



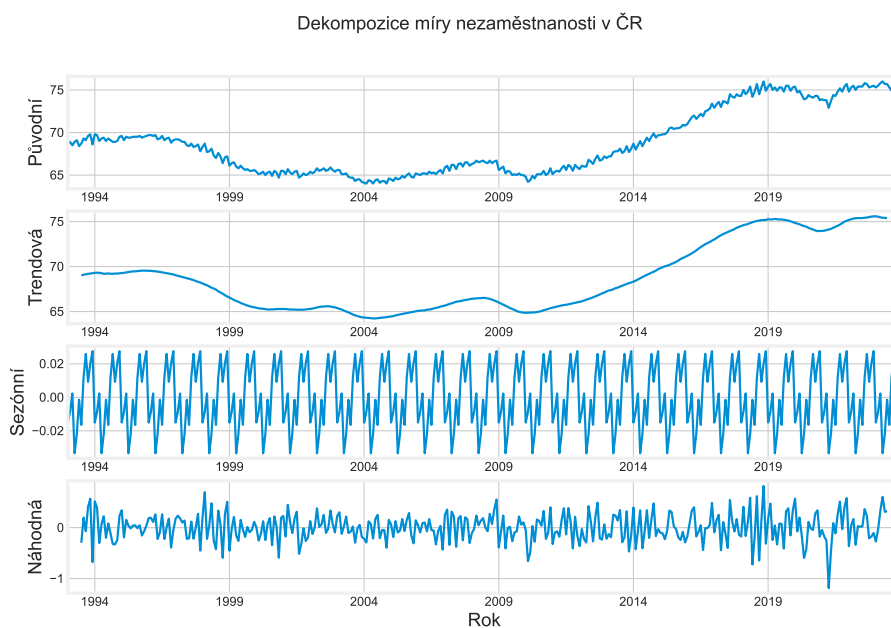
Obrázek 2.3: Klouzavý průměr řádu $m=30$ popisující míru nezaměstnanosti v ČR od roku 1993.

Na grafu výše lze vidět, že klouzavý průměr (červená čára) s řádem $m = 30$ vyhladil průběh časové řady od výkyvů a zvýraznil trend-cyklickou komponentu.

2.3.4 Klasická dekompozice

O něco sofistikovanější metodou je klasická dekompozice. Pro provedení samotné dekompozice je nutné dodat tři vstupy. Analyzovanou časovou řadu, informaci o tom jestli časová řada má model aditivního nebo multiplikativního charakteru a jak velkou má časová řada sezónní periodu m (např. týdenní - $m = 7$, měsíční - $m = 12$, atd.).

V implementační části pak jde o čtyři základní kroky. Výpočet trend-cyklické komponenty, o detrendizaci časové řady, aproximace sezónní složky a výpočet náhodné složky. Jednotlivé kroky se liší pro aditivní a multiplikativní model.



Obrázek 2.4: Klasická dekompozice s aditivním modelem pro data s měsíčními záznamy o míře nezaměstnanosti v ČR od roku 1993

V náhodné komponentě nám zbyla časová řada po odstranění trend-cyklické a sezónní komponenty. Někdy se jí místo náhodná říká detrendizovaná komponenta. V tomto případě model byl aditivní, tedy odstranění proběhlo jako $\epsilon_t = x_t - T_t - S_t$, pro multiplikativní model by vztah vypadal jako $\epsilon_t = \frac{x_t}{T_t S_t}$.

Z pohledu na graf výše je vidět, že řada má nějaký trend i sezónnost. Na grafu náhodné komponenty lze zaznamenat několik vysokých výkyvů. V roce 2008 náhodná komponenta ukázala výrazný nárůst nezaměstnanosti, který nelze vysvětlit trendem, ani cyklickou složkou. (V tomto roce začala tzv. Velká recese).

Další je vidět v roce 2019, kde Česká Republika měla silnou rostoucí ekonomiku. Hned po roce 2019 je vidět další výkyv, tentokrát negativní, způsobený pandemií.

3 PRŮZKUM SOUČASNÝCH TRENDŮ V OBLASTI PREDIKTIVNÍCH METOD

V současné době s rostoucím objemem sbíraných dat se stále více organizací zaměřuje na prediktivní modelování. Díky dnešním výkonným výpočetním technologiím je možné zužítkovat nasbíraná data a využít metody strojového učení pro přesné predikce a odhad budoucích trendů.

