FAKULTA MECHATRONIKY,
INFORMATIKY A MEZIOBOROVÝCH
STUDIÍ TUL

Bakalářská práce

Predikce profilů spotřeby elektrické energie

Studijní program: B0613A140005 – Informační technologie
Studijní obor: B0613A140005AI-80 – Aplikovaná informatika
Autor práce: **Pavel Vácha**
Vedoucí práce: Ing. Jan Kraus, Ph.D.

Liberec 2024

Tento list nahraďte
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.

16. 3. 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 účení

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 11

1 Úvod **12**

2 Časové řady **13**
2.1 Komponenty časové řady 13
2.1.1 Trendová komponenta 13
2.1.2 Sezónní komponenta 13
2.1.3 Cyklická komponenta 14
2.1.4 Náhodná komponenta 14
2.2 Stacionariata 14
2.3 Dekompozice časové řady 15
2.3.1 Aditivní model 15
2.3.2 Multiplikativní model 16
2.3.3 Klouzavý průměr 16
2.3.4 Klasická dekompozice 17

3 Průzkum současných trendů v oblasti prediktivních metod **18**
3.1 Regresní analýza 18
3.1.1 Regresní modely 19
3.2 ARIMA 19
3.2.1 Autoregresivní model 19
3.2.2 Model klouzavého průměru 20
3.2.3 ARIMA model 20
3.3 Rozhodovací stromy 21
3.3.1 Gradientní boostované stromy 21
3.4 Neuronové sítě 22
3.4.1 Perceptron 23
3.4.2 Vícevrstvé perceptrony 23
3.4.3 Rekurentní neuronové sítě 24
3.4.4 Long short-term memory networks 25
3.5 Konvoluční neuronové sítě 26
3.5.1 Konvoluční vrstva 26
3.5.2 Poolingová vrstva 27
3.5.3 CNN-LSTM 28

4 Aplikace vybraných metod **29**
4.1 Zdroj dat 30
4.1.1 IRIS Data Platform 32
4.2 Explorační analýza dat a jejich příprava 33
4.2.1 Předzpracování dat 33
4.2.2 Průměrné profily 35
4.2.3 Závislost počasí 38
4.3 Implementace jednotlivých modelů 41
4.3.1 XGBoost 41
4.3.2 Hyper-parametry 42
4.3.3 ARIMA 42
4.3.4 LSTM 42
4.3.5 CNN-LSTM 42
4.4 Metodika vyhodnocení 43

5 Výsledky **44**

6 Závěr **45**

7 ChangeLog **46**

Použitá literatura **48**

SEZNAM OBRÁZKŮ

2.1 Míra registrované nezaměstnanosti v ČR od roku 1993 14
2.2 Míra registrované nezaměstnanosti po spočtení první diference 15
2.3 Klouzavý průměr řádu m=30 popisující míru nezaměstnanosti v ČR od roku 1993. 16
2.4 Klasická dekompozice s aditivním modelem pro data s měsíčními záznamy o míře nezaměstnanosti v ČR od roku 1993 17

3.1 Ukázka rozhodovacího stromu 21
3.2 Logický součet pomocí biologických neuronů 22
3.3 Jedna buňka LSTM sítě (převzato a přeloženo z [8]) 25
3.4 Ukázka extrakce charakteristiky z obrazu pomocí konvoluce . 27
3.5 Ukázka maxpool operace s krokem $h = 2$ a $v = 2$ 27

4.1 Graf s počtem zapojených domácností v čase 34
4.2 Graf rozdělení zaznamenaných dní pro domácnosti 34
4.3 Hodinová spotřeba vybraných budov v pracovním týdnu 35
4.4 Hodinová spotřeba vybraných budov o víkendu 36
4.5 Hodinová spotřeba všech domácností z datové sady 36
4.6 Hodinová spotřeba vybraných domácností dle období 37
4.7 Denní průměrný profil vybraných domácností v týdnu 37
4.8 Denní průměrný profil vybraných domácností v roce 38
4.9 Vztah mezi průměrnou denní spotřebou a průměrnou denní teplotou 39
4.10 Ilustrační korelační matice zobrazující vztah mezi spotřebou a počasím 40

SEZNAM TABULEK

4.1 Struktura datové sady se spotřebou v domácnostech (Londýn) 30
4.2 Struktura datové sady s počasím pro Londýn 30
4.3 Struktura datové sady se spotřebou v domácnosti (Brno) . . . 31
4.4 Společné parametry pro všechny predikční modely 41
4.5 Zpožděné proměnné pro XGBoost model 42
4.6 Výsledná architektura LSTM sítě 43

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	Rectifed 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 největ-šími 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. klimatic-ké). 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 časovému okamžiku. Před samotným průzkumem prediktivních metod je nutné si definovat několik pojmů z oblasti analýzy časových řád, 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 zmíněna 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 STACIONARIATA

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.

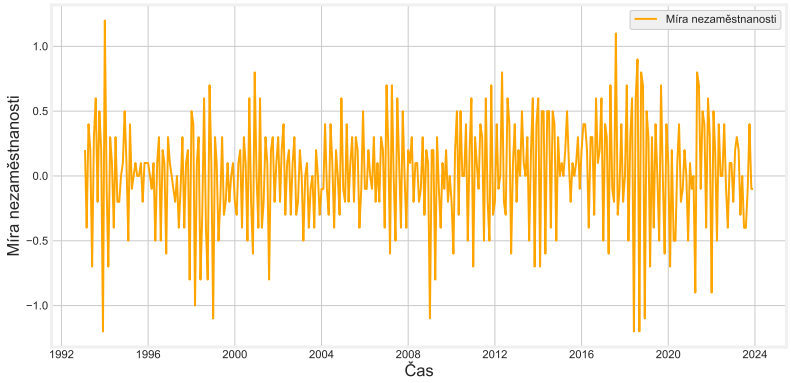
Nejjednodušší transformace řady na stacionární je pomocí diferencování. Tato transformace přispívá ke snížení trendové složky. Například diferenci prvního řádu se zapisuje jako:

$$\Delta x_t = x_t - x_{t-1} \tag{2.1}$$

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 diference 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 diference

Pro otestování zdali je řada stacionární existuje několik testů ze statistické analýzy. V implementační části této práci 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 dekompozice využívají k predikci, obzvlášť 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.

$$x_t = T_t + S_t + C_t + \epsilon_t \tag{2.2}$$

kde x_t je samotná hodnota časová řady v čase t.

2.3.2 Multiplikativní model

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

$$x_t = T_t * S_t * C_t * \epsilon_t \tag{2.3}$$

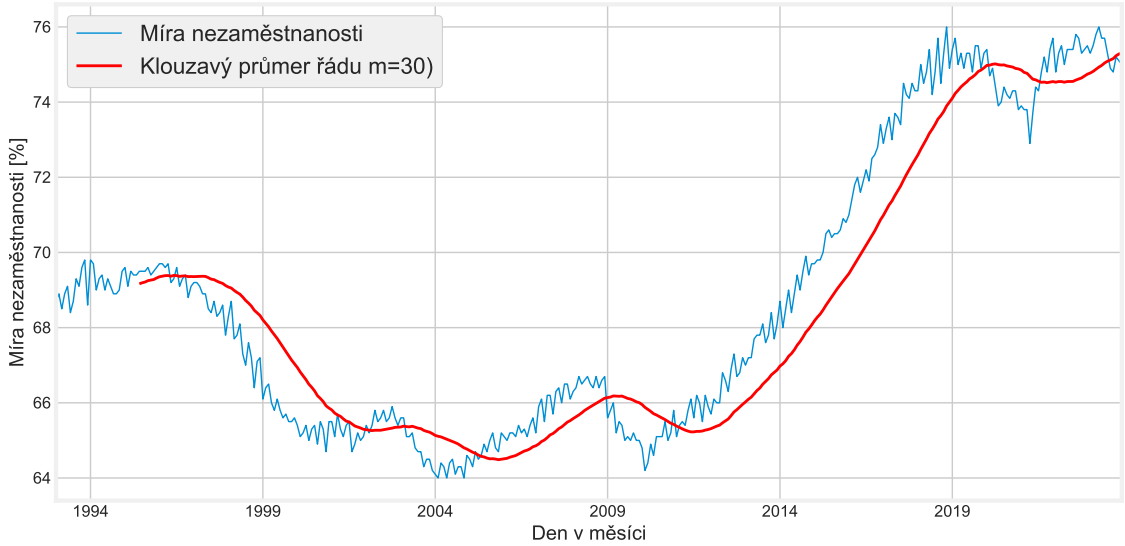
Tento model se využívá, pokud se výkyvy sezonní složky vůči trendu mění v čase.

