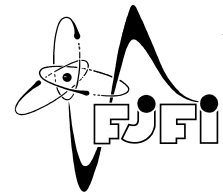




ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta jaderná a fyzikálně inženýrská



Vývoj a srovnání PM metod pro odhady elasticity materiálů

Development and comparison of PM methods for material elasticity estimation

Diplomová práce

Autor: **Erik Dolejš**
Vedoucí práce: **Ing. Václav Kůs, Ph.D.**
Akademický rok: 2021/2022

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Erik Dolejš
Studijní program: Aplikované matematicko-stochastické metody
Název práce (česky): Vývoj a srovnání PM metod pro odhady elasticity materiálů
Název práce (anglicky): Development and comparison of PM methods for material elasticity estimation

Pokyny pro vypracování:

- 1) Nastudujte principy metod rozpoznávání statistického modelu distribučních směsí v Preisach-Mayergoyzově (PM) prostoru metodami klasickými i strojovými.
- 2) Implementujte tyto metody a proveďte jejich vzájemné porovnání na základě generovaných dat, vyberte nejlepší kandidáty.
- 3) Zopakujte úlohu pro případ specifických typů směsí nebo neparametrických odhadů PM prostoru.
- 4) Predikujte Preisach-Mayergoyzův index elasticity pro neznámá zatížení s využitím předchozích analýz.
- 5) Vyberte nejlepší odhad hustoty pravděpodobnosti v Preisach-Mayergoyzově prostoru z dostupných reálných dat (zatížení + hystereze), např. z testování zemětřesných tlumičů.

Doporučená literatura:

- 1) I. D. Mayergoyz, Mathematical Models of Hysteresis and Their Applications. AcademicPress-Elsevier, 2003.
- 2) G. Celeux, S. Frühwirth-Schnatter, Ch. Robert, Model Selection for Mixture Models--Perspectives and Strategies. Handbook of Mixture Analysis, CRC Press, hal-01961077, 2018.
- 3) G. Malsiner-Wallia, S. Frühwirth-Schnatterb, B. GrünIdentifying, Mixtures of Mixtures Using Bayesian Estimation. Journal of computational and graphical statistics 26, 2017, 285--295.
- 4) E. Sgouritsa, D. Janzing, J. Peters, B. Scholkopf, Identifying Finite Mixtures of Nonparametric Product Distributions and Causal Inference of Confounders. Max Planck Institute for Intelligent Systems, Tubingen, Germany, Rep.Num. UAI-P-2013-PG-556-565, ArXiv:1309.6860, 2013.
- 5) O. Shojae, M. Asadi, M. Finkelstein, On Some Properties of α -Mixtures. Metrika 84, 2021, 1213--1240.
- 6) E. Suarez, A. Roldán, et. al., Entropy analysis for damage quantification of hysteretic dampers used as seismic protection of buildings. Applied Sciences 7(628), 2017, 1--14.

Jméno a pracoviště vedoucího diplomové práce:

Ing. Václav Kůs, Ph.D.

Katedra matematiky, FJFI ČVUT v Praze, Trojanova 13, 120 00 Praha 2

Jméno a pracoviště konzultanta:

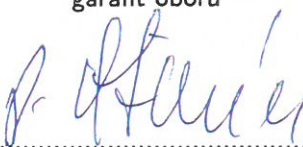
Datum zadání diplomové práce: 31.10.2021

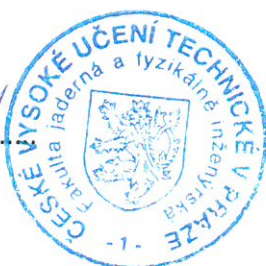
Datum odevzdání diplomové práce: 2.5.2022

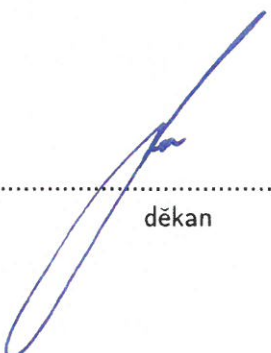
Doba platnosti zadání je dva roky od data zadání.

V Praze dne 01.11.2021


.....
garant oboru


.....
vedoucí katedry




.....
děkan

Poděkování:

Chtěl bych poděkovat svému školiteli Ing. Václavu Kůsovi, PhD. za odborné vedení, za pomoc a rady při zpracovávání této práce.

Čestné prohlášení:

Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl jsem všechnu použitou literaturu.

V Praze dne 2. května 2022

Erik Dolejš



Název práce:

Vývoj a srovnání PM metod pro odhady elasticity materiálů

Autor: Erik Dolejš

Program: Aplikované matematicko-stochastické metody

Druh práce: Diplomová práce

Vedoucí práce: Ing. Václav Kůs, Ph.D., Katedra matematiky, FJFI ČVUT

Abstrakt: Hlavním cílem této diplomové práce je schopnost nalezení vhodné hustoty pravděpodobnosti, resp. směsi distribucí z naměřené hysterezní křivky pro Preisach-Mayergoyzův (PM) model. Nejdříve je představena hystereze a její PM model. Dále jsou popsány nástroje pro identifikaci PM prostoru, kterými jsou pravděpodobnostní rozdělení, heuristické optimalizační algoritmy a divergenční míry. Dále jsou představeny běžné stromové metody strojového učení, stejně jako sofistikovanější metody hlubokých a konvolučních neuronových sítí. Všechny metody jsou jednak implementovány pro klasifikaci nejvhodnějšího pravděpodobnostního rozdělení z hysterezní křivky a dále jsou implementovány za účelem regrese zastoupení jednotlivých složek ve směsi distribucí. Na závěr jsou představeny speciální jádrové odhady pro popis PM prostoru, které dále slouží pro vyhodnocení elastických vlastností materiálů odpovídajících jejich naměřeným hysterezním smyčkám. Všechny metody jsou také aplikovány na reálná data z testování ocelových zemětřesných tlumičů.

Klíčová slova: divergenční míra, hluboká neuronová síť, Jaya algoritmus, jádrový odhad hustoty pravděpodobnosti, konvoluční neuronová síť, náhodný les, Preisach-Mayergoyzův model, XGBoost

Title:

Development and comparison of PM methods for material elasticity estimation

Author: Erik Dolejš

Abstract: The main objective of this thesis is the ability to find a suitable probability density, or a mixture of probability densities, from a measured hysteresis curve for the Preisach-Mayergoyz (PM) model. First, the hysteresis curve and its PM model are introduced. Then, the tools for identifying the PM space, which are probability distributions, heuristic optimization algorithms, and divergence measures, are described. Next, common tree-based machine learning methods, as well as more sophisticated deep and convolutional neural network methods are introduced. All the methods are firstly implemented to classify the best-fitting probability distribution from the hysteresis distribution and secondly implemented to regress the representation of the individual components in the mixture of distributions. Finally, special kernel estimators are presented to describe the PM space, which are further used to evaluate the elastic properties of materials corresponding to their measured hysteresis loops. All methods are also applied to real data from steel earthquake damper testing.

Key words: convolutional neural network, deep neural network, divergence measure, Jaya algorithm, kernel density estimation, Preisach-Mayergoyz model, random forest, XGBoost

Obsah

Úvod	11
1 Preisach-Mayergoyzův model hystereze	13
1.1 Hystereze a její vlastnosti	13
1.2 Preisach-Mayergoyzův model hystereze	15
1.3 Aplikace PM modelu	17
2 Identifikace PM prostoru	19
2.1 Pravděpodobnostní rozdělení	19
2.2 Divergenční míry	21
2.3 Heuristické algoritmy pro optimalizaci	23
2.3.1 Jaya algoritmus	23
2.3.2 aDE-Jaya	24
2.3.3 Softwarová implementace	25
3 Klasifikační a regresní metody strojového učení	27
3.1 Kritéria vyhodnocení kvality odhadu	28
3.2 Stromové a ensemble metody	30
3.3 Hluboké neuronové sítě	31
3.4 Konvoluční neuronové sítě	34
4 Implementace metod strojového učení	37
4.1 Předzpracování dat	37
4.2 Datasetsy a proces trénování	40
4.2.1 Trénování ensemble metod	44
4.2.2 Trénování NN a CNN metod	45
4.2.3 Trénování regresní přístupu jednotlivých metod	48
4.2.4 Kombinovaný regresor	51
4.3 Vyhodnocení na reálných datech I	52

5	Jádrové odhady hustoty pravděpodobnosti	63
5.1	Aplikace jádrových odhadů na Preisachův trojúhelník	66
5.2	Index elasticity	70
5.3	Vyhodnocení na reálných datech II	71
	Závěr	73

Úvod

Hysterezní chování nalezneme v mnoha fyzikálních, biologických, mechanických či ekonomických systémech. Vzhledem k jejímu nelineárnímu chování existuje mnoho matematických modelů. V první kapitole si představíme Preisach-Mayergoyzův (PM) model hystereze a ukážeme si jeho aplikaci na popis mechanické hystereze materiálů, konkrétně při nedestruktivním testování ocelových tlumičů, které jsou používány ve stavebních konstrukcích za účelem snížení škod při zemětřesení. Ve druhé kapitole si představíme klasické metody optimalizace PM modelu a tedy nalezení tzv. Preisachova prostoru tak, aby co nejlépe odpovídal naměřeným datům. Tento prostor budeme popisovat pomocí vhodných hustot pravděpodobnosti, jejichž parametry budou nalezeny pomocí heuristických algoritmů.

Ve třetí a čtvrté kapitole se zaměříme na metody strojového učení pro klasifikaci nejvhodnějšího pravděpodobností rozdělení pro Preisachův prostor z naměřených hysterezních smyček, resp. pro regresi zastoupení více rozdělení v distribuční směsi. Budou použity jednak jednodušší metody, jakými jsou např. náhodné lesy, ale také sofistikovanější ve formě hlubokých a konvolučních neuronových sítí. V poslední kapitole představíme specifické jádrové odhady hustoty pravděpodobnosti pro charakterizaci získaného Preisachova prostoru. Tento neparametrický odhad nám poslouží pro vyhodnocení elastických vlastností materiálu, tedy jak dobře pohlcuje mechanické deformace. Všechny výše zmíněné metody budou také vyhodnoceny na reálných datech z testování zemětřesných tlumičů.

Kapitola 1

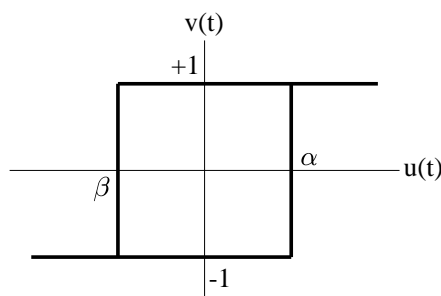
Preisach-Mayergoyzův model hystereze

V této části se seznámíme s termínem hystereze a rozebereme její vlastnosti. Vzhledem k nelineárnímu chování hystereze je zapotřebí pro její popis speciální matematický aparát. Jednou z možností je Preisach-Mayergoyzova (PM) model, který bude detailně rozebrán v první kapitole, neboť tento model budeme hojně používat ve zbytku práce. Na závěr ukážeme aplikaci PM modelu na vyhodnocení mechanické hystereze na reálných datech.