Tato kapitola si klade za cíl provést rešerši aktuálně používaných metod pro predikce. Výstupem by měl být výběr několika metod a jejich následná aplikace na datech o spotřebě.

3.1 REGRESNÍ ANALÝZA

V oblasti předpovědi spotřeby energie lze často najít modely, které byly vytvořeny pomocí modelování spotřeby energie touto technikou (nebo její podmnožinou). Jeden z důvodů, proč se používají je jednoduchost použití a interpretace.

Regresní model popisuje vztah mezi jednou nebo více nezávislými proměnnými a jednou závislou proměnnou. Cílem regresní analýzy je najít funkční vztah mezi závislými a nezávislými proměnnými, který je potom použit k predikci hodnoty závislé proměnné na základě hodnot nezávislých proměnných.

Regresní model s více než jednou proměnnou se zapisuje jako

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \epsilon \quad (3.1)$$

kde y je výstupní proměnná, ϵ chybový člen a $\beta_i, i=0, 1, 2 \dots p$ jsou regresní parametry.

Po získání regresních koeficientů je možné použít rovnici výše k predikci spojité hodnoty. Regresní koeficienty se dají odhadovat pomocí metody nejmenších čtverců. Tento postup spočívá v nalezení hodnot parametrů β takových, aby součet čtverců rozdílů mezi predikovanými hodnotami výstupní proměnné y a skutečnými hodnotami byl co nejmenší. [?]

Regresní analýza neříká nic o tom, jaký je mechanismus, kterým jsou tyto vztahy vysvětleny. To znamená, že regresní analýza může být velmi užiteč-

ným nástrojem pro předpovídání budoucích hodnot, ale nemůže sloužit jako nástroj pro vysvětlování příčin.

3.1.1 Regresní modely

Existuje několik druhů regresních modelů, jenž se používají v různých případech v závislosti na povaze dat a cíle analýzy. Jedním z nejprimitivnějších modelů je lineární regrese, zpravidla se používá jako validační model při práci s velkými a komplexními datovými sadami, jelikož nedosáhne stejných výsledků jako pokročilejší metody. Její princip spočívá v prokládání dat přímkou a hledáním parametrů dané přímky, aby co nejlépe modelovaly vztah mezi proměnnými.

Často se stává, že datové sady mají mnoho různých parametrů. Například v práci bude využita datová sada, která obsahuje informace o počasí v určitých časových intervalech. Ukáže se, že některé parametry mezi sebou vysoce korelují a to může mít vliv na stabilitu regresního modelu. V praxi proto lze pro predikci spotřeby najít ještě jeden užitečný regresní model s názvem ridge (hřebenový). Tento model přistupuje k problému vysoké korelace tím, že zahrnuje další regularizační parametr do regresního modelu, který penalizuje vysoké koeficienty (ty co spolu korelují).

3.2 ARIMA

ARIMA model je pravděpodobně mezi nejznámějšími modely pro predikci časových řad. Klíčové prvky modelu zahrnují autoregresi $AR(p)$, integraci $I(d)$ a klouzavý průměr $MA(q)$?? . Samotná rovnice ARIMA je tedy kombinací stacionarizovaného autoregresivního modelu a k tomu přidaný model s klouzavým průměrem.

3.2.1 Autoregresivní model

Autoregresivní model lze popsat jako funkci, jenž vrací predikovanou hodnotu x_t na základě svých předchozích hodnot x_{t-1}, \dots, x_{t-n} . Těmto hodnotám se říká zpožděné (lag) proměnné a slouží ke zvýšení přesnosti autoregresivního modelu.

Toho se dosahuje snižováním hodnoty t , která označuje počet kroků v časové řadě dat. Vyšší počet těchto kroků umožňuje modelu zachytit více předchozích předpovězených hodnot jako vstup. Například se dá rozšířit autoregresivní model tak, aby zahrnoval předpovězenou teplotu za posledních 7 dní až po posledních 14 dní, což může přispět k přesnějším výsledkům. ?? Obecně bychom mohli model zapsat jako funkci:

$$x_t = f(x_{t-1}, x_{t-2}, \dots, x_{t-n}, \epsilon_t) \quad (3.2)$$

kde ϵ_t je bílý šum, který platí pro data co mají průměr a variaci dat nulovou.