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:

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

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



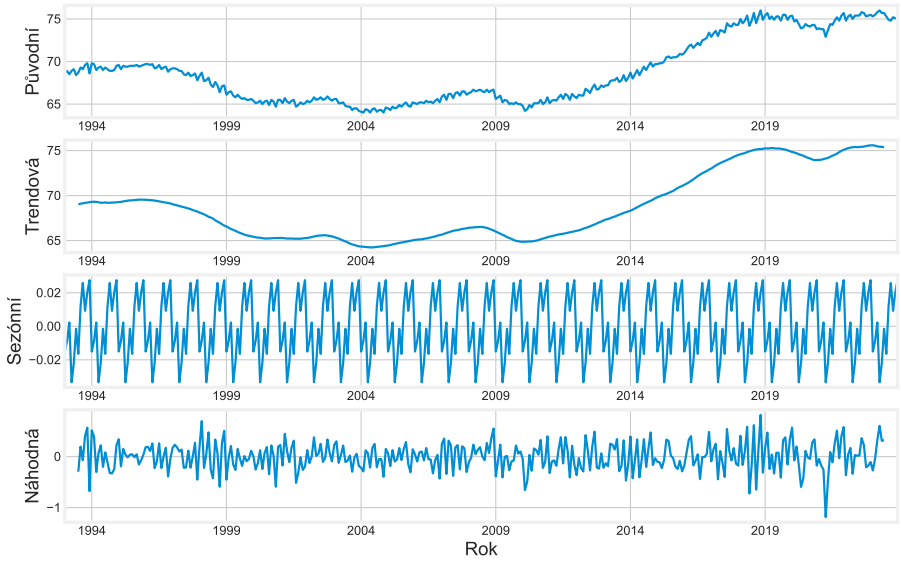
Obrázek 2.3: Klouzavý průměr řádu m=30 popisující míru nezaměstnanosti v ČR od roku 1993.

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 řády, 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 zužitkovat 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 předpovědí spotřeby energie lze často najít modely, které byly vytvořeny pomocí modelování spotřeby energie touto technikou (nebo její podmnožinou). Jeden z důvodu, 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 použít k predikci hodnoty závislé proměnné na základě hodnot nezávislých proměnných.

Regresní model s více než jednou proměnnou se zapisuje jako

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p + \epsilon \tag{3.1}$$

kde y je výstupní proměnná, ϵ chybový člen a β_i, i=0, 1, 2 ... 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. [1]

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

užitečným 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 a nejpoužívanějšími modely pro predikci časových řad. **??** Klíčové prvky modelu spojují; autoregresi AR(p), integraci I(d) a klouzavý průměr MA(q) **??**. Samotná rovnice ARIMA modelu je tedy kombinací stacionarizovaného autoregresního modelu a k tomu přidaný model s klouzavým průměrem.

3.2.1 Autoregresivní model

Autoregresní model lze popsat jako funkci, jenž vrací predikovanou hodnotu x_t na základě svých předchozích hodnot $x_{t-1}, ..., x_{t-p}$. Těmto hodnotám se říká zpožděné (lag) proměnné a slouží ke zvýšení přesnosti autoregresního 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 autoregresní 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:

$$x_t = f(x_{t-1}, x_{t-2}, ..., x_{t-p}, \epsilon_t) \tag{3.2}$$

kde ϵ_t je bílý šum, který platí pro data co mají průměr a variaci dat nulovou. Počet zpožděných proměnných určuje parametr p v AR(p).

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 autoregresní model s váženými parametry α bude vypadat následovně:

$$x_t = \sum_{i=1}^{p} \alpha_i x_{t-i} + \alpha_0 + \epsilon_t \tag{3.3}$$

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

3.2.2 Model klouzavého průměru

Ačkoliv model nese stejný název jako kapitola v dekompozici časových řad, nesmí se tyto dva pojmy zaměnit. V prvním případě se jedná o výpočet pro vyhlazení a získání trend-cyklické komponenty.

V tomto případě se jedná o model, který používá pro předpovězení nové hodnoty x_t předchozí hodnoty z náhodné komponenty $\epsilon_t, \epsilon_{t-1}, ..., \epsilon_{t-q}$. Počet těchto zpožděných hodnot ovlivňuje parametr q v MA(q). Stejně jako u autoregresního modelu i zde se počítá s váhami. Finální rovnice modelu se zapíše jako:

$$x_t = \sum_{i=1}^{q} \alpha_i \epsilon_{t-i} \tag{3.4}$$

3.2.3 ARIMA model

Pro analýzu časových řad se zpravidla vyžaduje stacionarita řady, ARIMA není výjimkou. Tato podmínka vyžaduje tedy stacionarizovat výše zmíněný autoregresní model pomocí diference.

Pokud se spojí tedy AR model po diferenci s modelem klouzavého průměru obdržíme rovnici ARIMA modelu:

$$x_t = \sum_{i=1}^{n} (\alpha_i \Delta x_{t-i}) + \sum_{i=1}^{n} (\alpha_i \epsilon_{t-i}) + \alpha_0 + \epsilon_t \tag{3.5}$$

kde x_t je nová hodnota. Odhad jednotlivých parametrů p a q je jedním z kroků používané Box-Jenkinsovy metody. **??** Tato metoda využívá dvě funkce. Autokorelační funkci využívá pro odhad zpožděných parametrů modelu klouzavého průměru a parciální autokorelační funkci pro odhad parametrů autoregresního modelu. Použití této metody bude ukázáno v praktické části této práce.

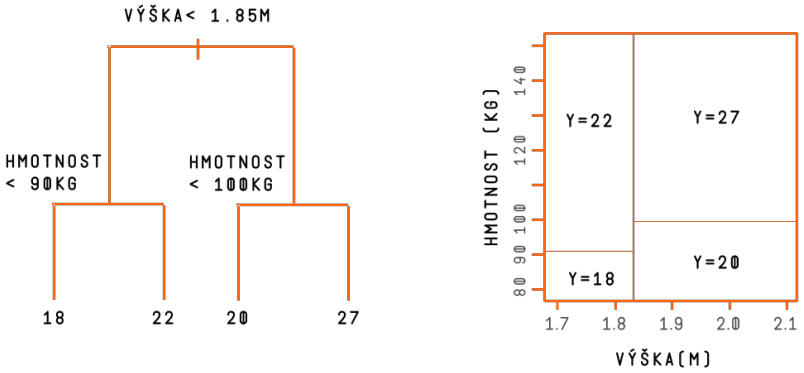
3.3 ROZHODOVACÍ STROMY

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 < 1.85.m” a dále se již větví na základě největšího poměrného *informačního zisku* (založeno na výpočtu entropie). [2]

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. [2]



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*. [3]

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 tuto práci byla zvolena implementace XGBoost v Pythonu pro svou přehlednou dokumentaci a jednoduchosti použití

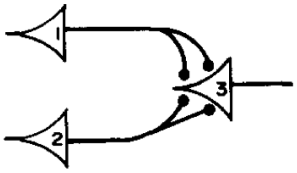
3.4 NEURONOVÉ SÍTĚ

V poslední letech si neuronové sítě vydobyly 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 neurofyziologii 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. [4]

$$y = \begin{cases} 1 & z \geq T \\ 0 & z < T \end{cases} \tag{3.6}$$

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



Obrázek 3.2: Logický součet pomocí biologických neuronů

Neuron označují jako N s indexem i, který je uvnitř neuronu. V tom případě na obrázku výše lze vidět $N_3(t) = N_1(t-1) \vee N_2(t-1)$, což znamená, že se jedná o logický součet. [4] 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. [5]. Aktivace neuronu se pak spočítá následovně:

$$z = \sum_{i=1}^{n} w_x x_i + b \tag{3.7}$$

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) [5] se spočítá jako:

$$Z = H(z) \tag{3.8}$$

kde

$$H(z) = \begin{cases} 1 & f(z) \geq 0 \\ 0 & f(z) < 0 \end{cases} \tag{3.9}$$

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

$$w_{n,i} = w_i + r(y_i - Z_i)x_n \tag{3.10}$$

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

Důležitým poznatkem je 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 nelze vyřešit. To vedlo k rozšíření této oblastí 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 vícevrstvými 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

$$H_i = \sigma(w_{ni} \cdot I_n + b_i) \tag{3.11}$$

$$Y = \sigma(w_{ij} \cdot H + b_{out}) \tag{3.12}$$

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

Ve více vrstvých 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:

$$Relu(z) = max(0, z) \tag{3.13}$$

a používá se především ve skrytých vrstvách. [6] Na výstupní vrstvě je zpravidla funkce softmax, definovaná jako:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \quad pro\ i = 1, 2, \ldots, n \tag{3.14}$$

Pro trénování pak již nelze použít klasický perceptronový algoritmus (viz 3.10). 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).

