České vysoké učení technické v Praze
Fakulta strojní
Ústav přístrojové a řídící techniky

Analýza křivek hystereze čerpacího modulu DNOX 5.3

Diplomová práce

Jonáš Cikhart

Magisterský program: Automatizační a přístrojová technika
Magisterský obor: Automatizace a průmyslová informatika
Vedoucí práce: Ing. Adam Peichl

Praha, leden 2022

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Cikhart** Jméno: **Jonáš** Osobní číslo: **467339**

Fakulta/ústav: **Fakulta strojní**

Zadávající katedra/ústav: **Ústav přístrojové a řídicí techniky**

Studijní program: **Automatizační a přístrojová technika**

Specializace: **Automatizace a průmyslová informatika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Analýza křivek hystereze čerpacího modulu DNOX 5.3

Název diplomové práce anglicky:

Analysis of the hysteresis curves of the pump module DNOX 5.3

Pokyny pro vypracování:

1. Popište čerpací modul a postup měření hystereze
2. Proveďte rešerši na téma shluková analýza
3. Implementujte alespoň 3 algoritmy (K-means a další dva) a aplikujte na naměřené křivky hystereze
4. Interpretujte výsledky a porovnejte implementované algoritmy

Seznam doporučené literatury:

[1] XU, Rui; WUNSCH, Donald. Survey of clustering algorithms. IEEE Transactions on neural networks, 2005, 16.3: 645-678.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Adam Peichl U12110.3

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **27.10.2022** Termín odevzdání diplomové práce: **26.01.2023**

Platnost zadání diplomové práce: _____

Ing. Adam Peichl
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

doc. Ing. Miroslav Španiel, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studenta

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Adama Peichla a s použitím literatury uvedené v přiloženém seznamu.

V Praze dne 26.01.2023 ……..

Bc. Jonáš Cikhart

Poděkování

Tímto bych rád poděkoval vedoucímu mé diplomové práce Ing. Adamovi Peichlovi za jeho konzultace, podněty a rady, které mi věnoval a které mi značně pomohly k vypracování této diplomové práce. Dále bych rád poděkoval mé rodině za podporu.

Abstrakt

Abstrakt

Tato diplomová práce se zabývá analýzou křivkových dat hystereze membránového čerpacího modulu. V úvodní části se zaměřuje na popis čerpacího modulu včetně propojení dostupných dat s fyzikální podstatou procesu. Na tuto část je navázáno rešerší na téma předzpracování dat, která je zaměřena především na redukci dimenzí pomocí analýzy hlavních komponent a autoenkodéru. Následuje rešerše v oblasti metod shlukování, ve které jsou představeny tři různé algoritmy. Poznatky získané touto rešerší jsou průběžně použity v praktické části, ve které jsou původní data předzpracována a následně použita v představených metodách shlukování. V poslední části této diplomové práce jsou prezentovány výsledky metod shlukování, po kterých následuje celkové zhodnocení.

Klíčová slova: křivková data, shlukování, datová analýza, Python

Abstract

This Master's thesis deals with the analysis of curve data of the hysteresis of a membrane pumping module. In the introductory part, it deals with the description of the pumping module including the connection of the available data with the physical nature of the process. This section is followed by research on the topic of pre-processing, which is mainly focused on dimension reduction through principal component analysis and autoencoder. This is followed by research on clustering methods in which three different algorithms are presented. The knowledge gained from this search is continuously applied in the practical part, in which the original data is preprocessed and then used in the presented clustering methods. In the last part of the thesis, the results of the clustering methods are presented, followed by an overall evaluation.

Key words: curve data, clustering, data analysis, Python

Seznam obrázků

Seznam tabulek

Seznam zkratek

NOx	oxidy dusíku
OK	označení dílu, který byl označen jako vhodný pro další kroky ve výrobě
NOK	označení dílu, který byl označen jako zmetek
PCA	analýza hlavních komponent
AE	autoenkodér

Obsah

Kapitola 1

Úvod

Automobilový průmysl je i přes rozšiřující se popularnost elektromobilů stále jedním z celosvětově nejrozšířenějších odvětví průmyslu. Zpřísňující se ekologické normy na motorová vozidla s sebou nesou potřebu neustálého vylepšování dostupných technologií, které se tímto problémem zabývají. Konkrétně obsah spalin ve výfukových plynech je jednou z vlastností, která je velmi často posuzovaná i širokou veřejností.

Z toho důvodu byl vynalezen produkt Denoxtronic, jehož cílem je snižovat obsah oxidů dusíku ve výfukových plynech v osobních, užitkových a nákladních vozidlech. Výroba takového produktu je velmi komplexní záležitostí, která se skládá z velkého množství dílčích procesů. Jedním z nich je i výroba čerpacího modulu, který je z mého pohledu jednou z nejdůležitějších částí celého produktu. Zajištění konstantní kvality produktu pro zákazníka je dnes naprostým standardem. To vytváří prostor pro experimentování s novými metodami, které mohou tento proces usnadnit.

Mezi tyto metody patří i analýza dat, která se stává stále důležitějším aspektem moderního průmyslu. Velmi často přidává nový pohled do konkrétní problematiky, který usnadňuje její pochopení, případně napomáhá rozhodovacímu procesu, který problematiku řeší. Nejčastější se s datovou analýzou člověk setkává ve formě vizualizace. Té ovšem standardně předchází analýza různé formy dat, ať už například skalárních, křivkových, kategorických nebo jejich kombinace. Právě křivková data nacházejí stále větší uplatnění, jelikož jejich reprezentace závislosti dvou nebo více veličin v čase umožňuje hlubší proniknutí do problematiky.

Cílem této diplomové práce je analýza doposud málo využívaných hysterezních křivek komponenty čerpacího modulu, které jsou dostupné již před jeho samotnou výrobou. Tato analýza by měla více přiblížit proces uvnitř čerpacího modulu z fyzikálního i matematického pohledu a potvrdit nebo vyvrátit hypotézu, zda lze na základě těchto křivek najít bližší závislost výsledku dostupného přímo z výroby čerpacího modulu. Dalším přínosem je rešerše dostupných metod, které budou s velkou pravděpodobností použity pro analýzy dalších křivek, které se ve výrobních procesech čerpacího modulu používají.

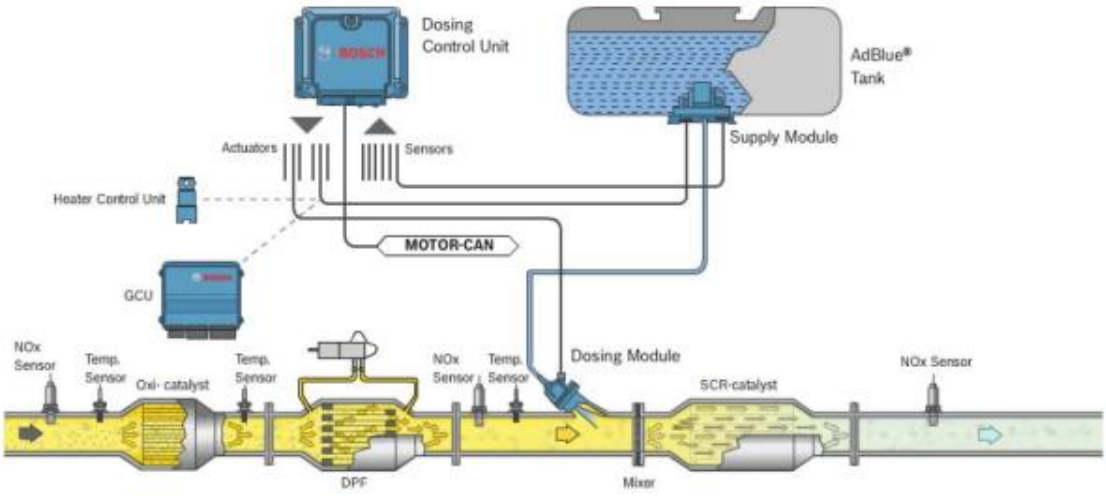
Má osobní motivace pro tuto práci je možnost uplatnit získané vědomosti ohledně fyzikálních procesů společně s programováním v jazyce Python, ve kterém chci navázat na základy, které jsem mohl během navazujícího magisterského studia získat. Velkou motivací mi byla také skutečnost, že se jedná o konkrétní a reálný problém, k jehož vyřešení by tato diplomová práce mohla výrazně přispět.

Kapitola 2

Systém Denoxtronic

Čerpací modul DNOX 5 byl vyvinut z důvodu neustále se zpřísňující emisní legislativy v Evropské Unii pro dodržování obsahu spalin ve výfukových plynech. Aktuálně je nutné, aby veškerá vozidla měla určitou formu aktivního čištění výfukových plynů.

Čištění probíhá vstřikováním směsi AdBlue, což je pracovní název pro směs 32,5% močoviny ve vodě, do výfukového proudu před katalyzátorem SCR. Močovina se pomocí termolýzy a hydrolýzy převede na amoniak, který uvnitř katalyzátoru SCR reaguje s oxidy dusíky, které redukuje na samostatnou vodu a dusík. Systém čištění s použitím Denoxtronic je znázorněn na obrázku 2.1. Surové emise NOx moderních dieslových motorů jsou oproti typickému dieselovému motoru z roku 1990 přibližně o 96 % nižší. Systém Denoxtronic dále snižuje tyto surové emise o dalších 95 %. Ukázka vývoje evropských norem pro osobní automobily [1] je na obrázku 2.2.



Obrázek 2.1 Systém čištění výfukových spalin s použitím Denoxtronic [2]

Uvnitř komponenty *Supply Module (napájecí modul)*, která se nachází v pravé horní části obrázku 2.1 je čerpací modul, který je klíčovým prvkem této diplomové práce a bude blíže představen na straně 4.

Čistící proces se dá rozdělit do čtyř kroků, které jsou popsány následujícími chemickými rovnicemi [2]:

1) Oxidace výfukových plynů

$$2NO + O_2 \rightarrow 2NO_2 \quad (2.1)$$
$$2CO + O_2 \rightarrow 2CO_2 \quad (2.2)$$
$$4HC + 3O_2 \rightarrow 2CO_2 + H_2O \quad (2.3)$$

2) AdBlue hydrolýza

$$CO(NH_2)_2 + H_2O \rightarrow 2NH_3 + CO_2 \quad (2.4)$$

3) Selektivní redukce

$$8NH_3 + 6NO_2 \rightarrow 7N_2 + 12H_2O \quad (2.5)$$
$$4NH_3 + 4NO + O_2 \rightarrow 4N_2 + 6H_2O \quad (2.6)$$
$$2NH_3 + NO + NO_2 \rightarrow 2N_2 + 3H_2O \quad (2.7)$$

4) Oxidace

$$4NH_3 + 3O_2 \rightarrow 2N_2 + 6H_2O \quad (2.8)$$

EU emission standards for passenger cars (Category M_1*)

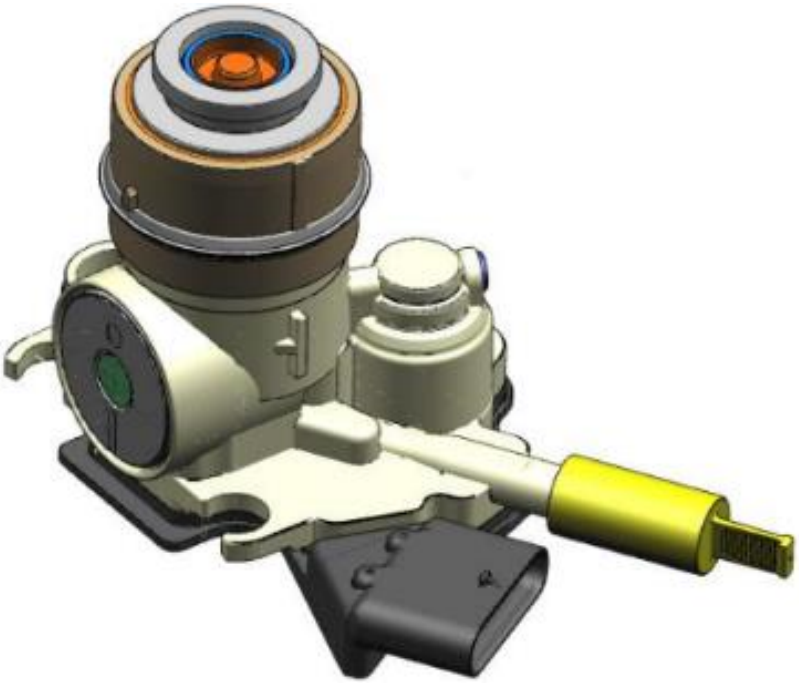
Stage	Date	CO	HC	HC+NOx	NOx	PM	PN
		g/km					*#/km*
Positive Ignition (Gasoline)							
Euro 1†	1992.07	2.72 (3.16)	-	0.97 (1.13)	-	-	-
Euro 2	1996.01	2.2	-	0.5	-	-	-
Euro 3	2000.01	2.30	0.20	-	0.15	-	-
Euro 4	2005.01	1.0	0.10	-	0.08	-	-
Euro 5	2009.09[b]	1.0	0.10[d]	-	0.06	0.005[e,f]	-
Euro 6	2014.09	1.0	0.10[d]	-	0.06	0.005[e,f]	6.0×10^{11} [e,g]
Compression Ignition (Diesel)							
Euro 1†	1992.07	2.72 (3.16)	-	0.97 (1.13)	-	0.14 (0.18)	-
Euro 2, IDI	1996.01	1.0	-	0.7	-	0.08	-
Euro 2, DI	1996.01[a]	1.0	-	0.9	-	0.10	-
Euro 3	2000.01	0.64	-	0.56	0.50	0.05	-
Euro 4	2005.01	0.50	-	0.30	0.25	0.025	-
Euro 5a	2009.09[b]	0.50	-	0.23	0.18	0.005[f]	-
Euro 5b	2011.09[c]	0.50	-	0.23	0.18	0.005[f]	6.0×10^{11}
Euro 6	2014.09	0.50	-	0.17	0.08	0.005[f]	6.0×10^{11}

* At the Euro 1..4 stages, passenger vehicles > 2,500 kg were type approved as Category N_1 vehicles
† Values in brackets are conformity of production (COP) limits
a. until 1999.09.30 (after that date DI engines must meet the IDI limits)
b. 2011.01 for all models
c. 2013.01 for all models
d. and NMHC = 0.068 g/km
e. applicable only to vehicles using DI engines
f. 0.0045 g/km using the PMP measurement procedure
g. 6.0×10^{12} 1/km within first three years from Euro 6 effective dates

Obrázek 2.2 Změny evropských norem pro emise [13]

2.1. Čerpací modul

Čerpací modul je základní součástí napájecího modulu, který hraje hlavní roli v provozu systému. Jeho primární funkcí je dopravovat AdBlue z nádrže do dávkovacího modulu, zároveň nasávat nevyužité AdBlue zpět do nádrže, když je motor vozidla vypnutý. Zpětný tok do nádrže je nezbytný, aby nedocházelo k zamrzání média v systému. Ke zmíněným funkcím využívá čerpací modul dvou čerpadel – hlavní a zpětné.

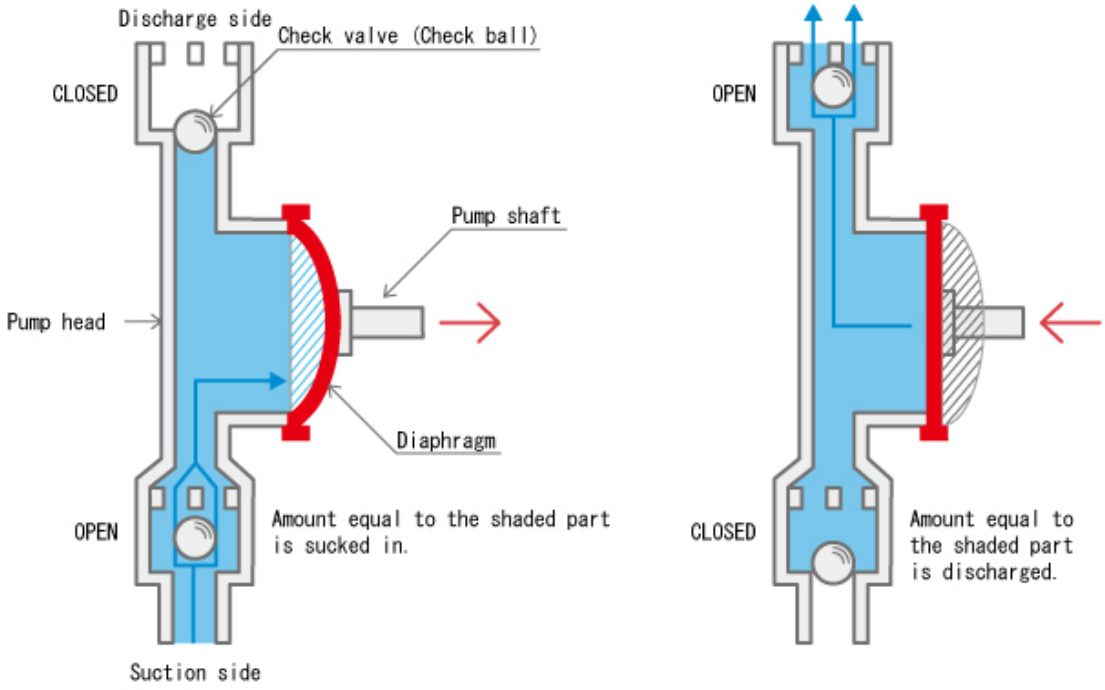


Obrázek 2.3 Čerpací modul DNOX 5.X

Hlavní čerpadlo

Hlavní čerpadlo je membránové čerpadlo ovládané přes solenoid, který využívá elektromagnetické pole pro ovládání polohy kotvy, která je mechanicky spojena s membránou čerpadla. Směr průtoku je zajištěn dvojicí protisměrných ventilů, u kterých je zajištěno, že bude v jeden moment vždy otevřen pouze jeden.

Na obrázku 2.4 je možné vidět schéma mechanické části procesu pumpování. Levá část obrázku naznačuje nasávání a čárkovaná oblast reprezentuje objem, který se do čerpadla nasaje. Pravá část obrázku pak ukazuje vytlačování media a čárkovaná oblast reprezentuje objem, který je z čerpadla vytlačen.



Obrázek 2.4 Schéma membránového čerpadla [14]

Zpětné čerpadlo

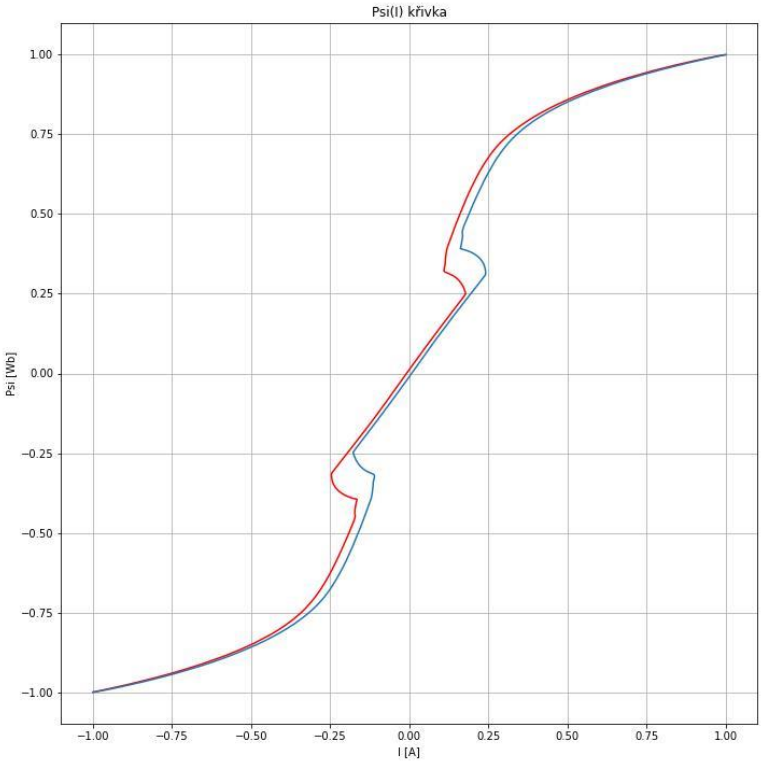
Zpětné čerpadlo je stejně jako hlavní čerpadlo solenoidové. Zajišťuje, aby při vypnutí motoru vozidla nezůstávalo žádné nevyužité AdBlue mimo nádrž systému. Toto je důležité zajistit z důvodu možného zamrznutí, což by mělo za následek zanesení systému. Dle zdroje [1] k tomuto dochází při teplotách nižších než -11 °C.

Kapitola 3

Použitá data a jejich předzpracování

Data používaná k další analýze se získávají přímo od výrobce magnetické části solenoidových čerpadel. Zaznamenává se závislost magnetického toku na proudu. Součástí surových dat je i prvotního zmagnetizování, které ovšem není pro další analýzu důležité. Z toho důvodu se proto budu zaměřovat jen na křivku pracovního cyklu.

V následujících odstavcích budu používat pojem spodní a horní křivka, respektive část křivky. Na obrázku 3.1 jsem horní křivku označil červenou barvou, spodní část barvou modrou. Toto rozdělení plyne z fyzikální podstaty procesu, kde horní křivka pro 1. kvadrant obrázku odpovídá nasávání, spodní křivka vytlačování media z čerpadla. Třetí kvadrant je principiálně identický, pouze označení křivek je zde naopak. Více toto rozdělení bude představeno v kapitole 3.1.



Obrázek 3.1 Ukázka obecné hysterezní křivky

Surová data každé křivky jsou z měřícího stroje výrobce ukládány do textových dokumentů, kde jsou jednotlivé části křivky a veličin rozděleny do 4 samostatných sloupců – proud a magnetický tok horní části a spodní části křivky. Jeden řádek odpovídá jednomu datovému vzorku. Spodní a horní křivky mají rozdílný počet vzorků, který není konzistentní přes celý data set. Současně v datech můžou chybět datové body, nebo se na konci datového souboru můžou objevit nulové hodnoty. To může být zapříčiněno chybou měřícího zařízení dodavatele. V případě přehlédnutí tohoto problému může dojít ke špatnému načtení křivek, což bude mít za následek špatné spárování datových bodů a povede k zavádějícím výsledkům. Na obrázku 3.1 je ukázána obecná křivka, na jejíž segmentech bude vysvětlena funkcionalita pumpy.

Rozsah obou os této křivky byl změněn z původních hodnot na rozmezí od -1 do +1 pomocí algoritmu MinMaxScaler [15] z důvodu zachování firemního tajemství výrobce. Toto škálování ovšem není nežádoucí, vzhledem k tomu, že se jedná o velmi častou metodu předzpracování dat v datové analýze. Škálování dat může být užitečné z mnoha důvodů. Jedním může být zefektivnění některých algoritmů strojového učení, které používají určitou formu měření vzdálenosti k porovnávání vzorků. Pokud nejsou data škálována, může dojít k upřednostnění určitých charakteristik na úkor jiných, což zhoršuje celkovou schopnost modelu. Dalším důvodem je omezení negativního dopadu odlehlých hodnot v původních datech. Tyto hodnoty mohou mít neúměrný vliv na schopnost modelu a škálování tomuto může výrazně pomoci.

Celkem je k dispozici 248 souborů se surovými daty křivek. Každý soubor má specifický identifikátor, pomocí kterého je možné ho spojit s unikátním DMC kódem čerpadla a dohledat konkrétní výrobní parametr, který reprezentuje funkčnost čerpadla v čerpacím modulu pro další použití. Dále jsou k dispozici informace o tom, kdy a na jaké stanici byl díl měřen a zda na stanici vyšel jako OK, nebo NOK. Důležité je také je zmínit, že tento parametr byl měřen na šesti různých strojích, které jsou na měřící lince k dispozici. Tyto stroje mají nezpochybnitelný vliv na výslednou hodnotu měřeného parametru, a proto je při další analýze důležité rozdělit původní data do podskupin, podle stroje, na kterém došlo k měření, aby byla analýza vhodná pro vyvození smysluplných závěrů a nedošlo k chybnému vyhodnocení.

3.1. Popis dat

V této podkapitole přiblížím části křivky zobrazené na obrázku 3.1. Vysvětlím, co určité části křivky znamenají a co konkrétně se děje uvnitř čerpadla. Díky hlubšímu pochopení dostupných křivek a fyzikálního procesu se budu moci přesněji zaměřit na části křivek, které mají větší význam a díky tomu upravit následnou analýzu tak, aby shlukování právě těchto nejdůležitějších částí bylo co nejlepší.

Proces v čerpadle je cyklický, tudíž není možné jednoznačně určit začátek, ale lze rozdělit do dvou hlavní částí – nasávání a vytlačování média čerpadlem. Na obrázku 3.1 je možné vidět dva pracovní cykly, které lze rozlišit kladnými a zápornými hodnotami proudu. Vytlačování média pro kladné hodnoty proudu odpovídá modré křivce, nasávání červené křivce a pro záporné hodnoty proudu je označení opačné.

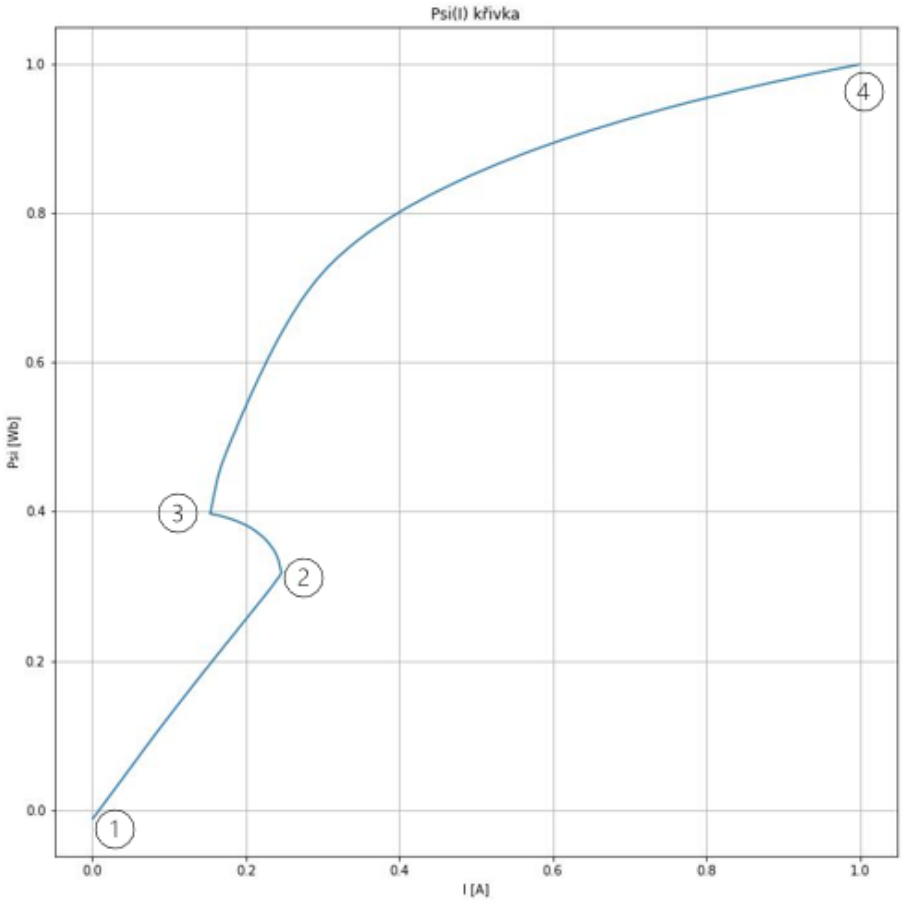
Důležité je ovšem zmínit, že tyto křivky jsou z měření dodavatelem. Při reálném provozu je čerpadlo napájeno pouze kladnými hodnotami proudu a do levé spodní části obrázku 3.1 se tedy proces nedostane.