1.1 Hystereze a její vlastnosti

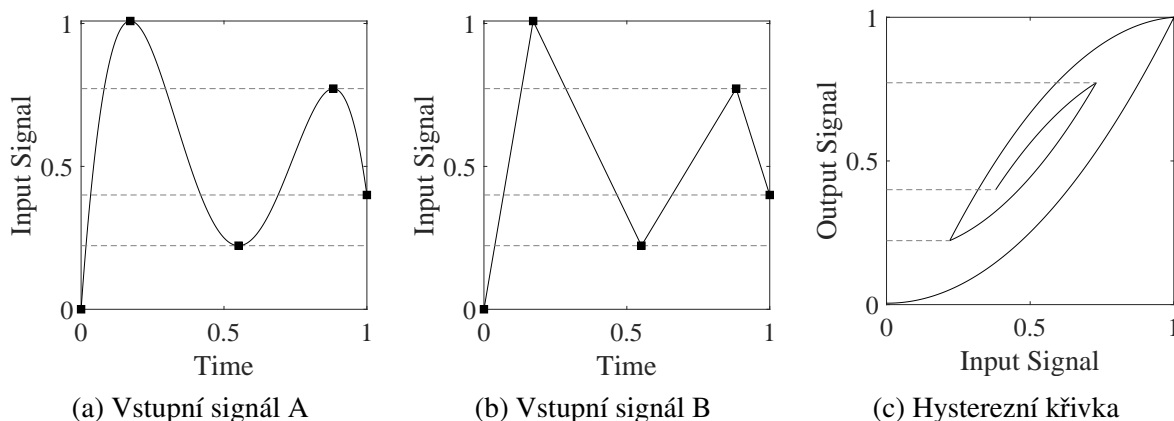
Pojmem hystereze označujeme takové chování dynamického systému, kde jeho stav závisí na historii předešlého vývoje. Tedy i kompletní znalost vstupních veličin do systému nám není schopná sama o sobě popsat jeho stav, pokud nemáme informaci o předchozím vývoji tohoto systému. Toto chování se objevuje v mnoha odvětvích, ať už v magnetismu nebo biologických systémech. V našem případě se bude jednat zejména o mechanickou hysterezi, avšak vzhledem k čistě matematickému přístupu lze naše použití upravit na odlišné problematiku. Grafy takového chování nazýváme hysterezní křivkou, které často tvoří uzavřenou nebo téměř uzavřenou smyčku.

Chceme-li ilustrovat hysterezi na jednoduchém případě, tak si můžeme představit termostat, který má za úkol udržovat teplotu vody v nádrži mezi danými hodnotami. A to tak, že se zapne ohřívání, pokud je teplota nižší, nebo se ohřívání naopak vypne, pokud teplota přesáhne horní mez. V tomto případě nejsme schopni z informace o teplotě vody říci, jestli je termostat ve



Obrázek 1.1: Přepínací relé s hysterezí

stavu zapnuto nebo vypnuto, ale musíme znát i teplotu v předchozím jednom nebo více okamžicích, abychom mohli určit, jestli se voda ochlazuje nebo ohřívá, a tím pádem i v jakém stavu je termostat. Takové hysterese chování můžeme vyobrazit např. pomocí relé na obrázku 1.1. Zde opět platí, že pokud je na začátku hodnota vstupního signálu $u(t)$ menší než α , tak hodnota závislé veličiny $v(t)$ je -1 . V případě, že dosáhne vstup hodnoty α , tak se výstupní stav přepne do hodnoty $+1$, ve kterém zůstává až do případu dosažení spodní hranice β . Hodnoty -1 a $+1$ mohou značit v našem ilustrativním případě stavy zapnuto a vypnuto, resp. další binární stavy systému např. zavřeno a otevřeno. Takové hysterese chování můžeme následně rozdělit do dvou kategorií podle toho, jestli navíc závisí nebo nezávisí na průběhu vstupní proměnné. V našem případě budeme uvažovat pouze takovou hysterese, která je tzv. *rate-independent*, neboli hodnoty výstupní proměnné závisí pouze na minimech a maximech vstupní proměnné, aniž by to ovlivňoval průběh mezi těmito hodnotami. To vidíme znázorněné na obr. 1.2, kde dva odlišné průběhy, které však mají stelné lokální extrémy, vytváří stejnou hysterese křivku. Zřejmě je takový model zjednodušením skutečného chování, avšak pro naše potřeby je to více nežli dostačující, neboť průběhy vstupní proměnné by se musely měnit s extrémně velkou frekvencí, aby se začalo toto chování v praxi projevovat. Obecně má tedy použití tohoto předpokladu své limity, a tedy například pro modelování vysokofrekvenční magnetické hysterese by bylo tím pádem zapotřebí použití *rate-dependent* modelu.



Obrázek 1.2: Rate-independence vlastnost hysterese

Je také důležité zmínit, že použití *rate-independence* předpokladu je aplikováno na třech rozdílných časových škálách. První je škála rychlé vnitřní dynamiky snímače. Druhá škála je časové měřítko, na kterém se provádí měření, která je mnohem větší než časová stupnice vnitřní dynamiky snímače, takže každé pozorování lze ztotožnit s konkrétní výstupní hodnotou snímače. Třetí časové škála odpovídá změnám vstupního signálu. Toto časové měřítko je opět větší než časová stupnice pozorování, a tudíž celkově lze každé naměřenou hodnotu hysterese ztotožnit s konkrétní hodnotou vstupního signálu. Také se budeme zabývat pouze skalárním modelem hysterese, kdy vstupní signál tvoří jednorozměrná vstupní proměnná $u(t)$, ale je možné i zobecnění této problematiky do dvou- či více-dimenzionálního vektorového modelu charakterizovaném vektorem vstupních veličin $\mathbf{u}(t)$ se stejnorozměrným výstupem $\mathbf{v}(t)$. Z dosavadního popisu lze tedy tvrdit, že skalární hysterese je nelinearita s pamětí, která se projevuje prostřednictvím vývoje průběhu hysterese křivky.

Rate-independent modely hystereze můžeme dále rozdělit do dvou skupin. Modely hystereze s lokální pamětí a modely s globální pamětí. Model nelinearity s lokální pamětí je charakterizován vlastností, že hodnota výstupu $v(t)$ v čase t_0 a hodnota vstupního signálu $u(t)$ v časech $t \geq t_0$ jednoznačně určují $v(t)$ ve všech následujících časech $t > t_0$. Tedy v tomto případě se historie bývalých stavů projevuje do budoucnosti pouze skrze současnou hodnotou výstupu. Naopak pro model nelinearity s globální pamětí nezávisí hodnoty $v(t)$ v časech $t > t_0$ pouze na současné hodnotě, ale také na předešlé dosažené extrémální hodnotě vstupního signálu. My se budeme zabývat modelováním hystereze s globální pamětí. Důvodem je, že vstupní signál není v praxi apriorně znám, neboť může pocházet ze senzoru měřícího v reálném čase, a proto nelze predikovat dopředu, jak bude vypadat hysterezní křivka. To je hlavní překážka při matematickém popisu hystereze. Proto jsou zapotřebí uvažovat modely, které samy díky své struktuře jsou schopné detekovat a zapamatovat extrémální hodnoty signálu a budou predikovat vhodné průběhy hysterezních křivek pomocí nahromaděných informací.

Existuje mnoho modelů pro popis hystereze, které mají své výhody pro specifické aplikace. Jedná se například o model M. Krasnoselkiiho a A. Pokrovskiiho ([11]), který navíc předpokládá, že hysterezní operátor neprobíhá skokovité, jak jsme viděli na obrázku 1.1, ale že přechod mezi stavy je lineární. Dále Bouc-Wenův model ([12]), který popisuje hysterezi pomocí systémem nelineárních rovnic. A dále např. Prandtlův-Išlinského model ([13]), který je založen na popisu složením jednoduchých operátorů, a je vlastně speciálním typem tzv. Preisach-Mayergoyzova (PM) modelu, který budeme pro popis hystereze používat my.

1.2 Preisach-Mayergoyzův model hystereze

Jak již bylo zmíněno, Preisach-Mayergoyzův (PM) model je založen na složení z jednoduchých operátorů, konkrétně hysterezních operátorů, které svojí definicí odpovídají relé na obrázku 1.1. Hysterezní operátor, neboli hysteron $\hat{\gamma}_{\alpha,\beta} \in \{-1, +1\}$, kde $\beta < \alpha$, matematicky definujeme jako

$$\hat{\gamma}_{\alpha,\beta} = \begin{cases} -1, & u(t) \leq \beta, \\ 1, & u(t) \geq \alpha, \\ \xi, & u(t) \in (\beta, \alpha), \end{cases} \quad (1.1)$$

kde

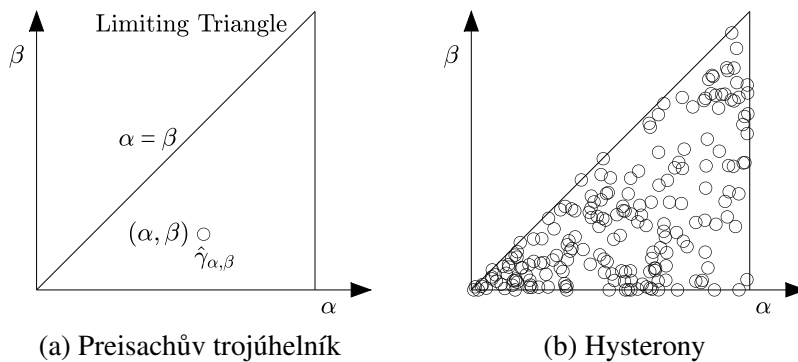
$$\xi = \begin{cases} 1, & \text{pokud } \exists t^* : u(t^*) > \alpha, \\ -1, & \text{pokud } \exists t^* : u(t^*) < \beta, \end{cases} \quad \text{kde } \forall \tau \in (t^*, t), u(\tau) \in (\beta, \alpha).$$

Tento zápis přímo odpovídá našemu ilustrativnímu případu, kdy pro hodnoty mimo interval (β, α) je hodnota výstupu jednoznačně dána a pro hodnoty uvnitř je zapotřebí znát hodnotu vstupního signálu předchozích časech. Speciálním případem je hysteron s hodnotami $\alpha = \beta$, který je idealizovaným případem, neboť dosahuje dokonalé schopnosti změny stavu při stejném vstupním signálu. Pokud budeme uvažovat množinu jednoduchých hysterezních operátorů, Preisach-Mayergoyzův model následně definujeme jako jejich superpozici

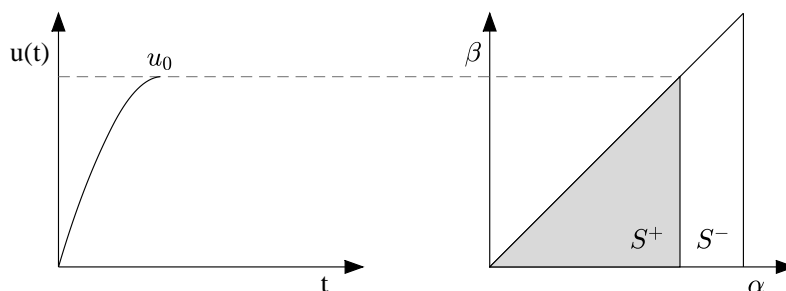
$$v(t) = \hat{\Gamma}[u(t)] = \iint_{\beta \leq \alpha} \mu(\alpha, \beta) \hat{\gamma}_{\alpha,\beta}(u(t)) d\alpha d\beta, \quad (1.2)$$

kde $\mu(\alpha, \beta)$ značí dvourozměrnou hustotu pravděpodobnosti, která není degenerovaná a kterou nazýváme Preisachovou funkcí. Tato funkce popisuje výskyt hysteronů v prostoru parametrů, kterými jsou jednotlivé hysteryony charakterizovány. Výstup modelu získáme tedy jako aplikaci vstupního signálu na výše zmíněnou superpozici $\hat{\Gamma}$. Vzhledem ke svému matematickému zavedení, má tento model schopnost obecně popisovat hysterezi na pestré škále aplikací.