Někdy se nehodí, aby veškeré zpožděné proměnné měly stejnou váhu. Model je připraven na použití vážených parametrů a díky tomu lépe reagovat na výkyvy v časové řadě. Výsledná rovnice pro autoregresivní model s váženými parametry α bude vypadat následovně:

$$x_t = \sum_{i=1}^n \alpha_i x_{t-i} + \alpha_0 + \epsilon_t \quad (3.3)$$

kde α_0 je výchozí konstantní bod pro predikce (může být nulový a tedy být zanedbán).

Model nelze aplikovat na plně náhodná data bez vzoru. Před použitím tohoto modelu je nutné zajistit stacionaritu časové řady, což lze dosáhnout diferencováním řady zmíněném v kapitole ??.

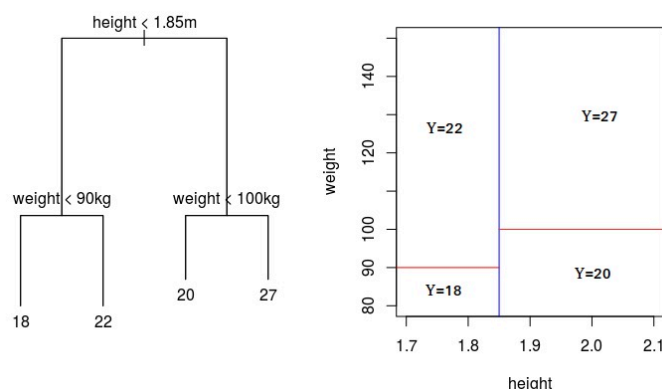
3.3 ROZHODOVACÍ STROMY

Pro strom existuje mnoho analogií a jedna z analogií se dostala i do modelování. Rozhodovací stromy můžou pomoci jak s klasifikačními, tak i s regresními problémy.

Jak název napovídá, při tvorbě rozhodovacího stromu je snaha rozdělit data na menší a jednodušší skupiny pomocí pravidel (segmentace dat), která se postupně aplikují na různé atributy dat. Cílem je vytvořit sadu pravidel, která umožní co nejpřesněji predikovat hodnotu požadované proměnné pro nově příchozí data.

Při tvorbě stromu se postupuje od shora dolů. V kořeni stromu je nutné zvolit jeden hlavní dělící parametr s nejvyšší schopností rozlišení dat, například "výška < 1.85m" a dále se již větví na základě největšího poměrného *informačního zisku* (založeno na výpočtu entropie). [?]

Pro tvorbu těchto stromů existuje mnoho metod. Zpravidla založených na statistických metodách. Nejčastějšími zástupci těchto metod jsou algoritmy **gradientní boostované stromy**, CHAID, C5.0 či náhodné lesy. [?]



Obrázek 3.1: Ukázka rozhodovacího stromu

3.3.1 Gradientní boostované stromy

Tato metoda je přesná a použitelná pro oba základní problémy, klasifikaci i regresi napříč různými odvětvími. Jelikož se tato práce zabývá predikcí spotřeby energie, což je spojitá veličina, budou prozkoumány možnosti regresního řešení problémů s pomocí GBT.

Hlavním úkolem boostování je sestavit z mnoha slabších modelů výrazně silnější model, takzvaně *ensemble model*. [?]

V každé iteraci se vytváří nový rozhodovací strom, který se zaměřuje na opravu chyb (reziduí) předchozího stromu. Tímto způsobem je postupně zlepšována přesnost predikce a minimalizována chybovost modelu.

Mezi nejpopulárnější implementace GBT patří například XGBoost, LightGBM a CatBoost. Tyto implementace využívají různé optimalizace a techniky

pro minimalizaci přeučení a maximalizaci přesnosti predikce.