Vytlačování média

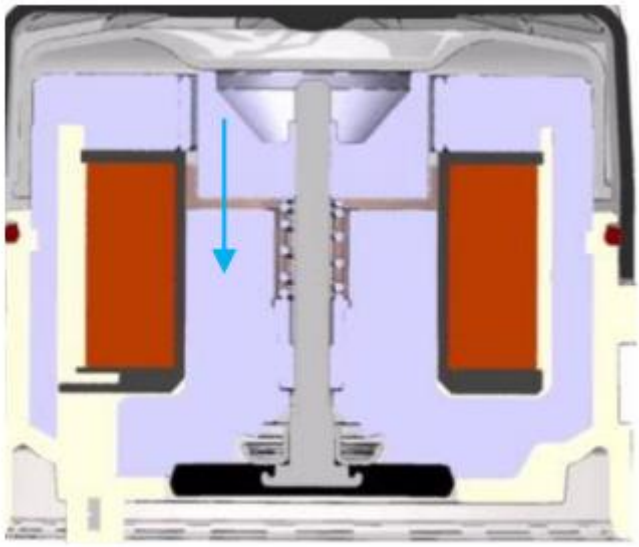
Na obrázku 3.2 je výsek hysterezní křivky při vytlačování média z pracovní komory a lze si zde všimnout několika důležitých vlastností. Pokud začneme v bodě 1 a pohybujeme se po křivce z leva doprava, dochází v této části k napájení magnetu. To znamená, že se v magnetu akumuluje energie ve formě elektromagnetického pole. Mezi hodnotami proudu 0,05A a 0,2A je velmi důležitý sklon křivky. Tento sklon nám popisuje dynamiku magnetu. Je to míra, jak efektivně se převádí elektrická energie do elektromagnetického pole. Čím je křivka strmější, tím dynamičtější je sestava – potřebuje méně času k naakumulování dostatku energie pro další stav.

V bodě 2 dojde ke změně, protože se v tomto momentu v cívce naakumulovalo dostatek energie pro to, aby byla překonána síla pružiny a aparát se rozpohyboval. V tomto momentu se energie cívky začíná přeměňovat na mechanický pohyb, což má za následek vytlačení média pryč. Tento stav trvá až do bodu 3, kdy membrána s kotvou dosáhne koncové polohy a zastaví se.

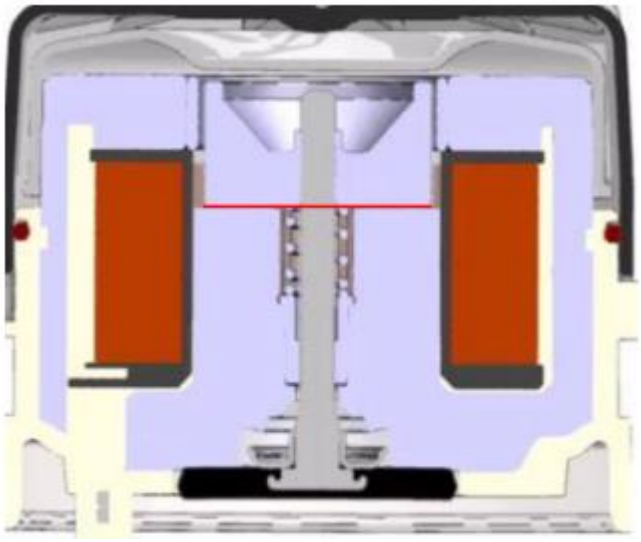
Mezi bodem 3 a 4 se nachází oblast exponenciálního růstu proudu až do hodnoty proudu 1 A, což je bod blízký saturaci cívky, a začne další pracovní krok. Dění při bodu 2 je ukázáno i na obrázku 3.3, kde je naznačen začátek mechanického pohybu čerpadla, který je popsán v předchozích odstavcích. Stejně tomu tak je pro bod 3 na obrázku 3.4, který ukazuje, jak stav vypadá při dojezdu membrány s kotvou na doraz.



Obrázek 3.2 Část spodní křivky



Obrázek 3.3 Začátek mechanického pohybu čerpadla



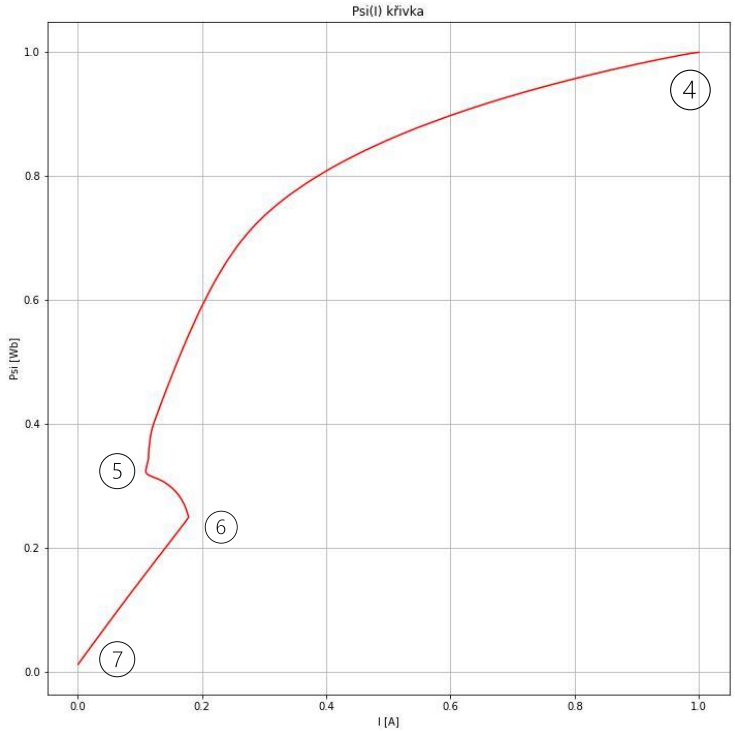
Obrázek 3.4 Konec mechanického pohybu čerpadla

Nasávání média

Na obrázku 3.5 je ukázán výsek křivky pro nasávání média, který navazuje na úsek z obrázku 3.2. Po bodu 4 dochází k úbytku proudu procházejícím cívkou. A to až do bodu 5.

V bodě 5 je síla pružiny dostatečná pro začátek pohybu a vrácení membrány s kotvou do původní polohy, která je reprezentována bodem 6. Tím dochází k nasání média do pracovní komory.

Mezi bodem 6 a 7 dochází k vybíjení naakumulované energie cívky a při dosažení nulových hodnot následuje nový pracovní cyklus.



Obrázek 3.5 Část horní křivky

3.2. Předzpracování dat

Předzpracování dat je jedním z prvních a zároveň nejdůležitějších kroků datové analýzy. Je to proces přípravy surových nezpracovaných dat pro analýzu za pomocí čištění, formátovaní, transformací a dalších metod, pro snadnější a efektivnější analýzu v pozdějších krocích.

Čištění dat je část předzpracování, která se zaměřuje na identifikaci a odstranění obecných chyb v datech. Do těch spadají chybějící záznamy v surových datech. Tento problém je velmi zásadním a jeho přehlédnutí může vést ke špatným výsledkům, nebo nestabilitě použitých algoritmů, které na chybějící data nemusí být připraveny. Dalším problémem mohou být chybějící, duplicitní, nebo již dříve zmíněná odlehlá data. Ty dělají analyzovaný datový set nekonzistentní a také mohou vést k nespolehlivým výsledkům.

Formátování dat je krokem, který nám usnadní následující práci s připravenými daty. To zahrnuje úkoly, jako je převod dat z textových souborů do formátu, se kterým lze následně snadno analyzovat a který je kompatibilní s používaným analytickým nástrojem. Zároveň s tím souvisí uložení ať už původních nebo částečně upravených dat do formátu, který usnadňuje opakované použití pro různé analytické metody.

Transformace dat je téma, do kterého spadá velká škála metod různých složitostí a jejich volba velmi záleží na konkrétním využití. Já se zaměřím konkrétně na dvě metody pro snížení počtu datových bodů, které v praktické části této diplomové práce použiji – analýza hlavních komponent a autoenkodér. Obě tyto metody se používají při práci s většími datovými sadami, což může být výpočetně a časově náročné, ale často i nevhodné pro určitý typ algoritmů. Jejich cílem je převzorkování datové sady, nebo snížení počtu datových bodů způsobem, který zachová co nejvíce možných informací.

Analýza hlavních komponent

Analýza hlavních komponent (PCA – principal component analysis) [3] je technika používaná ke snížení dimenze dat při zachování co největšího množství informací. Jedná se o široce používanou techniku v oblasti strojového učení, počítačového vidění a rozpoznávání vzorů, která je užitečná zejména při práci s velkými a složitými soubory dat. Základní myšlenkou PCA je nalezení nové sady os, tzv. hlavních komponent, které jsou lineárními kombinacemi původních rysů.

Tyto nové osy se volí tak, aby první hlavní komponenta měla největší možný rozptyl, druhá hlavní komponenta měla druhý největší rozptyl atd. Cílem je najít v datech nejvíce informativní směry, které lze použít k reprezentaci dat v kompaktnější podobě.

Jednou z hlavních výhod PCA je, že může pomoci vizualizovat vysoko rozměrná data tím, že je promítne do méně rozměrného prostoru. To může být užitečné zejména při práci s daty, která mají mnoho rysů, protože vizualizace a interpretace dat s vysokou dimenzí bývá obtížná, někdy nemožná. Kromě toho lze PCA použít ke snížení šumu v souboru dat a také k identifikaci vzorců a korelací v datech.

Při použití PCA se postupuje následujícími pěti kroky [4]:

1) Normalizace nebo standardizace dat

   PCA je algoritmus citlivý na měřítko vstupních dat, proto je nutné provést normalizaci dat. Častější variantou je ovšem standardizace. Ta se pro každý parametr provede spočtením průměru a směrodatné odchylky a jejich použitím v rovnici 3.1.1, kde $\bar{x}$ je průměr a σ směrodatná odchylka.

$$x_{new} = \frac{x - \bar{x}}{\sigma} \tag{3.1.1}$$

2) Výpočet kovarianční matice

   Výpočet kovariantní matice dle rovnice 3.1.2.

$$Cov(x, y) = \frac{\sum(x_i - \bar{x}) * (y_i - \bar{y})}{N - 1} \tag{3.1.2}$$

3) Výpočet vlastních čísel a vektorů

   Spočtení vlastních čísel a vektorů kovarianční matice. Vlastní vektory udávají směr hlavní komponenty, vlastní číslo pak její důležitost, respektive rozptyl.

4) Seřazení vlastních vektorů

   Sestupné seřazení vlastních vektorů podle velikosti vlastních čísel.

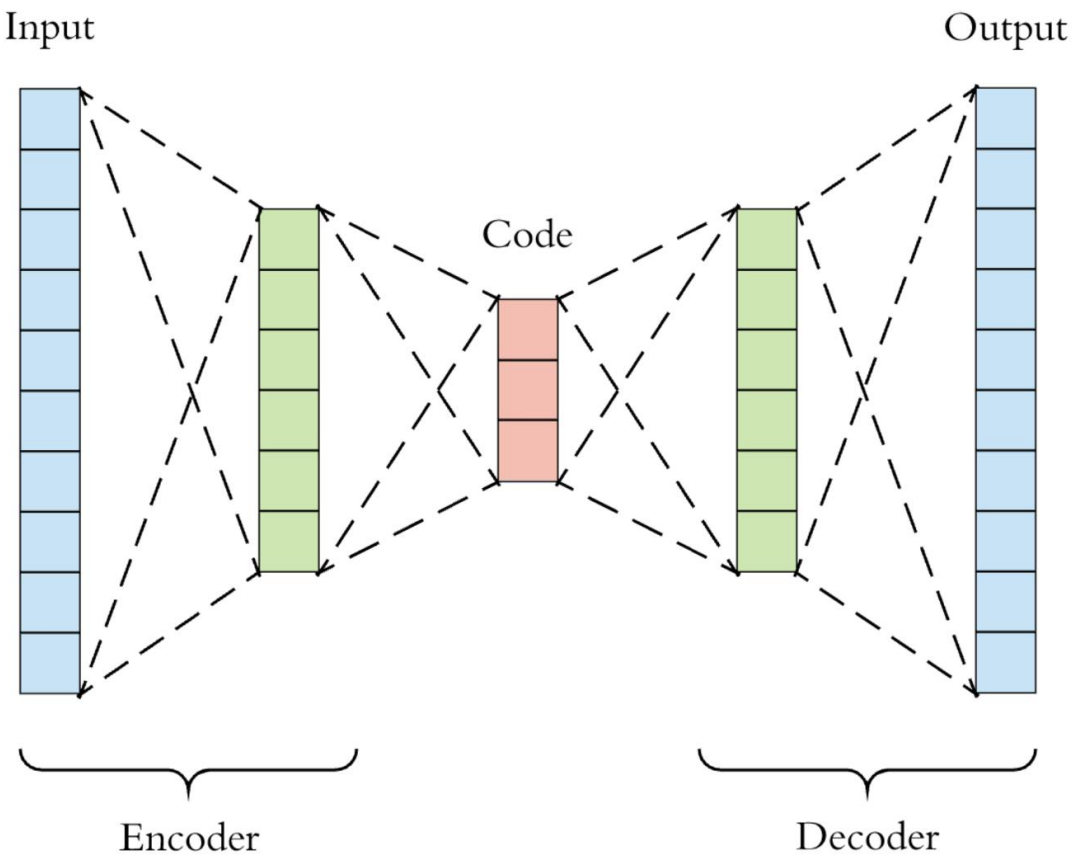
5) Použití vlastních vektorů s nejvyšším rozptylem

   Tímto se docílí zachování co nejvíce původních informací z dat, ale snížení celkového počtu dimenze. Analýza hlavních komponent pak v závislosti na velikosti vlastních čísel jednotlivých komponent umožňuje určit, jak velké procento původní informace se zachovalo, respektive ztratilo. V inženýrské praxi se často používá procentuální prahová hodnota, která udává, jak velké množství rozptylu zachovat, podle které je zvolen počet použitých komponent.

Autoenkodér

Autoenkodéry [21] jsou typem architektury neuronových sítí, které jsou navrženy tak, aby se naučily komprimovanou reprezentaci vstupních dat. Tyto sítě jsou natrénovány k rekonstrukci vstupních dat z komprimované reprezentace, známé také jako kódová vrstva. Díky tomu jsou autoenkodéry užitečné pro úlohy, jako je redukce dimenze dat, extrakce charakteristických vlastností a detekce anomálií. Důležitým rozdílem oproti analýze hlavních komponent je možnost zachytit i nelinearity ve zkoumaných datech.

Autoenkodér se skládá ze dvou hlavních komponent: kodéru a dekodéru. Kodér přijímá vstupní data a mapuje je do méně rozměrné reprezentace, zatímco dekodér přijímá tuto zakódovanou reprezentaci a pokouší se co nejpřesněji rekonstruovat původní vstupní data. Cílem trénování autoenkodéru je minimalizovat rozdíl mezi původními vstupními daty a rekonstruovanými daty. Schéma obecného autoenkodéru je ukázáno na obrázku 3.6.



Obrázek 3.6 Obecné schéma autoenkodéru [20]

Jednou z klíčových vlastností autoenkodérů je, že se jedná o algoritmy s učením bez učitele. To znamená, že nevyžadují označená data, aby se naučily užitečnou reprezentaci vstupních dat. Díky tomu jsou vhodným řešením pro širokou škálu úloh, kde je nedostatek označených dat nebo je jejich získání problematické.

Mezi důležitý faktor architektury autonekodéru patří výběr aktivační funkce každé vrstvy. Ty rozhodují o tom, zda má být konkrétní neuron v závislosti na vstupu aktivován nebo ne. Důvodem, proč jsou aktivační funkce tak důležité je, že umožňují modelu zachytit nelineární prvky. V případě, že by se aktivační funkce nepoužívaly, prakticky by se jednalo o implementaci analýzy hlavních komponent. Současně by vnitřní skryté vrstvy modelu neměli žádný smysl, jelikož složení jakýchkoliv dvou lineárních funkcí je samo o sobě také lineární funkcí a celý model by se tak stal pouze lineárním regresním modelem.

Takzvané učení autoenkodéru probíhá na základě účelové funkce l, která říká, jak moc je rekonstruovaný výstup rozdílný od původního vstupu. Nejčastěji bývá použita kvadratická střední chyba, viz rovnice 3.1.3, kde x_k odpovídá vstupní hodnotě, $\hat{x}_k$ odpovídající předpovídané hodnotě a n počtu pozorování.

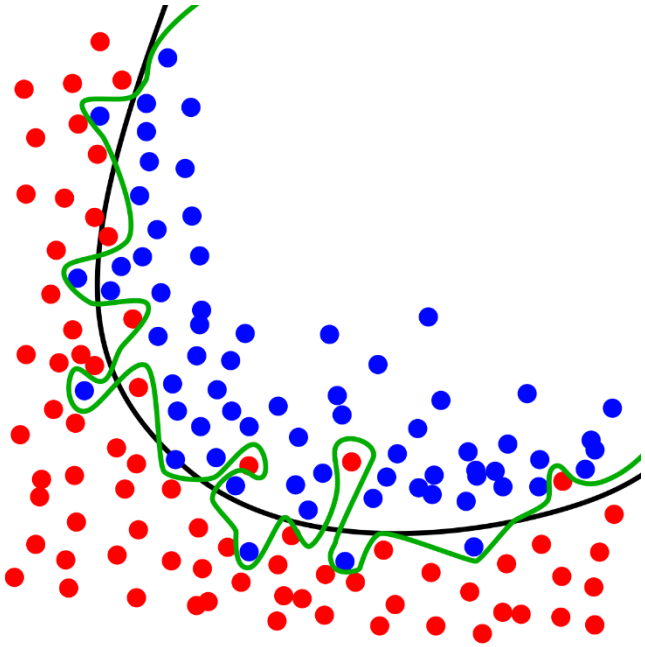
$$l(f(x)) = \frac{\sum_k (\hat{x}_k - x_k)^2}{n} \tag{3.1.3}$$

Důležitým parametrem učení je takzvaný optimalizátor. To je označení pro algoritmy, které se zaměřují na změnu atributů autoenkodéru, jako jsou váhy neuronů a míra učení, za účelem snížit ztráty.

Aktuálně nejčastěji používaný optimalizátor je „adam“ [22], jehož název byl odvozen z anglických slov adaptive moment estimation. Jedná se o algoritmus, který využívá kombinaci metod RMSprop a Stochastic Gradient Descent s hybností. Více o zmíněných metodách lze najít ve zdroji [22].

Pro samotný proces učení se následně používá nastavení počtu epoch, které říká, kolikrát mají veškerá data projít učícím procesem. Čím více epoch je použito, tím více se může model naučit konkrétní trénovací data.

Někdo by mohl namítnou, proč se tedy nepoužívá co největší množství epoch. Hlavní důvody jsou dva. Za prvé by to bylo příliš časově náročné, za druhé by mohlo dojít k takzvanému přeučení, které je naznačené na obrázku 3.7. Je vidět, že model se perfektně naučil rozpoznat trénovací data, ovšem současně výrazně ztratil svou robustnost a při styku s daty, se kterými se zatím nesetkal by aplikoval zeleně vyznačenou hranici na obrázku a rozpoznávání dat by mělo špatné výsledky.



Obrázek 3.7 Ukázka přetrénování modelu [23]

Dalším parametrem je batch size, což je parametr, který v tomto případě určuje, kolik křivek má učícím se modelem projít, než dojde k úpravě jeho vnitřních parametrů.

Validation split je parametr, který automaticky rozděluje vstupní data na trénovací a validační, které slouží pro ověření funkčnosti modelu. Při trénování se model testuje na datech, se kterými nikdy předtím nepřišel do styku, což umožňuje posoudit obecnost a potencionální přetrénování modelu na konkrétní trénovací dataset.

Existuje několik různých typů architektur autoenkodérů [6][21], z nichž každý má své silné a slabé stránky. Například autoenkodér k redukci dimenze je jednoduchá dopředná neuronová síť s jednou skrytou vrstvou, zatímco konvoluční autoenkodér je určen pro práci s obrazovými daty a využívá konvoluční vrstvy. Dalším populárním typem autoenkodéru je variační autoenkodér (VAE), který přidává schopnost generovat nové vzorky, a to vzorkováním z naučeného latentního prostoru. Pro moje účely stačilo použití základního autoenkodéru.

Kapitola 4

Shluková analýza

Shluková analýza [24] je často používanou technikou pro seskupování datových bodů do smysluplných podskupin nebo shluků na základě jejich podobnosti. K tomu se využívá velkého spektra algoritmů, které mají velmi různé charakteristiky, předpoklady pro správné seskupování a další parametry, které je důležité při používání algoritmů znát. Cílem shlukování je odhalit inherentní seskupení v datech do shluků tak, aby si položky v každém shluku byly navzájem podobnější než položky v jiných shlucích. Tato podobnost může být vyjádřena různými způsoby a kritérii a je to jedna z důležitých vlastností každého z algoritmu. Existuje několik způsobů, jak lze metody shlukování rozdělit a klasifikovat.

Jako první a velmi důležitý faktor zmíním typ dat. Metody shlukování, které pracují se spojitými daty jsou navrženy pro zpracování bodů s číselnou hodnotu na spojitém měřítku. Příkladem takových to dat můžou být naměřené hodnoty proudu, magnetického toku nebo objemu kapaliny.

Na druhou stranu metody shlukování, které pracují s kategorickými daty jsou navrženy tak, aby zpracovávaly datové body, které se skládají z diskrétních skupin a kategorií. Příkladem takových to dat může být informace, zda se jedná o správný nebo zmetkový díl nebo typ produktu.

Existují i metody shlukování, které jsou schopné zpracovat kombinaci spojitých a kategorických dat. Tyto metody mohou být použité například pro převod kategorických dat na číselné hodnoty, ale i přímo na shlukování kombinovaných typů dat.

Při používání metod shlukování je velmi důležité, zda máme informaci o tom, kolik shluků v analyzovaných datech máme, respektive očekáváme. Některé algoritmy, které v následujících podkapitolách představím, tuto informaci vyloženě potřebují pro to, aby je bylo možné použít. Jiné zvládnou o celkovém počtu shluků rozhodnout samy, ale mohou tak původní předpoklad potvrdit, nebo vyvrátit.

Při shlukování dat je důležité zvolit metodu, která je vhodná na typ analyzovaných dat. Některé metody nemusí být vhodné pro určitý typ dat nebo mohou vyžadovat různě složité kroky předběžného zpracování, které následnou práci algoritmu na datech umožní a pokud se algoritmus shlukování vybere bez předchozího zamyšlení, výsledné shluky nemusí mít jakoukoliv vypovídací hodnotu.

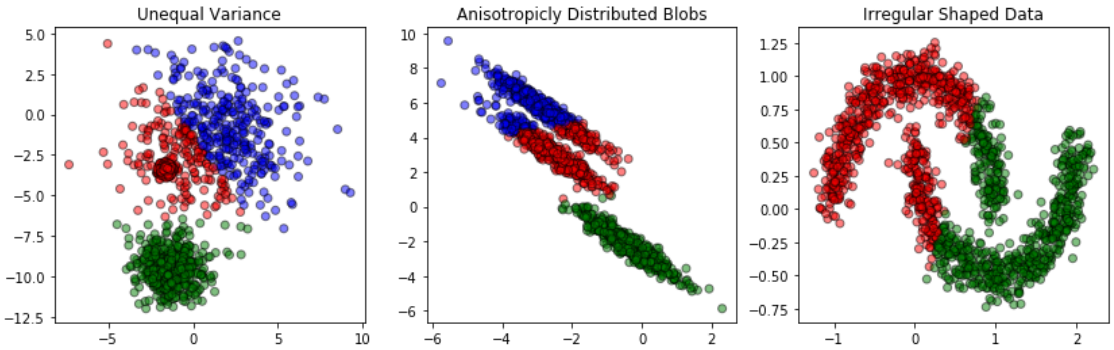
Výběr může usnadnit i faktor, zda disponujeme trénovacími daty, u kterých předem víme, do jakého shluku by měli patřit, nebo ne. S tím jsou spojené takzvané shlukování s učitelem, respektive trénování modelů s učitelem nebo bez učitele.

Pokud označenými daty disponujeme, lze zvážit použití klasifikačních metod. Ty zahrnují trénování modelu na označených datech, u kterých jsou již známé správné třídy. Model pak používá tato označená data jako vzor k učení charakteristik každé třídy a může díky tomu předpovídat výslednou skupinu neoznačených dat. Aktuálně nejpopulárnějšími klasifikačními metodami jsou neuronové sítě, které nacházejí stále větší uplatnění v průmyslu.

Na druhou stranu, pokud používáme data bez jakékoli předchozí znalosti správných seskupení, model musí objevit vzory a charakteristiky každého shluku sám. To do určité míry ztěžuje proces analýzy dat, na druhou stranu umožňuje algoritmu objevit spojitosti v datech, které pro člověka nemusí být na první pohled zřejmé. Jedním z neznámějších příkladů shlukování bez učitele je použití algoritmu k-means, který bude více přiblížen v následující kapitole.

4.1. K-means

K-means [6] se řadí mezi nejznámější a velmi populární algoritmy shlukování. Jedná se o algoritmus shlukování, který seskupuje vstupní data do shluků na základě jejich společných charakteristik. Je navržen tak, aby data rozděloval do určitého a předem známého počtu seskupení. Tento počet se označuje písmenem *K*. Cílem k-means je najít skupiny ve vstupních datech tak, aby si členové jedné datové skupiny byli navzájem podobnější než s členy v ostatních skupinách. Důležitý předpoklad tohoto algoritmu je, že očekává shluky, které mají kruhový tvar. To může v případu vstupních dat, které mají tvar jako na obrázku 4.1 znamenat nepřekonatelnou překážku a nutnou volbu jiného algoritmu.