Budeme-li chtít geometricky interpretovat PM model, tak jednotlivé hysteryony mohou být zaneseny do tzv. Preisachovy roviny podle hodnot parametrů α a β , jak můžeme vidět na obrázku 1.3. Požadavek $\beta < \alpha$, společně s předpokladem konečné maximální hodnoty a případem dokonalé hystereze $\alpha = \beta$ nám navíc dává omezení, podle kterého se budou všechny hysteryony vyskytovat uvnitř hraničního pravoúhlého rovnoramenného trojúhelníku. Mimo tento limitní trojúhelník je hustota výskytu hysteronů $\mu(\alpha, \beta)$ dodefinována nulou. Způsob jakým se projeví aplikace vstupního signálu $u(t)$ na libovolný PM prostor je pak znázorněna na obrázcích 1.4 až 1.6. Jsou-li na začátku všechny hysterezní operátory na počátku ve stavu -1 , tak při monotónním růstu vstupního signálu do hodnoty u_0 všechny hysteryony s hodnotou α menší než u_0 nabydou hodnoty $+1$, což je znázorněno šedou plochou S^+ . Tím pádem se nám Preisachův trojúhelník rozdělí na dvě části podle toho, ve kterém stavu jsou jednotlivé hysteryony (obr. 1.4).



Obrázek 1.3: Preisachova rovina a limitní trojúhelník



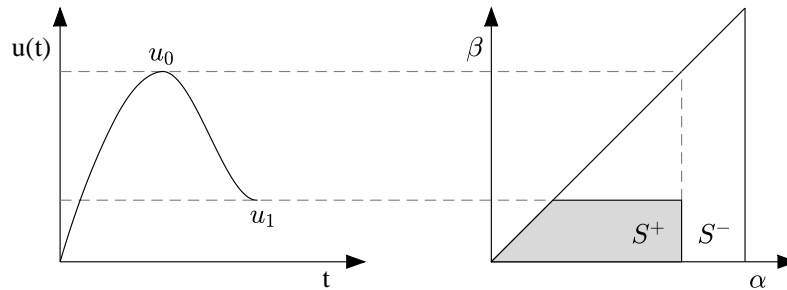
Obrázek 1.4: Vstupní zatížení a odpovídající Preisachův trojúhelník

Pokračuje-li vstupní signál monotónním pokles z u_0 na hodnotu u_1 , všechny hysteryony, které s hodnotami $\beta > 1$ přejdou zpět do stavu -1 (obr. 1.5). Tento proces následně může stejným způsobem pokračovat růstem vstupního signálu do hodnoty u_2 a opětovnou změnou příslušných hysteronů do stavu $+1$ (obr. 1.6) atd. Dělicí křivka $L(t)$ ploch S^+ a S^- odpovídá lokálním maximům a lokálním minimům vstupního signálu. Z tohoto vývoje je tedy vidět, že se jedná

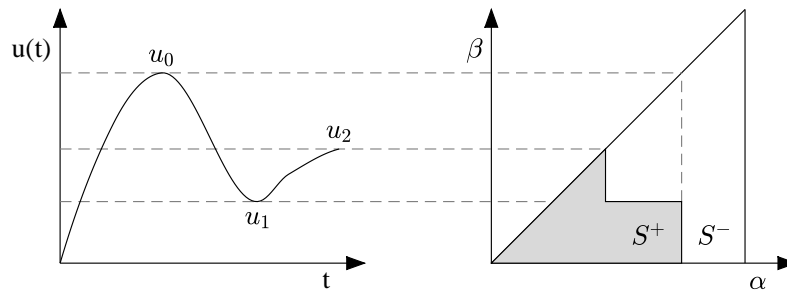
o model s globální pamětí, a navíc můžeme pomocí takto rozděleného PM prostoru na dvě části přepsat rovnici (1.2) pro výstup modelu na

$$v(t) = \hat{\Gamma}[u(t)] = \iint_{S^+(t)} \mu(\alpha, \beta) d\alpha d\beta - \iint_{S^-(t)} \mu(\alpha, \beta) d\alpha d\beta. \quad (1.3)$$

Hledání vhodného PM prostoru tedy přešlo na hledání pravděpodobnostního rozdělení $\mu(\alpha, \beta)$. V naší práci bude znamenat hledání toho Preisachova prostoru numerickou optimalizací vhodných hustot pravděpodobnosti. Vstupní signál společně s PM prostorem dohromady vytvoří výstup v podobě hysterezní křivky, a tudíž hledání PM prostoru je prováděno za účelem, aby výsledná hysterezní křivka vzniklá z Preisachova modelu co možná nejlépe odpovídala empiricky naměřené hysterezi.



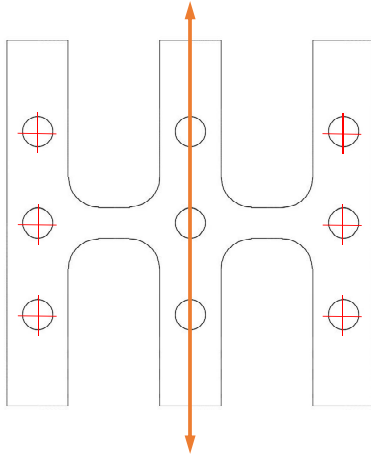
Obrázek 1.5: Pokles vstupního zatížení a odpovídající Preisachův trojúhelník



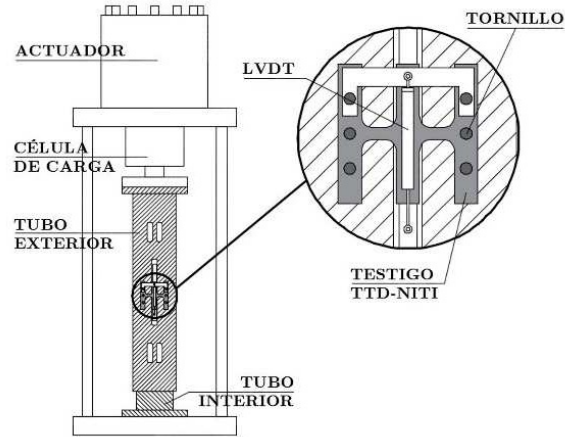
Obrázek 1.6: Opětovný nárůst vstupního zatížení a odpovídající Preisachův trojúhelník

1.3 Aplikace PM modelu

V této práci se budeme zabývat zejména mechanickou hysterezi materiálů a budeme ji využívat k vyhodnocení jejich elasticity, resp. poškození. V takovém případě hodnoty α a β budou značit hodnoty zavíracího a otevírací tlaku hysteronů. To, že je hysteron otevřený znamená, že při dosažení tohoto tlaku už hysteron ztratí schopnost pohltit další zatížení a už nenapomáhá tlumení. Při znázornění na Preisachově trojúhelníku (obr. 1.3) to bude znamenat, že hysterony blízko dokonale elastické diagonále $\alpha = \beta$ budou téměř dokonale pohlcovat veškeré mechanické zatížení, zatímco vzdálenější hysterony budou stále více poškozené. Budeme jednak zpracovávat uměle vytvořená data, na kterých budou vytvářeny jednotlivé postupy, ale také reálná data



Obrázek 1.7: Schéma působení vstupního zatížení (převzato z [14])



Obrázek 1.8: Obecné schéma testovaného tlumiče (převzato z [14])

z testování disipativních tlumičů, které pohlcují mechanické deformace při zemětřesení a snaží se tak ochránit budovu před poškozením. Tyto pasivní ochranné prostředky získávají v posledních letech na popularitě, neboť jejich výměna je snadnější a levnější než oprava nosných konstrukcí. Avšak je zapotřebí efektivní vyhodnocení stupně jejich poškození, o což se budeme snažit. Data pochází nedestruktivního testování tlumičů, provedeném Univerzitou v Granadě ve spolupráci s Univerzitou v Madridu. Podrobnější informace o experimentu jsou k nalezení v [14]. Systém tlumičů byl složen z hlavního rámu s sérií ocelových tlumičů (obr. 1.8) umístěných v železobetonové konstrukci, přičemž celá tato soustava byla podrobena mechanickému zatížení napodobujícímu deformace při zemětřesení.

Kapitola 2

Identifikace PM prostoru

Mezi hlavní problémy používání PM formalismu je rychlé a dostatečně přesné nalezení odpovídajícího PM prostoru, který přísluší dané dvojici vstupního zatížení a hysterenzní křivky. V následujících sekcích této kapitoly si proto představíme důležité nástroje pro tento účel. Nejprve představíme vhodná rozdělení pravděpodobnosti, které dobře popisují rozložení hysteronů v materiálu. Dále představíme divergenční míry, které budou použity k vyhodnocení rozdílnosti mezi naměřenými a optimalizací získanými hysterenzními křivkami. A na závěr popíšeme použité optimalizační algoritmy, které byly implementovány pro nalezení parametrů rozdělení.

2.1 Pravděpodobnostní rozdělení

Nejdříve začneme představením pravděpodobnostních distribucí, která jsou typicky používána pro nehomogenní materiály. Jedná se o rozdělení, která představili Guyer a McCall ([15]), a rozdělení Koen. Následuje jejich kombinace v podobě rozdělení Guyer3, které bylo představeno v práci ([7]). V naší aplikaci bude hodnotu vstupního zatížení $u(t)$ zastávat tlak P , který působí na materiál. Konkrétně budou pomocí pravděpodobnostních rozdělení popsány hodnoty zavíracího a otevírací tlaku hysteronů, ozn. P_α a P_β , které charakterizují vlastnosti těchto hysterenzních operátorů. Dále M značí maximální hodnotu dosaženého vstupního tlaku, a r bude značit náhodné číslo, které je zvlášť pro P_α , P_β každého hysteronu rovnoměrně generováno z intervalu $(0, 1)$.

Rozdělení Guyer1

$$P_\alpha = M \cdot r^\eta, \quad P_\beta = P_\alpha \cdot r^\kappa, \quad (2.1)$$

kde $\eta, \kappa \geq 0$ jsou parametry rozdělení.

Rozdělení Guyer2

$$P_\alpha = M \cdot r^\eta, \quad P_\beta = P_\alpha \cdot r^{0.25+0.75\kappa}, \quad (2.2)$$

kde $\eta, \kappa \geq 0$ jsou parametry rozdělení.

Rozdělení Koen

$$P_\alpha = M \cdot r, \quad P_\beta = \left(\frac{P_\alpha}{\lambda}\right)^\kappa \cdot r, \quad (2.3)$$

kde $\kappa, \lambda \geq 0$ jsou parametry rozdělení.

Rozdělení Guyer3

$$P_\alpha = M \cdot r^\eta, \quad P_\beta = P_\alpha \cdot (\lambda r)^\kappa, \quad (2.4)$$

kde $\eta, \kappa \geq 0$, $\lambda \in (0, 1)$ jsou parametry rozdělení. Dále budou představena standardně používaná rozdělení, která jsou useknuta tak, že jejich definiční obor je omezen na $x_1, x_2 \in [0, M]$ a $x_1 \geq x_2$, aby otevírací a zavírací hodnoty ležely v Preisachově trojúhelníku. Kvůli těmto omezením bude porušena normalizace hustot pravděpodobností, ale pro charakter naší úlohy je definice takovýchto rozdělení dostačující. Bude se jednat o dvourozměrné normální rozdělení, které generuje přímo body v PM rovině, a které se osvědčilo na reálných datech v práci [8]. Kromě výše zmíněných rozdělení, kde obě složky spolu souvisí, použijeme také další pravděpodobností rozdělení, podle kterých však budou nezávisle generovány hodnoty obou složek, tedy zavíracího a otevíracího tlaku P_α a P_β .

Gaussovské normální rozdělení (useknuté)

$$(P_\alpha, P_\beta)^T = \frac{1}{2\pi \sqrt{|\mathbb{C}|}} \exp\left(-\frac{1}{2}(x - \mu)^T \mathbb{C}^{-1}(x - \mu)\right), \quad (2.5)$$

kde parametry jsou vektor středních hodnot $\mu = (\mu_1, \mu_2)$, $\mu_1, \mu_2 > 0$ a pozitivně definitní a symetrická korelační matice \mathbb{C} , přičemž $x = (x_1, x_2)$.