Pro tento projekt byla zvolena implementace XGBoost v Pythonu pro svou přehlednou dokumentaci a jednoduchosti použití

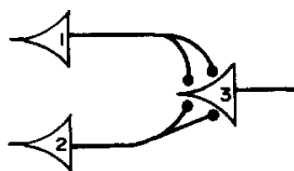
3.4 NEURONOVÉ SÍTĚ

V poslední letech si neuronové sítě vydobýly své postavení na poli analýzy a predikci dat. Neuronové sítě fungují dobře v problematice predikce hodnot, kde nejsou známy matematické vzorce a předchozí znalosti o vztazích mezi vstupy a výstupy. Fungují dobře v této problematice díky svému původu a díky inspiraci v neurofyzilogii lidského mozku.

Na první pohled to může vypadat, že neuronové sítě jsou novinka posledního století. Není tomu tak, je to pouze posunutím vývoje výkonného hardwaru kupředu. Neuronové sítě tu jsou již od 50. let, kdy Warren McCulloch a matematik Walter Pitts vydali publikaci, ve kterém modelovali zjednodušený model neuronu s elektrickým obvodem, jehož aktivační funkce byla pouze binární. [?]

$$y = \begin{cases} 1 & z \geq T \\ 0 & z < T \end{cases} \quad (3.4)$$

Autoři práce také ukázali, jak se dají sestavit pomocí neuronu jednoduché logické funkce:



Obrázek 3.2: Logický součet pomocí biologických neuronů

Neuron označme jako N s indexem i , který je uvnitř neuronu, v tom případě na obrázku výše vidíme $N_3(t) = N_1(t-1) \vee N_2(t-1)$, což znamená, že se jedná o logický součet. [?] Autoři v práci dále naznačili, jak se dají skládat tyto neurony do sítě.

Toto položilo první základy na rozvoj neuronových sítí v roce 1957 Frankem Rosenblattem, který začal práci na perceptronu.

3.4.1 Perceptron

Perceptron je nejjednodušší model dopředné neuronové sítě, který se skládá z jednoho jediného neuronu.

Perceptron přijímá na vstupu vektory x_i , ke kterým je přiřazena váha w_i . b je potom práh (bias), jenž ovlivňuje nakolik musí být suma vah větší než 0, aby se perceptron aktivoval. [?]. Aktivace neuronu se pak spočítá následovně:

$$z = \sum_{i=1}^n w_i x_i + b \quad (3.5)$$

A samotný výstup perceptronu (neuronu) Z , kde oproti jednoduchému neuronu nemusí být pouze binární přenosová funkce, ale libovolná přenosová funkce (zpravidla skoková nebo sigmoidální pro jeden perceptron) [?] se spočítá jako:

$$Z = H(z) \quad (3.6)$$

kde

$$H(z) = \begin{cases} 1 & f(z) \geq 0 \\ 0 & f(z) < 0 \end{cases} \quad (3.7)$$

Učení perceptronu probíhá jednoduše. Na vstup přivádíme (x, y) z trénovací sady (kde x jsou příznaky a y požadované výstupy) a aktualizujeme váhy perceptronu dle rovnice:

$$w_{n,i} = w_i + r(y_i - Z_i)x_n \quad (3.8)$$

kde r je parametr učení, který nám určuje, jak rychle se mění váha a kde n je n -tý vstup.

Důležité je poznamenat to, že tento perceptronový algoritmus je jednoduchý a efektivní pro řešení lineárně rozdělitelných problémů, ale není schopný řešit problémy, které nejsou lineárně rozdělitelné. Například funkci XOR, která se chová nelineárně, již tímto způsobem nevyřešíme. To vedlo k rozšíření této oblasti a vznikly tak vícevrstvé perceptrony, kde se pro učení používá algoritmus zpětného šíření.

3.4.2 Vícevrstvé perceptrony

Vícevrstvé perceptrony mají na rozdíl od perceptronu ještě skryté vrstvy, které se nachází mezi vstupní a výstupní vrstvou. Skrytých vrstev může být více nebo jenom jedna, to už záleží na volbě programátora a na typu problému. Neuronové sítě s jednou vrstvou se nazývají mělké. Sítě s více skrytými vrstvami se nazývají hluboké sítě. V souvislosti s vícevrstvámi perceptrony se mluví také o dopředné síti (feedforward). Vstupní vektor dat $I = (x, y)$ jde do skrytých vrstev H a poté na výstup Y . Vektor hodnot z výstupní vrstvy označíme jako Y .