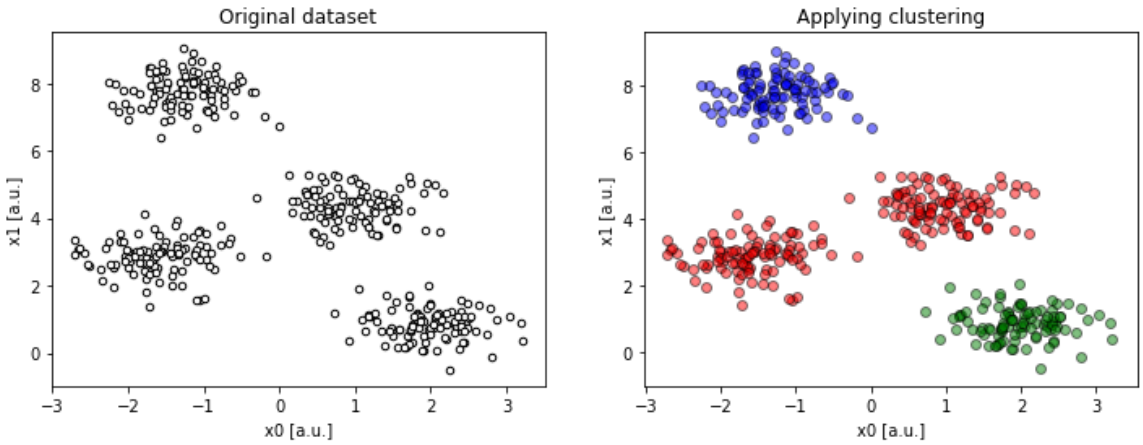
Obrázek 4.1 Ukázka špatného shlukování k-means

Princip algoritmu

Průběh algoritmu se dá rozdělit do 3 základních kroků, po kterých dochází k rozdělení vstupních dat do „*K*“ seskupení.

1) Zvolení počtu seskupení

To je jedno z nejdůležitějších nastavení pro tento algoritmus. Následky špatně zvoleného počtu seskupení je možné vidět na jednoduché ukázce na obrázku 4.2. Na něm je znázorněné, že při zvolení příliš mnoho nebo příliš málo seskupení jsou i na první pohled snadno oddělitelná data rozdělena nepřesně. Bližší informace k volení správného počtu budou více rozebrány na stránce 21.

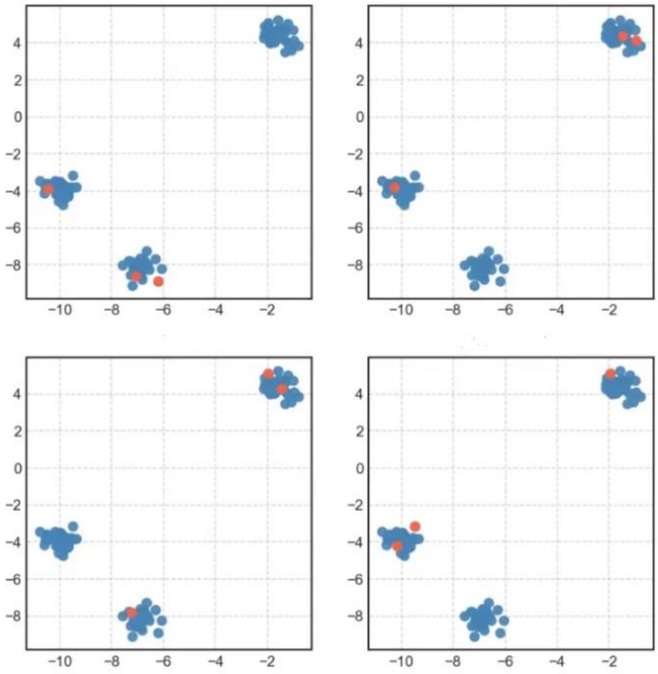


Obrázek 4.2 Ukázka špatně zvoleného počtu shluků

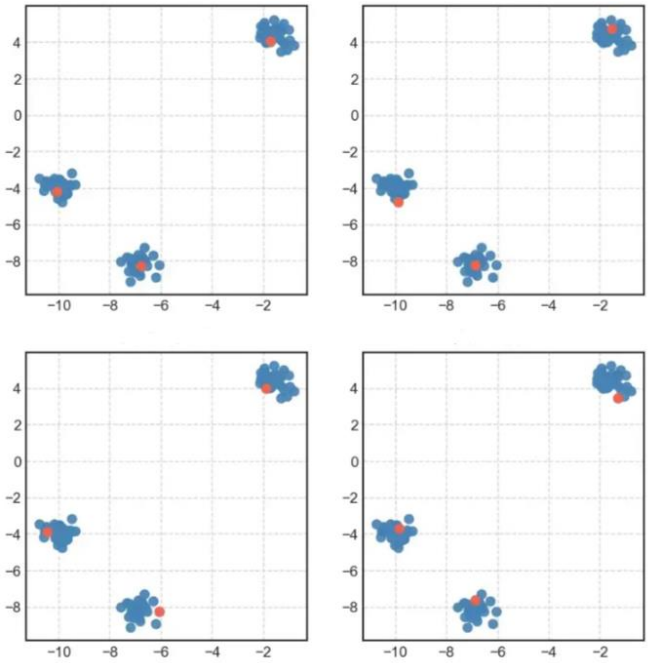
2) Určení pozic těžišť

Algoritmus vybere počáteční pozici těžišť. Tyto počáteční pozice jsou náhodně vybrané. Tento fakt způsobuje, že při opakovaném aplikování se k-means může se stejnými daty dostat k jinému finálnímu rozdělení. Tento případ nemusí nastat pokaždé a záleží na typu dat, zda se jedná o potencionální problém nebo ne. Jedním z možných řešení je nahrazení náhodného vybírání těžišť inicializací [7] těžišť s určitými podmínkami. Jako příklad uvedu k-means++, což je inicializační metoda těžišť pro k-means. Tato metoda pozici prvního těžiště vybere náhodně a ty následující vybírá na základě maximální čtvercové vzdálenosti. Jejím cílem je odsunout jednotlivá těžiště co nejdále sebe.

Na obrázku 4.3 je možné vidět 4 iterace náhodné inicializace k-means, na obrázku 4.4 pak 4 iterace inicializace k-means++. Při porovnání je zřejmé, že inicializace k-means++ byla úspěšnější v prvotním rozřazení těžišť, což by umožňovalo větší opakovatelnost procesu. Nevýhodou této metody je ovšem rychlost algoritmu, která se umisťováním těžišť na specifické pozice zmenšila.



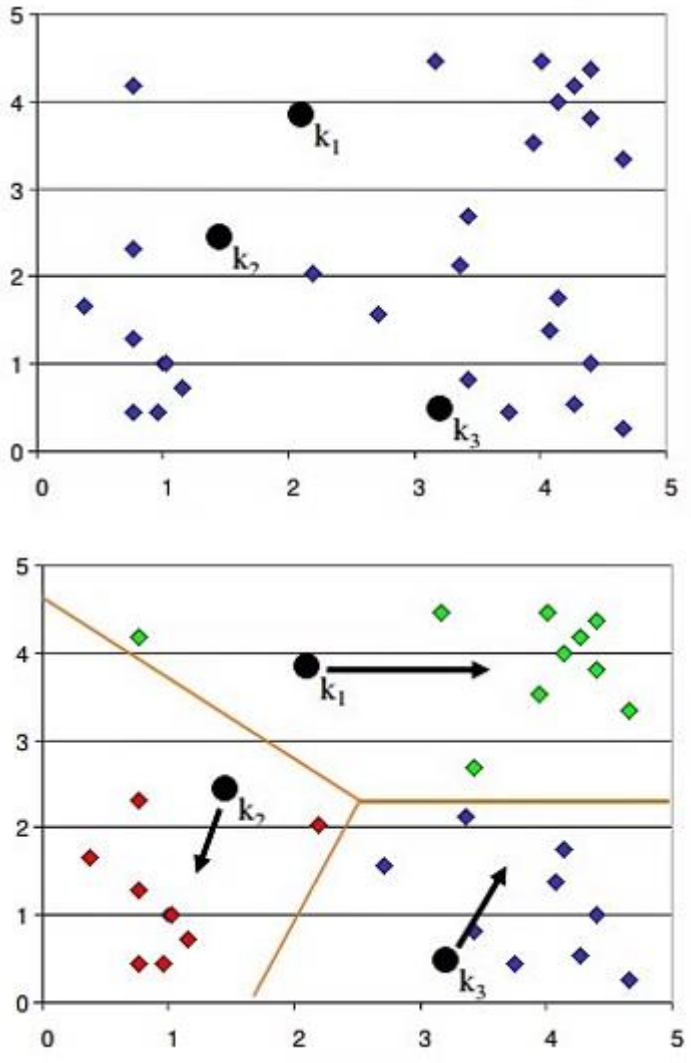
Obrázek 4.3 Čtyři iterace inicializace k-means [7]



Obrázek 4.4 Čtyři iterací inicializace k-means++ [7]

3) Určení pozic těžišť

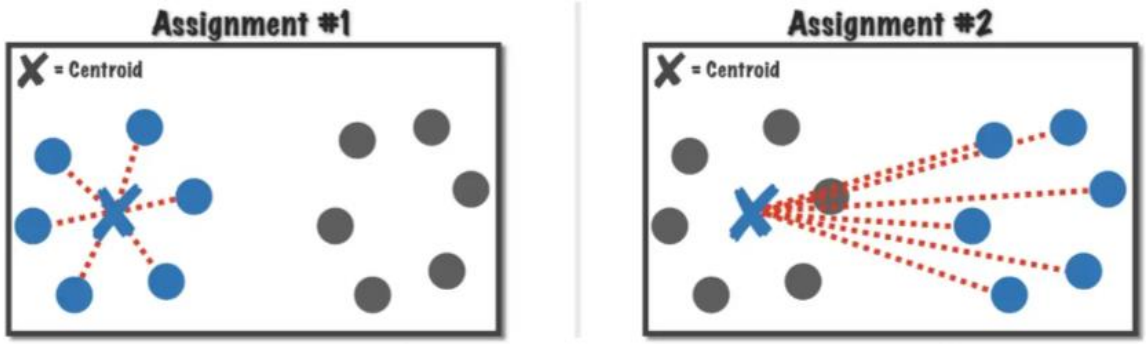
V tomto kroku dochází k iterativní úpravě polohy těžiště. Jako první se každý datový bod přiřadí k odpovídajícímu těžišti. Následně se pro každý takto vytvořený shluk vypočítá nová poloha jeho těžiště, jako průměr všech bodů, které do shluku aktuálně náleží. Toto je i s původním náhodným výběrem těžišť naznačeno na obrázku 4.5. Tento krok se opakuje do té doby, než jednotlivé body nepřestanou měnit svoji skupinu, což má za následek, že těžiště shluků už zůstává stejné a nemění svou polohu.



Obrázek 4.5 Úprava poloh těžišť algoritmu k-means [16]

Metrika vyhodnocení

Pokud se pohybujeme ve dvou rozměrném prostoru, který je znázorněn i na předešlých obrázcích, nebo tří rozměrném prostoru, není pro člověka intuitivně těžké určit, do jakého shluku má datový bod patřit. Pro k-means algoritmus je ovšem potřeba toto určení kvantifikovat a zvolit určitý typ metriky, podle které bude schopný toto rozdělení provézt. Na obrázku 4.1.6 jsou ukázány dva případy přiřazení datových bodů k těžišti.



Obrázek 4.6 Dva případy přiřazení datových bodů [6]

Na první pohled je zřejmé, že přiřazení vlevo je vhodnější než přiřazení vpravo. Pokud bychom se zaměřili na červeně čárkované úsečky na obrázku, je vidět, že délka těchto úseček je u přiřazení vlevo obecně nižší než na přiřazení vpravo. Obecně lze tedy říct, že chceme najít těžiště takové, aby průměrná vzdálenost bodů z příslušného shluku byla co nejmenší.

Jinak řečeno, hledáme takové těžiště C, které minimalizuje účelovou funkci J, která je definována jako celková suma kvadratických odchylek polohy datových bodů $x_i \in \{x_1, x_2, \dots, x_m\}$ od těžiště. To lze matematicky zapsat dle rovnice 4.1.1.

$$J(x) = \sum_{i=1}^{m} \|x_i - C\|^2 \tag{4.1.1}$$

Polohu těžiště C, která byla v předchozích odstavcích slovně definována jako průměrná hodnota bodů x_i je možné popsat rovnicí 4.1.2 níže. [6]

$$C = \frac{\sum_{i=1}^{m} x_i}{m} \tag{4.1.2}$$

Těmito rovnicemi jsme schopni získat polohu jednoho těžiště dle algoritmu k-means. Cílem je ovšem minimalizovat kvadratickou odchylku všech těžišť $c_j \in \{c_1, c_2, \dots, c_k\}$ pro všechny pozorování $x_i \in \{x_1, x_2, \dots, x_m\}$ zároveň. Proto hledáme minimum účelové funkce J dle rovnice 4.1.3. [6]

$$J(x) = \sum_{j=1}^{k} \sum_{i=1}^{m} \left\| x_i^{(j)} - c_j \right\|^2 \tag{4.1.3}$$

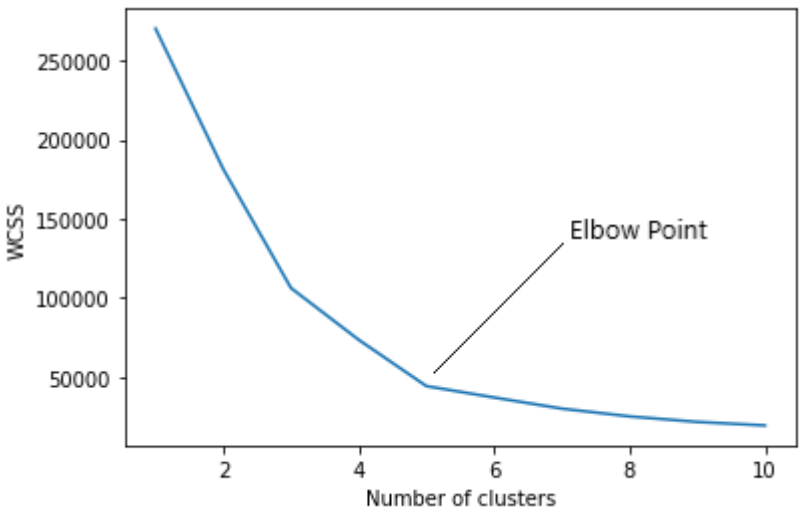
Volba počtu shluků

Jak již bylo zmíněno na začátku podkapitoly 4.1, volba počtu shluků pro shlukování k-means je důležitým parametrem algoritmu, protože nevhodný počet shluků může mít za následek špatné rozřazení vstupních dat do skupin a sníženou schopnost odhalit vzory v datech.

Důležité je zmínit, že vždy je lepší použít znalosti o vstupních datech, případně kombinaci více metod pro správnou volbu vstupního počtu seskupení, aby byla zvýšena pravděpodobnost správných výsledků.

1) Elbow method

Jedním přístupem k určení počtu shluků je takzvaná „elbow method“ [17], což je metoda, která vychází z myšlenky, že s rostoucím počtem shluků se bude kvadratická odchylka v rámci shluku stále snižovat, ale s klesající rychlostí. Při použití této metody u k-means se algoritmus spustí s postupně rostoucím počátečně zvoleným počtem shluků, a pro každý takto zvolený počet shluků se vypočítá suma kvadratických odchylek. Výsledkem je míra vzdálenosti datových bodů v rámci shluků k jejich těžišti, vynesená do grafu v závislosti na zvoleném vstupním počtu shluků. Ukázku takového grafu je možné vidět na obrázku 4.1.7. Správný počet shluků této metody bude odpovídat bodu, kde dojde k největšímu zlomu křivky, přesněji řečeno, kde dojde k největšímu zpomalení poklesu sumy kvadratických odchylek. Na ukázce na obrázku 4.7 to odpovídá bodu 4 na x-ové ose.



Obrázek 4.7 Elbow method [17]

2) Metoda siluety

Dalším přístupem k určení počtu shluků je metoda siluety [8][18]. Pro použití této metody se pro každý datový bod vypočítá takzvaný koeficient siluety. Ten je měřítkem toho, jak podobný je datový bod ostatním datovým bodům v jeho shluku ve srovnání s datovými body v jiných shlucích, a počítá se následovně:

- Pro každý datový bod se vypočítá průměrná vzdálenost ke všem ostatním datovým bodům ve stejném shluku (a).

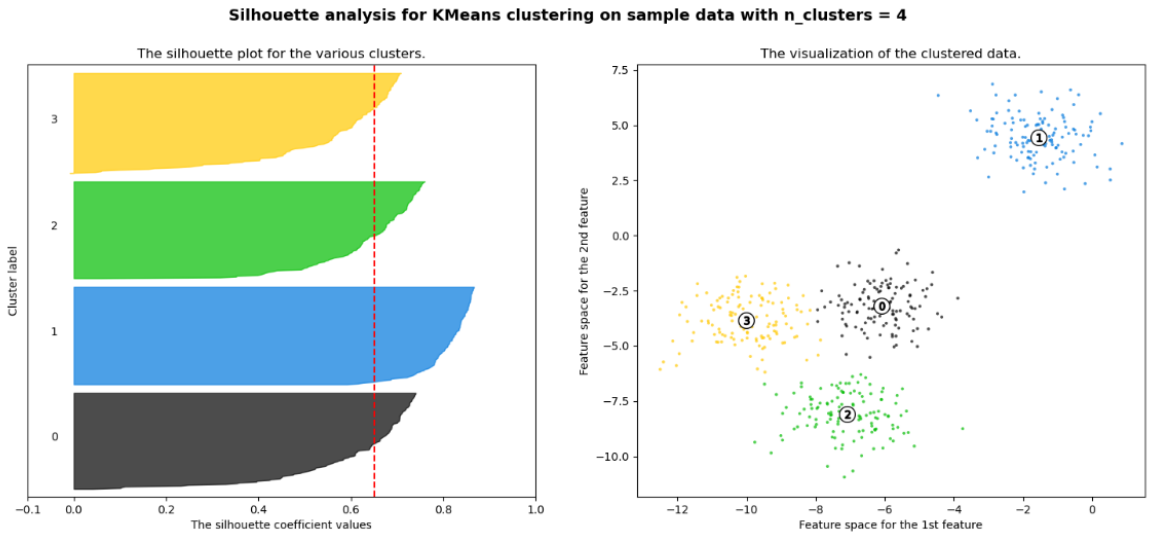
- Pro každý datový bod se vypočítá průměrná vzdálenost ke všem datovým bodům v nejbližším shluku (b).

Koeficient siluety S pro datový bod se pak vypočítá jako:

$$S = \frac{b - a}{\max(a,\ b)} \qquad (4.1.4)$$

Jeho hodnota se pohybuje od -1 do 1, přičemž hodnoty blížící se 1 znamenají, že datový bod dobře zapadá do svého shluku a hodnoty blížící se -1 znamenají, že datový bod je se svým shlukem sladěn špatně. Po provedení výpočtu dle rovnice 4, se vypočítá průměrný koeficient pro všechny datové body a nejvyšší hodnota se použije k určení vhodného počtu shluků.

Na obrázku 4.8 je možné vidět výsledek této metody se zvolenými 4 počátečními shluky. Hodnoty koeficientu siluety jsou podobné napříč všemi shluky a jejich průměrná hodnota je přibližně 0,65, což je hodnota koeficientu blížící se spíše k 1, která říká, že jednotlivé shluky jsou relativně správně odděleny. Analogicky se metoda aplikuje i na jiné hodnoty počátečního počtů shluků a výsledné grafy se porovnají. Na základě toho je pak možné zvolit jejich správný počáteční počet.



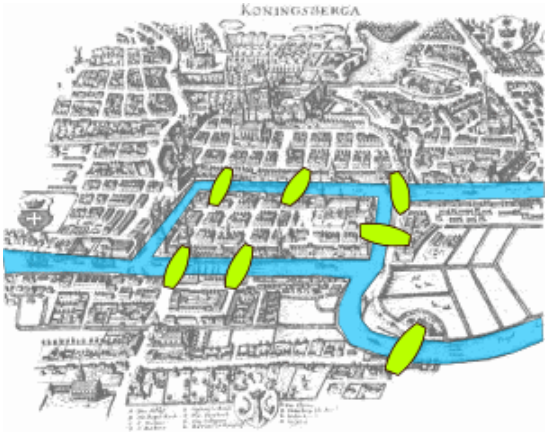
Obrázek 4.8 Silhuette method [18]

4.2. Spektrální shlukování

Spektrální shlukování [9] je metoda, která je postavená na základech teorie grafů. Jednou z jejich hlavních výhod je, že dokáže najít shluky libovolného tvaru, i když nejsou jednotlivé shluky lineárně oddělitelné. Kromě toho jej lze také použít ke shlukování dat, která nejsou reprezentována jako tradiční pole bodů, ale jako matice podobnosti nebo afinity. Spektrální shlukování je proto vhodné pro širokou škálu datových typů a často se také používá při analýze obrazu a sítí.

Základ teorie grafů a spektrálního shlukování

Teorie grafů je velmi široké téma a mnozí se s ním mohli setkat u Eulerovi úlohy z 18. století, která je známá také jako úloha sedmi mostů města Královce. Její zadání zní, zda je možné právě jednou projít přes všechny mosty ve městě, které jsou znázorněny na obrázku 4.9, a vrátit se zpět do původního místa.

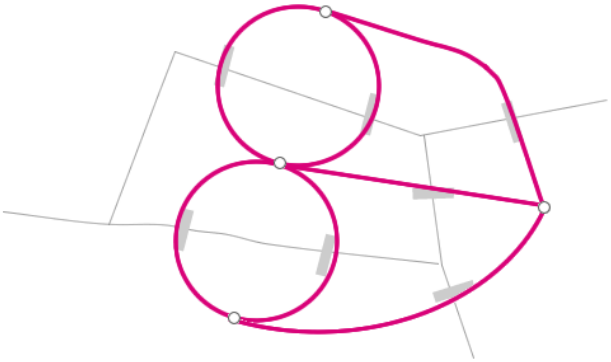


Obrázek 4.9 Úloha sedmi mostů města Královce [9]

Euler tuto úlohu přeformuloval do matematické reprezentace grafu, kde každý ostrov představuje vrchol a každý most představuje hranu, která tyto ostrovy spojuje, tak jak je ukázáno na obrázku 4.10.

Dále dokázal, že pokud graf splňuje následující dvě podmínky, jedná se o graf eulerovský, který má řešení pro kreslení jedním tahem, odborně řečeno eulerovský tah:

- Sudý stupeň vrcholů – z každého vrcholu vychází právě sudý počet hran
- Souvislý graf – pro každé dva vrcholy A a B platí, že v grafu existuje cesta z A do B



Obrázek 4.10 Přetvoření úlohy sedmi mostů města Královce [9]

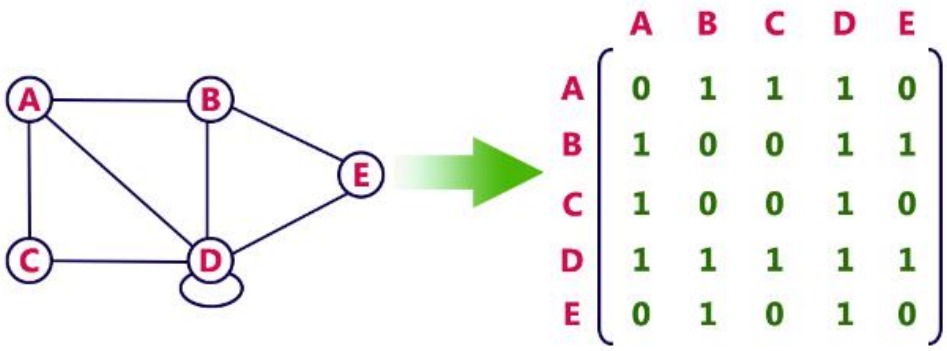
Z toho vychází, že úloha výše nemá řešení, protože i když je graf souvislý, všechny jeho vrcholy mají naopak stupeň lichý.

Nejtypičtějším zápisem grafu je zápis pomocí matice sousednosti [9]. Ta je symetrická, čtvercová a její rozměr je dán počtem vrcholů. Prvním krokem pro její vytvoření je náhodného očíslování všech vrcholů. Následně každý řádek a sloupec reprezentuje určitý vrchol a hodnota na daném místě, zda a kolikrát jsou vzájemné vrcholy spojeny. Matematická definice [9] je následující:

> *Nechť $G = (V, E)$ je graf s n vrcholy. Označme vrcholy $v_1,\ldots,v_n$ v libovolném pořadí. Matice sousednosti grafu G je čtvercová matice $\boldsymbol{A_G} = (a_{ij})_{i,j=1}^{n}$ definovaná předpisem:*

$$a_{ij} = \begin{cases} 1 \; pro \; \{v_i, v_j\} \in E \\ 0 \; jinak \end{cases}$$

Na obrázku 4.11 je možné vidět, že bod A je spojen s vrcholy B, C, D. To znamená, že se pohybujeme v prvním řádku, respektive sloupci matice a pro příslušné vrcholy B, C, D bude hodnota v matici 1. Analogicky lze pak vyplnit celou matici sousednosti.



Obrázek 4.11 Ukázka vytvoření matice sousednosti [19]

Další důležitou maticí je matice stupňů D [9], která je čtvercová a na hlavní diagonále má sumu příslušné řady z matice sousednosti. Říká nám, kolik má každý z vrcholů dohromady hran. Příklad takovéto matice je na uveden níže a odpovídá matici sousednosti z obrázku 4.11.

$$\boldsymbol{D} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} \tag{4.2.1}$$

Pomocí těchto dvou matic jsme schopní získat Laplacián grafu [10]. To je další matice, která původní graf reprezentuje a vypočítá se dle rovnice 4.2.1.

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A_G} \tag{4.2.2}$$

Pokud vypočítáme vlastní čísla Laplaciánu, jsme schopni o původním grafu získat velké množství informací, které jsou k následujícímu shlukování velmi užitečné. Pro použitý příklad je Laplacián spočítán na obrázku 4.2.5.

$$\boldsymbol{L} = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 \\ -1 & 3 & 0 & -1 & -1 \\ -1 & 0 & 2 & -1 & 0 \\ -1 & -1 & -1 & 4 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix} \tag{4.2.3}$$

Princip algoritmu

Velmi často nejsou vstupní data interpretována jako graf popsaný na předchozí stránce, ale jedná se o samostatné datové body. Nejjednodušším způsobem, jak data na graf pro spektrální shlukování [10] převést je použití metody k-nearest neighbors. Tato metoda bere každý datový bod jako samostatný vrchol. Hrany jsou vykresleny vždy k dalším „k“ nejbližším datovým bodům.

Pokud máme připravený graf, můžeme postupovat dle následujících bodů:

1) Získání matice sousednosti
2) Získání matice stupňů
3) Spočtení Laplaciánu
4) Spočtení vlastních čísel a vlastních vektorů Laplaciánu

Hodnoty vlastních čísel nám o původním grafu mohou říct mnoho zajímavých informací. Počet nulových hodnot vlastních čísel sdělují, z kolika samostatných komponentů se graf skládá.

První nenulové vlastní číslo se nazývá jako spektrální mezera a sděluje, jak hustě jsou mezi sebou jednotlivé vrcholy spojené.

Druhé nejmenší vlastní číslo se nazývá jako Fiedlerova hodnota a udává, kolik přibližně hran bychom museli přerušit, abychom graf rozdělili na dvě spojené komponenty. To, do jaké z dvou komponent by jednotlivé vrcholy patřili se dá vyčíst z vlastního vektoru tohoto vlastního čísla.