Exponenciální rozdělení (useknuté)

$$P_\alpha = \mu_1 \exp(-\mu_1 x_1), \quad P_\beta = \mu_2 \exp(-\mu_2 x_2), \quad (2.6)$$

kde $\mu_1, \mu_2 > 0$ jsou parametry rozdělení.

Weibullovo rozdělení (useknuté)

$$P_\alpha = \frac{\kappa_1}{\lambda_1} \left(\frac{x_1}{\lambda_1}\right)^{\kappa_1-1} \exp\left(-\left(\frac{x_1}{\lambda_1}\right)^{\kappa_1}\right), \quad P_\beta = \frac{\kappa_2}{\lambda_2} \left(\frac{x_2}{\lambda_2}\right)^{\kappa_2-1} \exp\left(-\left(\frac{x_2}{\lambda_2}\right)^{\kappa_2}\right), \quad (2.7)$$

kde $\kappa_1, \kappa_2, \lambda_1, \lambda_2 > 0$ jsou parametry rozdělení.

Laplaceova rozdělení (useknuté)

$$P_\alpha = \frac{1}{2\lambda_1} \exp\left(-\frac{|x - \mu_1|}{\lambda_1}\right), \quad P_\beta = \frac{1}{2\lambda_2} \exp\left(-\frac{|x - \mu_2|}{\lambda_2}\right), \quad (2.8)$$

kde $\lambda_1, \lambda_2, \mu_1, \mu_2 > 0$ jsou parametry rozdělení.

Mimo samotných pravděpodobnostních rozdělení budeme pro modelování PM prostorů používat také distribuční směsi ([16]), které v praxi lépe popisují chování hystereze. Vzhledem k dvourozměrné úloze Preisachovy roviny budeme následující definici aplikovat pro $d = 2$.

Definice 2.1.1. (Distribuční směs). Distribuční směsí nazveme každou konvexní kombinaci, pro kterou

$$p(\mathbf{x}|\boldsymbol{\alpha}) = \sum_{i=1}^k \alpha_i p_i(\mathbf{x}), \quad \sum_{i=1}^k \alpha_i = 1, \quad \alpha_i > 0, \quad (2.9)$$

kde $k \in \mathbb{N}$ značí počet komponent, $\mathbf{x} \in \mathbb{R}^d$, $d \in \mathbb{N}$, $p_i(\mathbf{x})$ jsou pravděpodobnostní rozdělení jednotlivých komponent směsi.

Mimo této základní kombinace budeme pracovat i se zobecněním směsi směsí ve znění použitém ve [3]. V dosavadním přístupu bylo ve směsi vždy dané pravděpodobnostní rozdělení použito maximálně jednou, proto pomocí této specifické směsi chceme přidat možnost opakování daného rozdělení a zároveň zachovat kontrolu nad možnými kombinacemi prvků ve směsi, neboť slepé prohledávání všech možných kombinací s opakováním by s rostoucím počtem prvků ve směsi bylo výpočetně náročnější.

Definice 2.1.2. (Distribuční směs směsí). Distribuční směsí směsí nazveme každou konvexní kombinaci, pro kterou

$$p(\mathbf{x}|\boldsymbol{\lambda}, \boldsymbol{\alpha}) = \sum_{j=1}^K \lambda_j p_j(\mathbf{x}|\boldsymbol{\alpha}), \quad \sum_{j=1}^K \lambda_j = 1, \quad \lambda_j > 0, \quad (2.10)$$

kde $K \in \mathbb{N}$ značí počet hlavních komponent, $\mathbf{x} \in \mathbb{R}^d$, $d \in \mathbb{N}$ a $p_j(\mathbf{x}|\boldsymbol{\alpha})$ jsou klasické distribuční směsi z definice 2.1.1, tedy

$$p_j(\mathbf{x}|\boldsymbol{\alpha}) = \sum_{i=1}^k \alpha_{ji} p_i(\mathbf{x}), \quad \sum_{i=1}^k \alpha_{ji} = 1, \quad \lambda_i > 0, \quad \forall j \in \{1, \dots, K\}, \quad (2.11)$$

kde $k \in \mathbb{N}$ je počet sub-komponent a $p_i(\mathbf{x})$ jsou jednotlivá pravděpodobnostní rozdělení.

2.2 Divergenční míry

V další sekci této kapitoly se zaměříme na způsoby vyhodnocení odlišnosti mezi hysterezními křivkami. Tento nástroj využijeme kromě optimalizací také v dalších částech této práce. Při optimalizaci se budeme snažit minimalizovat rozdíl mezi změřenou hysterezní křivkou a

křivkou získanou z PM prostoru. Základním přístup je pomocí L_2 , resp. L_1 metriky, které jsou definované na intervalu (a, b) pro spojité funkce f a g jako

$$\|f - g\|_2 = \sqrt{\int_a^b (f(x) - g(x))^2 dx}, \quad \|f - g\|_1 = \int_a^b |f(x) - g(x)| dx, \quad (2.12)$$

Dále budeme používat tzv. ϕ -divergence, které sice nesplňují požadované vlastnosti pro metriky, ale naše problematika nebrání jejich uplatnění.

Definice 2.2.1. (ϕ -divergence). Buď P_1 a P_2 jsou distribuční funkce pravděpodobnostního rozdělení na nějakém měřitelném prostoru $(\mathcal{X}, \mathcal{A})$ vzhledem k nějaké σ -konečné míře na μ . Potom ϕ -divergenci definujeme jako

$$D_\phi(P_1, P_2) = \int p_1 \phi\left(\frac{p_1}{p_2}\right) d\mu, \quad (2.13)$$

kde p_1 a p_2 jsou hustoty těchto distribucí a ϕ konvexních funkce na intervalu $[0, \infty)$ a navíc ryze konvexní na okolí 1 splňující vlastnosti

$$\phi(1) = 0, \quad \phi(0) = \lim_{t \rightarrow 0_+} \phi(t), \quad 0\phi\left(\frac{0}{0}\right) = 0, \quad 0\phi\left(\frac{a}{0}\right) = \lim_{t \rightarrow \infty} \frac{\phi(t)}{t}, \quad a > 0.$$

Pro ϕ -divergence lze dokázat různé užitečné vlastnosti, jako například monotonii, nebo pro zpřísnění předpokladů také symetrii. Takové chování je důležité při aplikaci, neboť požadujeme, aby se jejich hodnota zvyšovala při zvětšující se vzdálenosti mezi zkoumanými distribucemi. Více k vlastnosti ϕ -divergencí nalezneme např. v [17]. Přejdeme-li k použitým ϕ -divergencím, tak první z nich, Hellingerova divergence, je dána předpisem

$$H^2(P_1, P_2) = \frac{1}{2} \int (\sqrt{p_1} - \sqrt{p_2})^2 d\mu, \quad (2.14)$$

pro kterou platí $\phi(t) = (1 - \sqrt{t})^2$. Hellingerova divergence navíc splňuje definici metriky. Dále použijeme LeCammovu divergenci ve tvaru

$$LC^2(P_1, P_2) = \int \frac{(p_1 - p_2)^2}{p_1 + p_2} d\mu, \quad (2.15)$$

která vzniká pomocí generující konvexní funkce

$$\phi(t) = \frac{1 - t}{2t + 2}. \quad (2.16)$$

Očividně, pro LeCammovu divergenci je splněna symetrie i reflexivita. Po odmocnění platí trojúhelníková nerovnost, a tedy LeCammova divergence je také metrika. Při aplikaci těchto divergencí, budeme používat místo distribucí běžné funkce.

2.3 Heuristické algoritmy pro optimalizaci

Pro nalezení nejvhodnějšího pravděpodobnostního rozdělení a jeho parametrů budeme využívat zejména Jaya algoritmus ([18]), což je meta-heuristický optimalizační algoritmus představený roce 2016. Kromě tohoto budeme používat i modifikovanou variantu aDE-Jaya ([19]), která spojuje výhody standardní verze Jaya algoritmu a evoluční optimalizace, konkrétně diferenciální evoluce (DE), viz [20]. Srovnáním těchto algoritmů včetně simulovaného žíhání na naší problematiku jsme provedli v práci [7] a vzhledem k jejich osvědčení je budeme používat i nadále.

2.3.1 Jaya algoritmus

Tato heuristická metoda je založená na myšlence, že v průběhu optimalizace by se výsledek dané úlohy měl přibližovat k nejlepšímu řešení a souběžně vzdalovat od toho nejhoršího. Pro běh tohoto algoritmu nejsou nutné téměř žádné vstupní parametry. K zapotřebí je pouze počet jedinců v populaci (NP), potažmo podmínky pro zastavení (dosažená přesnost výsledku, počet iterací). Úkolem je optimalizovat danou účelovou funkci $f(\mathbf{x})$ s d -dimenzionální nezávislou proměnnou \mathbf{x} . Algoritmus 2.1 začne nejdříve tak, že náhodně vygeneruje NP -rozměrnou populaci hodnot vstupní proměnné \mathbf{x} funkce f . Dále v populaci nalezne nejlepší (*best*) a nejhorší (*worst*) výsledek a zapamatuje si jejich indexy. Poté již v každé iteraci pomocí Jaya operátoru jsou modifikovány jednotlivé komponenty každého členu prvku iterace dokud není dosaženo zastavovací podmínky podle následujícího předpisu

$$x_{i,j}^{(k)} = x_{i,j}^{(k-1)} + r_1 \left(x_{best,j}^{(k-1)} - |x_{i,j}^{(k-1)}| \right) - r_2 \left(x_{worst,j}^{(k-1)} - |x_{i,j}^{(k-1)}| \right), \quad (2.17)$$

kde $i \in \{1, \dots, NP\}$ značí index prvku populace, $j \in \{1, \dots, d\}$ je index složky \mathbf{x} , k je číslo iterace a r_1, r_2 jsou náhodně generovaná čísla z uniformního rozdělení v intervalu $[0, 1]$ zvláště $\forall i, j, k$. Člen, který přičítáme, zastupuje posouvání se k nejlepšímu výsledku, zatímco člen se záporným znamínkem symbolizuje vzdalování se toho nejhoršího. Pokud nové hodnoty prvků vstupní proměnné daného členu populace jsou lepší než v předchozí iteraci, použijeme novou hodnotu. Jinak si ponecháme hodnoty z předchozí iterace, tedy $x_i^{(k)} = x_i^{(k-1)}$. Na závěr dané iterace jsou nalezeny nové indexy členů populace s nejlepším a nejhorším výsledkem. Celý cyklus opakujeme, dokud není dosažena podmínka pro zastavení, kterou může být počet iterací nebo požadovaná přesnost řešení.