Výstup Y poté získáme pomocí rovnice níže

$$H_i = \sigma(w_{ni} \cdot I_n + b_i) \quad (3.9)$$

$$Y = \sigma(w_{ij} \cdot H + b_{out}) \quad (3.10)$$

kde σ označuje přenosovou funkci a w_{ni} váhu spojení mezi n -tým vstupním a i -tým výstupním neuronem.

Ve více vrstevných sítích se používají jiné přenosové funkce než v samostatném perceptronu. V současnosti vyčnívá jedna velmi používaná funkce ReLU (rectified linear unit), která se definuje jako:

$$Relu(z) = \max(0, z) \quad (3.11)$$

a používá se především ve skrytých vrstvách. [?] Na výstupní vrstvě je zpravidla funkce softmax, definovaná jako:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}} \quad pro \ i = 1, 2, \dots, n \quad (3.12)$$

Pro trénování pak již nelze použít klasický perceptronový algoritmus (viz 3.8). Nejčastěji se používá algoritmus zpětného šíření (back-propagation).

Při použití algoritmu zpětného šíření se výstupní chyba sítě propaguje zpět k vstupní vrstvě, přičemž se upravují váhy všech spojů v síti tak, aby se minimalizovala tato chyba. Tento proces se opakuje v průběhu několika iterací (epoch).

Dále bychom mohli mluvit o optimalizačních metodách, chybových funkcích a o tom jak jednotlivé vrstvy fungují, ale to je nad rámec tohoto projektu. Pro projekt je důležité zmínit *long short-term memory networks*, které vycházejí z rekurentních sítí, na které se zaměříme v další kapitole.

3.4.3 Rekurentní neuronové sítě

Zatím jsme zmínili pouze sítě, kde výstup závisí pouze na aktuálním vstupu. V našem případě časová řada spotřeby energie nezávisí pouze na aktuálním stavu a ke správnému výstupu by bylo vhodné zahrnout data předchozí.

Rekurentní sítě nám toto umožňují tím, že obsahují zpětná spojení, která fungují jako vnitřní stav (paměť). Nejjednodušší příklad rekurentní neuronové sítě může být síť, co má jeden vstup, výstup a jeden neuron ve skryté vrstvě. Když tento neuron bude mít spojení sám do sebe, tak při předložení vstupu neuron dostane kromě vstupu ještě aktivaci $h_{(t-1)}$ z minulého vstupu. Nej názornější je ukázat jak se změní výpočet výstupu z neuronu, když přidáme zpětná spojení.

$$h_t = \sigma(w_{hh}h_{t-1} + w_{hx}x_t) \quad (3.13)$$

kde h_t je aktuální stav neuronu, x_t je vstupní sekvence v čase t , w_{hh} váha rekurentního neuronu a w_{xh} váha vstupního neuronu. Zásadní změna oproti aktivaci v jednoduchém perceptronu (viz 3.4.1) je, že aktivace h_t závisí i na předchozí aktivaci neuronu h_{t-1} .

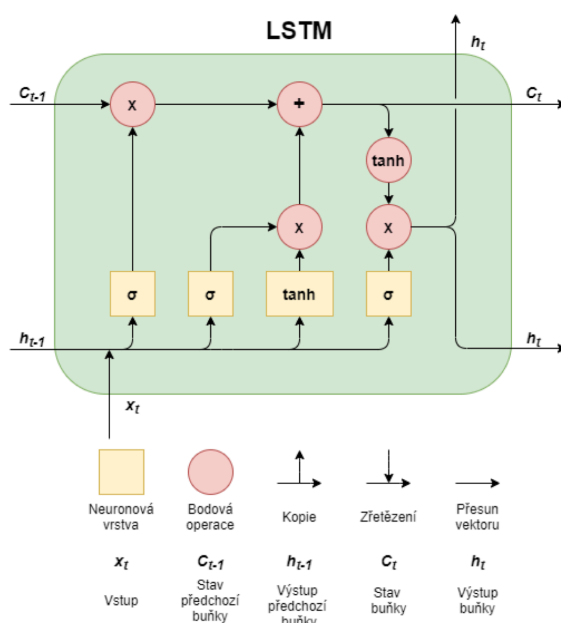
Trénování sítě probíhá opět algoritmem zpětného šíření, avšak lehce modifikovaným. Jmenuje se algoritmus zpětného šíření v čase (backpropagation through time). S trénováním rekurentních sítí se můžou vyskytnout dva problémy, díky tomu jak se zpětné šíření počítá. [?] Gradient se vždy násobí váhou. Pokud je tato rekurentní váha, tak se opakovaně násobí stejnou hodnotou. V případě, že tato hodnota bude větší než 1, dojde k problému explodujících gradientů, kdy hodnoty nekontrolovatelně rostou. V opačném případě, kdy je hodnota menší než 0, tak dochází k problému s mizejícími gradienty.