Obecně se dá říct, že z hodnot vlastních čísel je možné odhadnout, kolik shluků v grafu očekávat a jaké obecné vlastnosti graf má a z vlastních vektorů následně určit, jaké body do daných shluků náleží.

5) Provedení shlukování určených vlastních vektorů

V tomto bodu již lze použít algoritmus jako je například K-means, do kterého se jako vstupní data použijí určité vlastní vektory spočítané z Laplaciánu.

4.3. DBSCAN

DBSCAN [11][12] (Density-Based Spatial Clustering of Applications with Noise) je algoritmus shlukování založený na hustotě, který se používá k identifikaci shluků bodů v souboru dat. Hlavní myšlenkou DBSCAN je, že shluky bodů jsou tvořeny oblastmi s vysokou hustotou, přičemž hustota je definována počtem bodů v určité vzdálenosti od daného bodu. To je rozdíl od jiných shlukovacích algoritmů, jako je k-means, které k definování shluků používají metriku vzdálenosti.

DBSCAN je založen na dvou parametrech. Epsilon je maximální vzdálenost mezi dvěma body, které jsou považovány za stejné shluky. Druhý parametr je minimální počet bodů, které musí být od daného bodu vzdáleny do vzdálenosti epsilon, aby byl považován za jádrový bod. Shluk se vytvoří spojením všech jádrových bodů, které jsou od sebe vzdáleny do vzdálenosti epsilon. Body, které nejsou ve vzdálenosti epsilon od žádného jádrového bodu, se považují za šum.

Jednou z hlavních výhod metody DBSCAN je, že na rozdíl od metody k-means, která identifikuje pouze kulové shluky, dokáže identifikovat shluky libovolného tvaru. Kromě toho DBSCAN nevyžaduje, aby uživatel určil počet shluků v datech, což může být obtížné nebo nemožné předem zjistit.

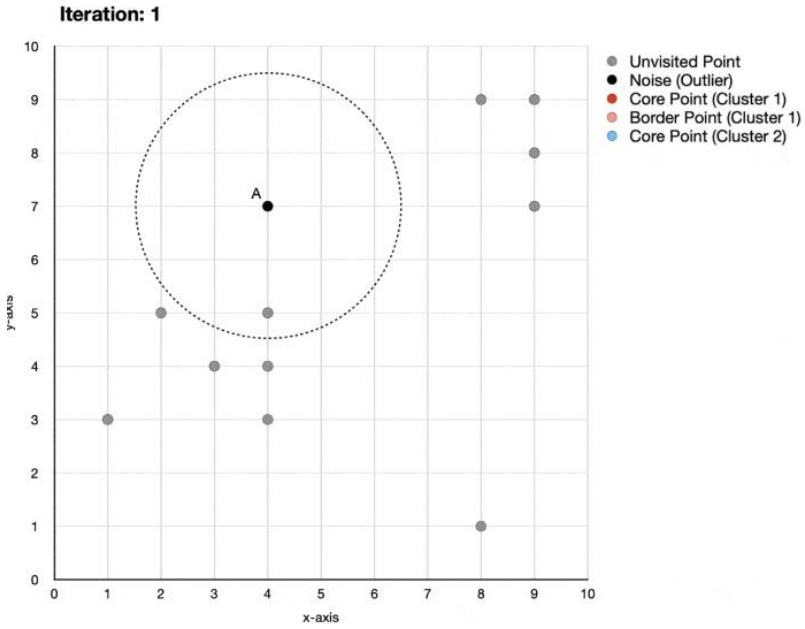
DBSCAN má však také některá omezení. Jedním z omezení je, že volba parametrů epsilon a minima bodů může výrazně ovlivnit výsledky algoritmu a může být obtížné najít optimální hodnoty těchto parametrů. Kromě toho může být DBSCAN citlivý na hustotu dat a nemusí dobře fungovat v souborech dat s různou hustotou.

Závěrem lze říct, že DBSCAN je shlukovací algoritmus založený na hustotě, který lze použít k identifikaci shluků bodů v souboru dat. Jeho hlavní výhodou je, že dokáže identifikovat shluky libovolného tvaru a nevyžaduje, aby uživatel určoval počet shluků v datech. DBSCAN má však také některá omezení, například citlivost na hustotu dat a volba hodnot epsilon a minima bodů může výrazně ovlivnit výsledky. DBSCAN se široce používá v různých oblastech, jako je počítačové vidění, rozpoznávání vzorů a zpracování obrazu, shlukování a může být cenným nástrojem pro analýzu a pochopení dat.

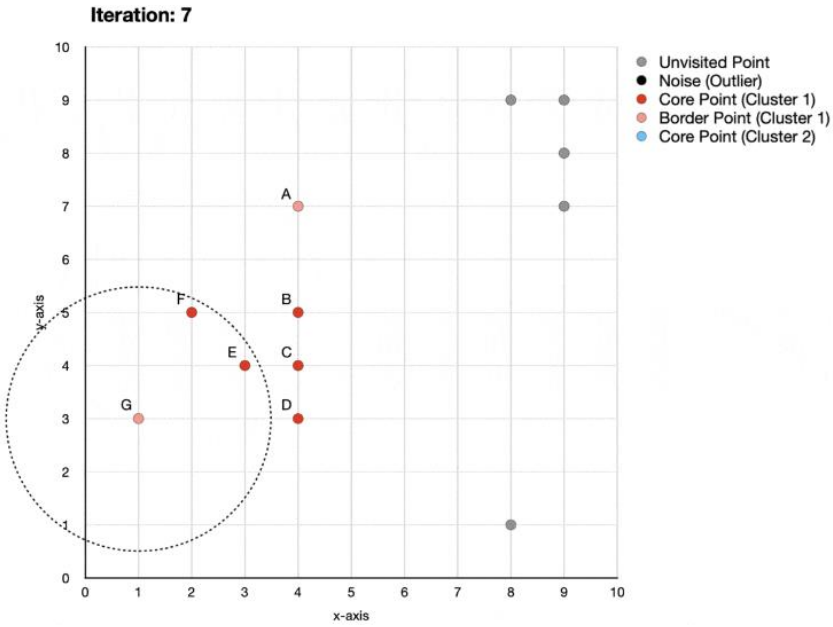
Princip algoritmu

DBSCAN [11] postupně prochází veškeré datové body a kontroluje, zda jsou splněny podmínky pro minimum bodů a vzdálenost epsilon. Na následujících čtyřech obrázcích je postupně ukázán proces rozdělování ukázkového data setu do shluků. Epsilon je nastaveno na hodnotu 2,5, minimum bodů pro jádrový bod 4.

Na obrázku 4.12 a 4.13 je zkoumán bod A a G, který má v okolí epsilon 1 respektive 2 další body, takže nesplňují podmínku minima bodů, a proto jsou zobrazeny jako okrajové body shluku 1. Body B, C, D, E a F splňují podmínky okolí epsilonu i minima bodů, tudíž jsou přiřazeny do jednoho shluku.

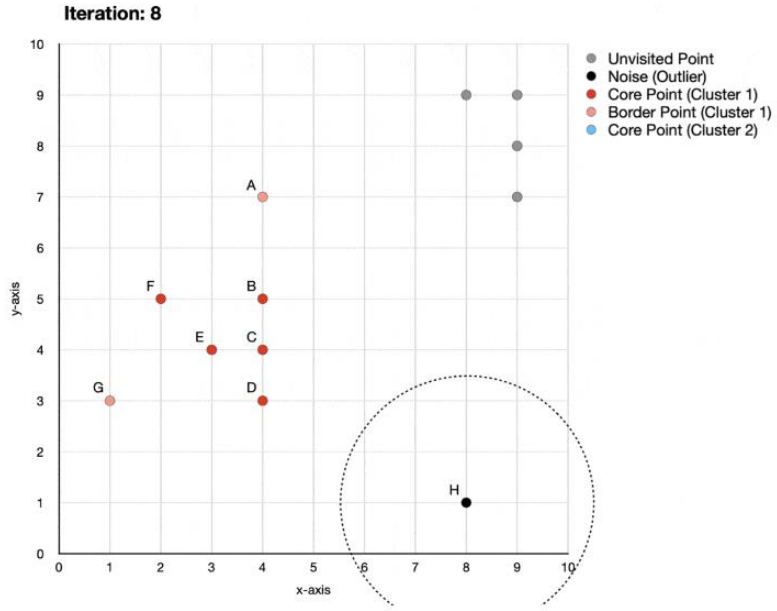


Obrázek 4.12 První iterace DBSCAN [12]

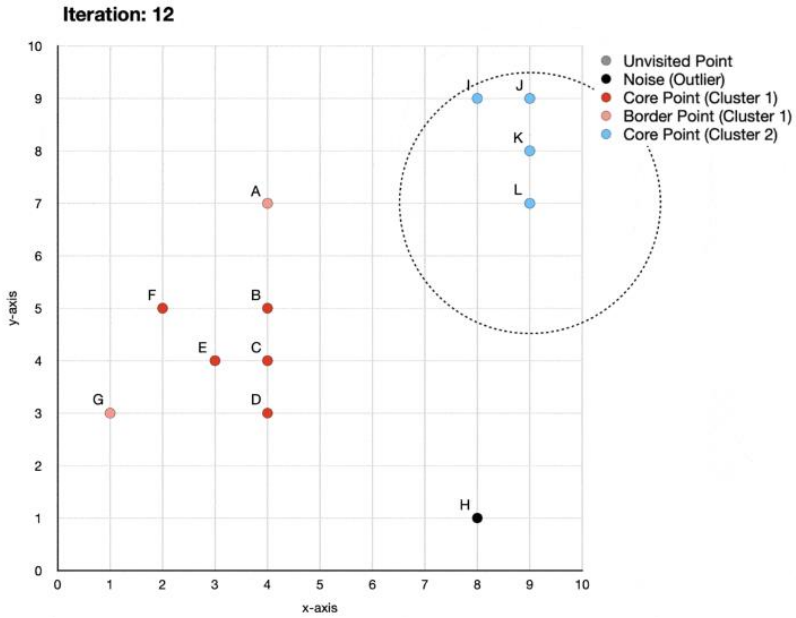


Obrázek 4.13 Sedmá iterace DBSCAN [12]

Na obrázku 4.13 je testován bod H, který nemá v okolí epsilon jediný další bod a proto je označen jako šum. Na obrázku 4.14 byly otestovány body I, J, K, L, které splňují podmínky okolí epsilonu i minima bodů, a proto tvoří další shluk. Analogicky by se pokračovalo i s větším datovým setem, dokud by nebyly zanalyzovány veškeré body.



Obrázek 4.14 Osmá iterace DBSCAN [12]



Obrázek 4.15 Dvanáctá iterace DBSCAN [12]

Kapitola 5

Praktická část – analýza dat

Celá analýza je provedena v programovacím jazyku Python [19], verze 3.10. Jeho výhodou je jednoduchá a čitelná syntaxe, díky níž se v něm mohou snadno zorientovat a používat ho i lidé s malými nebo žádnými zkušenostmi s programováním. Python je vysokoúrovňový programovací jazyk, který je vynikající pro analýzu dat. Má rozsáhlý ekosystém výkonných knihoven a frameworků určených speciálně pro analýzu dat a jejich manipulaci.

Jako příklad bych uvedl knihovnu NumPy, která poskytuje podporu pro vícerozměrná pole a matice číselných dat, což výrazně usnadňuje provádění matematických operací.

Dále knihovnu Pandas, která velmi usnadňuje čištění a transformaci dat. Nabízí flexibilní datové struktury – dataframy, které mají podobnou strukturu jako relační databáze a standardně se používají pro načítání, ukládání a předzpracování dat v datové analýze.

Matplotlib je knihovna pro vytváření statických, animovaných a interaktivních vizualizací dat.

Kromě těchto knihoven obsahuje Python také několik dalších knihoven a frameworků, které se běžně používají při analýze dat, například scikit-learn, knihovnu pro strojové učení a analýzu dat, a Tensorflow, framework pro hluboké učení neuronových sítí.

5.1. Načtení potřebných dat

Základem každé analýzy dat je prvotní prozkoumání, s jakými daty se pracuje. Jak bylo zmíněno v kapitole 3.4, k dispozici je 248 textových souborů s jednotlivými částmi křivek, které jsou lokálně uložené na pevném disku počítače. Název každého souboru obsahuje číselný identifikátor od 1 od 248. Dále je k dispozici textový soubor, ve kterém je ke každému tomuto číselnému identifikátoru DMC kód konkrétního dílu.

Pro získání parametrů z výroby jsem vytvořil MySQL dotaz, který jsem použil pro stáhnutí dat z časového období, ze kterého křivky pochází. Pro tento účel byla využita knihovna Pandas, která nabízí metodu *Pandas.read_sql*. Ta umožňuje specifikací

adresy a jména databáze, a způsobu přihlášení stáhnout výsledek dotazu přímo do objektu dataframu.

Následně jsem iterativně nahrával jednotlivé textové soubory s křivkami a pároval je pomocí identifikátoru s daty z databáze. Vše jsem postupně ukládal do slovníku, což je datový typ pythonu, který je charakteristický párováním klíče a hodnoty. Struktura tohoto slovníku je naznačena níže.

- Slovník
  - Identifikátor křivky
    - Data horní křivky elektrického proudu
    - Data horní křivky magnetického toku
    - Data spodní křivky elektrického proudu
    - Data spodní křivky magnetického toku
    - Datum průchodu výrobou
    - Číslo výrobní stanice
    - Výsledek z výroby
    - Parametr z výroby

Důvod, proč jsem data doposud nijak nepředzpracovával je, že tento krok bych musel dělat pokaždé, co bych zkoušel jiný přístup nebo nový krok analýzy. Tudíž takto uložená data jsou univerzální pro navazující práci.

Pro uložení jsem použil knihovnu pickle, která se zaměřuje na serializaci a deserializaci objektové struktury Pythonu. Jinak řečeno se jedná o proces převodu objektu Pythonu na bajtový proud, který ho ukládá do souboru. Následně lze tento proud pak použít k opětovnému vytvoření hierarchie původních objektu.

5.2. Základní analýza dat

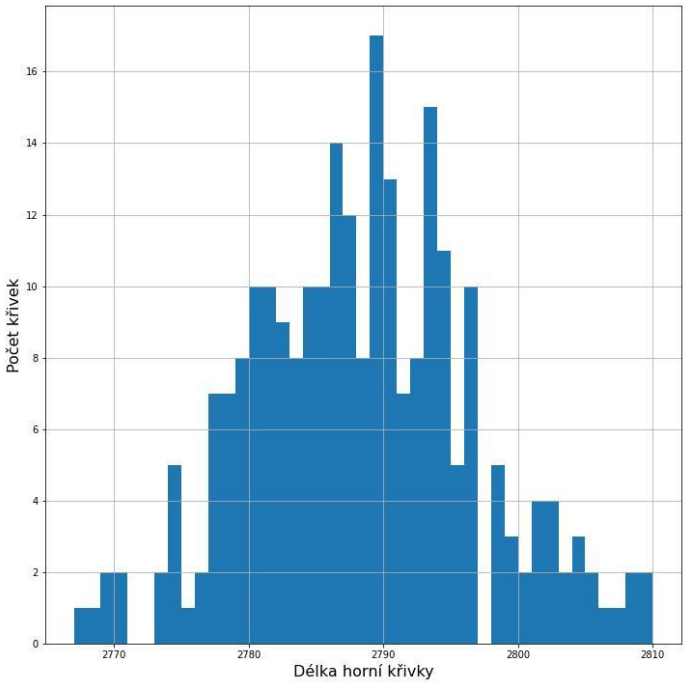
V rámci analýzy jsem se rozhodl vyzkoušet tři různé přístupy, jakými vstupní křivky použít pro následné shlukování a porovnat jednotlivé výsledky. Cílem je získat přehled o důležitosti předzpracování křivek s ohledem na fyzikální pozadí procesu. Přístupy předzpracování pro shlukování jsou následující:

1) Použít celé křivky
2) Použít určitou část nebo části křivky
3) Použít nejdůležitější body křivky

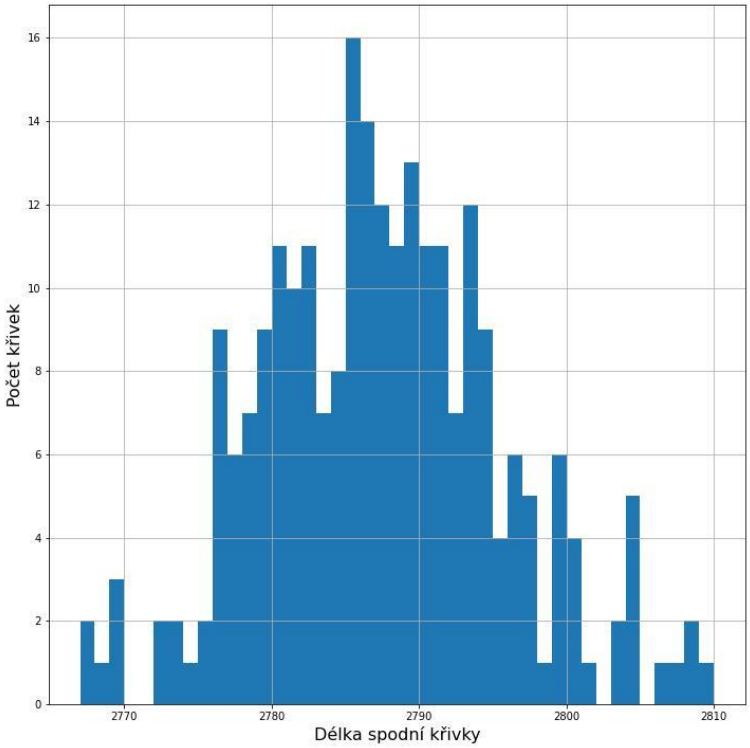
V kapitole 3 bylo zmíněno, že jednotlivé křivky mají různou délku. To je důležité především pro přístupy 1) a 2) uvedené v předchozím odstavci. Na obrázku 5.1 je ukázán histogram, který znázorňuje jednotlivé délky horních křivek. Na obrázku 5.2 je obdobný histogram pro jednotlivé délky spodních křivek. Součástí skriptu, který

délky křivek zpracovával byla i část, která kontrolovala, že proudová část křivky má stejný počet záznamů jako odpovídající část magnetického toku.

Z obrázků je vidět, že délky křivek jsou rozdělené dle Gaussovského rozdělení a pro použití dalších metod a algoritmů bude potřeba zajistit jejich konstantní délku napříč daty.



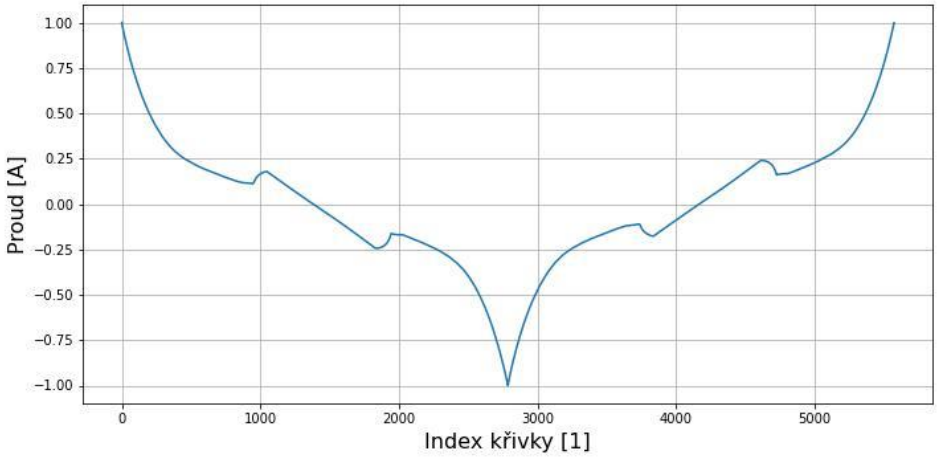
Obrázek 5.1 Histogram délek horních křivek



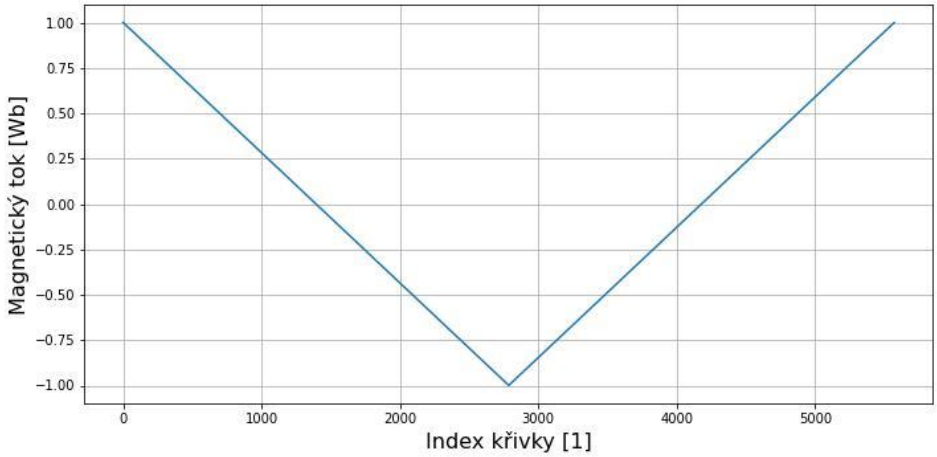
Obrázek 5.2 Histogram délek spodních křivek

Doposud jsem ukazoval křivky pouze v závislosti magnetického toku na proudu. Pro datovou analýzu tento formát ovšem není nejlepší, jelikož standardně se jako vstup do většiny algoritmů shlukování nebo modelů strojového učení používá jeden vektor.

Z toho důvodu jsem vykreslil zvlášť křivku proudu a magnetického toku, jak je možné vidět na obrázku 5.3, respektive 5.4. Na druhém z nich je možné si všimnout, že magnetický tok horní a spodní části křivky má ryze lineární průběh.

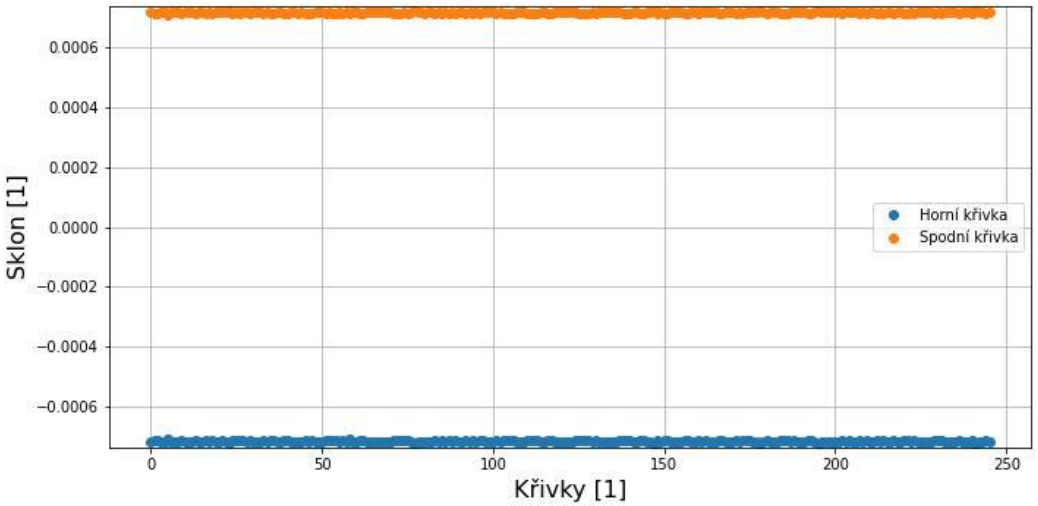


Obrázek 5.3 Ukázka průběhu proudu



Obrázek 5.4 Ukázka průběhu magnetického toku

Na obrázku 5.5 jsou následně vykresleny sklony klesající a stoupající části každé z křivek magnetického toku. Rozdíl mezi minimální a maximální hodnotou sklonu pro horní i spodní křivku je 1.54%. Počáteční body všech křivek jsou vzhledem k předchozímu škálování identické. Z toho vyplývá, že variabilita magnetického toku napříč křivkami je velmi malá, což dělá úkol shlukování složitější. Na základě toho se budu zaměřovat primárně pouze na proudové křivky, které jsou odlišnější, a tudíž zprostředkovávají pro roztřídění do shluků více informací.



Obrázek 5.5 Sklony křivek magnetického toku

Kapitola 6

Praktická část – předzpracování dat

V této kapitole navážu na téma předzpracování dat představené v teoretické části této diplomové práce a postupně přiblížím předzpracování dat pro tři hlavní přístupy, které byli popsány na stránce 32.

Metody pro předzpracování jsou pro jednotlivé přístupy podobné nebo stejné. Proto v rámci podkapitol vždy přiblížím princip a metodiku postupu a následně se na tento popis budu pouze odkazovat, případně ho upravím pro konkrétní přístup.

Jednou z nejčastějších metod, jak docílit vyrovnání délky vstupních dat je takzvaný „padding“. Obecně se jedná o metodu, která je určená k vytváření určitého okolí a používá se mimo datovou analýzu také například během tvoření webových stránek, nebo zpracování obrazu.

V datové analýze se jedná o metodu, která po najití nejdelší křivky v datech všechny ostatní doplní na stejnou délku. Kterou hodnotou jsou data doplněny se může lišit a záleží na přístupu a rozhodnutí datového analisty. Mezi nejčastější patří takzvaný „zero-padding“, který doplní pouze nulové hodnoty. Další možností je duplikování poslední hodnoty křivky, dokud délka nebude dorovnána.

Vzhledem k tomu, že rozsah křivek je od -1 do 1, využiji druhou možnost paddingu z předchozího odstavce. Pro horní klesající křivku tak kratší křivky prodloužím hodnotou -1 a horní křivky prodloužím hodnotou 1. Tímto způsobem budou všechny křivky stejné velikosti a nebudou v nich žádné skokové změny, které by s použitím zero-paddingu nastaly.

Nejdelší z horních křivek má délku 2810 datových bodů. Ke všem ostatním přidám tedy hodnotu -1 tolikrát, dokud této délky také nedosáhne.

Nejdelší ze spodních křivek má délku také 2810 datových bodů, a proto proces budu opakovat, tentokrát ovšem s hodnotou 1.