Algoritmus 2.1: Pseudokód pro Jaya algoritmus

- 1 Náhodné vygenerování členů počáteční populace $\mathbf{x}_i^{(1)}$, $\forall i \in \{1, \dots, NP\}$
 - 2 Nalezení nejlepšího $\mathbf{x}_{best}^{(1)}$ a nejhoršího prvku $\mathbf{x}_{worst}^{(1)}$ v populaci
 - 3 **while** k splní zastavovací podmínku **do**
 - 4 **for** $i = 1$ to NP **do**
 - 5 **for** $j = 1$ to d **do**
 - 6 $x_{i,j}^{(k)} = x_{i,j}^{(k-1)} + r_1 \left(x_{best,j}^{(k-1)} - |x_{i,j}^{(k-1)}| \right) - r_2 \left(x_{worst,j}^{(k-1)} - |x_{i,j}^{(k-1)}| \right)$
 - 7 **if** $f(\mathbf{x}_i^{(k)}) \geq f(\mathbf{x}_i^{(k-1)})$ **then** $\mathbf{x}_i^{(k)} = \mathbf{x}_i^{(k-1)}$
 - 8 Nalezení $\mathbf{x}_{best}^{(k)}$ a $\mathbf{x}_{worst}^{(k)}$
-

2.3.2 aDE-Jaya

Kromě klasické varianty Jaya algoritmu použijeme také jeho upravenou variantu, která vznikla přidáním mutačního operátoru diferenciální evoluce (DE). Hybridní adaptivní diferenciální evoluce a Jaya algoritmus (aDE-Jaya) byl navržen v [19] pro podobnou úlohu hledání parametrů pravděpodobnostních rozdělání pro popis hysterezního chování materiálu. Ačkoliv byl navržen pro tzv. Bouc-Wenův model hystereze, v práci [7] se nám jeho použití osvědčilo i pro použití v Preisachově modelu hystereze, a proto ho budeme aplikovat i zde. Jak již bylo zmíněno, tento přístup kombinuje při vytváření nové hodnoty parametru pomocí adaptivní kontroly klasický Jaya operátor společně s operátorem DE. Podle autorů je takový přístup schopen lepšího kompromisu mezi nalézáním globálních a lokálních minim a tím pádem rychlejšího dosažení lepšího výsledku než běžný Jaya algoritmus. Operátor diferenciální evoluce vytváří novou hodnotu parametru v k -té iteraci podle pravidla

$$x_{i,j}^{(k)} = x_{i_1,j}^{(k-1)} + F(x_{i_2,j}^{(k-1)} - x_{i_3,j}^{(k-1)}) \quad (2.18)$$

kde $i \in \{1, \dots, NP\}$ značí index prvku populace, $j \in \{1, \dots, d\}$ je index složky \mathbf{x} , $i_1 \neq i_2 \neq i_3 \neq i$, kde $i_1, i_2, i_3 \in \{1, \dots, NP\}$ jsou náhodně zvolené indexy členů populace, která určují prvky populace použité při mutaci a F je parametr škálování, který je náhodně volen z intervalu $[0, 1]$.

Algoritmus 2.2: Pseudokód modifikace i -tého prvku aDE-Jaya algoritmu

```

1   $j_{rand} = randint[1, d]$ 
2   $CR = rand[0.7, 1.0]$ 
3   $F = rand[0.4, 1.0]$ 
4  for  $j = 1$  to  $d$  do
5      if  $rand[0, 1] < CR$  or  $j = j_{rand}$  then
6          if  $rand[0, 1] > 0.3$  then
7              Náhodná volba indexů  $i_1 \neq i_2 \neq i_3 \neq i$ , kde  $i_1, i_2, i_3 \in \{1, \dots, NP\}$ 
8               $x_{i,j}^{(k)} = x_{i_1,j}^{(k-1)} + F(x_{i_2,j}^{(k-1)} - x_{i_3,j}^{(k-1)})$ 
9          else
10              $x_{i,j}^{(k)} = x_{i,j}^{(k-1)} + r_1(x_{best,j}^{(k-1)} - |x_{i,j}^{(k-1)}|) - r_2(x_{worst,j}^{(k-1)} - |x_{i,j}^{(k-1)}|)$ 
11             if  $f(x_i^{(k)}) \geq f(x_i^{(k-1)})$  then  $x_i^{(k)} = x_i^{(k-1)}$ 
12         else
13              $x_i^{(k)} = x_i^{(k-1)}$ 

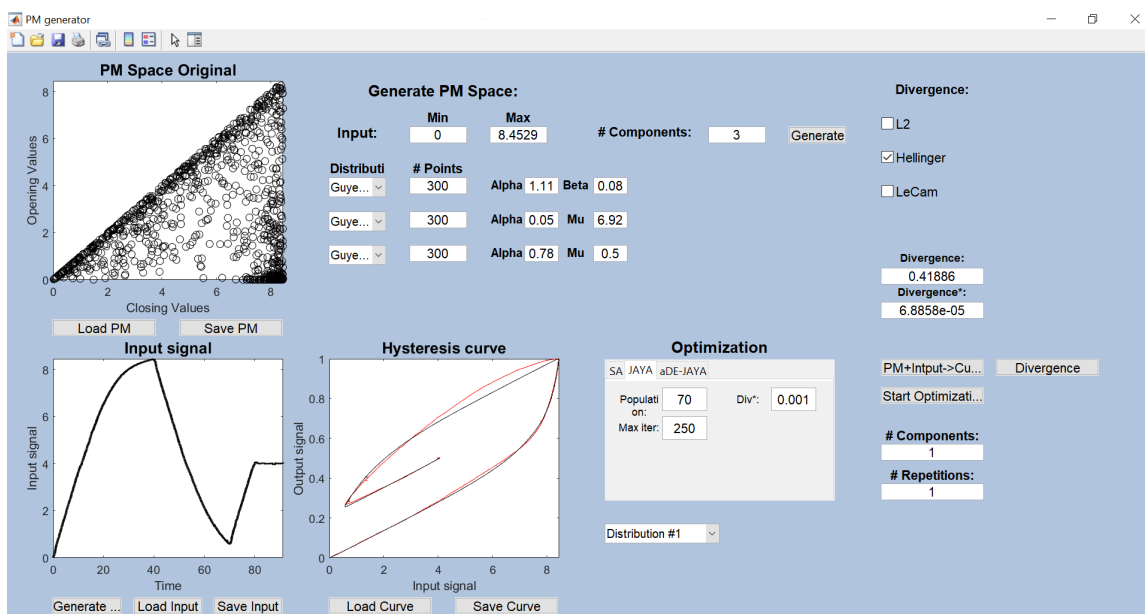
```

Adaptivní kontrola je v aDE-Jaya algoritmu implementovaná tak, že při vytváření nové hodnoty složky proměnné je v přibližně 70% použita DE mutace (2.18) a ve zbylých případech úprava provedena podle Jaya algoritmu (2.17). Další nově zavedený kontrolní parametr je parametr míry překročení CR (*crossover rate*), který rozhoduje o tom, jestli bude pro danou složku proměnné vytvářena nová proměnná nebo ne. Tato procedura tedy svojí strukturou odpovídá algoritmu 2.1 s tím rozdílem, že vytváření nové hodnoty na řádcích pět až sedm je nahrazeno kódem 2.2. Stejně jako pro poměr mezi použitím DE a Jaya operátoru navrhuje autoři po testování tohoto přístupu (viz [19]), aby kontrolní parametry byly pro každý člen populace náhodně rovnoměrně generovány a to pro CR z intervalu $[0.7, 1]$ a pro F z intervalu $[0.4, 1]$, čímž mají

být zaručeny lepší schopnosti algoritmu pro hledání lokálních a globálních extrémů. Také je to účelné pro používání různých směrů hledání, a proto tuto konfiguraci převezmeme i pro tuto práci.

2.3.3 Softwarová implementace

V pracích [7] a [8] byl v softwaru Matlab postupně vytvářen program pro hledání odpovídajícího Preisachova prostoru ze zadaného vstupního zatížení a naměřené hysterezní křivky, který používá pro hledání parametrů pravděpodobnostních rozdělání a jejich zastoupení ve směsi distribucí heuristické algoritmy a ϕ -divergence ve formě, v jaké byly představeny výše v této kapitole. Uživatelské prostředí tohoto softwaru můžeme vidět na obrázku 2.1. Po nahrání vstupního zatížení a odpovídající hysterezní smyčky můžeme dole uprostřed zvolit heuristický algoritmus, který chceme použít a zadat jeho parametry. Dále po výběru počtu složek, o jaká rozdělání se jedná a jejich parametrů a zastoupení můžeme pomocí tlačítka *Start Optimization* spustit hledání parametrů. Můžeme dále zvolit kolikrát se má daná optimalizace opakovat a jakou divergenční míru má použít. Po doběhnutí programu můžeme vidět vykreslený PM prostor, přikreslení k němu odpovídající hysterezní křivky a vpravo uprostřed hodnotu divergence mezi původní a nalezenou hysterezní smyčkou. Tato hodnota je pod ní ještě normována počtem bodů, ze kterých se skládá vstupní zatížení, pro účely možného porovnávání mezi různými daty.



Obrázek 2.1: Grafické rozhraní implementovaného softwaru

Kromě této hlavní funkce, zde můžeme dále například uprostřed nahoře pomocí tlačítka *Generate* generovat Preisachovy prostory z vybraných pravděpodobnostních rozdělání a zadaných hodnot parametrů. Dále pomocí tlačítka *PM+Input->Curve* můžeme vykreslit hysterezní smyčku, která odpovídá aplikaci nahraného zatížení na současně používaný PM prostor.

Tlačítko *Divergence* napočítá vybranou ϕ -divergenci mezi nahranou a vygenerovanou hysterezní smyčkou. Samozřejmostí je možnost nahrát nebo uložit jak vstupní zatížení a hysterezní smyčku, tak i PM prostor.

Kapitola 3

Klasifikační a regresní metody strojového učení

Ačkoliv přístup pomocí heuristických optimalizací dosahuje slibných výsledků, jak se již ukázalo v pracích [7] a [8], tak zůstává problém s neefektivností. Dosavadní systém předpokládal pro nalezení výsledku použití optimalizace zvlášť pro všechna pravděpodobnostní rozdělení, resp. pro všechny jejich směsi a z nich pak vybrat nejlepší výsledek. Proto si v této kapitole představíme klasifikační a regresní metody strojového učení, které následně použijeme pro určení vhodného pravděpodobnostního modelu PM prostoru, tzn. směsi distribucí z dostupných dat. Budeme se zabývat pouze metodami strojového učení s učitelem (supervised machine learning), kdy máme k dispozici vstupní data a očekávaný výstup. Učení takové metody můžeme obecně popsat následovně.

Definice 3.0.1. Necht' \mathcal{X} , \mathcal{Y} a \mathcal{Z} jsou měřitelné prostory. Při učení máme data ze \mathcal{Z} a ztrátovou funkci $\mathcal{L} : \mathcal{M}(\mathcal{X}, \mathcal{Y}) \times \mathcal{Z} \rightarrow \mathbb{R}$, kde $\mathcal{M}(\mathcal{X}, \mathcal{Y})$ je množina měřitelných funkcí z \mathcal{X} do \mathcal{Y} . Cílem je zvolit množinu funkcí $\mathcal{F} \subset \mathcal{M}(\mathcal{X}, \mathcal{Y})$ a zkonstruovat učící algoritmus, tedy zobrazení

$$\mathcal{A} : \bigcup_{m \in \mathbb{N}} \mathcal{Z}^m \rightarrow \mathcal{F}, \quad (3.1)$$

kteří používá trénovací data $s = (z^{(i)})_{i=1}^m \in \mathcal{Z}^m$ k nalezení modelu $\varphi = \mathcal{A}(s) \in \mathcal{F}$, který dosahuje dobrých výsledků na trénovacích datech s a tedy dosahuje požadovaně malé hodnoty \mathcal{L} , ale také dobře zobecňuje pro dosud neznámá data $z \in \mathcal{Z}$, na kterých se hodnota \mathcal{L} příliš nezhorší oproti trénovacím datům.