V kontextu práce by se dalo psát o optimalizačních metodách, chybových funkcích a o tom jak jednotlivé vrstvy fungují, ale to je nad rámec této rešerše. Pro tuto práci je důležité zmínit *long short-term memory networks*, které vycházejí z rekurentních sítí, které budou popsány v další kapitole.

3.4.3 Rekurentní neuronové sítě

Zatím byly zmíněny pouze sítě, kde výstup závisí pouze na aktuálním vstupu. V této práci je vstupem časová řada popisující spotřebu energie, která nezávisí pouze na aktuálním stavu a ke správnému výstupu by bylo vhodné zahrnout data předchozí.

Rekurentní sítě 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. Nejnázornější je ukázat jak se změní výpočet výstupu z neuronu, když se přidají zpětná spojení.

$$h_t = \sigma(w_{hh} h_{t-1} + w_{xh} x_t) \tag{3.15}$$

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}.

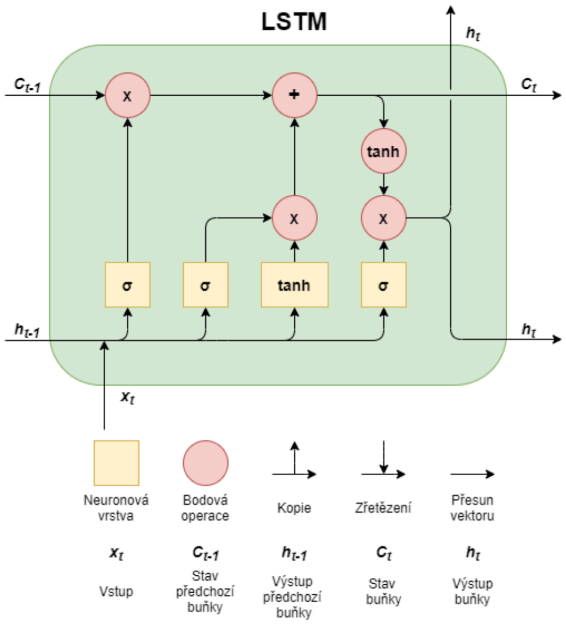
Trénování sítě probíhá opět algoritmem zpětného šíření, avšak lehce modifikovaným. Jmenuje se algoritmus zpětného šíření v čase (backpropagation throguh time). S trénováním rekurentních sítí se můžou vyskytnout dva problémy, díky tomu jak se zpětné šíření počítá. [7] Gradient se vždy násobí váhou. Pokud je toto 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 [8])

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 dá vyhnout 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 autor práce narazil také na kombinaci více sítí v jednu. Autoři jedné z prací zabývající se predikcí spotřeby energie v domácnosti [9] 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. [10]

3.5 KONVOLUČNÍ NEURONOVÉ SÍTĚ

Již z názvu vyplývá, že konvoluční neuronové sítě (CNN) jsou podmnožinou neuronových sítí. Zpravidla se vyznačují využitím konvolučních a poolingových vrstev. Tyto vrstvy umožňují efektivnější zpracování velkých vstupů s menším počtem potřebných parametrů nežli klasická neuronová síť.

Toto je jeden z důvodu, proč je tento typ sítí široce využíván v oblasti počítačového vidění. **??**

V porovnání s klasickými metodami, jako jsou autoregresivní modely nebo regresní metody, mohou konvoluční neuronové sítě poskytovat lepší výsledky při zachycování nelineárních a lokálních vzorů v datech. Tyto vlastnosti mohou být klíčovými faktory pro úspěšnou predikci profilů spotřeby energie.

Typická konvoluční neuronová síť se tedy skládá z několika klíčových prvků. Hlavním stavebním kamenem je operace zvaná konvoluce, ta se využívá v ostatních vrstvách jako jsou konvoluční a poolingové vrstvy, které se starají o extrakci příznaků.

3.5.1 Konvoluční vrstva

V této vrstvě se děje matematická operace zvaná konvoluce. Základem je takzvané konvoluční jádro, což je obyčejná matice o rozměrech m a n. V oblasti počítačového vidění se toto konvoluční jádro postupně přikládá na vstupní obrazovou matici, poté se pronásobí na sobě ležící prvky a jejich suma se zapíše do výstupní matice. Následuje posunutí jádra o definovaný krok a tento proces se opakuje. Konvoluce s jednotkovým krokem se dá tedy zapsat jako:

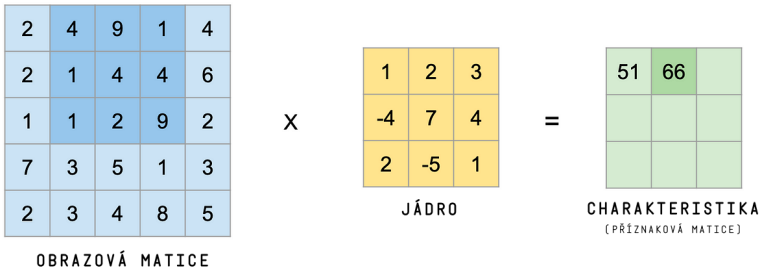
$$y_{i,j} = \sum_{a=1}^{m} \sum_{b=1}^{n} f_{a,b} x_{i+a,j+b} \tag{3.16}$$

kde $f_{i,j}$ jsou prvky konvolučního jádra, $x_{i,j}$ prvky vstupní matice a $y_{i,j}$ prvky výstupní matice. zjednodušeně se konvoluce zapisuje jako:

$$Y = X \otimes F \tag{3.17}$$

kde F je konvoluční jádro, X vstupní matice a Y výstupní matice.

Tato vrstva dokáže extrahovat lokální charakteristiky vstupní matice (obrazu) při schopnosti zachovat informaci o jejich pozici. Mezi tyto extrahované charakteristiky můžou patřit rohy či barvy obrazu.**??**



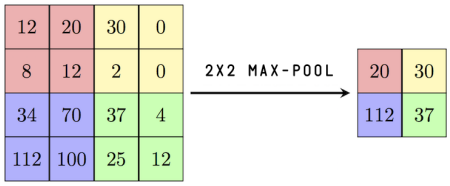
Obrázek 3.4: Ukázka extrakce charakteristiky z obrazu pomocí konvoluce

Opakem konvoluční vrstvy je dekonvoluční vrstva, která provádí inverzní operaci ke konvoluci.

3.5.2 Poolingová vrstva

Tato vrstva je vždy přímo za konvoluční vrstvou a slouží k snížení velikosti vstupů z předchozí vrstvy, což má za následek snížení výpočetní náročnosti (snížením počtu parametrů) a zlepšení distribuce informace.

Nejčastěji lze potkat maxpool vrstvu, která má tři základní parametry. Velikost jádra m a n, krok na ose x h a krok v po ose y. Vrstva vždy vybere maximální hodnotu v daném překrytí obrazu jádrem a vloží její výstup do výstupní matice a posune se dále.



Obrázek 3.5: Ukázka maxpool operace s krokem $h = 2$ a $v = 2$

3.5.3 CNN-LSTM

Pro analýzu časových řad se využívá hybridního modelu, který je spojením konvoluční neuronové sítě a LSTM sítě. Tato metoda se snaží využít silných stránek obou modelů.

Konvoluční vrstvy v síti jsou schopné extrahovat lokální vzorce chování spotřeby energie. V praxi to znamená, že konvoluční vrstvy mohou identifikovat specifické charakteristiky, jako jsou špičky či útlumy, což je klíčové pro porozumění dynamiky spotřeby energie v čase.

Přidáním LSTM sítě lze zachytit na druhou stranu dlouhodobé vzorce chování v časové řadě, díky své rekurentní architektuře. Například mohou úspěšně identifikovat sezónní trendy, které se opakují v pravidelných intervalech, a analyzovat dlouhodobé změny v chování spotřeby, což může být zásadní zejména v oblasti plánování a optimalizace energie.

V předchozích kapitolách bylo uvedeno, že CNN se většinou používá pro 2D obrazová data. Časové řady jsou jednodimenzionální, takže klasická 2D konvoluční vrstva nemůže být využita.

Tomuto problému se dá vyhnout pomocí 1D konvoluční vrstvy, kde se dělá konvoluce pro jednu dimenzi. **??**

Druhý způsob je provést časově frekvenční transformaci signálu. **??** Výsledkem této transformace je rozmístění frekvencí signálu v čase. Zpravidla se jedná o spektogram pomocí Fourierovy transformace a nebo se může jednat o vlnkovou transformaci, kterou zobrazuje scalogram.