Jedno z možných řešení je použít další typ sítě. Tento typ sítě se jmenuje long-short term memory network a problém řeší tak, že rekurentní váha se zafixuje na hodnotu 1 a práce se stavem se provádí explicitně.

3.4.4 Long short-term memory networks

LSTM sítě jsou speciální druh rekurentních sítí, které dokáží zpracovat bez problému jednorozměrné signály a časové řady. Klíčovým prvkem těchto sítí jsou tzv. LSTM buňky, které nahrazují každý neuron.

Tyto buňky se skládají typicky z pěti prvků. Mezi tyto prvky patří input gate, forget gate, cell gate, output gate a samotná cell, která je jako aktuální stav ovlivňovaná vyjmenovanými prvky.



Obrázek 3.3: Jedna buňka LSTM sítě (převzato a přeloženo z [?])

LSTM mají řešit problém klasických RNN a to explodující a mizející gradienty. Problém je řešen tak, že přenos informací mezi časovým krokem probíhá pomocí stavu, s kterým není spojená žádná váha a díky tomu se při propagaci chyby vyhneme násobení, které by mohlo vyvolat tento problém. Díky této vlastnosti si LSTM sítě umí zapamatovat delší posloupnosti vstupů než základní rekurentní neuronové sítě.

Při rešerši jsem narazil také na kombinaci více sítí v jednu. Autoři jedné z prací zabývající se predikcí spotřeby energie v domácnosti [?] využívají například Bi-LSTM (obousměrná LSTM síť), CNN-LSTM (konvoluční neuronová síť s LSTM sítí) a kombinaci všeho - CNN-Bi-LSTM síť.

Kromě LSTM sítě existuje ještě její zjednodušená verze, které chybí output gate. Nazývá se gated recurrent unit (GRU) a v některých případech dosahuje vyšších přesností než LSTM, hlavně na menších datových sadách a převážně v oblast zpracování přirozeného jazyka. [?]

3.5 KONVOLUČNÍ NEURONOVÉ SÍTĚ

Co to je more? Proč to je tu vole

3.5.1 CNN-LSTM

Určitě někde bude [3.4.4](#)

4 APLIKACE VYBRANÝCH METOD

V této kapitole bude popsán způsob aplikace metod z předchozích kapitol. Po rešerši bylo rozhodnuto, že budou aplikovány dvě metody. První zvolenou metodou budou rozhodovací stromy, přesněji gradientní boostavené stromy za pomoci knihovny XGBoost. Druhá vybraná metoda je postavená na rekurentních neuronových sítích, tedy výše popisovaná LSTM síť. Zde využijeme bohaté knihovny Tensorflow od společnosti Google.

Před samotnou aplikací těchto metod se zaměříme na základní datovou analýzu, která nám pomůže odhalit nejlepší možné parametry pro výsledný model. Tato analýza nám umožní lépe porozumět datům, identifikovat vzorce a trendy v spotřebě elektrické energie a získat představu o nejdůležitějších faktorech ovlivňujících spotřebu.

Výstupem této kapitoly by měly být natrénované modely, které budou sloužit k predikci budoucích hodnot spotřeby elektrické energie. Jako jedno rozšíření této práce bude následovat implementace těchto modelů do informačního systému NEO.

Nakonec budou tyto modely vyhodnoceny na základě metrik přesnosti, které jsou uvedeny v kapitole 4.4. Tím získáme objektivní měřítko úspěšnosti a spolehlivosti našich modelů.

4.1 ZDROJ DAT

V práci byly využity dva zdroje dat. První ze zdrojů je veřejný a jedná se o kolekci 5567 domácností, které se zúčastnili projektu UK Power Networks vedené společností Low Carbon London mezi roky 2011 a 2014.