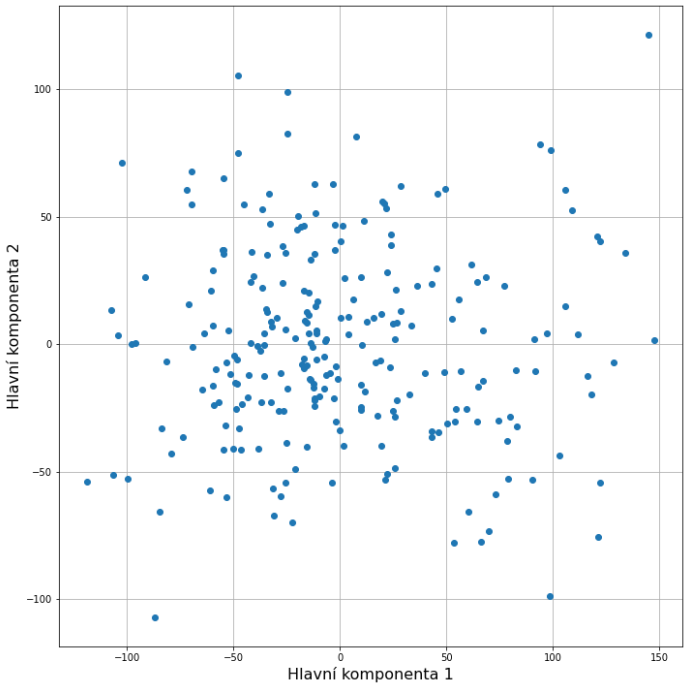
6.1. Použití celé křivky

Všechny křivky mají nyní 5620 datových bodů. To by znamenalo, že při shlukování by 246 křivek reprezentovaly jednotlivé pozorování a 5620 datových bodů by byly parametry každého z těchto pozorování. Celý dataset pro další zpracování je tedy nyní k dispozici v jediném poli o rozměru (246, 5620). Takové to množství parametrů může být pro algoritmy představené v kapitole 4 problém. Z toho důvodu jsem použil algoritmy pro redukci vstupních dimenzí, viz kapitola 3.3.

Ty použiji prvně pro redukci dimenzí z původních 5620 hodnot na 2 a 3 dimenze, speciálně pro možnost vizualizace ve grafu. Takto velká redukce ovšem povede ke ztrátě informací a s největší pravděpodobností bude potřeba zvolit více parametrů. Proto následně provedu redukci na 4, 8 a 16 parametrů a porovnám výsledky pro následující použití do shlukovacích algoritmů.

Analýza hlavních komponent

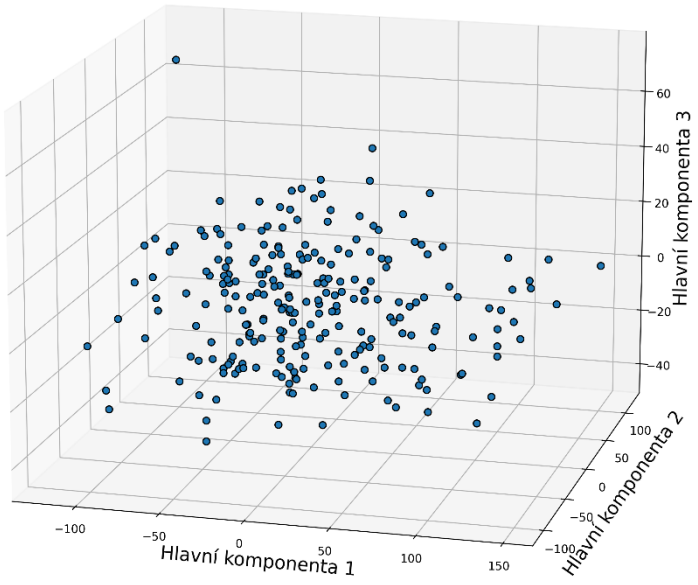
Pomocí analýzy hlavních komponent zredukuji postupně vstupní dimenze dat. Při použití je možné přímo zjistit konkrétní množství zachovaného rozptylu. Ten odpovídá poměru sumy použitých vlastních čísel po redukci dimenze vůči celkové sumě všech vlastních čísel kovarianční matice. Pro výpočet jsem použil implementaci algoritmu knihovny sklearn [21].



Obrázek 6.1 Dvě hlavní komponenty pro celé křivky

Pro dimenzi 2 bylo zachováno 79,37 % celkového rozptylu. Na obrázku 6.1 je možné si všimnout, že data na první pohled netvoří rozlišitelné shluky, ale jsou rovnoměrně rozprostřeny po ploše, kterou hlavní dvě komponenty tvoří.

Pro dimenzi 3 bylo zachováno 85,98% původní variance. Na vizualizaci na obrázku 6.2 je vidět, že ani tato redukce s největší pravděpodobností nebude mít při shlukování smysluplné výsledky. To potvrdilo můj předpoklad, vzhledem k podobnosti veškerých křivek. Proto budu postup opakovat pro vyšší dimenze, které už z pochopitelných důvodů vizualizovat nelze.



Obrázek 6.2 Tři hlavní komponenty pro celé křivky

Pro vyšší dimenze pak bylo zachování rozptylu následující:

- 4 dimenze – 89,60 %
- 8 dimenzí – 96,43 %
- 16 dimenzí – 98,68 %

V tomto kroku analýzy předpokládám, že nejlepší řešení je použít redukci na 8 dimenzí, které výrazně zmenšuje velikost vstupních dat, ale současně zachovává 96,43 % původního rozptylu.

Autoenkodér

Pro snížení dimenze použiji také symetrický autoenkodér, jehož struktura je naznačena na obrázku 6.3. Vstupní vrstva autoenkodéru bere jako vstup 5620 parametrů. Dalších 5 vrstev má vstup přibližně poloviční oproti té předchozí a postupně tak snižují počet parametrů, se kterými pracují. Za poslední vrstvou kódovací části autoenkodéru je kódová vrstva, která má pouze 8 parametrů. Následuje dekódovací část, které je symetrická s kódovací a jejím výstupem je tak opět 5620 parametrů.

Layer (type)	Output Shape	Param #
encoder10 (Dense)	(None, 1405)	7897505
encoder20 (Dense)	(None, 702)	987012
encoder30 (Dense)	(None, 351)	246753
encoder40 (Dense)	(None, 175)	61600
encoder50 (Dense)	(None, 87)	15312
latent_layer (Dense)	(None, 8)	704
decoder10 (Dense)	(None, 87)	783
decoder20 (Dense)	(None, 175)	15400
decoder30 (Dense)	(None, 351)	61776
decoder40 (Dense)	(None, 702)	247104
decoder50 (Dense)	(None, 1405)	987715
output_layer (Dense)	(None, 5620)	7901720

Total params: 18,423,384
Trainable params: 18,423,384
Non-trainable params: 0

Obrázek 6.3 Struktura autoenkodéru pro celé křivky

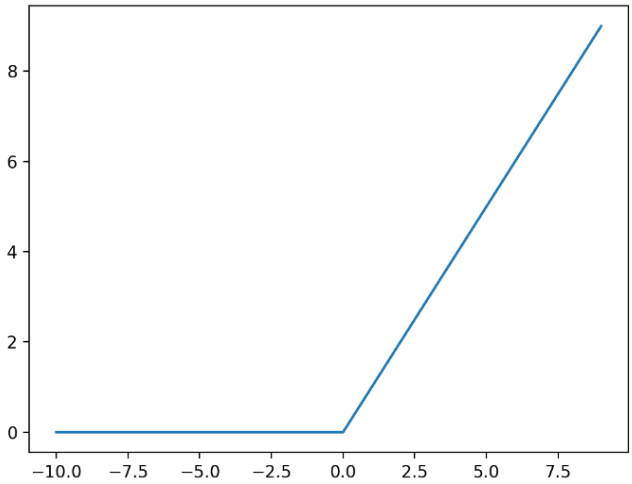
Aktivační funkce v autoenkodéru jsou použity dvě – lineární a relu. Jejich střídání napříč vrstvami se mi v minulosti osvědčilo, a proto jsem postup v této diplomové práci zopakoval. Nicméně přepokládám, že volba jiných aktivačních funkcí nebo jejich umístění může vést k potencionálnímu zlepšení modelu.

Lineární aktivační funkce je definována jako přímka se sklonem jedna dle rovnice 6.1.

$$f(x) = x \tag{6.1.1}$$

Druhou použitou aktivační funkcí je relu, která je definována rovnicí 6.1.2 a její průběh je ukázán na obrázku 6.4.

$$f(x) = \max(0, \mathrm{x}) \tag{6.1.2}$$



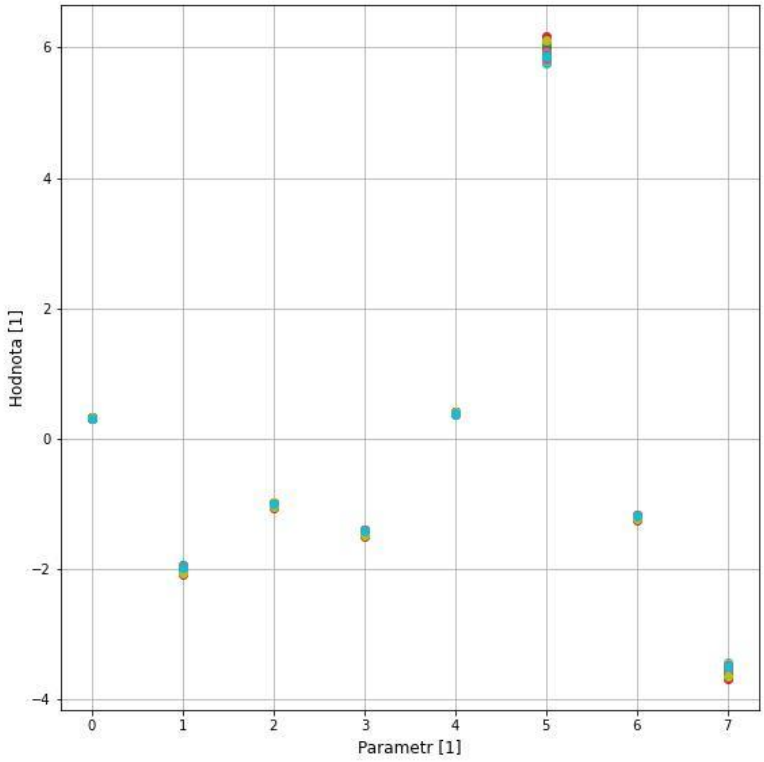
Obrázek 6.4 Průběh relu funkce

Význam parametrů pro kompilaci a učení byl představen v teoretické části o autoenkodéru, viz strana 15. Jejich nastavení je následující:

- Kompilace
  - Optimizer – adam
  - Ztrátová funkce – střední kvadratická chyba
- Učení
  - Počet epoch – 16
  - Batch size – 4
  - Validation split – 0.2

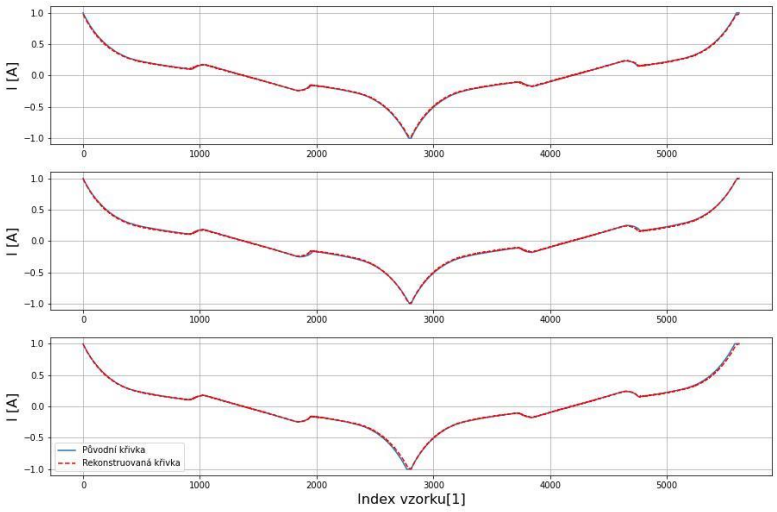
Po dokončení učení mohu ze struktury autoenkodéru extrahovat kódovou vrstvu, díky které jsem schopný získat 8 parametrů pro každou z křivek, ze kterých je autoenkodér schopen rekonstruovat její původní tvar.

Na obrázku 6.5 je ukázána hodnota parametrů 30 křivek. Je možné si všimnout, že některé z parametrů napříč křivkami prakticky nemění svou hodnotu, například parametr 0. Naopak u parametru 5 a 7 je rozptyl větší a bude tak při shlukování přínosnější.



Obrázek 6.5 Parametry kódové vrstvy pro celé křivky

Pro bližší ukázku funkce autoenkodéru jsou na obrázku 6.6 ukázány tři porovnání původní křivky s její rekonstrukcí.



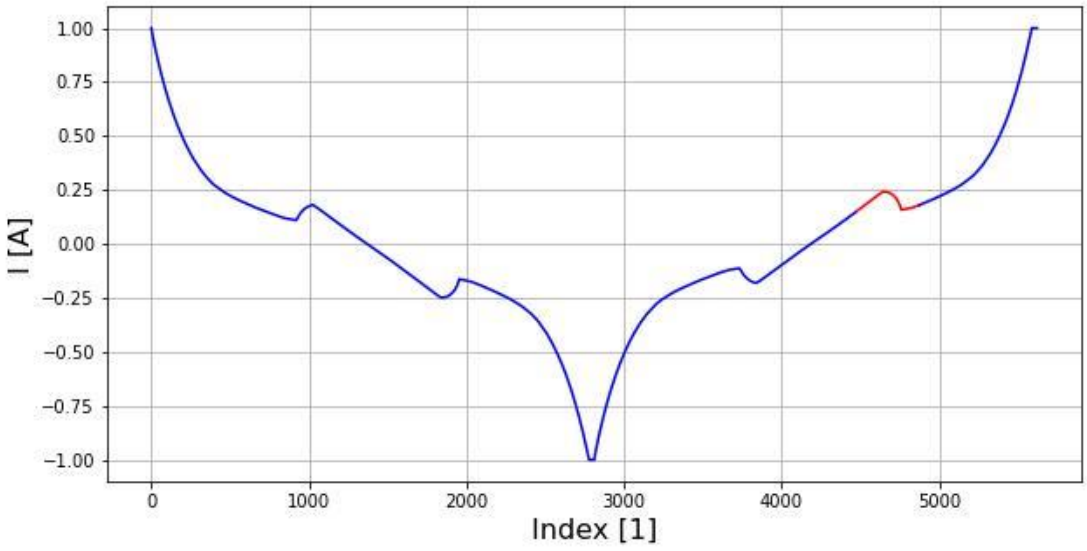
Obrázek 6.6 Porovnání celé křivky s predikcí autoenkodéru

6.2. Použití části křivky

Při fyzikální analýze křivek v kapitole 3.1 jsem popisoval část procesu nasávání. Zmiňoval jsem zde důležitost této části křivky, proto zkusím analýzu na tuto konkrétní část zaměřit, jelikož začátek a konec mechanického pohybu vytlačování by mohl přímo ovlivňovat výsledek z výrobních stanic. Pro analýzu tedy použiji určitou část křivky, u které by mohly být jednotlivé odlišnosti více koncertované, oproti použití celých křivek.

Je důležité rozhodnout, jakou metodikou části křivek vybírat napříč dostupnými daty. Začátek části bude dán hodnotou proudu 0.1 A a fixním počtem 400 následujících bodů. Ukázka takovéto části je červeně znázorněna na obrázku 6.7.

Druhou možností volby začátku by byl fixní index datového vzorku na křivce. To ovšem vzhledem k použitému předzpracování paddingem může být zavádějící, a proto metodiku v předchozím odstavci považuji za vhodnější.



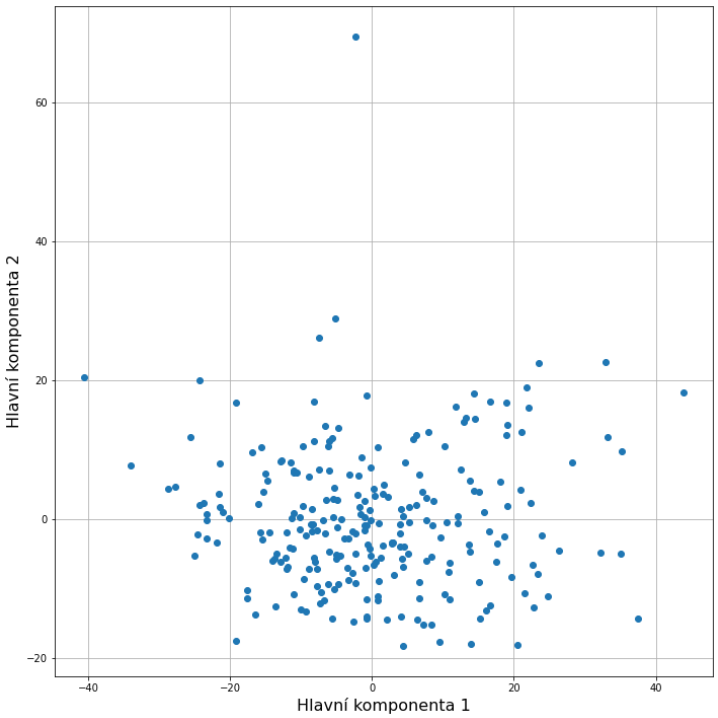
Obrázek 6.7 Použitá část křivky

Analýza hlavních komponent

Při použití analýzy hlavních komponent na popsanou část křivky a snížení původních 500 vstupních parametrů na 2 došlo k zachování 86.26 % rozptylu. Ukázka je možná vidět na obrázku 6.8. Je možné si všimnou výskytu několika odlehlých hodnot, které ve vizualizaci celé křivky nebyly. To naznačuje, že je možné očekávat kvalitnější shlukování v této oblasti křivky.

Analýzu hlavních komponent aplikuji i pro snížení na 3, 4, 8 a 16 dimenzí, pro něž je zachování rozptylu následující:

- 3 dimenze – 86,66 %
- 4 dimenze – 91,18 %
- 8 dimenzí – 97,72 %
- 16 dimenzí – 99,25 %



Obrázek 6.8 Dvě hlavní komponenty pro část křivky

Autoenkodér

Pro snížení dimenze použiji opět i autoenkodér. Jeho struktura je naznačena na obrázku 6.9 a je velmi podobná struktuře z kapitoly 6.1.

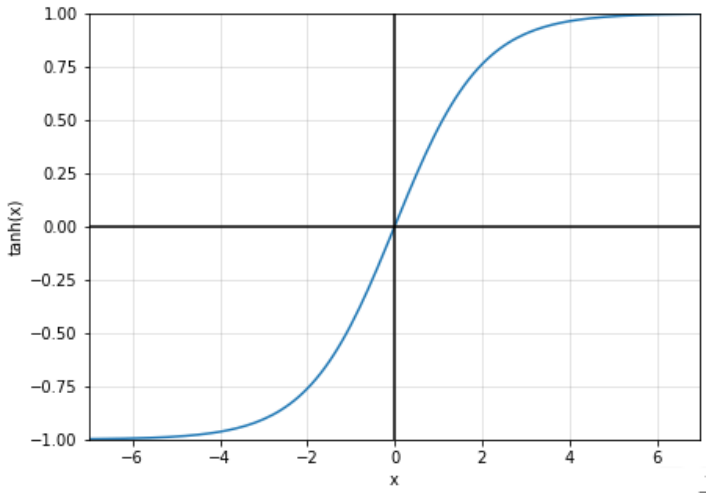
Layer (type)	Output Shape	Param #
encoder10 (Dense)	(None, 200)	80200
encoder20 (Dense)	(None, 100)	20100
encoder30 (Dense)	(None, 50)	5050
encoder40 (Dense)	(None, 25)	1275
encoder50 (Dense)	(None, 12)	312
latent_layer (Dense)	(None, 8)	104
decoder10 (Dense)	(None, 12)	108
decoder20 (Dense)	(None, 25)	325
decoder30 (Dense)	(None, 50)	1300
decoder40 (Dense)	(None, 100)	5100
decoder50 (Dense)	(None, 200)	20200
output_layer (Dense)	(None, 400)	80400

Total params: 214,474
Trainable params: 214,474
Non-trainable params: 0

Obrázek 6.9 Struktura autoenkodéru pro část křivky

V tomtu modelu jsem jako aktivační funkci použil tanh, jelikož jsem dosahoval lepších výsledků rekonstrukce. Její předpis je definován rovnicí 6.2.1 a průběh je naznačený na obrázku 6.10.

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{6.2.1}$$

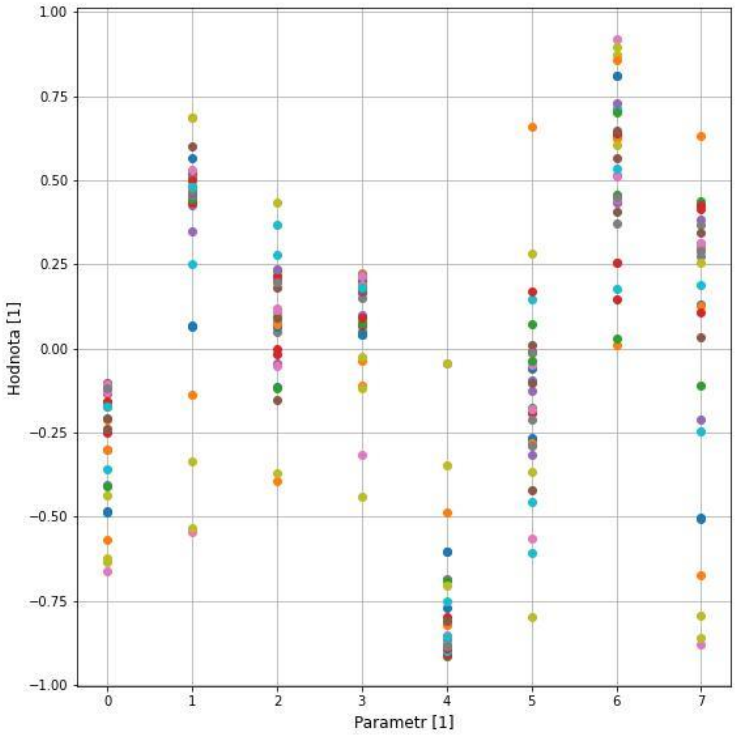


Obrázek 6.10 Průběh tanh funkce

Nastavení parametrů pro kompilaci a učení je následující:

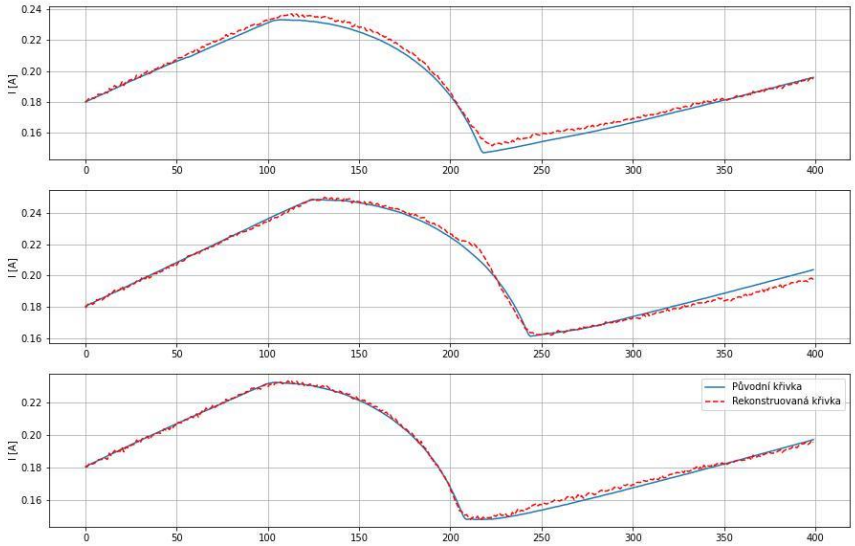
- Kompilace
  - Optimizer – adam
  - Ztrátová funkce – střední kvadratická chyba
- Učení
  - Počet epoch – 512
  - Batch size – 4
  - Validation split – 0.2

Na obrázku 6.11 jsou opět znázorněny parametry pro 30 různých křivek. Lze si oproti použití celých křivek všimnout výrazně většího rozptylu křivek. Tento fakt je samozřejmě spojený s použitím jiné aktivační funkce v struktuře autoenkodéru, nicméně z pohledu shlukování lze očekávat, že kombinace parametrů umožní lepší shlukování.



Obrázek 6.11 Parametry kódové vrstvy pro část křivky

Při predikci 3 náhodných křivek na obrázku 6.12 je možné si všimnout, že není úplně dokonalá. Domnívám se, že to při shlukování to nebude znamenat problém, ale je důležité na tuto skutečnost brát zřetel a v případě neuspokojivých výsledků je tato část jedním z míst pro potencionální vylepšení.



Obrázek 6.12 Porovnání predikce části křivky

6.3. Použití nejdůležitějších bodů

V této části se zaměřím pouze na tři fyzikálně nejdůležitější hodnoty při nasávání. Cílem je porovnat důležitost použití celé, respektive části křivky oproti minimu ručně vybraných hodnot, o kterých ovšem víme, že mají pro proces veliký fyzikální význam.

Prvním bodem je hodnota sklonu před začátkem mechanického pohybu. Zjištění hodnoty provedu odečtením hodnot magnetického toku pro proud 0.05 A a 0.2 A a vypočtením sklonu mezi nimi dle rovnice 6.3.1.

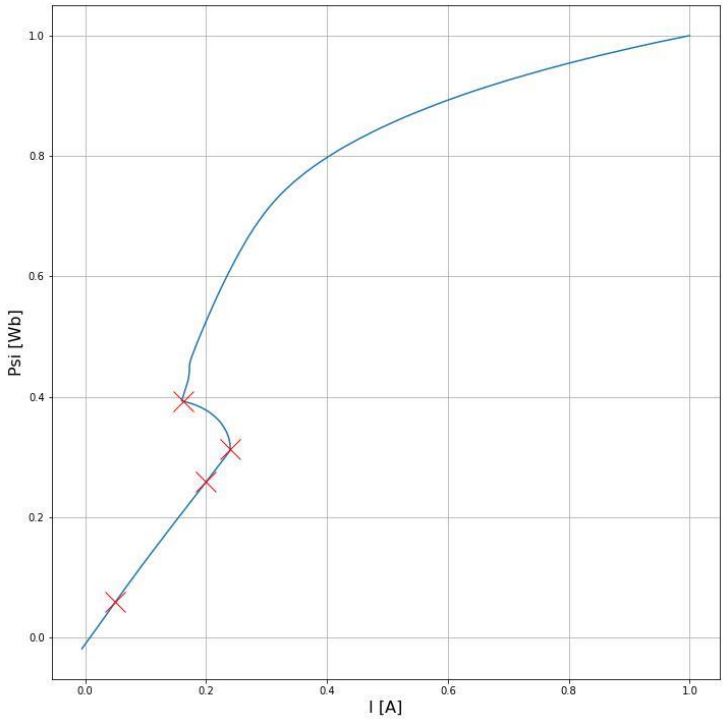
$$m = \frac{y_2 - y_1}{x_2 - x_1} \tag{6.3.1}$$

Druhým bodem je bod začátku mechanického pohybu nasávání. Ten mohu najít jako první tři po sobě klesající hodnoty proudu v části nasávání čerpadla.

Třetím bodem je konec mechanického pohybu nasávání. Při jeho hledání mohu navázat na předchozí bod, a naopak najít prvních 5 po sobě stoupajících hodnot proudu po začátku mechanického pohybu.