4 APLIKACE VYBRANÝCH METOD

V této kapitole bude popsán způsob aplikace metod z rešeršní části této práce. Jednotlivé metody byly vybrány na základě dostupnosti knihoven a očekávaných výsledků.

První zvolená metoda jsou rozhodovací stromy, pro konkrétní implementaci byly zvoleny gradientní boostované stromy, pro které je dostupná knihovna XGBoost. Jako druhou metodou byl zvolen klasický statistický přístup, konkrétně statistický model ARIMA, jež se standardně používá pro predikci časových řad. Další metody jsou postavené na rekurentních neuronových sítích, tedy výše popisovaná LSTM síť a kombinace konvoluční neuronově sítě s LSTM sítí. Zde bude využito bohaté knihovny Tensorflow od společnosti Google

Před samotnou aplikací těchto metod bude kladen důraz na základní datovou analýzu, která má klíčový význam při identifikaci optimálních parametrů pro finální modely. Tato analýza umožní lépe porozumět datům, identifikovat vzorce a trendy ve spotřebě elektrické energie a získat ucelenou představu o nejdůležitějších faktorech ovlivňujících spotřebu.

Výsledkem této kapitoly budou natrénované modely, jež budou využívány k predikci budoucích hodnot spotřeby elektrické energie pomocí vybraných metod. Jako jedno z rozšíření této práce v budoucnu je integrace těchto modelů do informačního systému NEO.

Nakonec budou tyto modely podrobeny důkladnému hodnocení založenému na metrikách přesnosti, které jsou uvedeny v kapitole 4.4. Vyhodnocení na základě těchto metrik bude klíčovým krokem při posuzování celkové úspěšnosti modelů a pomůže identifikovat případné oblasti, které vyžadují zdokonalení. Vyhodnocení povede také k informovanému rozhodování o jejich budoucí integraci v informačním systému NEO.

V rámci implementace byl využit programovací jazyk Python a základní statistické knihovny, jako jsou numpy, pandas a statsmodels, spolu s knihovnami specializovanými pro strojové učení, jako jsou Keras, Scikit-learn a další. Tato kombinace nástrojů poskytla robustní prostředí pro vývoj a testování predikčních modelů.

4.1 ZDROJ DAT

V práci byly využity dva klíčové 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.

Cílem tohoto projektu byla inovace distribuční soustavy v Londýně a zvýšení využívání nízkouhlíkových technologií pro vytápění. V průběhu počáteční fáze projektu byly domácnosti vybaveny chytrými elektroměry, a následně byl spolu s příslušnou studií zveřejněn vzorek dat [11]

Naměřená data jsou dostupná v půlhodinových intervalech a obsahují informace o spotřebě v kWh, unikátní identifikátor domácnosti (LCLid) a časové razítko. Celá datová sada má rozsah přes 10 GB a obsahuje více než 160 milionů záznamů.

Struktura datové sady s energetickou spotřebou v domácnostech v Londýně je detailně popsána následovně:

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

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

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 je nezbytné provést transformaci dat o počasí, aby byla dosažena konzistence a kompatibilita s datovou sadou pro Londýn. Tato transformace spočívá v rozdělení časových údajů o počasí do intervalů po půl hodinách. Bez této transformace by nebylo možné využít datovou sadu pro Londýn v kombinaci s daty o počasí, aniž by se neztratila informace o měřené spotřebě.

V dalším kroku je nezbytné provést analýzu klíčových sloupců datové sady o počasí, aby bylo dosaženo přesnějších výsledků. Předpokládá se, že některé sloupce, například sloupec *icon*, nemají signifikantní vliv na určení spotřeby energie. Tato analýza umožní identifikovat klíčové proměnné, které mají významný dopad na spotřebu energie, a tím zlepšit přesnost modelu.

Druhý zdroj dat, který je využíván, již není veřejně dostupný. Jedná se o komerční data, která jsou získávána společností Albistech s.r.o. Autor práce se jako zaměstnanec podílí na vývoji informačního systému NEO, který umožňuje komplexní sbírání dat získaných při měření v uzlových bodech distribuční soustavy. Systém NEO následně tato data zpracovává a poskytuje užitečné informace a přehledy.

Pro tuto práci byla zvolena lokální distribuční soustava, kterou provozuje zákazník LDEnergy. Náhodně bylo zvoleno několik odběrných míst v Brně a byly dohledány identifikátory příslušných elektroměrů. Všechna získaná data byla anonymizována a obsahují pouze časová razítka a naměřenou spotřebu v kWh s čtvrthodinovým rozlišením.

Tabulka 4.3: Struktura datové sady se spotřebou v domácnosti (Brno)

Sloupec	**Popis**
wstime	Časové razítko ve formátu ddddd,sssss
value	Spotřebovaná energie v kWh

Struktura datové sady se spotřebou v domácnosti v Brně je popsána v tabulce 4.3, kde sloupec ”wstime” reprezentuje takzvaný horolog. Sloupec ”value” udává spotřebovanou energii v kWh.

Pro získání tohoto zdroje dat bylo nutné se seznámit s proprietární databázovou platformou IRIS od společnosti Intersystems, což zahrnuje pokročilejší orientaci v specifických prostředcích této platformy. Tato znalost byla nezbytná pro úspěšné zpracování a interpretaci dat v této práci.

4.1.1 IRIS Data Platform

IRIS je vysoce výkonná multimodelová databáze, která se vyvinula z databáze jménem Caché, vytvořené společností Intersystems. Tato databáze, díky svým unikátním vlastnostem a širokým možnostem integrace, nachází své uplatnění zejména ve zdravotnickém a finančním sektoru, kde je potřeba pracovat s velkými a zároveň různorodými daty.

Klíčovou součástí celého ekosystému IRIS je modul, který se stará o ukládání dat, co mají být persistentně uložena (storage engine). Data jsou uspořádána v multidimenzionálních polích nazývaných ”globály”, které si lze představit jako hierarchickou strukturu ve formě stromu. Globály jsou řídká pole a databáze zde udržuje pouze tolik položek, kolik je prvků. To má za následek výrazně nižší hardwarové nároky oproti systému s relačními databázemi.

Díky skvělé implementaci se dají s globály tvořit komplexní struktury, které mohou čítat několik stovek gigabajtů, a i přesto mít minimální odezvu na čtení a zápis dat. [12] Globály, také podporují transakce, mechanismy pro žurnálování a replikaci dat. Data jsou nepřetržitě k dispozici objektovým přístupem, pomocí SQL či pomocí přímého přístupu do globálu. Její hlavní výhodou na rozdíl od jiných platforem, je ta, že data se uloží jednou a dále se k nim může přistoupit libovolným způsobem, aniž by bylo nutné provádět duplikaci dat nebo například object-to-relational-mapping. [12] Tato flexibilita dává vývojářům a datovým modelářům možnost zvolit optimální přístup nebo kombinaci přístupů, aby byla zajištěna maximální efektivita bez zbytečné komplexity.

K datům uloženým v IRIS je možné přistoupit několika způsoby. Prvním způsobem je použití jazyka Objectscript, který pohání celou datovou platformu IRIS. Tento jazyk umožňuje psát aplikační logiku, kterou následně platforma vystaví prostřednictvím REST API pro komunikaci.

Druhým možným přístupem je využití nativního rozhraní, které IRIS nabízí. V současné době poskytuje společnost InterSystems rozhraní pro Javu, .NET, Node.js a Python.[12] Pro tuto práci hraje klíčovou roli Python rozhraní, které umožňuje vytvářet aplikační logiku přímo v Pythonu nebo volat již existující logiku napsanou v Objectscriptu na straně IRIS.

Pro samotné získání dat k analýze byla v Objectscriptu vytvořena metoda *GetData*, která přijímá identifikátor elektroměru jako parametr a vrací seznam časových razítek a hodnot elektroměrů od počátku zapojení, dle struktury zmíněné v tabulce 4.3. Při implementaci této metody bylo nutné pochopit komplexní strukturu několika globálů, provést iteraci přes globály a vhodným způsobem tato data vrátit.

V Pythonu je poté pomocí nativního rozhraní metoda pro získání dat spuštěna a následně jsou data uložena do JSON souboru s příslušným identifikátorem a časovým razítkem pro další zpracování.

4.2 EXPLORAČNÍ ANALÝZA DAT A JEJICH PŘÍPRAVA

Jeden z mnoha klíčových bodů úspěšného projektu, jenž využívá metody strojového učení, je provést exploratční analýzu dat (dále již jen EDA z Exploratory Data Analysis). Při provádění této analýzy se zkoumají předběžně vlastnosti dané datového sady. Před použitím metod strojového učení je vhodné znát vlastnosti, charakteristiku a jednotlivé závislosti mezi atributy, které budou zkoumány. Porozumění datům dokáže odhalit skryté vzorce, které nejsou na první pohled vidět.