Projekt si kladl za cíl inovovat Londýnskou distribuční soustavu a zvýšit pokrytí nízkouhlíkovým vytápěním. V rámci počáteční studie projektu byly domácnosti vybaveny novými chytrými elektroměry a následně byl spolu se studií zveřejněn vzorek dat. [?]

Naměřená data jsou v půlhodinových intervalech a obsahují spotřebu v kWh, unikátní identifikační číslo domácnosti a časové razítko. Soubor má zhruba 10 GB dat a obsahuje přes 160 milionů řádků

Pro tvorbu robustnějších modelů a podrobnějších analýz byl také zveřejněn datový soubor s hodinovými informacemi o počasí získaný z *Dark Sky* API. Tato sada má následující strukturu:

Tabulka 4.1: Struktura datové sady se spotřebou v domácnostech (Londýn)

Sloupec	Popis
LCLid	Unikátní identifikátor domácnosti
tstp	Časové razítko ve formátu YYYY-MM-DD hh:mm:ss
energy(kWh/hh)	Spotřebovaná energie v kWh

Tabulka 4.2: Struktura datové sady s počasím pro Londýn

Sloupec	Popis
visibility	Maximální viditelnost
windBearing	Směr, ze kterého fouká vítr
temperature	Skutečná teplota ve stupních Celsia
time	Časové razítko ve formátu YYYY-MM-DD hh:mm:ss
dewPoint	Rosný bod
pressure	Aktuální atmosférický tlak měřený v hektopascálech.
apparentTemperature	Pocitová teplota, která je kombinací vlhkosti a reálné teploty
windSpeed	Rychlost větru měřená v km/h
precipType	Typ srážky (sníh nebo déšť)
icon	Název ikony signalizující oblačnost, slunečno, atd.

Pro účely této práce bude nutné transformovat data o počasí na intervaly po půl hodinách. Pokud by se neprovedla transformace, nebylo by možné využít datovou sadu pro Londýn v kombinaci s daty o počasí, aniž by se neztratila informace o měřené spotřebě.

Dále bude potřeba provést analýzu klíčových sloupců pro přesnější výsledky. Předpokládá se totiž, že například sloupec *icon* nebude signifikantní pro určení spotřeby energie.

4.1.1 IRIS Data Platform

4.2 EXPLORAČNÍ ANALÝZA DAT A JEJICH PŘÍPRAVA

4.2.1 Předzpracování dat

4.2.2 Denní průměrné profily

Projekt + udělat denní průměr na kompletně celým datasetu.

4.2.3 Závislost počasí

4.2.4 Korelační matice

4.3 IMPLEMENTACE JEDNOTLIVÝCH MODELŮ

Nakonec se vybrali následující modely. ARIMA, LSTM, LSTM-CNN, XGBOOST

4.3.1 Hyper-parametry

4.4 METODIKA VYHODNOCENÍ

Pro kvalitní určení, jak jsou modely úspěšné, je potřeba zavést nějaké metriky. Rozhodnutí bylo ve prospěch tří standardních veličin z matematické statistiky. Jedná se o střední kvadratickou chybu (dále již jen MSE).

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2 \quad (4.1)$$

a též směrodatná odchylka pro správné jednotky:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2} \quad (4.2)$$

Třetí zvolenou veličinou je průměrná absolutní odchylka, která je definována jako:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |x_i - y_i| \quad (4.3)$$

a pro vyjádření MAE v procentech:

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{x_i - y_i}{x_i} \right| \quad (4.4)$$

Použití těchto veličin umožňuje poskytnout užitečnou zpětnou vazbu vzhledem k přesnosti predikce a identifikovat případné nedostatky v modelu. Je důležité vyhodnocovat výsledky predikce pomocí více než jedné veličiny, aby se zajistila celková robustnost výsledného modelu.

5 VÝSLEDKY

6 ZÁVĚR

Tato práce si kladla za cíl ...

7 CHANGELOG

- 22.01. 2024 - Rozvržení kapitol
- 08.02. 2024 - Kapitola časové řady
- 09.02. 2024 - Podkapitola o stacionaritě a ukázkový skript
- 10.02. 2024 - Dokončena kapitola o časových řadách
- 11.02. 2024 - ARIMA
- 12.02. 2024 - CNN a CNN-LSTM SNAD?