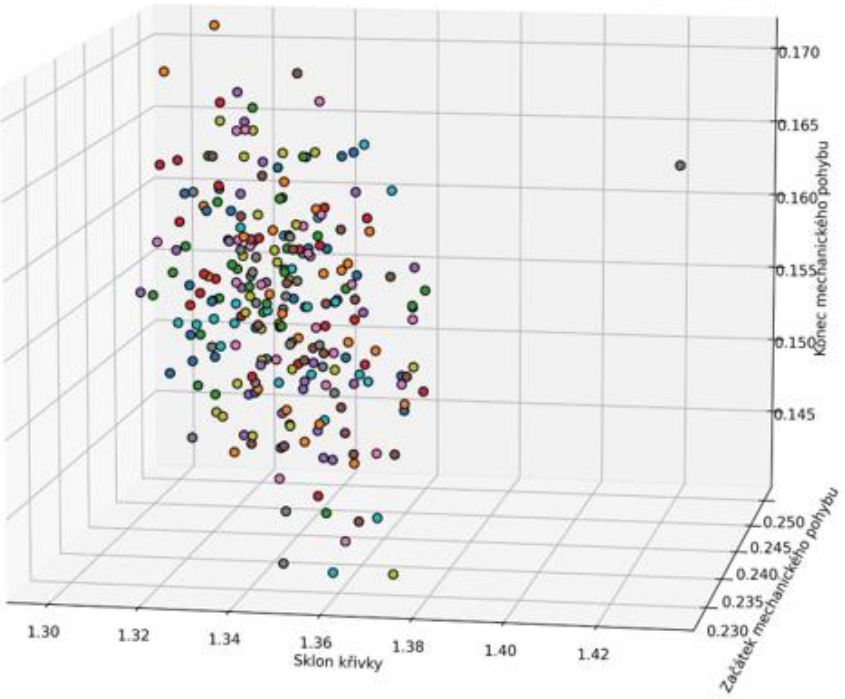
Pro hledání druhé a třetího bodu touto metodou bylo nutné prvně křivku vyhladit, jelikož nebylo možné zaručit konstantní růst nebo pokles hodnot. K tomu jsem použil digitální filtr Savitzky-Golay [25], který je založen na principu prokládání menších výseků dat polynomiální křivkou. Velikost výseku a řád polynomiální křivky lze v této implementaci definovat jako vstup metody.

Funkčnost vybírání bodů jsem experimentálně ověřil a získal 100 % úspěšnost. Ukázka bodů pro jednu konkrétní křivku jsou na obrázku 6.13.



Obrázek 6.13 Ukázka použitých bodů

Na obrázku 6.14 jsou tři získané parametry vizualizované do tří osového grafu. Tyto parametry mohu následně přímo aplikovat do algoritmů shlukování.



Obrázek 6.14 Vizualizace třech použitých parametrů

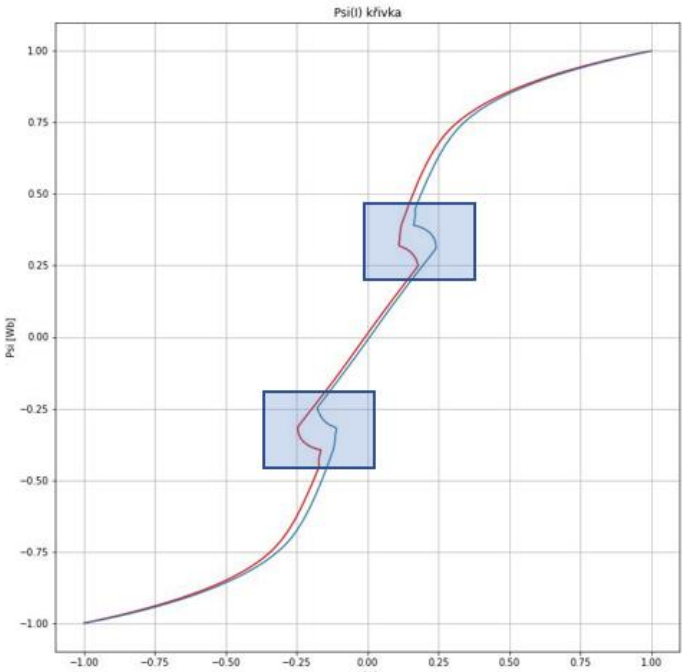
Kapitola 7

Praktická část – shlukování

V této kapitole použiji algoritmy shlukování na všechny tři postupy z kapitoly 6 o předzpracování dat. Cílem je porovnat výsledky jednotlivých přístupů v kombinaci s metodami shlukování a zároveň zjistit, zda je možné vytvořené shluky nějakým způsobem použít pro odhad měřené veličiny na výrobních stanicích.

Vzhledem k použití dvou různých metod předzpracování, třech různých přístupů a třech různých metod shlukování by vykreslení každého obrázku dělalo tuto kapitolu až příliš nepřehlednou. Z toho důvodu zde popíši principálně použití a vykreslím ty grafy, které budou dle mého názor pro čtenáře nejobsáhlejší. Všechny ostatní grafy jsou dostupné v souboru *clustering.ipynb* a příloze diplomové práce. V rámci shlukování jsem ve veškerých náhodných jevech používal náhodný seed pro opakovatelnost shlukování. Pro přístupy 1) a 2) použiji reprezentaci 8 parametrů z analýzy hlavních komponent i autonekodéru. Implementaci algoritmů shlukování jsem použil z knihovny scikit [26].

V rámci vizualizace se nejvíce zaměřuji na 4 oblasti mechanického pohybu membrány čerpadla, které jsou znázorněny na obrázku 7.1. V jedné vizualizaci zpravidla zobrazuji okolo 20 křivek, jelikož větší počet vede k nepřehlednosti.



Obrázek 7.1 Zvýraznění oblastí mechanického pohybu membrány

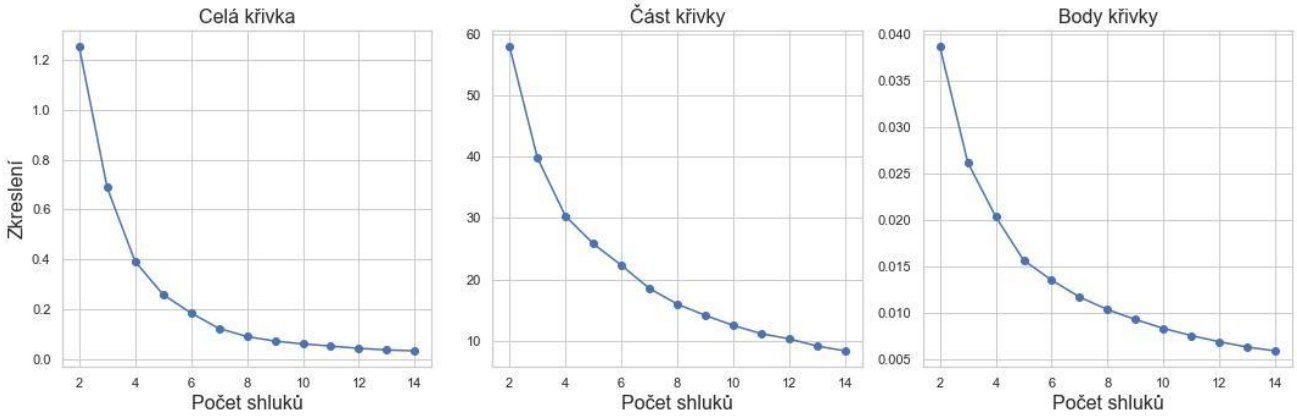
Druhou částí vizualizace je graf jádrového odhadu hustoty (KDE) s parametry výrobní stanice. Jedná se o metodu vizualizace rozložení pozorování v souboru dat, obdoba histogramu. KDE reprezentuje data pomocí spojité křivky hustoty pravděpodobnosti v jedné nebo více dimenzích. Graf lze připodobnit histogramům, jejichž horními body je proložena spojitá křivka.

V tomto grafu je vždy šest obdobných grafů pod sebou, kde každý odpovídá jedné výrobní stanici. V ideálním případě by se křivky neměly překrývat a shluky by jednoznačně oddělovaly určité hodnoty výrobního parametru. V realitě budou křivky s největší pravděpodobností překryté.

7.1. Počet shluků

Při výběru počtu shluků jsem porovnával všechny tři přístupy, s tím, že jako reprezentaci celé křivky a její části používám parametry autoenkodéru, a to kvůli jeho schopnosti zachytit nelinearity křivek.

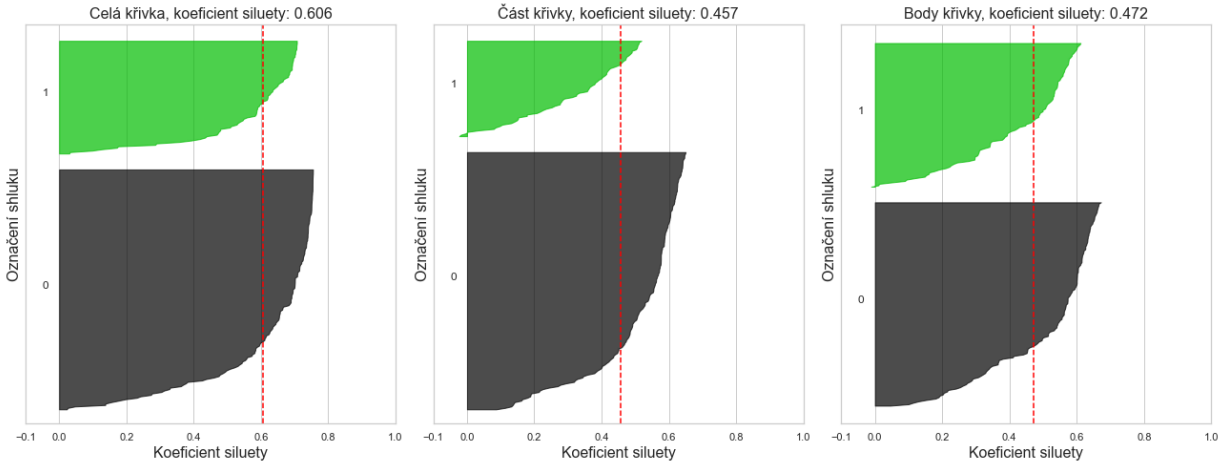
Na obrázku 7.2 je ukázána elbow method pro tři přístupy a rozmezí shluků od 2 do 14. Vzhledem k vyšším dimenzím a velké podobnosti jednotlivých pozorování má křivka exponenciální tvar a není jednoznačně jasné, kde se bod zlomu nachází. Čistě na základě této metody bych určil jako ideální počet shluků číslo 6.



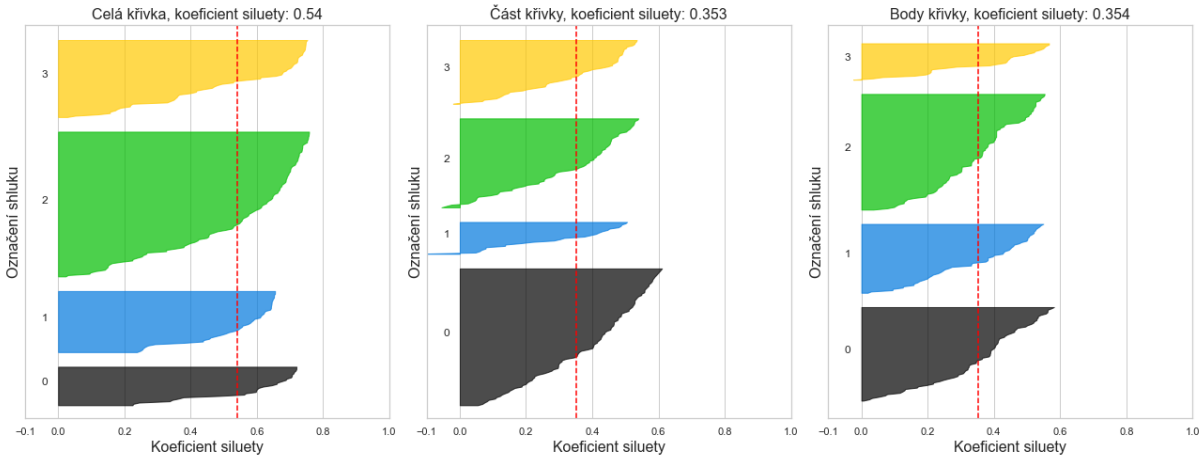
Obrázek 7.2 Porovnání elbow method pro jednotlivé přístupy

Předchozí odstavec ukázal, proč je důležité používat pro zjištění počtu shluků více metod. Ne vždy je jedna metoda dostatečně jednoznačná. Metoda siluety bývá obecně brána jako přesnější, obzvlášť pro takové to použití, kde shluky mají velikou podobnost.

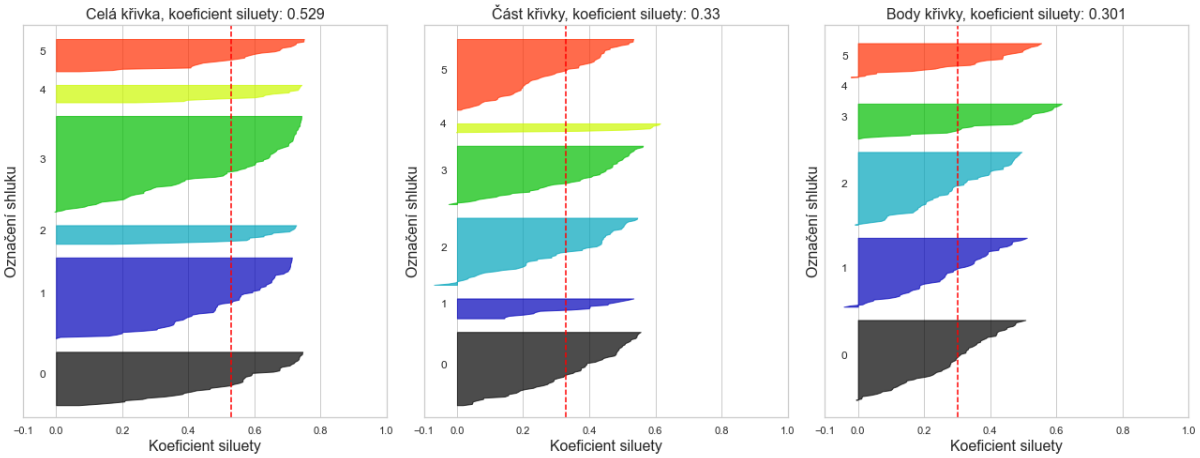
Na obrázcích 7.3 až 7.5 jsou ukázány tři výsledky metody siluety, u kterých byl formát vstupních dat stejný jako u elbow method. V každém nadpisu konkrétního grafu je uveden průměr koeficientu siluety, na základě kterého se posuzuje správnost rozdělení.



Obrázek 7.3 Metoda siluety pro dva shluky



Obrázek 7.4 Metoda siluety pro čtyři shluky



Obrázek 7.5 Metoda siluety pro šest shluků

Seznam průměrů koeficientu siluety pro počet shluků od 2 do 8 je v následující tabulce 7-1:

Počet shluků	Celá křivka	Část křivky	Body křivky
2	0,606	0,457	0,472
3	0,520	0,366	0,406
4	0,540	0,353	0,354
5	0,543	0,359	0,355
6	0,529	0,330	0,301
7	0,553	0,340	0,312
8	0,573	0,346	0,306

Tabulka 7-1 Seznam průměrů koeficientů siluety

V tabulce 7-1 je možné se všimnout, že největší průměr koeficientu siluety napříč přístupy je pro dva shluky, dále pak pro tři shluky, nicméně výsledky vyšších počtů shluků jsou již podobné. Ačkoliv se výsledek této metody liší od výsledků elbow method, dávám ji na základě rešerše a vlastních zkušeností větší váhu, a proto se v dále budu primárně věnovat shlukování s dvěma a třemi počty shluků. Nicméně v rámci analýzy jsem všechny možnosti otestoval.

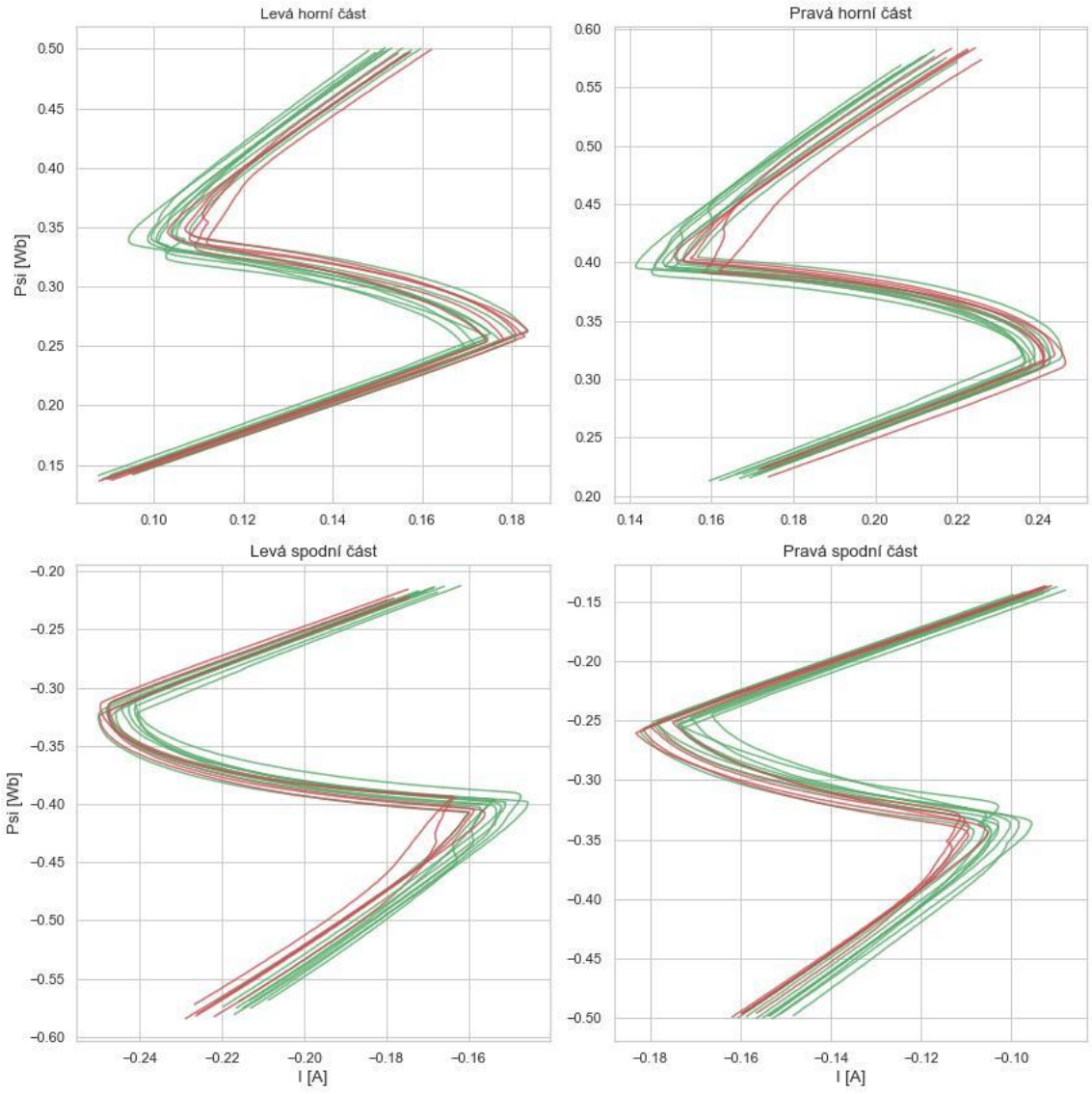
7.2. Celá křivka

Nejlepších výsledků jsem pro shlukování celých křivek dosahoval při použití parametrů z autoenkodéru ve spojení s algoritmem k-means a spektrálním shlukováním. Na obrázku 7.6 je možné vidět výsledek shlukování na křivkách a na obrázku 7.7 výsledek shlukování s hodnotou parametrů v rámci jednotlivých výrobních stanic. Je možné vidět, že shlukování proběhlo primárně po okrajích křivek. Tento jev se objevuje napříč všemi metodami.

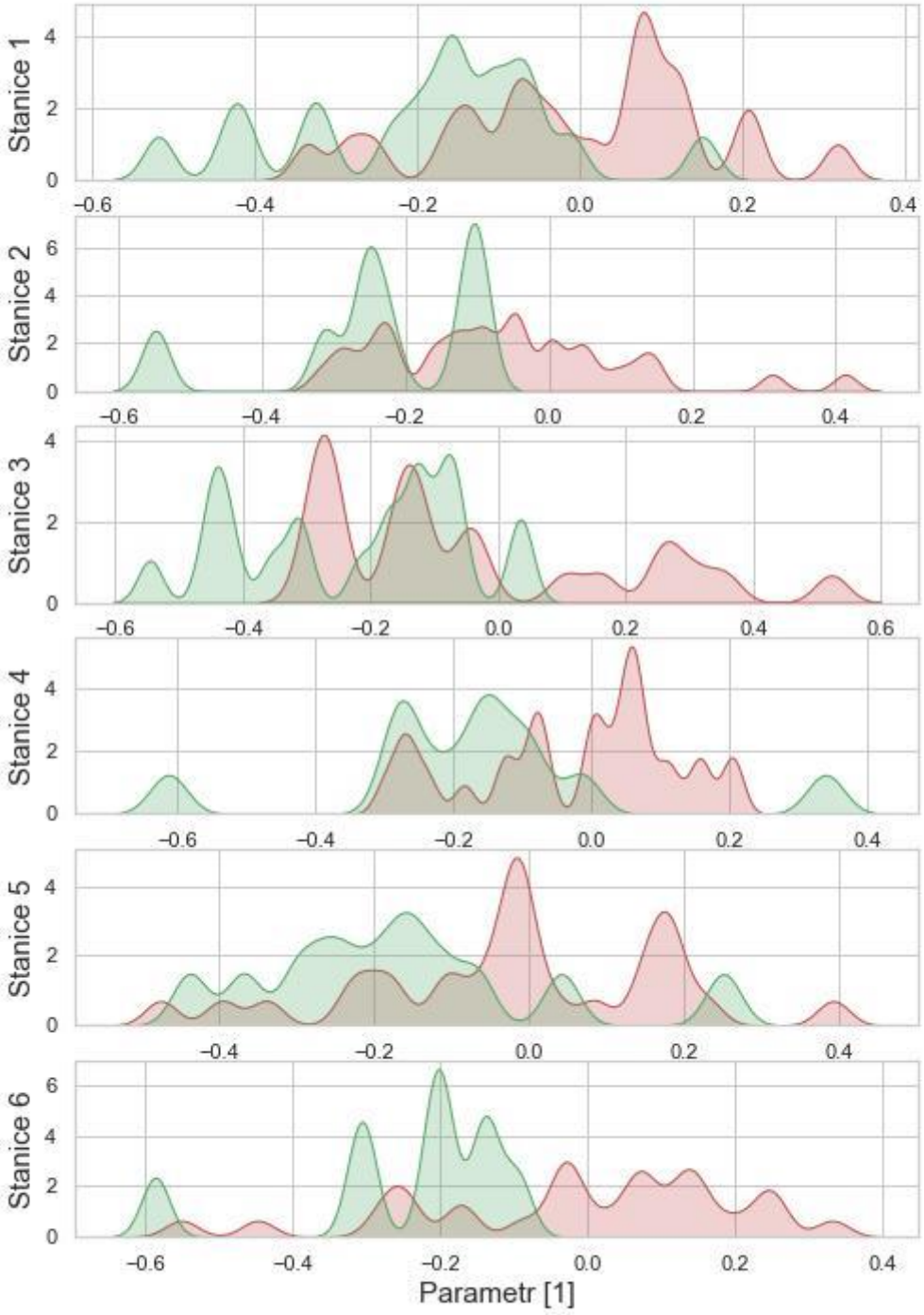
Výrobní parametr lze pomocí shluků hrubě rozdělit do kladné a záporné části. Toto rozdělení není stoprocentní, ale napříč stanicemi se ve většině případů opakuje.