Právě výsledky na testovacích datech, které nebyly použity při trénovacím procesu, budou hlavním ukazatelem pro vyhodnocování dané metody. Stejně obecně můžeme popsat predikci takové metody. Pro predikci mějme $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$, neboli trénovací data $s = ((x^{(i)}, y^{(i)}))_{i=1}^m$, která se skládají z prediktorů $x^{(i)} \in \mathcal{X}$ a k nim odpovídajících příznaků (tzv. label) $y^{(i)} \in \mathcal{Y}$. Cílem je poté hledat takové $\varphi : \mathcal{X} \rightarrow \mathcal{Y}$, které pro neznámý pár $(x, y) \in \mathbb{R}^d \times \mathcal{Y}$ přiřadí predikci $\hat{y} = \varphi(x)$, která co možná nejlépe odpovídá skutečné hodnotě příznaku y . Konkrétně budeme v této práci hledat klasifikátor, resp. regresor $\varphi : \mathcal{X} \rightarrow \mathcal{Y}$, kde navíc prediktory budou pocházet z euklidovského prostoru, tedy $\mathcal{X} \subset \mathbb{R}^d$, kde $d \in \mathbb{N}$. Ohledně volby ztrátové funkce \mathcal{L} , kde vždy N bude značit

počet vzorků, budeme pro klasifikační úlohu používat křížovou entropii (*cross entropy*), kterou pro případ binární klasifikace, tedy $\mathcal{Y} = \{0, 1\}$, definujeme jako

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^N y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i). \quad (3.2)$$

Tu můžeme zobecnit pro případ klasifikace do více tříd, tedy $\mathcal{Y} = \{0, 1, \dots, C - 1\}$, kde $C \in \mathbb{N}$ je počet tříd, jako

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log(\hat{y}_{i,c}), \quad (3.3)$$

kde $y_{i,c}$ je myšleno indexování přes počet vzorků a přes počet tříd. Speciálním případem jsou stromové metody, které používají funkci vyhodnocující kvalitu větvení v uzlu, kterou nazveme impurity. Nejpoužívanějšími je tzv. Gini impurity definovaná jako

$$\mathcal{L}_{Gini} = \sum_{c=1}^C p_c (1 - p_c) \quad (3.4)$$

a dále Entropy impurity ve tvaru

$$\mathcal{L}_{Entropy} = -\sum_{c=1}^C p_c \log_2(p_c) \quad (3.5)$$

kde p_c je pravděpodobnost výskytu třídy c v uzlu. Stromové metody rozebereme podrobněji dále v této kapitole. Obdobně pro případ jednodimenzionální regrese, tedy $\mathcal{Y} \subset \mathbb{R}$, budeme používat dvě ztrátové funkce, jednak střední kvadratickou chybu (*mean squared error*, MSE)

$$\mathcal{L}_{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (3.6)$$

a střední absolutní chybu (*mean absolute error*, MAE)

$$\mathcal{L}_{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|. \quad (3.7)$$

3.1 Kritéria vyhodnocení kvality odhadu

Začneme-li klasifikací, tak nejdříve si představíme nástroje pro vyhodnocení binární klasifikace, kdy přiřazujeme data pouze do dvou tříd a následně je zobecníme pro případ více tříd.

Definice 3.1.1. Necht' $\mathbf{x}_i = (x_{i1}, \dots, x_{id})$, kde $x_{ij} \in \mathbb{R}$, $i \in \{1, \dots, N\}$, $j \in \{1, \dots, d\}$ je realizace vektoru náhodných veličin (X_1, \dots, X_d) a necht' jsou dány disjunktní třídy A a B. Necht' $l(x)$ je věrohodnostní poměr

$$l(\mathbf{x}_i) = \frac{p(\mathbf{x}_i|A)}{p(\mathbf{x}_i|B)}. \quad (3.8)$$

Pak binárním klasifikátorem rozumíme funkci $\varphi : \mathbb{R}^d \rightarrow \{0, 1\}$ takovou, že platí

$$\varphi(\mathbf{x}_i) = \begin{cases} 1, & \text{když } l(\mathbf{x}_i) > \frac{P(B)}{P(A)} \\ 0, & \text{jinak.} \end{cases} \quad (3.9)$$

Obě hodnoty $P(A)$ a $P(B)$ jsou pro daný dataset typicky neznámé a proto je třeba výraz $\frac{P(B)}{P(A)}$ nahradit prahem δ a tuto hodnotu optimalizovat vůči zvolenému kritériu. Ozn. δ^* jako argument maxima zvoleného kritéria binární klasifikace.

Pro binární klasifikaci existují různé metriky kvality modelu závislé na počtu nulových hypotéz, které jsou ve skutečnosti pravdivé/neppravdivé, vyhodnocené jako pravdivé/neppravdivé (*true positive* = TP / *true negative* = TN) nebo vyhodnocené nesprávně (*false positive* = FP / *false negative* = FN). Nyní si představíme hodnotící metriky použité v této práci.

Definice 3.1.2. Veličinou Accuracy nazveme

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (3.10)$$

veličinou Precision nazveme

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (3.11)$$

veličinou Recall (nebo také True positive rate) nazveme

$$\text{Recall} = \text{TPR} = \frac{TP}{TP + FN}, \quad (3.12)$$

veličinou F1 score nazveme

$$\text{F1} = 2 \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (3.13)$$

Pokud budeme chtít zobecnit metriky z definice 3.1.2, tak nejpřímočařejší řešení je pro accuracy. Jedná se o podíl všech správně přiřazených tříd vůči celkovému počtu všech vzorků. Když budeme uvažovat o třídách místo o jednotlivých prvcích, budou existovat třídy s velkým zastoupením a jiné s malým počtem prvků. V této situaci budou mít třídy s vysokým počtem členů vyšší váhu v porovnání s těmi nejmenšími. Accuracy má tedy tendenci skrývat silné klasifikační chyby u tříd s malým zastoupením, protože tyto třídy jsou méně relevantní ve srovnání s těmi největšími. Pro ostatní metriky existuje několik způsobů průměrování. Prvním průměrováním je tzv. *macro* průměrování, které vezme danou metriku vůči jednotlivým třídám a následně ji zprůměruje do výsledné metriky. Obdobně můžeme zadefinovat *micro* průměrování. V tomto případě používáme celkový počet TP , TN , FP a FN napříč všemi třídami. To je ovšem stejná definice jako accuracy pro více tříd, a tedy micro-precision, micro-recall ani micro-F1 nepřinášejí nové užitečné informace. Posledním způsobem je *weighted* (vážené) průměrování, které stejně jako pro macro napočítá metriku vůči jednotlivým třídám, ale výsledek vznikne jako vážený průměr vůči počtu zastoupení správně klasifikovaných prvků v každé třídě.

Dalšími nástroji pro vyhodnocení kvality klasifikace jsou tzv. ROC křivky (*receiver operating characteristic*) a AUC skóre (*area under curve*), která udává obsah pod ROC křivkou. Tyto

jsou opět zdefinovaly pro binární klasifikaci, neboť ROC křivka znázorňuje závislost true positive rate a false positive rate pro různé hodnoty prahu $\delta \in (0, 1)$, podle kterého se řídí přiřazení do jedné ze dvou daných tříd. Nicméně, obdobně můžeme zobecnit pro více tříd pomocí macro nebo weighted průměrování AUC skóre pro každou dvojici tříd. Pro regresi budeme používat pro vyhodnocení predikce opět MSE a MAE definované v rovnicích (3.6) a (3.7).

3.2 Stromové a ensemble metody

Stromové ensemble metody, kterými jsme se zabývali již v práci [8], jsou oblíbeným nástrojem zejména kvůli dobrému výkonu a nízké výpočetní náročnosti a také kvůli snadné interpretovatelnosti výsledků. Začneme nejdříve rozhodovacími stromy a následně pro ně představíme ensemble metody, přičemž některé z nich použijeme i pro vytvoření meta-modelu tvořeném z metod, které se ukáží jako nejperspektivnější. Rozhodovací strom (decision tree) je algoritmus, který provádí rozhodnutí na základě posloupnosti pravidel uspořádaných do stromové struktury. Skládá se z nelistových uzlů, ve kterých se testuje podmínka pro dané příznaky, větvi, které znázorňují možné výsledky testu, a listů reprezentující výslednou příslušnost do dané třídy nebo predikovanou hodnotu. Rozhodování takového klasifikátoru funguje tak, že nové pozorování prochází od kořene k listům natrénovanými rozhodovacími pravidly.

Jako náhodný les (random forest) nazveme metodu skládající se ze souboru rozhodovacích stromů, při jejichž vytváření jsou používány metody náhodného výběru příznaků a tzv. *bootstrap aggregating* neboli *bagging* vzorkovací metoda [25], kterou si nyní představíme.

Definice 3.2.1. Necht' $M, N, N' \in \mathbb{N}$ a je dána množina $(\mathbf{X}, \mathbf{y}) = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$. Pak baggingem nazveme vytváření množin $\alpha_1, \dots, \alpha_M$ z množiny (\mathbf{X}, \mathbf{y}) , kde pro $m \in \{1, \dots, M\}$ je $\alpha_m = \{(\mathbf{x}_{j_1}, y_{j_1}), \dots, (\mathbf{x}_{j_{N'}}, y_{j_{N'}})\}$, přičemž uspořádané dvojice $(\mathbf{x}_{j_k}, y_{j_k})$ jsou vybrány z původní množiny (\mathbf{X}, \mathbf{y}) náhodně rovnoměrně s opakováním.

Bagging lze použít i při vytváření meta-modelu, kdy vytvoříme více modelů, které jsou natrénovány na náhodném vzorku $(\mathbf{x}_{j_k}, y_{j_k})$ z původních dat (\mathbf{X}, \mathbf{y}) a kombinuje výsledky všech základních modelů do finálního klasifikátoru. Pro určení výsledku se používá tzv. hlasování (*voting*). Pro klasifikaci může být voting dvojího druhu a to hard voting, kdy za výsledek bereme tu třídu, která vyšla nejčastěji mezi základními klasifikátory, a soft voting, pro který se nejdříve zprůměrují pravděpodobnosti tříd jednotlivých klasifikátorů a třídu s nejvyšší pravděpodobností bereme jako výsledek. Pro regresi probíhá voting opět jako průměrování výsledků základních modelů. Samostatný voting v těchto variantách může hrát roli ensemble nástroj pro vytváření meta-modelu.

Ačkoliv takováto varianta náhodného lesa je robustním modelem vůči šumu a přefitování, tedy špatné generalizaci na neznámých datech, existují další alternativy, které mohou pro různé aplikace dosahovat lepších výsledků. První z nich je tzv. AdaBoost boosting, která je založena na kombinování mnoha slabých prediktorů, které jsou pouze mírně lepší než čistě náhodná predikce a ze kterých je pomocí speciální vážené lineární kombinace vytvořen výsledný model. AdaBoost aplikuje trénovací algoritmus na vážené vzorky $(\mathbf{x}_{j_k}, y_{j_k})$, přičemž v každé iteraci roste váha špatně predikovaných prvků a je naopak snížena u správných předpovědí. Výsledný model je lineární kombinací všech modelů získaných v různých iteracích.

Poslední představenou ensemble metodou je tzv. *extreme gradient boosting* neboli XGBoost. Motivací vzniku tohoto algoritmu byla maximalizace prediktivních schopností stromových metod, a proto spojuje dohromady více přístupů. Jako boosting uvažujeme optimalizační úlohu, která minimalizuje ztrátovou funkci postupným přidáváním slabého prediktoru do ensmbly. Algoritmus postupně minimalizuje chybu předešlého modelu ve směru gradientu a vytváří tak model nový. XGBoost zahrnuje regularizace ztrátové funkce v L_1 a L_2 normě, čímž se snaží omezit přefitování. Dále obsahuje strukturu pro paralelní výpočty v rámci jednoho stromu pro urychlení výpočtů, nástroje pro efektivní práci s nevyváženými daty a další.