Zpravidla se k analýze používá statistických nástrojů, které analytikům pomáhají vizualizovat závislosti a chování dat. Nejčastěji se k vizualizaci využívají krabicové grafy (box plot), histogramy nebo třeba teplotní mapy (heat mapy).

Při analýze je kladen důraz na snížení dimenze dat či na odhalování atributů, co nesou signifikantnější informaci než ostatní atributy. K tomu bude v následující kapitole sloužit analýza hlavních komponent (Principal Component Analysis, PCA).

Je též důležité věnovat pozornost identifikaci odlehlých hodnot (outlierů) v datech. Odlehlé hodnoty mohou výrazně ovlivnit výsledky analýzy či modelování, a tím měnit celkový obraz datové sady. Proto bude provedena analýza odlehlých hodnot.

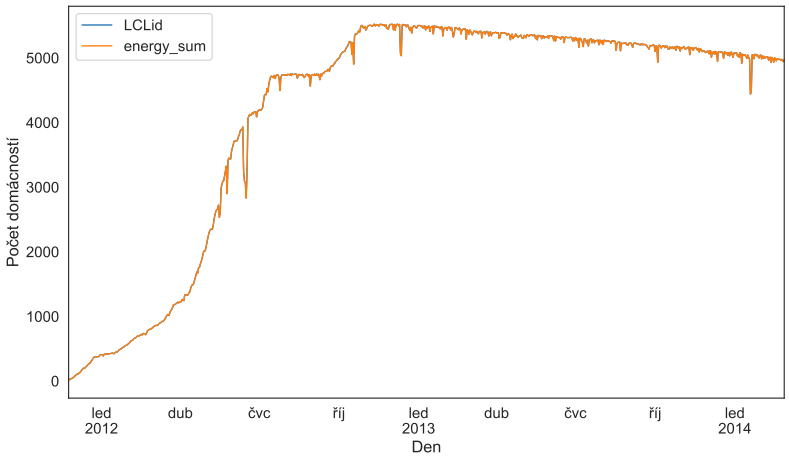
4.2.1 Předzpracování dat

Pro další analýzu či zpracování je nutné mít data očištěna o možné anomálie a výpadky v měření. Pro datovou sadu z Londýna je potřeba vybrat reprezentativní vzorek pro analýzu, jelikož ne všechny domácnosti se zapojily ve stejný čas a proto je vhodné vybrat společný časový úsek pro všechny domácnosti, to pomůže také při analýze jednotlivých komponent časové řady. Datová sada od Albistechu bude použitá v celém jejím objemu.

Pro následující graf tedy byly zvoleny denní průběhy s platnými daty o měření. Data byla seskupena podle dnů v roce a jednotlivé unikátní identifikátory sečteny pro získání počtu domácností v čase. To umožnilo najít nejvhodnější časový úsek společný pro všechny domácnosti.

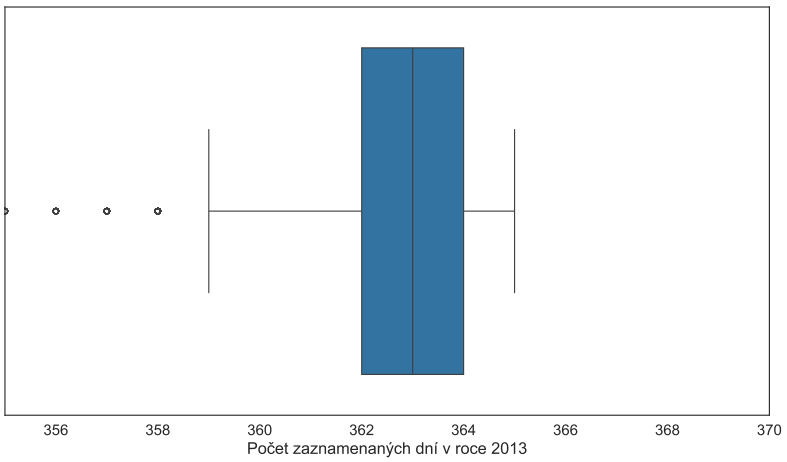
Následně je nutné vybrat pouze ty domácnosti, které obsahují nejvíce záznamu v průběhu nalezeného časového úseku. Pro tento úkol byl vytvořen distribuční graf na základě seskupení dnů v roce a zároveň identifikátoru domácnosti *LCLid*.

Na prvním grafu 4.1 níže je vidět rychlý nárůst zapojených domácností až do roku 2013, kde je maximální počet připojených domácností do projektu. Na základě této informace byl vybrán časový úsek od začátku roku 2013 počínaje lednem až do konce projektu v březnu roku 2014.



Obrázek 4.1: Graf s počtem zapojených domácností v čase

Po seskupení dnů v roce s identifikátorem domácnosti a zobrazení tohoto rozdělení na krabicovém grafu vypadají výsledky následovně:



Obrázek 4.2: Graf rozdělení zaznamenaných dní pro domácnosti

Z analýzy grafu výše vyplývá, že v rozsahu od 357 do 365 zaznamenaných dní dochází statisticky nejvýznamněji. S cílem zachovat kvalitu dat bylo rozhodnuto v datové sadě omezit výběr na domácnosti s alespoň 357 záznamy. Toto kritérium eliminuje 38 záznamů, což představuje pouze 0,68 % z celkového objemu datové sady. Tímto opatřením lze zajistit relevantnější analýzu v rámci sledovaného rozsahu zaznamenaných dní.

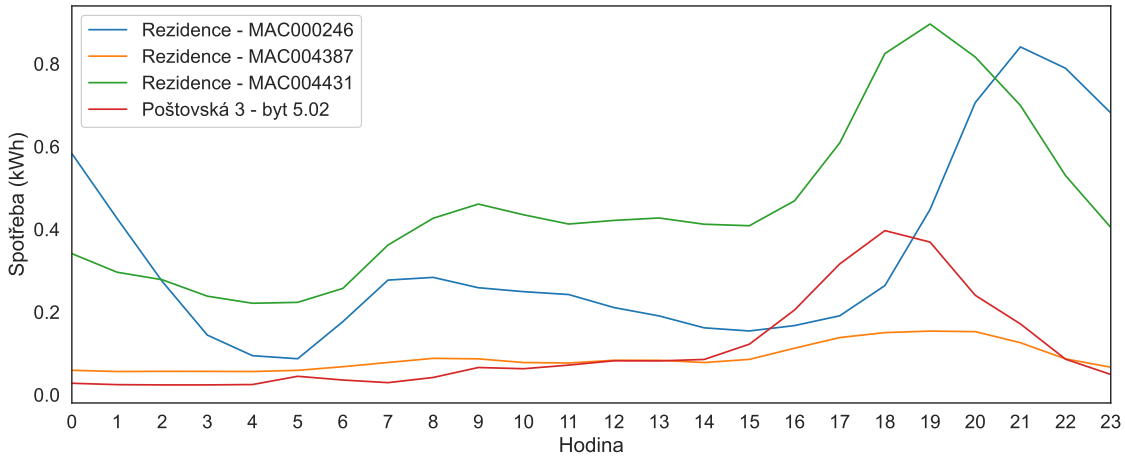
TODO: POPŘEMÝŠLET O OUTLIERECH

4.2.2 Průměrné profily

Jako první byly analyzovány hodinové průměry čtyř domácností. Byly vybrány tři domácnosti z datové sady z Londýna a jedna domácnost z Brna. Výsledky ukázaly, že průměrná spotřeba energie během dne výrazně kolísá mezi jednotlivými domácnostmi. To je způsobeno rozdílným chováním a užíváním spotřebičů obyvateli domácností.

Z grafu 4.3 níže lze vidět, že u rezidence s identifikátorem 246 členové domácnosti začínají vstávat mezi 5-6. hodinou do práce. Tato aktivita se odráží v nárůstu spotřeby energie v těchto časných hodinách. Domu se vracejí kolem šesté, kde se pravděpodobně zapíná více spotřebičů (televize, mikrovlnná trouba, atd.). Podobný vzor lze nalézt i u rezidence 4431.

U zbytků rezidencí lze dle křivky předpokládat, že obyvatelé se vracejí domu kolem 15. hodiny a vstávají mezi 7-9. hodinou.