Parametry shlukování:

- Předzpracování – autoenkodér
- Použitý algoritmus – k-means
- Počet shluků – 2



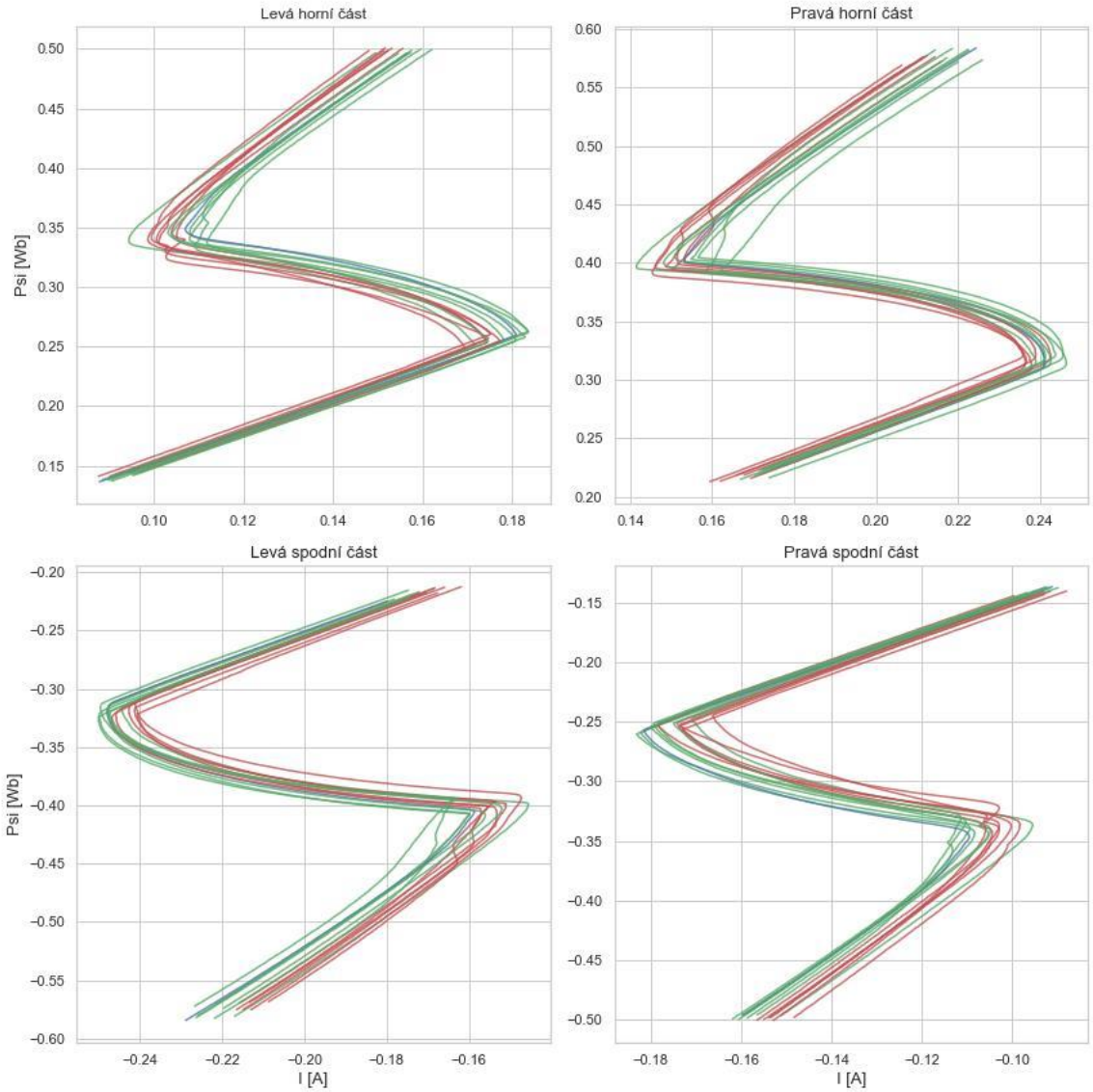
Obrázek 7.6 Shlukování celé křivky algoritmem k-means



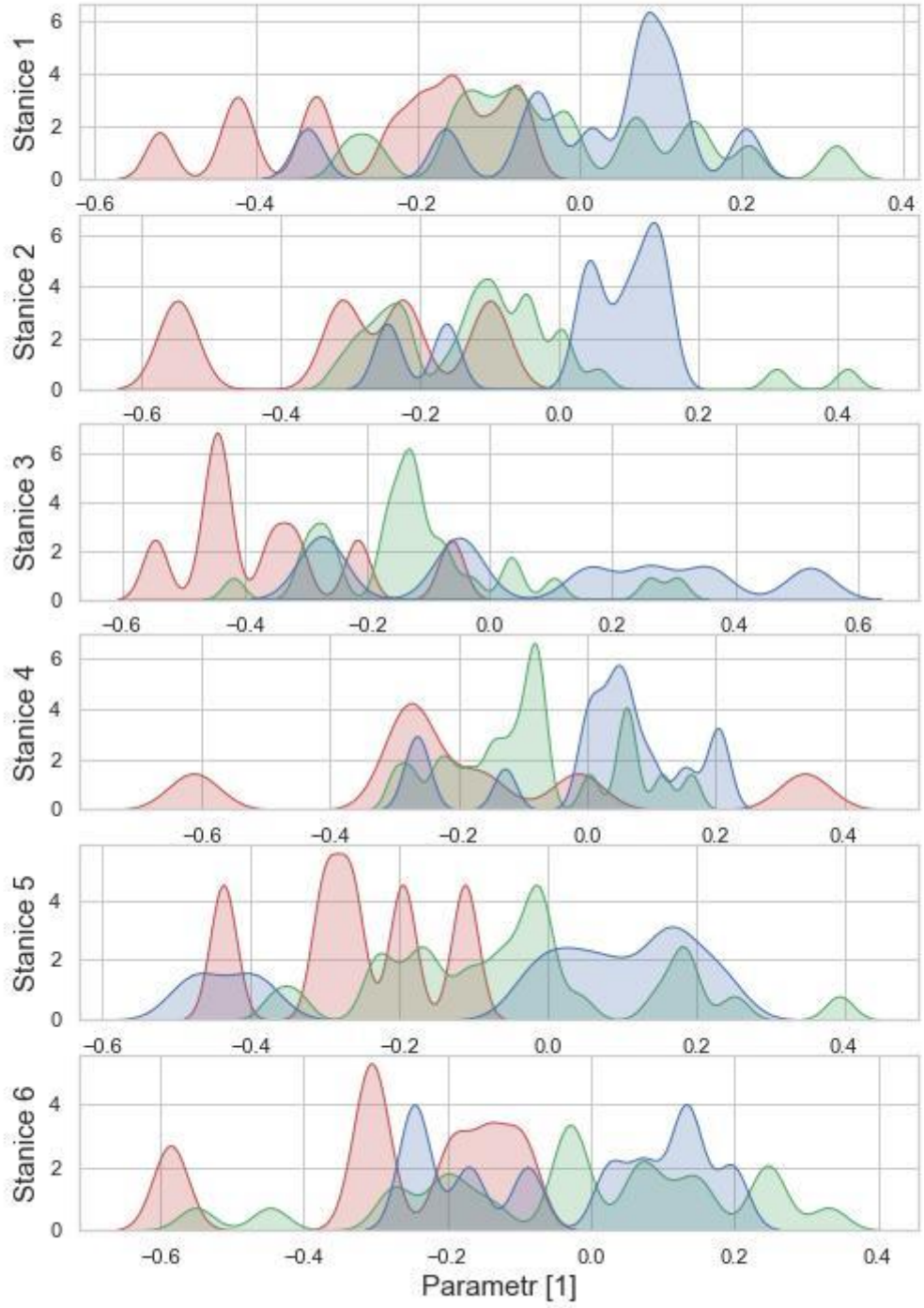
Obrázek 7.7 Výsledky shlukování celé křivky algoritmem k-means

Parametry shlukování:

- Předzpracování – autoenkodér
- Použitý algoritmus – spektrální shlukování
- Počet shluků – 3



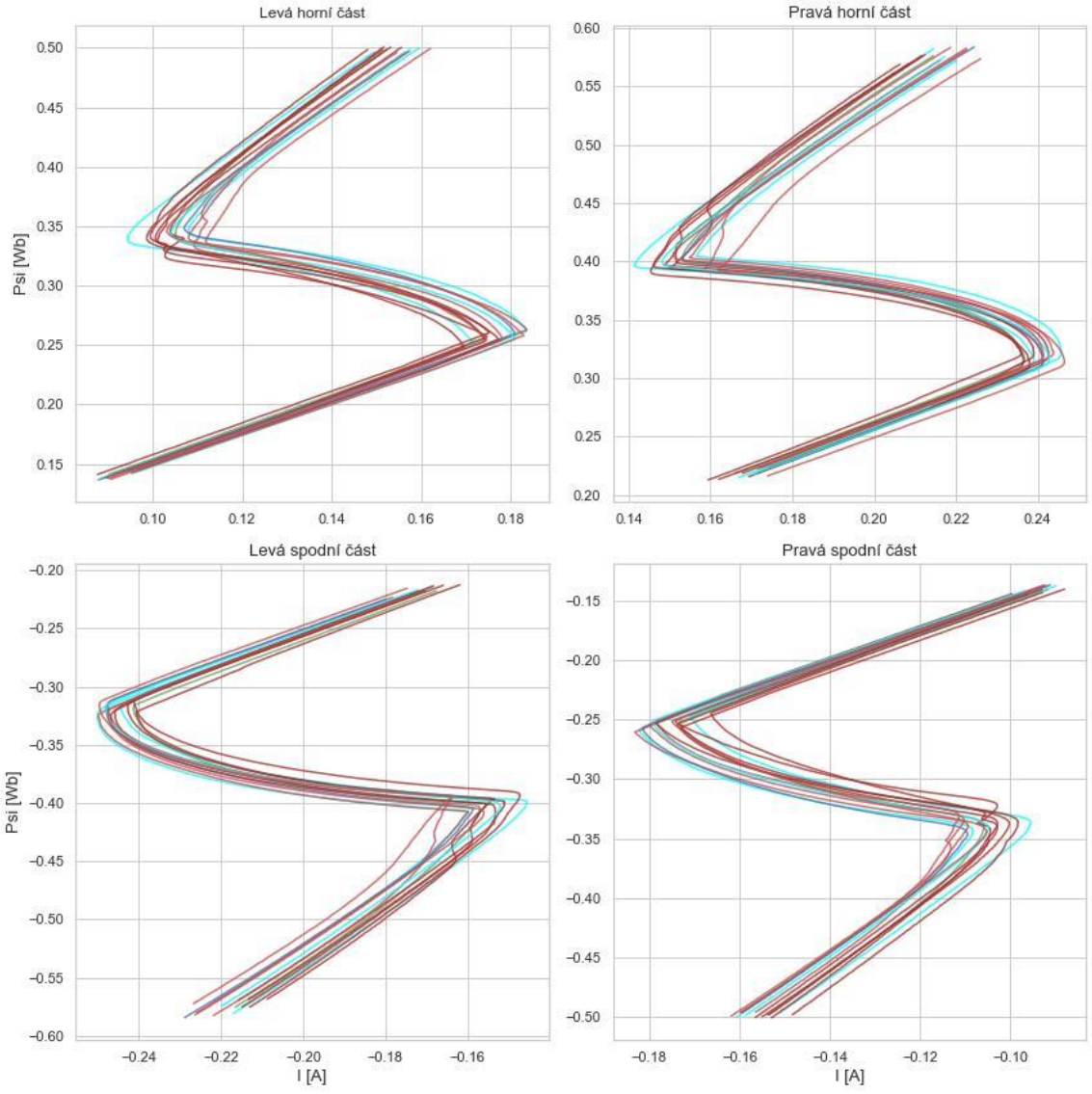
Obrázek 7.8 Shlukování celé křivky algoritmem spektrálního shlukování



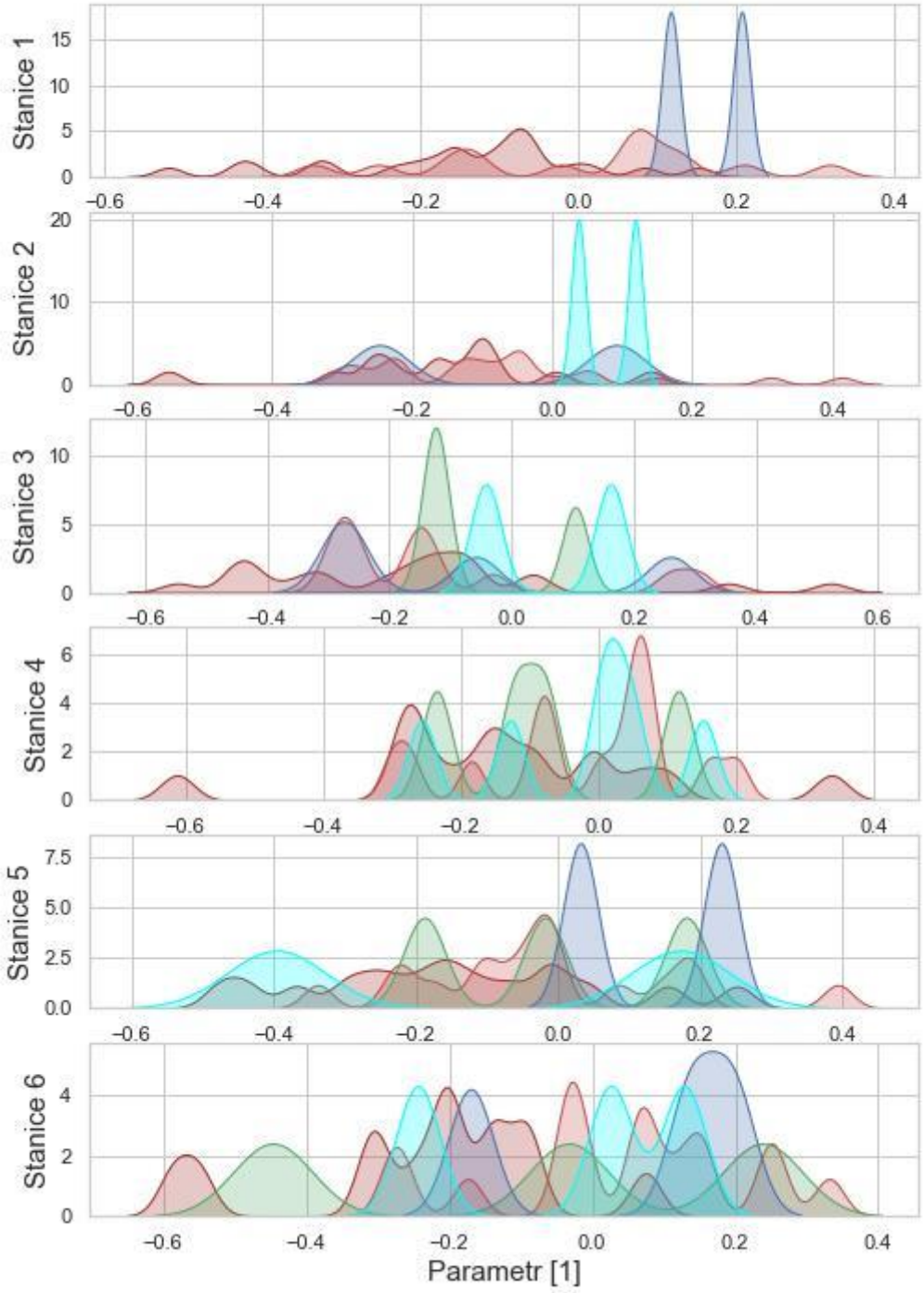
Obrázek 7.9 Výsledky shlukování celé křivky algoritmem spektrálního shlukování

Parametry shlukování:

- Předzpracování – autoenkodér
- Použitý algoritmus – DBSCAN
- Minimum bodů – 10
- Epsilon – 0,01



Obrázek 7.10 Shlukování celé křivky algoritmem DBSCAN



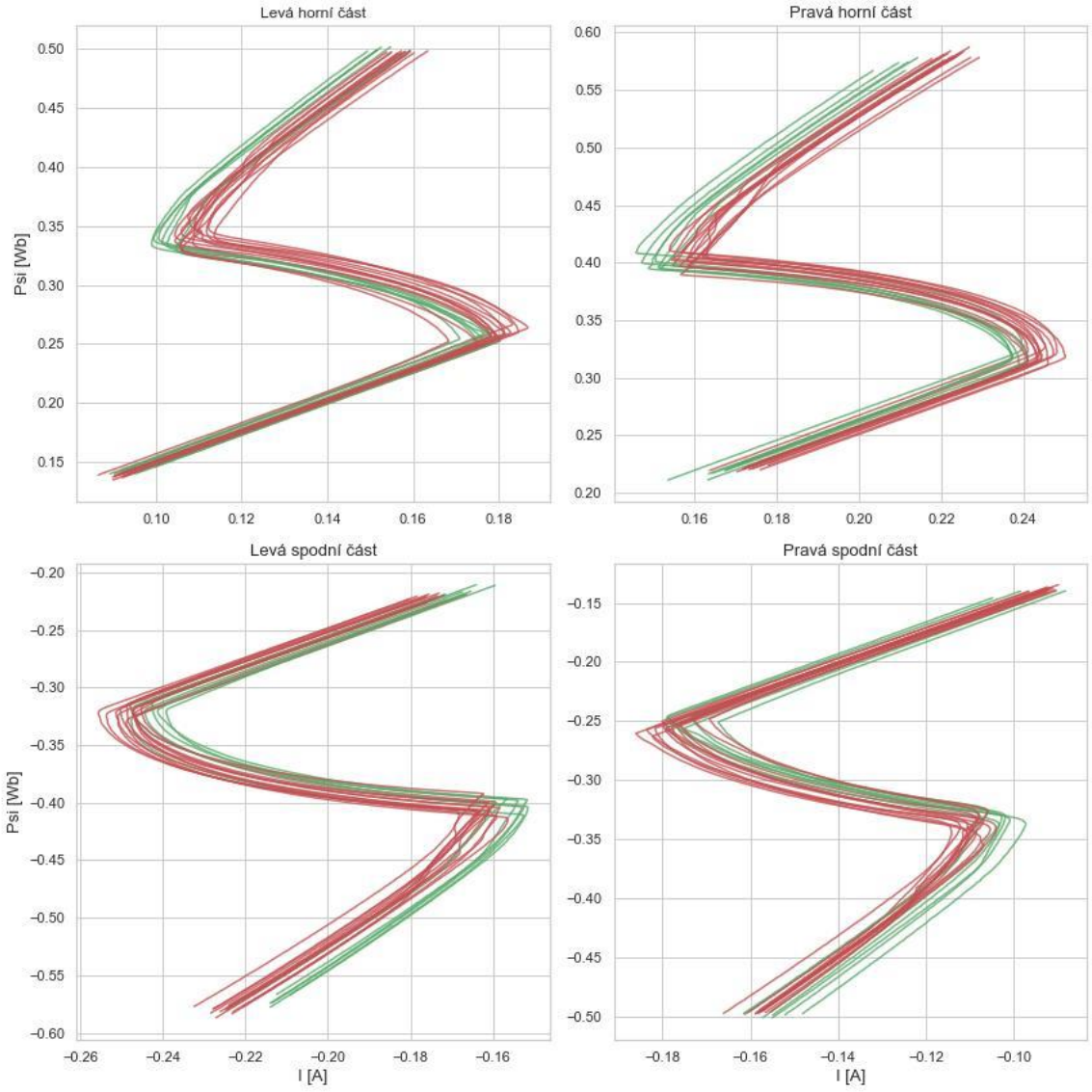
Obrázek 7.11 Výsledky shlukování celé křivky algoritmem DBSCAN

7.3. Část křivky

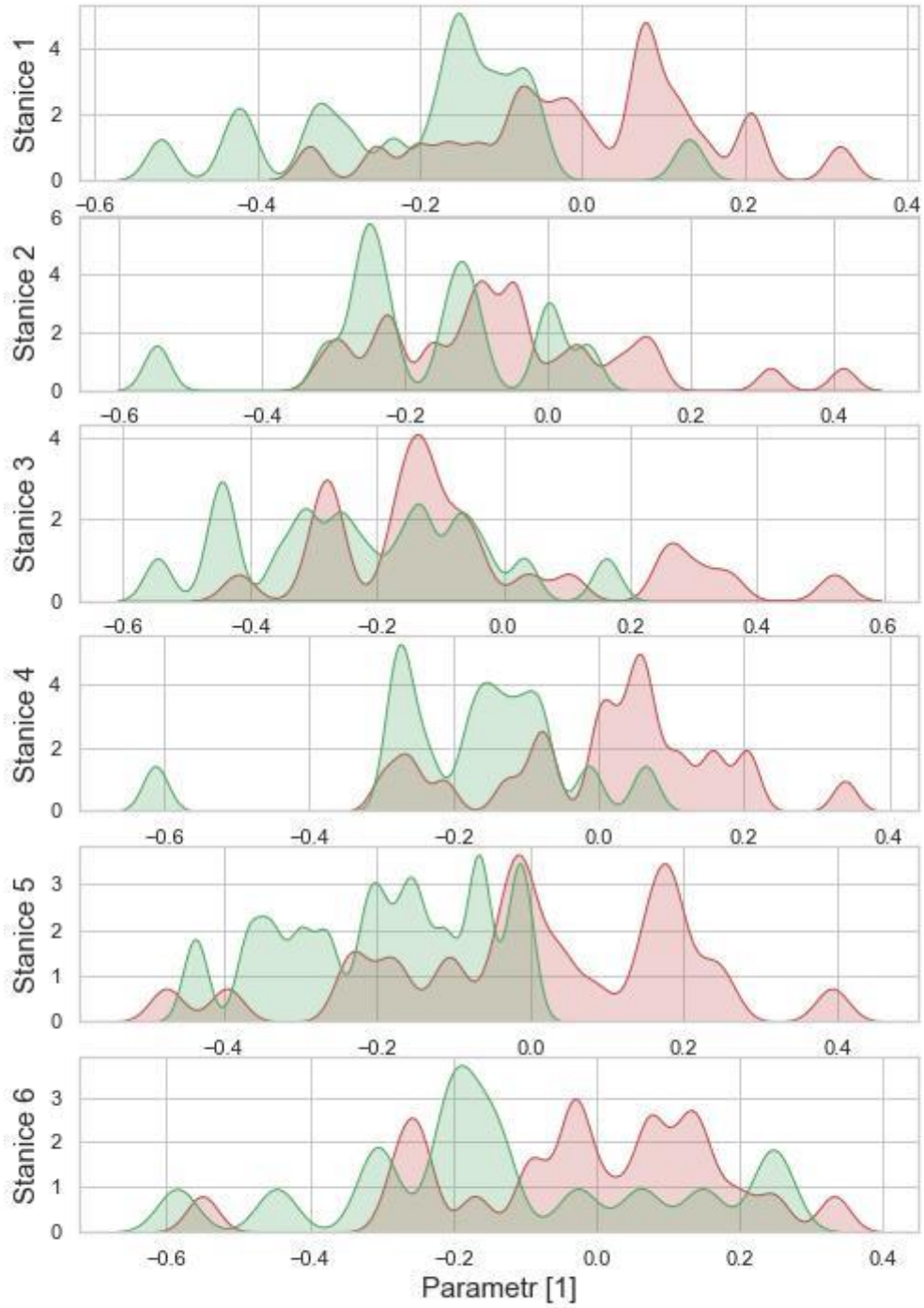
Při shlukování části křivky jsem se samozřejmě nejvíce zaměřoval na shlukování pravé horní části křivky obrázku 7.12, která byla vstupem algoritmů. Charakterem procesu ovšem proběhlo shlukování křivek velmi dobře po celé délce. Při pohledu na grafy rozdělení parametrů podle shluků v rámci stanic se grafy podobají těm z kapitoly 7.1.

Parametry shlukování:

- Předzpracování – analýza hlavních komponent
- Použitý algoritmus – k-means
- Počet shluků – 2



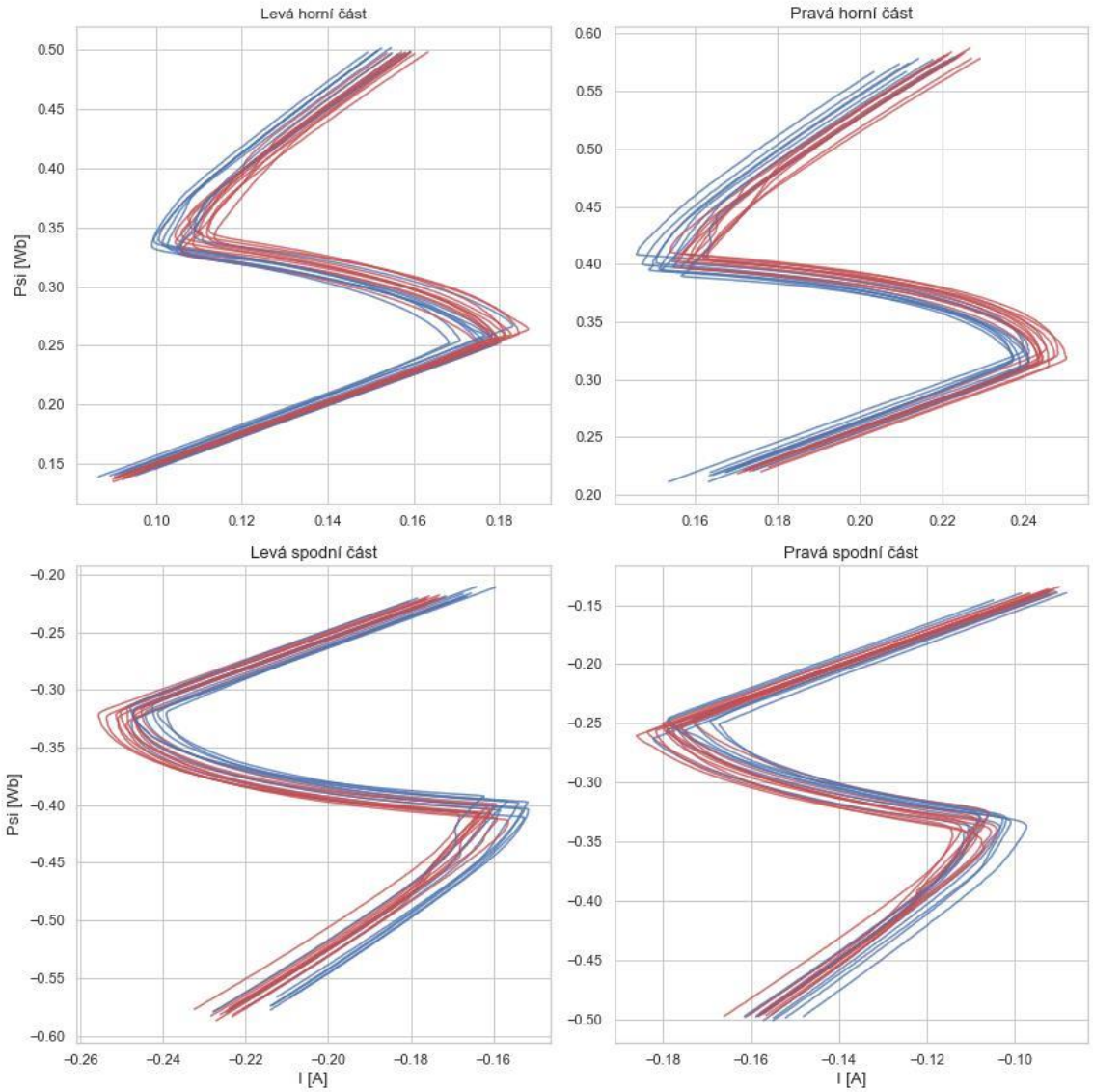
Obrázek 7.12 Shlukování části křivky algoritmem k-means



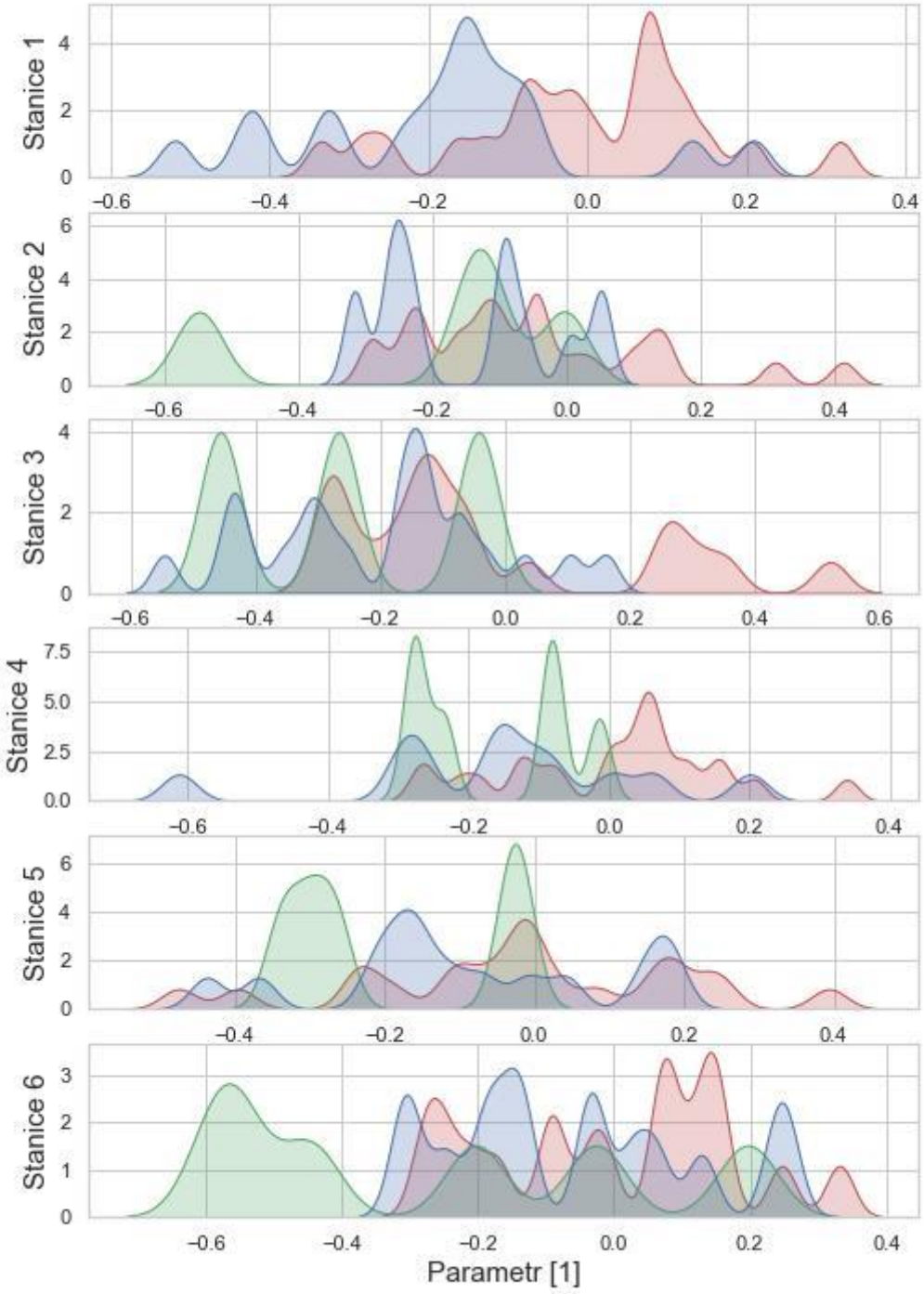
Obrázek 7.13 Výsledky shlukování části křivky algoritmem k-means