3.3 Hluboké neuronové sítě

Nově použijeme pro účely klasifikace také hluboké neuronové sítě. S rozvojem výpočetní techniky se hluboké neuronové sítě staly hojně používanou metodou a toto dominantní postavení si získaly díky široké škále aplikovatelností, přestože jejich interpretovatelnost není již tak jednoduchá jako je tomu u stromových metod. Pro hluboké neuronové sítě se množina prediktorů skládá z realizací neuronových sítí (*neural networks*) $\Phi_a(\cdot, \theta)$, $\theta \in \mathcal{P}$, s danou architekturou a a množinou parametrů \mathcal{P} . Obecně se pojmem neuronová síť označuje řada funkcí, které lze reprezentovat pomocí acyklického orientovaného grafu, jehož vrcholy odpovídají elementárním, skoro všude diferencovatelným funkcím parametrizovaných pomocí $\theta \in \mathcal{P}$ a hrany odpovídají složení těchto funkcí. Zadefinujme si nyní variantu plně propojené dopředné (*fully connected feedforward*) neuronové sítě.

Definice 3.3.1. Plně propojená dopředná neuronová síť je dána svojí architekturou $a = (N, \rho)$, kde $N \in \mathbb{N}^{L+1}$, $L \in \mathbb{N}$ a $\rho : \mathbb{R} \rightarrow \mathbb{R}$, přičemž ρ nazveme aktivační funkcí a L značí počet vrstev. Dále označíme N_0 , N_L a N_i , kde $i \in \{1, \dots, L-1\}$, jako počet neuronů na vstupní (*input*), výstupní (*output*) a v i -té skryté vrstvě. Jako počet parametrů této sítě definujeme

$$P(N) := \sum_{i=1}^L N_i N_{i-1} + N_i \quad (3.14)$$

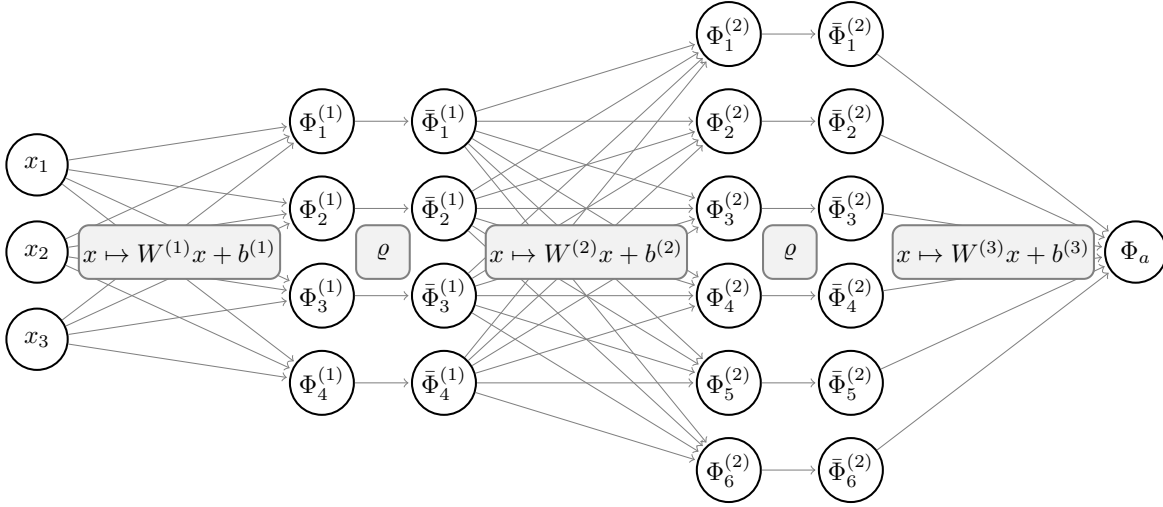
a definujeme realizační funkci $\Phi_a : \mathbb{R}^{N_0} \times \mathbb{R}^{P(N)} \rightarrow \mathbb{R}^{N_L}$, která pro každý prediktor $\mathbf{x} \in \mathbb{R}^{N_0}$ a parametry

$$\theta = (\theta^{(i)})_{i=1}^L = ((\mathbf{W}^{(i)}, \mathbf{b}^{(i)}))_{i=1}^L \in \prod_{i=1}^L (\mathbb{R}^{N_i \times N_{i-1}} \times \mathbb{R}^{N_i}) \cong \mathbb{R}^{P(N)} \quad (3.15)$$

splňuje, že $\Phi_a(\mathbf{x}, \theta) = \Phi^L(\mathbf{x}, \theta)$, kde

$$\begin{aligned} \Phi^1(\mathbf{x}, \theta) &= \mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)}, \\ \bar{\Phi}^i(\mathbf{x}, \theta) &= \rho(\Phi^i(\mathbf{x}, \theta)), \quad i \in \{1, \dots, L-1\}, \\ \Phi^{i+1}(\mathbf{x}, \theta) &= \mathbf{W}^{(i+1)} \bar{\Phi}^i(\mathbf{x}, \theta) + \mathbf{b}^{(i+1)}, \quad i \in \{1, \dots, L-1\}, \end{aligned} \quad (3.16)$$

přičemž aktivační funkce ρ je aplikovaná po složkách a prvky $\mathbf{W}^{(i)} \in \mathbb{R}^{N_i \times N_{i-1}}$ a $\mathbf{b}^{(i)} \in \mathbb{R}^{N_i}$ jsou matice vah a bias. Dále $\bar{\Phi}^i$ a Φ^i jsou aktivační a pre-aktivační funkce N_i neuronů v i -té vrstvě. Architekturu nazveme hlubokou (*deep*), pokud $L > 2$, a mělkou, pokud $L = 2$.



Obrázek 3.1: Schéma neuronů (šedá) a (pre-)aktivační funkce neuronů (bílá) hluboké plně propojené dopředné neuronové sítě $\Phi_a : \mathbb{R}^3 \times \mathbb{R}^{53} \rightarrow \mathbb{R}$ s architekturou $a = ((3, 4, 6, 1), \rho)$ a parametry $\theta = (\mathbf{W}^{(i)}, \mathbf{b}^{(i)})_{i=1}^3$ (převzato z [29]).

Příklad hluboké plně propojené neuronové sítě můžeme vidět na obrázku 3.1. Mezi používané aktivační funkce patří lineární

$$\rho(x) = x, \quad (3.17)$$

dále Heaviside funkce

$$\rho(x) = \theta(x) = \mathbb{I}_{(0, +\infty)}(x), \quad (3.18)$$

kde \mathbb{I} značí charakteristickou funkci. Díky dobrým výsledkům je stále častěji používána ReLU (*rectified linear unit*) dána jako

$$\rho(x) = \max\{0, x\}, \quad (3.19)$$

nebo její varianty, například parametrická leaky ReLU s parametrem α ve tvaru

$$\rho(x) = \begin{cases} \alpha x, & x < 0, \\ x, & x \geq 0. \end{cases} \quad (3.20)$$

Pro případ klasifikační úlohy se jako aktivační funkce v poslední vrstvě používá pro binární klasifikaci sigmoida ve tvaru

$$\rho(x) = \frac{1}{1 + \exp(-x)}, \quad x \in \mathbb{R} \quad (3.21)$$

a pro více tříd její zobecnění v podobě softmax funkce, která je po složkách pro $c \in \{1, \dots, C\}$, C je počet tříd, dána jako

$$\rho(x)_i = \frac{\exp(x_i)}{\sum_{c=1}^C \exp(x_c)}. \quad (3.22)$$

S ohledem na takto zavedenou neuronovou síť můžeme konkretizovat definici 3.0.1 pro naše účely v následující podobě.

Definice 3.3.2. Necht' $a = (N, \rho)$ je architektura neuronové sítě s dimenzí vstupní vrstvy $N_0 = d$, výstupní vrstvy $N_L = 1$ a měřitelnou aktivační funkcí ρ . Pro regresní úlohu je odpovídající množina hypotéz ve tvaru

$$\mathcal{F}_a = \{\Phi_a(\cdot, \theta) : \theta \in \mathbb{R}^{P(N)}\} \quad (3.23)$$

a pro klasifikační úlohu ve tvaru

$$\mathcal{F}_{a, \text{argmax}} = \{\text{argmax}(\Phi_a(\cdot, \theta)) : \theta \in \mathbb{R}^{P(N)}\}. \quad (3.24)$$

Ačkoliv existuje mnoho různých metod řešení minimalizačního problému, u neuronových sítí jsou nejběžnější metody založené na gradientu. Hlavním důvodem je skutečnost, že můžeme přesně a efektivně napočítávat bodové derivace $\nabla_{\theta} \Phi_a(\cdot, \theta)$ s ohledem na všechny proměnné, tedy prvky všech matic vah v síti. Specifickou formu takového algoritmu nazýváme backpropagation. Použitím těchto bodových derivací se obvykle snažíme minimalizovat ztrátovou funkci \mathcal{L} upravováním parametrů θ pomocí vhodného algoritmu. Na rozdíl od aktualizace parametrů po každém vzorku x se v praxi navíc pro větší stabilitu a trénovací rychlost používá metoda *mini-batche*, kdy bereme více vzorků naráz. Nejpoužívanějšími algoritmy jsou varianty stochastického gradientního sestupu (*stochastic gradient descent*, SGD), jež upravuje hodnoty parametrů v závislosti na rychlosti učení η_k pro k -tý krok iterace vztahem

$$\theta^{(k)} = \theta^{(k-1)} - \eta_k \nabla_{\theta} \mathcal{L}(\Phi_a(\cdot, \theta^{(k-1)})), \quad (3.25)$$

což můžeme upravit pro mini-batch jako

$$\theta^{(k)} = \theta^{(k-1)} - \frac{\eta_k}{m'} \sum_{z \in S'} \nabla_{\theta} \mathcal{L}(\Phi_a(\cdot, \theta^{(k-1)})), \quad (3.26)$$

kde S' je mini-batch o velikosti $|S'| = m'$ vybraný rovnoměrně z trénovacích dat s , přičemž iterace probíhají do dosažení požadované hodnoty ztrátové funkce \mathcal{L} , resp. dosažení maximálního počtu iterací K nebo jiné zastavovací podmínky. Výstupem hlubokého učícího algoritmu \mathcal{A} je potom

$$\phi_s = \mathcal{A}(s) = \Phi_a(\cdot, \bar{\theta}), \quad (3.27)$$

kde $\bar{\theta}$ může být zvolena jako realizace poslední iterace $\theta^{(K)}$ nebo jako konvexní kombinace všech iterací $(\theta^{(k)})_{k=1}^K$. Další možností je algoritmus AdaGrad, který generalizuje rychlost učení a navíc ji rozlišuje pro každý individuální parametr θ_i , kde $i \in \{1, \dots, P(N)\}$, tedy pokud budeme již uvádět předpis pro daný krok pouze verze bez mini-batchů, tak

$$\theta_i^{(k)} = \theta_i^{(k-1)} - \frac{\eta_k}{\sqrt{G_i^{(k)} + \epsilon}} \nabla_{\theta} \mathcal{L}(\Phi_a(\cdot, \theta_i^{(k-1)})), \quad (3.28)$$

kde $G \in \mathbb{R}^{P(N)}$ je rovno součtu druhých mocnin derivací ztrátové funkce \mathcal{L} vzhledem k θ_i v iteraci k , tedy

$$G_i^{(k)} = G_i^{(k-1)} + \left(\frac{\partial \mathcal{L}(\Phi_a(\cdot, \theta_i^{(k-1)}))}{\partial \theta_i^{(k-1)}} \right)^2, \quad (3.29)$$

a ϵ je regularizační parametr zamezující dělení nulou. Tento optimalizační algoritmus však může mít problém s tzv. mizející rychlostí učení, neboť bude-li síť příliš hluboká, pak bude

výsledný gradient téměř nulový, neboť v řetězovém pravidle pro derivace budeme mezi sebou násobit mnoho téměř nulových čísel. Ve snaze zabránit tomuto problému byl navržen algoritmus RMSProp, který navíc přidává do AdaGradu koeficient úpadku $\gamma \in (0, 1)$ ve smyslu

$$G_i^{(k)} = \gamma G_i^{(k-1)} + (1 - \gamma) \left(\frac{\partial \mathcal{L}(\Phi_a(\cdot, \theta_i^{(k-1)}))}{\partial \theta_i^{(k-1)}} \right)^2 \quad (3.30)$$

a samotné iterační pravidlo je poté stejné jako v rovnici (3.28). RMSProp byl poté dále zobecněn do algoritmu Adam, který kromě koeficientu γ přidává další koeficient úpadku $\delta \in (0, 1)$ a pro každou iteraci vektor pomocných proměnných $M^{(k)}$ ve smyslu

$$M_i^{(k)} = \delta M_i^{(k-1)} + (1 - \delta) \left(\frac{\partial \mathcal{L}(\Phi_a(\cdot, \theta_i^{(k-1)}))}{\partial \theta_i^{(k-1)}} \right)^2. \quad (3.31)$$

Samotná aktualizace pak v Adam algoritmu probíhá pomocí

$$\theta_i^{(k)} = \theta_i^{(k-1)} - \frac{\eta_k}{\sqrt{G_i^{(k)} + \epsilon}} M_i^{(k-1)}. \quad (3.32)$$

Ačkoliv jsme se celou dobu věnovali plně propojeným neuronovým sítím, v praxi budeme používat i sítě, které nemají propojené všechny neurony napříč vrstvami. Základním způsobem, jak toho dosáhnout, je tzv. *dropout*, kdy ve vrstvách 1 až $L - 1$ s danou pravděpodobností p vynecháme jednotlivé neurony.