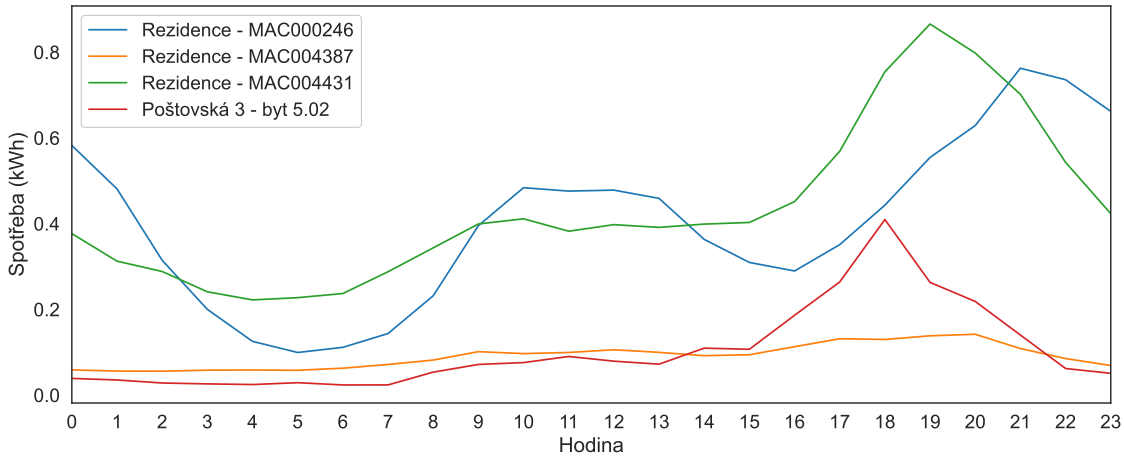


Obrázek 4.3: Hodinová spotřeba vybraných budov v pracovním týdnu

Samotná informace o tom, jak vypadá hodinová spotřeba může být zajímavá v případě nějaké predikce chování uživatele, ovšem pro predikci spotřeby energie to příliš nepomůže. Proto ještě byl analyzován rozdíl mezi hodinovým průběhem pracovního týdne a víkendu.

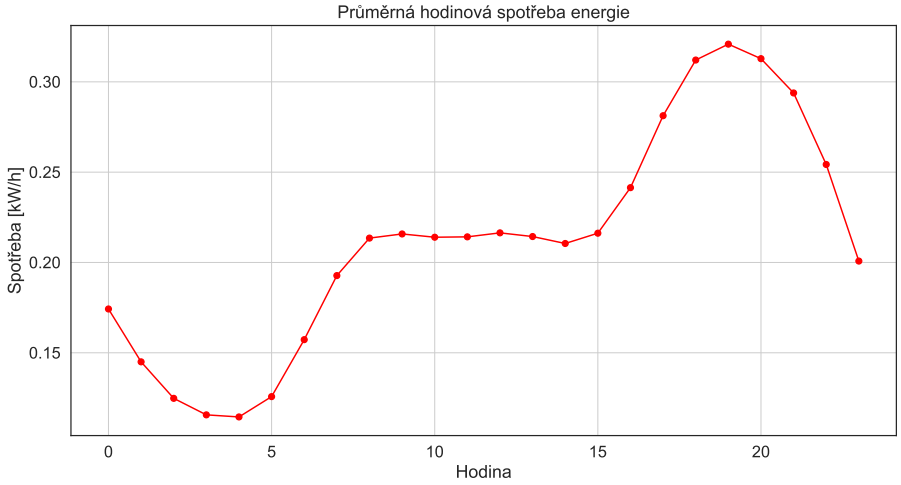
Na grafu 4.4 níže lze zaznamenat signifikantní zvýšení spotřeby v průběhu dne o proti pracovním týdnu (hlavně na rezidenci 246). To se dá vysvětlit tím, že obyvatelé objektu pravděpodobně o víkendu nepracují a tráví raději čas doma.

Tento zjištěný rozdíl v hodinové spotřebě mezi pracovním týdnem a víkendem představuje jeden z klíčových příznaků *is_weekend* pro dále trénované modely. Kombinace více příznaků totiž umožňuje vytvořit sofistikovanější model, který lépe predikuje a přizpůsobuje se dynamice spotřeby energie podle konkrétních situací, což může mít pozitivní dopad na efektivitu energetického řízení dané domácnosti.



Obrázek 4.4: Hodinová spotřeba vybraných budov o víkendu

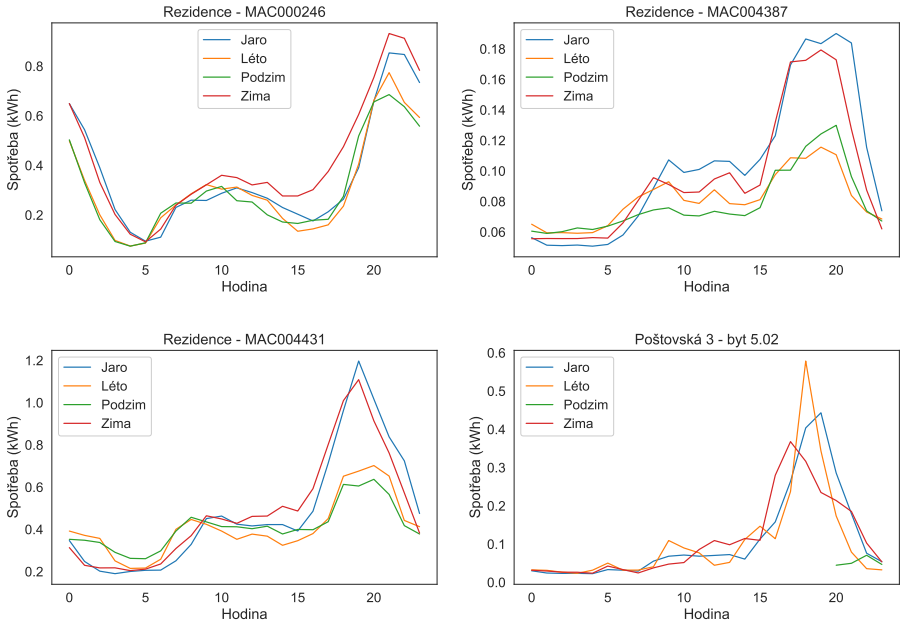
Pro úplnost byla provedena analýza hodinového profilů pro celou 10GB datovou sadu. Z grafu 4.5 je patrné, ve kterých hodinách obyvatelé domácnosti tráví čas doma. Tato analýza potvrzuje výsledky z hodinových průměrů vybraných čtyř domácností, které se od průměru příliš neliší.



Obrázek 4.5: Hodinová spotřeba všech domácností z datové sady

V rámci analýzy hodinových profilů bylo rovněž zkoumáno, zda roční období ovlivňuje spotřebu energie během jednotlivých dní. Výsledky odhalily souvislosti mezi sezónními změnami a spotřebou energie. Během zimních měsíců je zaznamenáván větší nárůst denní spotřeby energie, což může být způsobeno intenzivnějším využíváním vytápění v domácnostech.

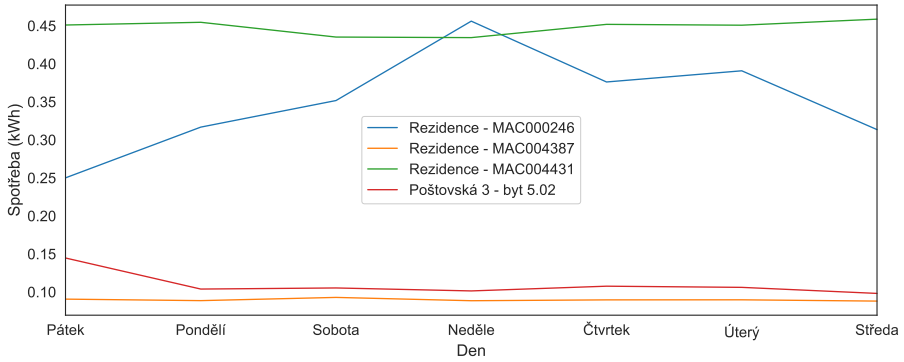
Naopak během letních měsíců je zaznamenáván větší nárůst denní spotřeby energie, což může být způsobeno intenzivnějším využíváním klimatizačních systémů a dalších chladicích zařízení v domácnostech.



Obrázek 4.6: Hodinová spotřeba vybraných domácností dle období

Na základě této provedené analýzy lze zařadit parametr *quarter* jakožto indikátor ročního období mezi další příznaky, jenž bude model pro predikce využívat.

Další ze zkoumaných parametrů je závislost na dni v týdnu. Byl tedy vytvořen průměrný týdenní profil vybraných domácností. Tento profil by mohl obsahovat důležité informace pro lepší pochopení dynamiky spotřeby energie v průběhu týdne.