Parametry shlukování:

- Předzpracování – autoenkodér
- Použitý algoritmus – k-means
- Počet shluků – 3



Obrázek 7.14 Shlukování části křivky algoritmem k-means



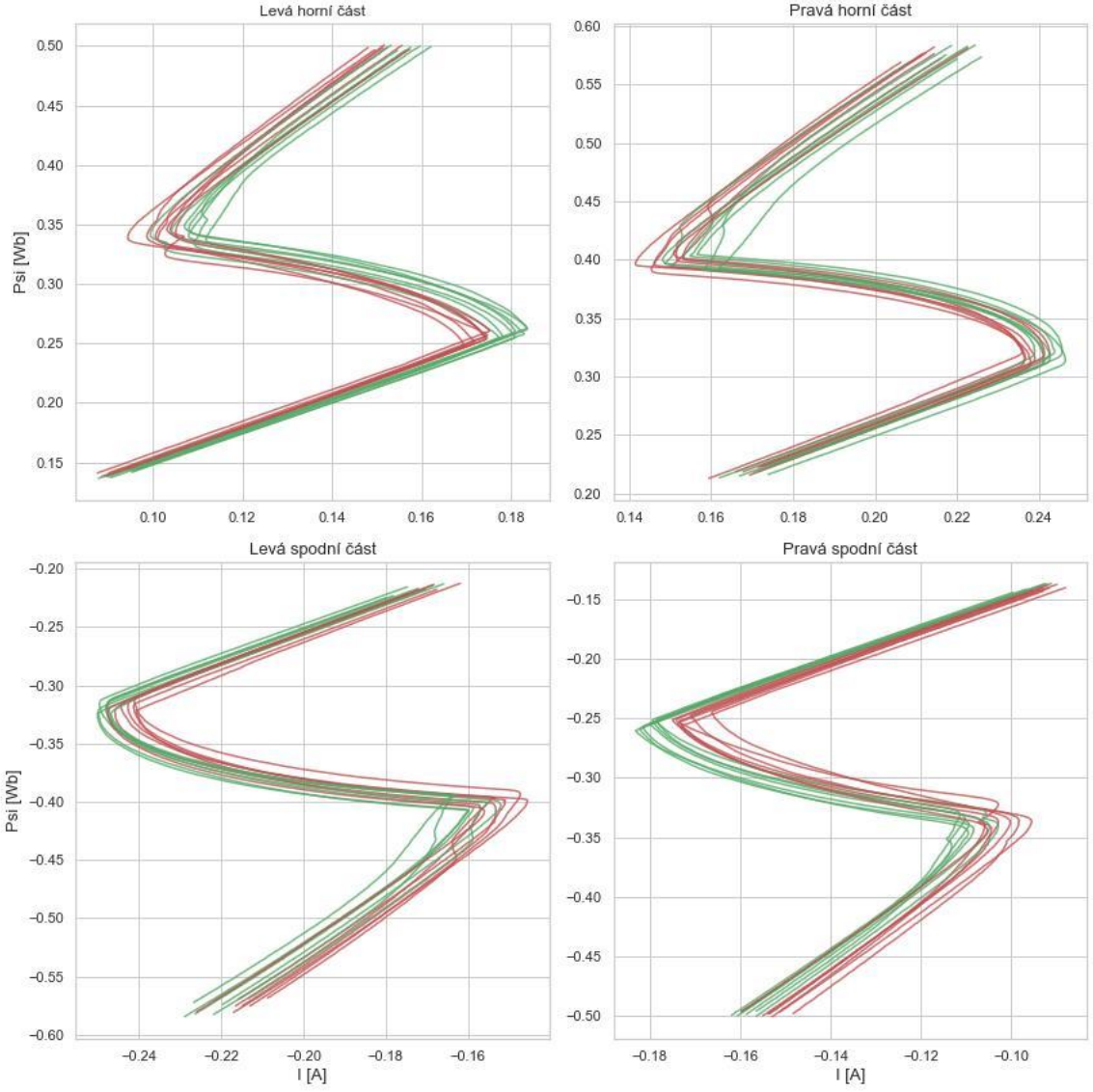
Obrázek 7.15 Výsledky shlukování části křivky algoritmem k-means

7.4. Jednotlivé body křivky

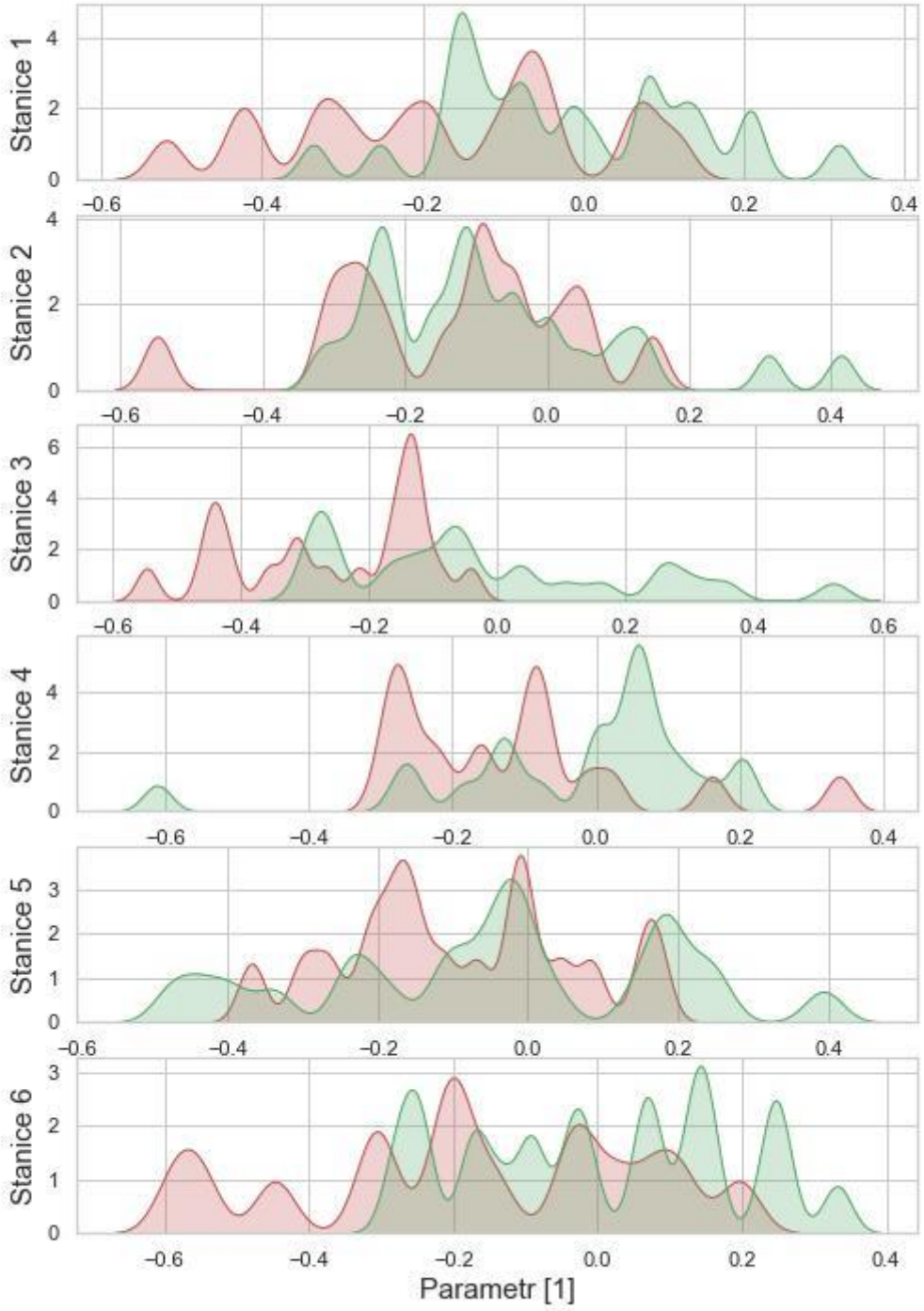
Pro charakteristické body křivky nebylo potřeba použití žádné metody snižování dimenzí. Rozdělení parametrů v rámci stanic ovšem není příliš dobré. Shluky se navzájem překrývají více než v kapitolách 7.2 a 7.3. Nejlepších výsledků jsem dosáhl použitím k-means a spektrálního shlukování.

Parametry shlukování:

- Předzpracování – algoritmus pro vybrání specifických bodů
- Použitý algoritmus – k-means
- Počet shluků – 2



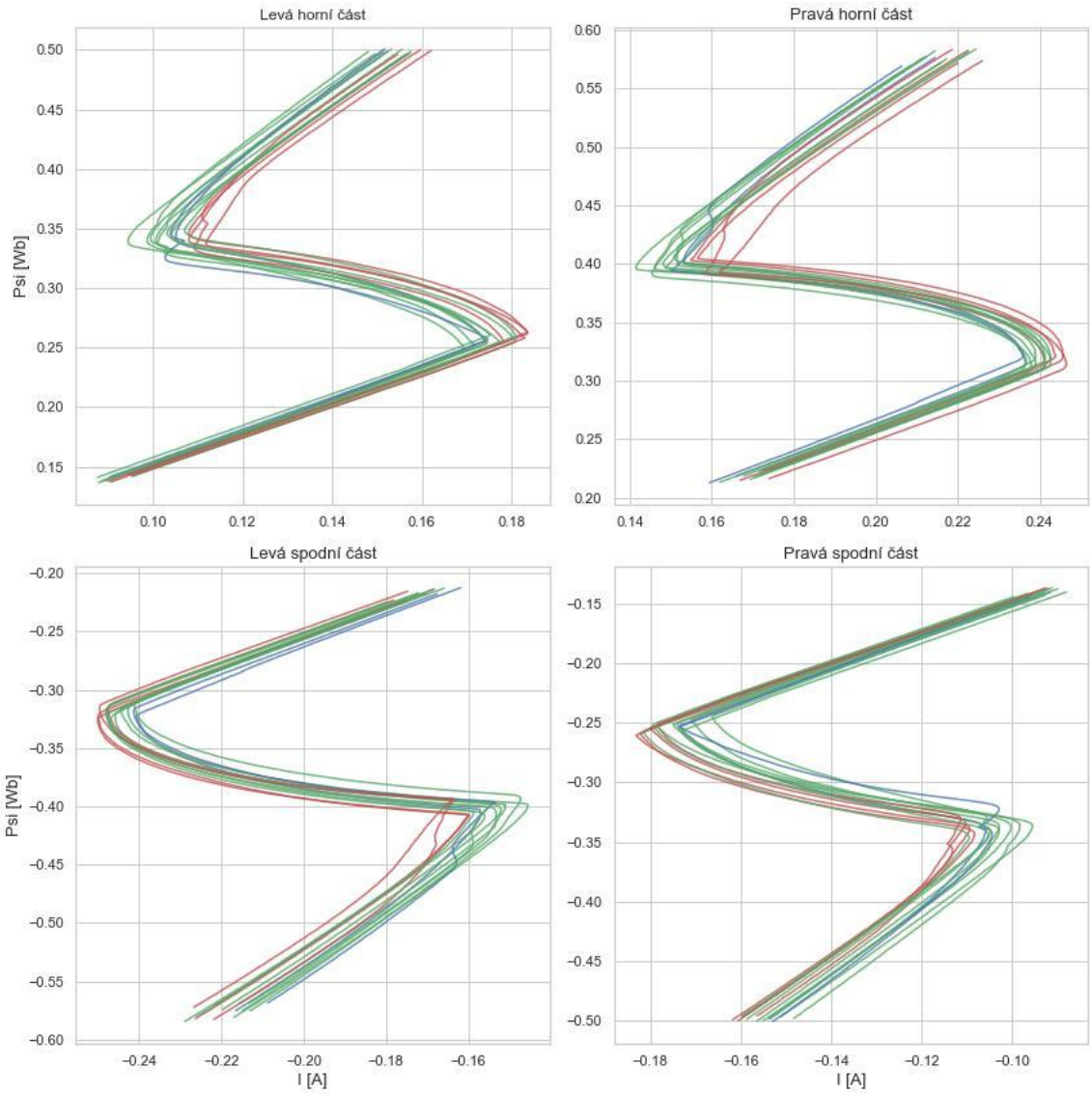
Obrázek 7.16 Shlukování konkrétních bodů křivky algoritmem k-means



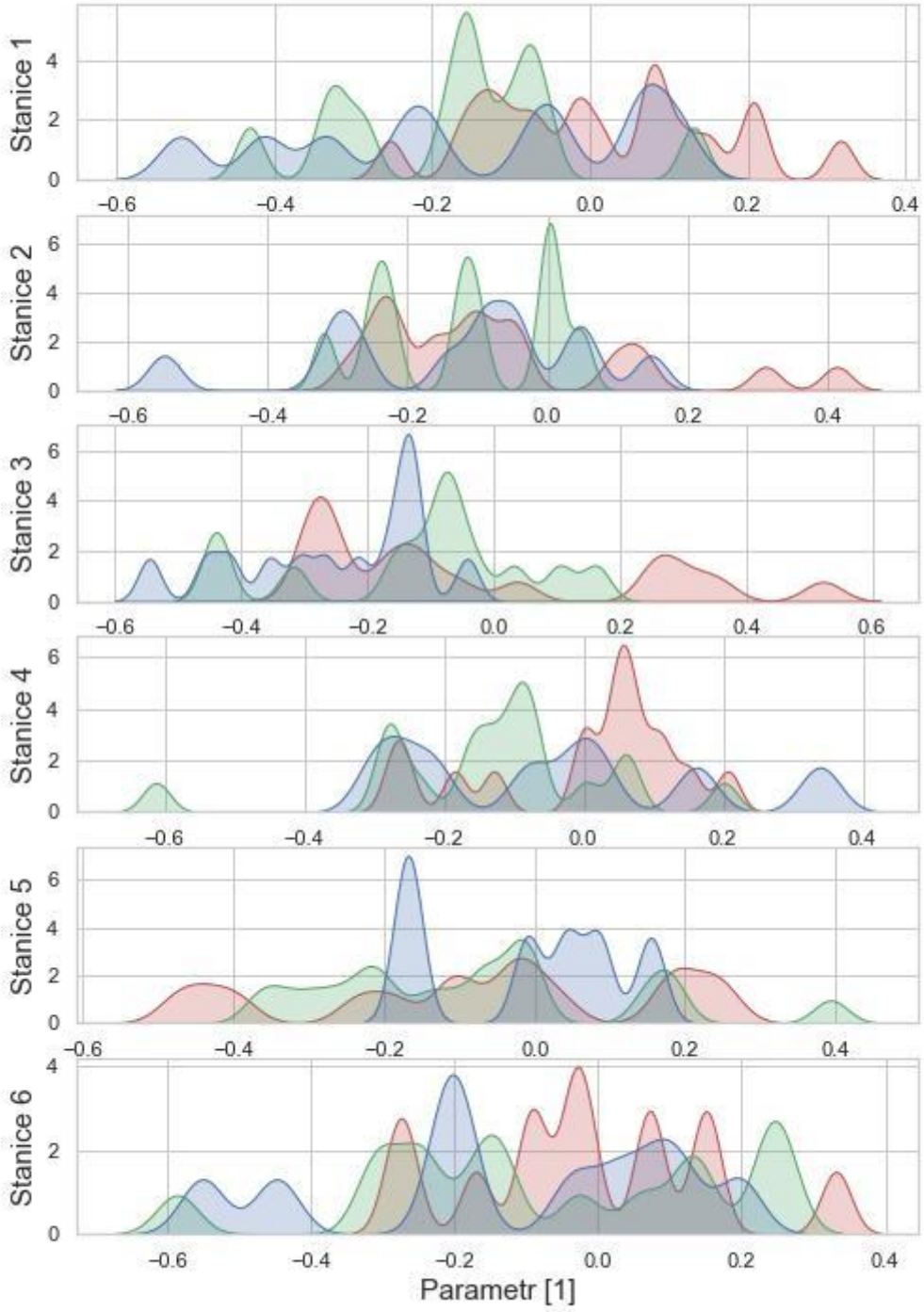
Obrázek 7.17 Výsledky shlukování konkrétních bodů křivky algoritmem k-means

Parametry shlukování:

- Předzpracování – algoritmus pro vybrání specifických bodů
- Použitý algoritmus – spektrální shlukování
- Počet shluků – 3



Obrázek 7.18 Shlukování konkrétních bodů křivky algoritmem spektrálního shlukování



Obrázek 7.19 Výsledky shlukování konkrétních bodů křivky algoritmem spektrálního shlukování

7.5. Shrnutí

Výsledky jednotlivých shlukování ukazují, že pro dva shluky lze velmi často odlišit křivky, které mají zápornou, respektive kladnou hodnotu výrobního parametru.

Pro tři shluky toto rozdělení bývá viditelné pouze v některých případech a není konzistentní přes všechny výrobní stanice, tudíž nelze s jistotou tvrdit, že je správné, a proto považuji lepší shlukovat pouze do dvou shluků, což by odpovídalo výsledkům metody siluety.

Jako nejlepší se ukázal přístup použití celé křivky a její části. Jejich dimenze se pomocí autoenkodéru zredukovala a následně použila do algoritmu k-means.

Použití analýzy hlavních komponent a spektrálního shlukování v různých kombinacích mělo dobré výsledky, ale použití metod z předchozího odstavce bylo nejstabilnější.

Algoritmus DBSCAN se mi v rámci tohoto použití podařilo optimalizovat pouze pro konkrétní případy. Pro lepší použití bych navrhl nastudování a implementaci algoritmů, které pomáhají s volbou parametrů *min_samples* a *eps*.

Důležité je si ovšem uvědomit, že konkrétní výsledky z výrobní stanice mohou být výrazně ovlivněny současným stavem stroje. Zároveň mezi měřením dodavatele a měřením použitého parametru je prakticky celý výrobní proces čerpadla, což do tohoto parametru přidává další vlivy. Pro úplně ověření spojitosti tvaru křivek s výrobním parametrem je dle mého názoru nutné získat více dat z delšího časového období, čímž by se hypotéza mohla potvrdit, nebo vyvrátit.

Kapitola 8

Závěr

Tato práce se zaměřuje na přiblížení tématu předzpracování a shlukování křivkových dat reálného výrobku Denoxtronic. Rostoucí populárnost křivkových dat ve výrobním průmyslu otevírá prostor pro testování a vylepšování dostupných technologií a metodik jejich analýzy.

V teoretické části této diplomové práce jsem se zabýval popisem čerpacího modulu včetně fyzikálního procesu, ke kterému se dostupná data vztahují. Představil jsem základní princip předzpracování, kde se konkrétně zabývám redukcí dimenze pomocí analýzy hlavních komponent a autoenkodéru. Rešerše pokračuje přiblížením třech metod shlukování – k-means, spektrální shlukování a DBSCAN. Pro tyto metody jsem popsal jejich základní vlastnosti, princip a způsob použití.

V praktické části jsem nabyté znalosti z rešerše využil na dostupných datech a navrhl jsem tři způsoby pro řešení úlohy, kde jsem se postupně zaměřil na celou křivku, její část, a nakonec na její samostatné body. Následně jsem použil algoritmy shlukování včetně metod pro získání předpokládaného počtu shluků a jejich výsledky jsem interpretoval ve dvou různých typech grafů.

Jako nejlepší kombinaci jsem určil použití kombinace autoenkodéru pro snížení dimenze společně s algoritmem k-means, nicméně použití analýzy hlavních komponent a spektrálního shlukování nemělo v tomto případě znatelně horší výsledky. Algoritmus DBSCAN se mi pro tento typ úlohy dařilo nastavit pouze v některých případech a výsledky bývaly oproti zbylým algoritmům horší, nicméně to mohlo zavinit nesprávné nastavení parametrů algoritmu.

Validaci výsledků jsem primárně prováděl na základě vizuální kontroly roztřídění křivek do jednotlivých shluků v jednotlivých částech křivky. Problémem vizualizace takového to množství dat je, že při zobrazení větších desítek křivek se vizualizace stává velmi málo přehledná. Proto jsem kontrolu musel provádět iterativně napříč křivkami.

Z výsledků jsem byl schopný pozorovat, že výsledný parametr z výrobních stanic má určitou spojitost s tvarem křivek hystereze. Tuto spojitost nelze na základě výsledků přesně popsat, ale pro dva shluky se daly přibližně rozdělit kladné a záporné hodnoty parametru. Současně tato analýza poskytuje některé informace, které budou užitečné pro zvážení dalších kroků.

Osobně bych v návaznosti na tuto diplomovou práci rád implementoval LSTM neuronovou síť, která se svojí strukturou hodí na zpracování sekvenčních křivkových dat. Výsledkem by mohl být algoritmus, který je na základě vstupní křivky hystereze, která od dodavatele přijde dříve než samotná komponenta čerpadla, předpovědět hodnotu na výrobní stanici, což by umožňovalo v případě očekáváných výpadků provést potřebná opatření.

Bibliografie

[1] *Diesel Systems Denoxtronic 5 – Urea Dosing System for SCR systems* [online]. [cit. 2023-01-11]. Dostupné z: http://www.bosch.co.jp/tms2015/en/products/pdf/DS_ProductDatasheet_Denoxtronic5_EN.pdf

[2] FRIEDL, MICHAL. *BOSCH DNOX 5.X – OPTIMALIZACE ZKOUŠEK* [online]. Brno, 2015 [cit. 2023-01-11]. Dostupné z: https://core.ac.uk/download/pdf/30309502.pdf. Diplomová práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ.

[3] JOLLIFE, Ian T. a Jorge CADIMA. *Principal component analysis: a review and recent developments*. Dostupné z: doi: 10.1098/rsta.2015.0202

[4] *Understanding Principle Component Analysis(PCA) step by step.* [online]. [cit. 2023-01-15]. Dostupné z: https://medium.com/analytics-vidhya/understanding-principle-component-analysis-pca-step-by-step-e7a4bb4031d9

[5] MICHELUCCI, Umberto. *An Introduction to Autoencoders*. Dostupné z: doi:10.48550/arXiv.2201.03898

[6] *K-Means: A complete Introduction.* [online]. [cit. 2023-01-15]. Dostupné z: https://towardsdatascience.com/k-means-a-complete-introduction-1702af9cd8c

[7] *K-Means Clustering: Comparison of Initialization strategies* [online]. [cit. 2023-01-15]. Dostupné z: https://medium.com/analytics-vidhya/comparison-of-initialization-strategies-for-k-means-d5ddd8b0350e

[8] *Silhouette Coefficient* [online]. [cit. 2023-01-15]. Dostupné z: https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c

[9] *Teorie grafů* [online]. [cit. 2023-01-15]. Dostupné z: https://teorie-grafu.cz/zakladni-pojmy/matematicka-definice-grafu.php

[10] *Spectral Clustering* [online]. [cit. 2023-01-15]. Dostupné z: https://towardsdatascience.com/spectral-clustering-aba2640c0d5b

[11] *How DBSCAN works and why should we use it?* [online]. [cit. 2023-01-15]. Dostupné z: https://towardsdatascience.com/spectral-clustering-aba2640c0d5b

[12] *One Minute Overview of the DBSCAN Clustering Algorithm* [online]. [cit. 2023-01-15]. Dostupné z: doi:One Minute Overview of the DBSCAN Clustering Algorithm

[13] *Emission standards* [online]. [cit. 2023-01-16]. Dostupné z: https://dieselnet.com/standards/eu/ld.php

[14] *Diaphragm Pump: What Is a Diaphragm Pump?* [online]. [cit. 2023-01-16]. Dostupné z: https://www.tacmina.com/learn/basics/01.html

[15] *MinMaxScaler* [online]. [cit. 2023-01-16]. Dostupné z: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[16] *Advanced Introduction to Machine Learning* [online]. In: . s. 10-11 [cit. 2023-01-16]. Dostupné z: https://www.cs.cmu.edu/~epxing/Class/10715-14f/lectures/EM.pdf

[17] *In-depth Intuition of K-Means Clustering Algorithm in Machine Learning* [online]. [cit. 2023-01-17]. Dostupné z: https://www.analyticsvidhya.com/blog/2021/01/in-depth-intuition-of-k-means-clustering-algorithm-in-machine-learning/

[18] *Selecting the number of clusters with silhouette analysis on KMeans clustering* [online]. [cit. 2023-01-17]. Dostupné z: https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

[19] *Adjacency matrix* [online]. [cit. 2023-01-17]. Dostupné z: https://www.oreilly.com/library/view/php-7-data/9781786463890/6bddc759-aa6e-4f1a-a2b3-a5460b5de121.xhtml

[20] *Towards Data Science: Comprehensive Introduction to Autoencoders* [online]. [cit. 2023-01-23]. Dostupné z: https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368

[21] BANK, Dor, Noam KOENIGSTEIN a Raja GIRYES. *Autoencoders*. Dostupné z: doi:doi:arXiv:2003.05991v2 [cs.LG] 3 Apr 2021

[22] *Adam — latest trends in deep learning optimization.* [online]. [cit. 2023-01-23]. Dostupné z: https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c

[23] *Overfitting* [online]. [cit. 2023-01-23]. Dostupné z: https://en.wikipedia.org/wiki/Overfitting#/media/File:Overfitting.svg

[24] YOUGUO, Li a Wu HAIYAN. *A Clustering Method Based on K-Means Algorithm*. Dostupné z: doi:10.1016/j.phpro.2012.03.206

[25] *Scipy.signal.savgol_filter* [online]. [cit. 2023-01-25]. Dostupné z: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.savgol_filter.html

[26] *Clustering* [online]. [cit. 2023-01-25]. Dostupné z: https://scikit-learn.org/stable/modules/clustering.html

Přílohy

- diplomova_prace.zip
  - Points_verification
    - Pozn. – 10 obrázků pro ověření algoritmu volení charakteristických bodů
  - Plots
    - Pozn. – 62 obrázků vytvořených k použití v diplomové práci
  - clustering.ipynb
  - data.pickle
  - diplomova_prace.pdf
  - preprocessing.ipynb
  - X_clustering.pickle