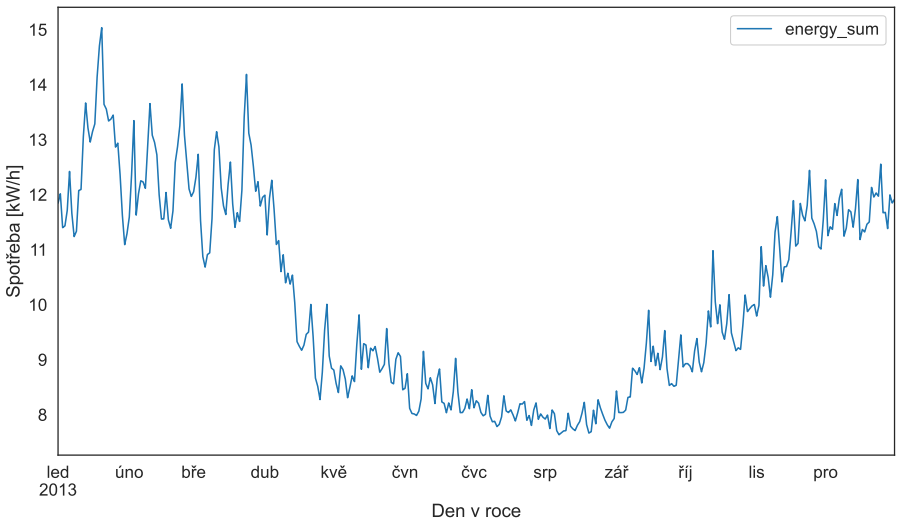


Obrázek 4.7: Denní průměrný profil vybraných domácností v týdnu

Z grafu 4.7 je na první pohled vidět, že někteří obyvatelé rádi tráví čas o víkendech doma (obyvatelé domácnosti 246) a jiní raději tráví čas mimo domov (obyvatelé domácnosti 4431), což vysvětluje snížení či zvýšení spotřeby o víkendu. To znamená, že je to jeden z relevantních indikátoru pro odhadování spotřeby energie. Z tohoto důvodu bude i příznak *week_day* použit v následujících modelech pro predikci.

Poslední příznak z časově závislých parametrů, jenž byl zkoumán je závislost spotřeby na dni v roce. Zde se předpokládá sezonní trend, obdobný jako příznak *quarter*, který vykazoval v zimních a letních měsících variaci oproti zbytku.

Pro každou domácnost v analýze byl tedy proveden denní průměr spotřebované energie přes celý rok 2013.



Obrázek 4.8: Denní průměrný profil vybraných domácností v roce

Z grafu 4.10 se potvrzuje, že zde existuje závislost mezi spotřebovanou energií a dnem v roce. Sezonní efekt lze vidět v měsících, kde je zpravidla větší zima, což by indikovalo, že mnoho lidí využívá elektrický zdroj pro vytápění své domácnosti. Parametr *day* bude zařazen do seznamu příznaků v modelech.

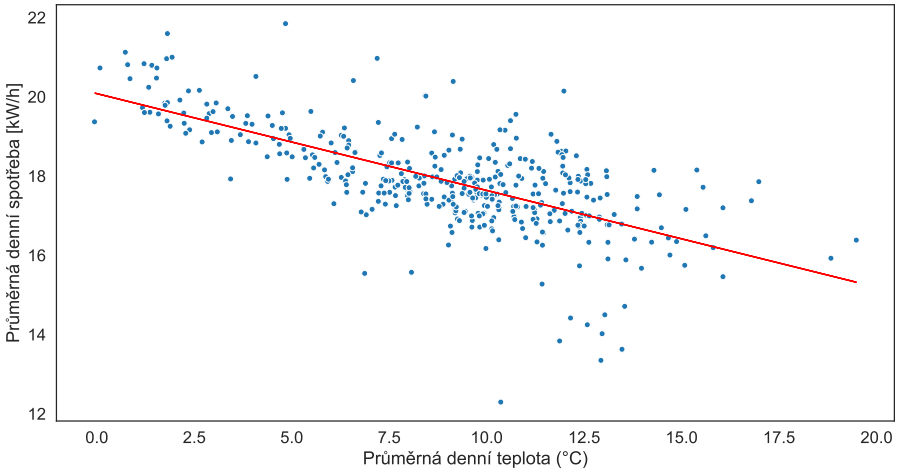
4.2.3 Závislost počasí

V předchozí kapitole byla odkryta závislost na ročním období, proto je vhodné prozkoumat závislosti spotřeby na parametrech z datové sady o počasí popsanou v tabulce 4.4.

K tomu, aby bylo možné provést takovou analýzu, je nutné spojit obě datové sady do jedné. Taktéž musí být vzato v úvahu, že údaje o spotřebě jsou měřeny v půlhodinových intervalech, zatímco meteorologická data jsou dostupná v hodinových intervalech. S cílem zabránit ztrátě informací, bude i počasí interpolováno na půlhodinové intervaly.

Pro tento účel byla využita lineární interpolace, přičemž pokud hodnota bude v datové sadě chybět, doplní poslední změřenou hodnotu teploty.

Analýza byla provedena na celé datové sadě. Byla spočítána průměrná denní teplota a průměrná denní spotřeba energie. Graf zobrazující tento vztah vypadá následovně:



Obrázek 4.9: Vztah mezi průměrnou denní spotřebou a průměrnou denní teplotou

Z grafu lze na první pohled vidět, že zde je trend a to klesající, jinými slovy, se zvyšující se teplotou klesá spotřebovaná energie. To může být způsobeno tím, že objekty v datové sadě využívají elektrického tepelného vytápění. Jelikož z grafu je patrný trend, byl příznak teploty zařazen pro použití ve finálních modelech.

Datová sada s počasím má ovšem mnohem více parametrů, které by mohly být důležité pro predikci. Pro identifikování parametrů, které by měly být zařazeny do finálního modelu, bylo využito výpočtu pro změření velikosti lineární závislosti mezi jednotlivými parametry. Bylo využito Pearsonova korelačního koeficientu. Korelační koeficient se počítá dle následujícího vzorce:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{4.1}$$

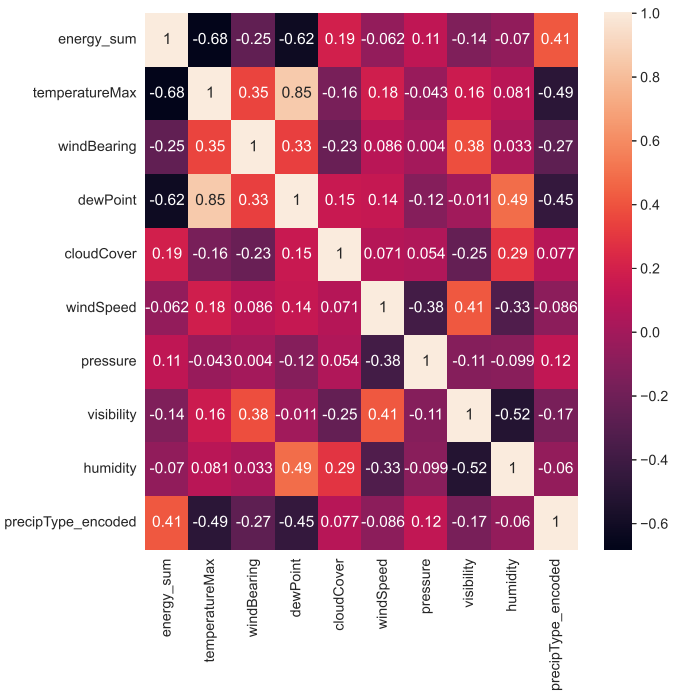
kde ve jmenovateli jsou násobeny směrodatné odchylky obou proměnných a v čitateli se počítá kovariance. Jelikož se zde dělí směrodatnou odchylkou, výsledný koeficient r vyjde jako normované bezrozměrné číslo v rozsahu -1 až 1.

Interpretace hodnoty r spočívá v tom, že čím blíže je hodnota korelačního koeficientu k 1 nebo -1, tím silnější je lineární vztah mezi dvěma proměnnými. Hodnota r blízká 0 naznačuje, že mezi proměnnými není lineární vztah.

Kromě toho, že tento koeficient umožňuje náhled na proměnné, které by mohly být užitečné, tak umožňuje i redukci dimenze vstupních dat pro predikční model. Pomocí korelační matice je totiž možné zobrazit všechny korelační koeficienty pohromadě a rozhodnout, které proměnné v datové sadě jsou redundantní na základě znalosti hodnoty r.

V této práci bylo využito implementace korelačního koeficientu v knihovně *pandas*. Sloupce, které jsou textovém tvaru (např. $precipType$),

byli převedeny na číselné hodnoty 0 až n, kde n je počet unikátních položek. Korelační matice zobrazující vztah mezi spotřebou energie a počasím vypadá takto:



Obrázek 4.10: Ilustrační korelační matice zobrazující vztah mezi spotřebou a počasím

Z matice vyplývá, že *vlhkost* a *rychlost větru* nemají téměř žádnou závislost na spotřebě energie. Naopak proměnná *dewPoint*, neboli rosný bod, vykazovala výraznější vztah ke spotřebě energie, avšak její korelace s teplotou byla příliš vysoká. Proto byla tato proměnná vynechaná z finálního seznamu proměnných, aby se předešlo redundanci a zjednodušila se interpretace výsledků.

Na základě této analýzy byly zvoleny parametry *temperature*, *pressure*, *windBearing*, *precipType* a *visibility*.

Dále byl proveden výpočet korelační matice i pro proměnné, které byly zkoumány v předchozích fázích analýzy (TODO: velká matice do přílohy). Ná základě hodnot koeficientu r vyšlo, že parametr *quarter* je nadbytečný, jelikož silně koreluje s parametrem *month* ($r = 0.98$) a silně také koreluje s parametrem *day* ($r = 0.97$). Proto byl tento parametr ze vstupních dat pro predikční modely odstraněn. Další analýza dat, jako je například autokorelace, bude provedena v následujících kapitolách zabývajících se samotnou implementací modelů.

4.3 IMPLEMENTACE JEDNOTLIVÝCH MODELŮ

Na základě předchozí explorační analýzy byly zvoleny následující společné proměnné pro všechny modely.

Tabulka 4.4: Společné parametry pro všechny predikční modely

Sloupec	Popis
visibility	Maximální viditelnost
windBearing	Směr, ze kterého fouká vítr
temperature	Skutečná teplota ve stupních Celsia
pressure	Aktuální atmosférický tlak měřený v hektopascálech.
precipType	Typ srážky (sníh nebo déšť)
hour	Hodina, kdy byl vytvořen záznam
minute	Minuta, kdy byl vytvořen záznam
month	Měsíc, kdy byl vytvořen záznam
year	Rok, kdy byl vytvořen záznam
energy(kWh/hh)	Spotřebovaná energie v kWh

Tyto proměnné slouží pouze jako základ pro veškeré modely. Některé z modelů využívají navíc ještě uměle vytvořené proměnné (např. zpožděné proměnné u xgboost modelu), pro zpřesnění predikcí.

Výsledné modely jsou serializované v souboru s příponami *.h5* pro LSTM a *.pkl* pro XGBoost. Tato serializace umožňuje uchování modelů pro opakované a přenosné použití bez nutnosti opakovaného trénování.

4.3.1 XGBoost

V kapitole 3.3.1 v rešeršní části práce bylo zmíněno, že xgboost využívá *ensemble* metody, které kombinují slabší modely, jenž jsou na sobě trénovány nezávisle. Z toho plyne, že se zde nevychází z časového uspořádání, což je pro přesnou predikci časové řady klíčové.

Proto bylo nejdříve nezbytné najít řešení pro včlenění časového uspořádání do modelu XGBoost, který standardně nezohledňuje informace o času. Jednou z metod, jak toho dosáhnout, je vytvoření umělých zpožděných proměnných (lag features). Tyto proměnné obsahují informaci o minulých hodnotách cílové proměnné nebo jiných relevantních faktorech v daných časových oknech.

Pro tento účel byl proveden experiment, ve kterém byly postupně vybírány vhodné zpožděné proměnné. Tyto proměnné byly vytvořeny na základě explorační analýzy datové sady. Do modelu byly zahrnuty proměnné, které obsahují informaci o minulých hodnotách vybraných

proměnných v různých časových oknech, což umožňuje modelu lépe zachytit trendy časové řady a vzory v datech.

Výsledná tabulka se zpožděnými proměnnými vypadá následovně:

Tabulka 4.5: Zpožděné proměnné pro XGBoost model

Parametr	Popis
energyMean6	průměrná hodnota spotřeby energie v posledních šesti hodinách
energyMean12	průměrná hodnota spotřeby energie v posledních dvanácti hodinách
energyMean24	průměrná hodnota spotřeby energie za poslední den
energyMax6	maximální hodnota spotřeba energie za posledních 6 hodin
energyMin6	minimální hodnota spotřeby energie za posledních 6 hodin

Trénovací a testovací data byla rozdělena na 80 % trénovacích a zbylých 20 % na testování. Kromě toho existuje měsíc dat, který je kompletně mimo datovou sadu (out-of-sample data).

Volba hyperparametrů pro výsledný model probíhala experimentální metodou. Nakonec došlo k volbě následujících parametrů:

- booster - *gbtree*
- evalmetric - *mse*
- objective - *reg:linear*

TODO: ZBYTEK

4.3.2 Hyper-parametry

říct neco co to vůbec je a proč se to optimalizuje ffs vypínám :(

4.3.3 ARIMA

Ukázat jak byla udělána autokorelace a výběr jednotlivých parametrů....

4.3.4 LSTM

TODO: Začátek a opravit tu debilní tabulku ať se neroztahuje.

TODO: zbytek

4.3.5 CNN-LSTM

TODO: Sepsat architekturu sítě

Tabulka 4.6: Výsledná architektura LSTM sítě

Vrstva	Neurony	Výstupní rozměr	Počet parametrů	Ostatní nastavení
LSTM	50	(None, 1, 50)	19 800	returnSequences=False
Dropout	-	(None, 1, 50)	0	-
LSTM	75	(None, 1, 75)	37 800	returnSequences=False
Dropout	-	(None, 1, 75)	0	-
LSTM	50	(None, 50)	25 200	returnSequences=False
Dropout	-	(None, 50)	0	-
Dense	-	(None, 1)	51	Dense(1)

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).

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2 \tag{4.2}$$

a též směrodatná odchylka pro správné jednotky:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2} \tag{4.3}$$

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

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i| \tag{4.4}$$

a pro vyjádření MAE v procentech:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - y_i}{x_i}\right| \tag{4.5}$$

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
23. 02. 2024 - Konvoluce a max-pool
25.02. 2024 - Seznamy tabulek, obrázků, bibliografie + kapitola zdroj dat a dál už mě to nebaví, je to na githubu ten changelog

POUŽITÁ LITERATURA

1. HANOUSEK, Jan; CHARAMZA, Pavel. *Moderní metody zpracování dat: matematická statistika pro každého*. 1. vyd. Praha: Grada, 1992. ISBN 80-85623-31-5.
2. TOMŠÍK, Jan. *Matematické modelování v energetice [online]*. 2016 [cit. 2023-05-29]. Dostupné také z: https://theses.cz/id/yatg62/. Rigorózní práce. Masarykova univerzita, Přírodovědecká fakultaBrno. SUPERVISOR:
3. CHEN, Tianqi; GUESTRIN, Carlos. XGBoost: A Scalable Tree Boosting System. 2016, s. 785–794. ISBN 9781450342322. Dostupné také z: https://dl.acm.org/doi/10.1145/2939672.2939785.
4. MCCULLOCH, W.S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*. 1943, roč. 5, s. 115–133. Dostupné z DOI: 10.1007/BF02478259.
5. PILÁT, Martin. *Neuronové sítě: Úvod k perceptronu* [online]. n.d. [cit. 28. 05. 2023]. Dostupné z: https://martinpilat.com/cs/prirodou-inspirovane-algoritmy/neuronove-site-uvod.
6. BROWNLEE, Jason. A Gentle Introduction to the Rectified Linear Unit (ReLU). *Machine Learning Mastery* [online]. 2019 [cit. 08. 04. 2023]. Dostupné z: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/.
7. PILÁT, Martin. *Neuronové sítě: RBF sítě, rekurentní sítě* [online]. n.d. [cit. 28. 05. 2023]. Dostupné z: https://martinpilat.com/cs/prirodou-inspirovane-algoritmy/neuronove-site-rbf-site-rekurentni-site.
8. OLAH, Christopher. *Understanding LSTM Networks* [online]. 2015-08. [cit. 28. 05. 2023]. Dostupné z: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.
9. GAUR, K.; SINGH, S. Kumar. CNN-Bi-LSTM Based Household Energy Consumption Prediction. *IEEE Xplore* [online]. 2021 [cit. 28. 05. 2023]. Dostupné z: https://ieeexplore.ieee.org/document/9451797.
10. KYNYCH, František. *Využití neuronových sítí pro automatickou fonetickou transkripci*. 2018. Bakalářská práce.

11. GREATER LONDON AUTHORITY. *Smartmeter Energy Use Data in London Households* [online]. 2015. [cit. 28. 05. 2023]. Dostupné z: https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households.
12. INTERSYSTEMS CORPORATION. *Data Model* [online]. 2020-03. [cit. 28. 05. 2023]. Dostupné z: https://docs.intersystems.com/irislatest/csp/docbook/Doc.View.cls?KEY=PAGE_multimodel.