3.4 Konvoluční neuronové sítě

Ačkoliv se jedná o speciální případ neuronových sítí, vzhledem k jejich významnému postavení zejména ve zpracování obrazu je rozebereme podrobněji v této sekci. Tento přístup je vhodný zejména pro data s vysokou dimenzí, kde navíc informace ve vstupních datech jsou prostorově propojené. Typicky v obrázcích očekáváme, že sousední pixely budou sdílet určitou informaci a jejich prostorová blízkost by proto měla být v architektuře zohledněna. Je proto rozumné mít neuronovou síť, která sbírá informace z prostorově blízkých vstupních dat. Tato metoda pro klasifikaci i regresi je vhodným nástrojem vzhledem k invarianci objektů v obrázku vůči translaci, avšak postrádají invarianci vůči rotaci. Tato nevýhoda se dá však omezit úpravou dat, kdy kopie trénovacích obrázků různě natočíme.

Konvoluční neuronová síť (*convolutional neural network*, CNN) odpovídá několika konvolučním blokům, které jsou speciálním typem vrstvy. Tato architektura vrstvy vychází z matematické operace konvoluce, kterou pro dvě jednorozměrné funkce $f(x)$ a $g(x)$ definujeme vztahem

$$(f * g) = \int_{-\infty}^{\infty} f(\alpha) g(x - \alpha) d\alpha. \quad (3.33)$$

V konkrétních případech se používá diskrétní varianta konvoluce, a to nejčastěji přes množinu $G \subset \mathbb{Z}$ v jedné dimenzi nebo přes množinu $G \subset \mathbb{Z}^2$ ve dvou dimenzích, ale je možné ji zobecnit

i do vyšších dimenzích. Pro náš případ dvourozměrné konvoluce pro dvě matice A a B platí po složkách

$$(A * B)_{m,n} = \sum_{k,l \in G} A_{k,l} \cdot B_{m-k,n-l}, \quad (3.34)$$

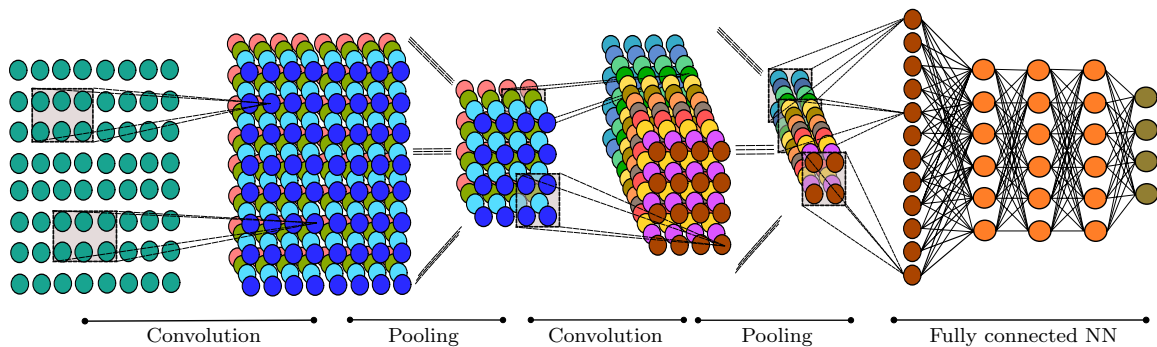
kde G je v praxi konečná matice o rozměrech konvoluční masky W , též nazývaného jako konvoluční jádro, která je maticí s rozměry pro i -tou vrstvu o šířce W_i a výšce H_i . Masky bývají většinou malé matice s rozměry řádově (3×3) nebo (5×5) vzhledem k výpočetní náročnosti a také kvůli tomu, že se zajímáme pouze o okolí daného bodu. V případě, že je navíc vstupní obrázek vícekanálový, resp. je vstupem tensor, tak můžeme provádět jednotlivé konvoluce pro každý kanál zvlášť, nebo je můžeme sloučit do jednoho či více kanálů na výstupu. To znamená, že pro počet vstupních kanálů $D_i \in \mathbb{N}$ na vstupu i -té vrstvy, $d_{out} \in \{1, \dots, D_{i+1}\}$, pro konvoluční masku $w_i \in \mathbb{R}^{H_i \times W_i}$ a vstupní obrázek $X^{(i)}$, dostáváme výstup $X^{(i+1)}$ konvoluční vrstvy jako

$$(X^{(i+1)})_{m,n,d_{out}} = \sum_{k=1}^{H_i} \sum_{l=1}^{W_i} \sum_{d_{in}=1}^{D_i} w_{k,l,d_{in},d_{out}}^{(i)} \cdot X_{m-k,n-l,d_{in}}^{(i)}. \quad (3.35)$$

Vzhledem k potřebnému snižování objemu informace při provádění konvolučních vrstev je pro CNN typické použití tzv. poolingových vrstev. Poolingový operátor p lze obecně definovat jako zobrazení $p : \mathbb{R}^G \rightarrow \mathbb{R}^{\tilde{G}}$, přičemž v praxi se nejčastěji používá tzv. podvzorkování (*subsampling*) např. o rozměrech 2×2 , kdy v podstatě krokujeme ob jeden prvek, tedy provádíme zobrazení

$$(X^{(i)})_{m,n} \rightarrow (X^{(i+1)})_{2m-1,2n-1}. \quad (3.36)$$

Alternativně můžeme použít average pooling nebo max pooling, kdy vždy okolí dané velikosti nahrazujeme průměrnou nebo maximální hodnotou.



Obrázek 3.2: Schéma konvoluční neuronové sítě s dvou-dimenzionálními konvolučními bloky a subsamplingem 2×2 v poolingové vrstvě (převzato z [29]).

Konvoluční neuronová síť, jak již bylo zmíněno, se skládá z několika konvolučních vrstev naskládaných za sebe. Po vhodném množství těchto vrstev můžeme poslední výstup zploštit (tzv. *flattening*), tedy převést na vektor, který dále můžeme použít jako vstup do klasické neuronové sítě. Příklad takové sítě můžeme vidět na obrázku 3.2. Vzhledem k tomu, že na konvoluční jádro w klademe speciální vlastnosti a konvoluce je lineární operací, můžeme konvoluční architekturu označit za speciální dopřednou síť. Navíc můžeme definovat tzv. *padding* pro projíždění

konvolučním jádrem, kdy můžeme použít buď tzv. *valid padding* a vstupní obrázek v plné velikosti projíždíme jádrem, čímž vznikne menší obrázek na výstupu. Nebo pro *half padding* původní obrázek na vstupu zvětšíme přidáním dalších pixelů okolo původního tak, aby při projíždění konvolučním jádrem vznikl stejně velký obrázek jako ten původní na vstupu. Zde může být přidávání pixelů provedeno buď jednoduše nulami, nebo replikováním hraniční hodnoty, anebo zrcadlením vůči hraničními pixelu.

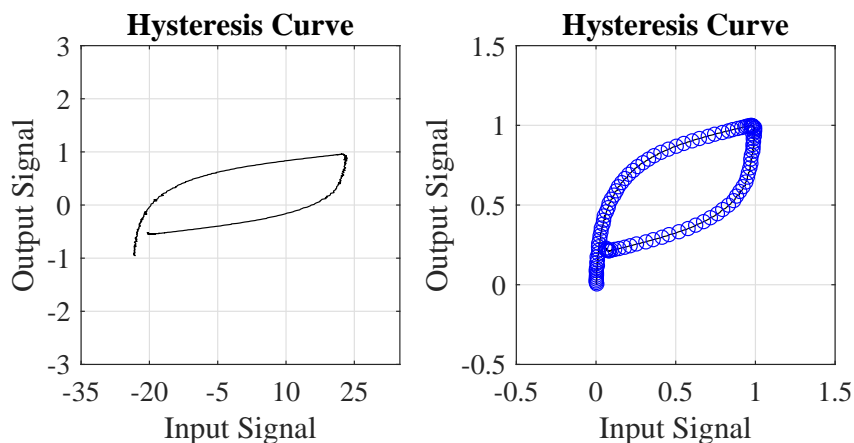
Kapitola 4

Implementace metod strojového učení

Cílem použití strojového učení na defektoskopická data je klasifikace, potažmo regrese vhodného pravděpodobnostního rozdělení, resp. jejich směsi, ze znalosti hysterezní smyčky. Jak již bylo zmíněno, optimalizační úloha hledání vhodného PM prostoru je časově náročná a pokud bychom mohli předem určit o jakou distribuci se jedná, případně směs distribucí a jejich zastoupení, tak se úloha redukuje na nalezení parametrů těchto pravděpodobnostních rozdělení, oproti dosavadnímu přístupu, kdy tento proces musí být proveden pro každé uvažované rozdělení a jejich různé kombinace ve směsích.

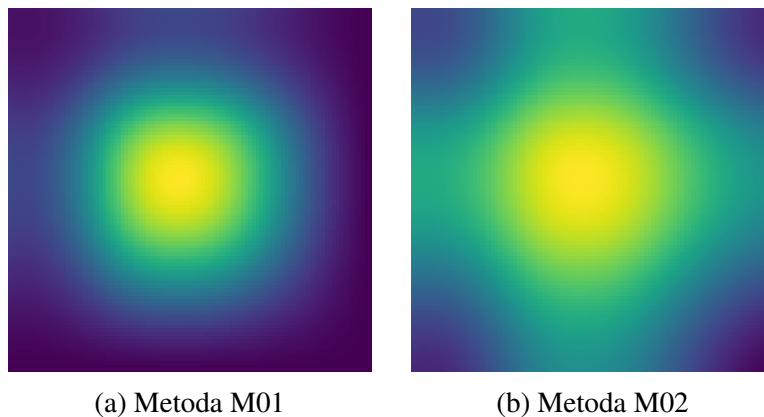
4.1 Předzpracování dat

Předpokládejme, že hysterezní křivka byla naměřena tak, že mezi jednotlivými body byl ekvidistantní časový odstup. Pak pro predikci budeme používat jednu hysterezní smyčku, kterou přeškálujeme na jednotkovou velikost a zredukujeme na 100 bodů s rovnoměrnými časovými rozestupy, jak je znázorněno na obrázku 4.1. Hodnoty výstupního zatížení v těchto bodech, samozřejmě standardizované, budou sloužit jako naše prediktory. Pro stromové metody a pro běžné neuronové sítě bude použit rovnou tento vektor x . Pro konvoluční neuronové sítě však



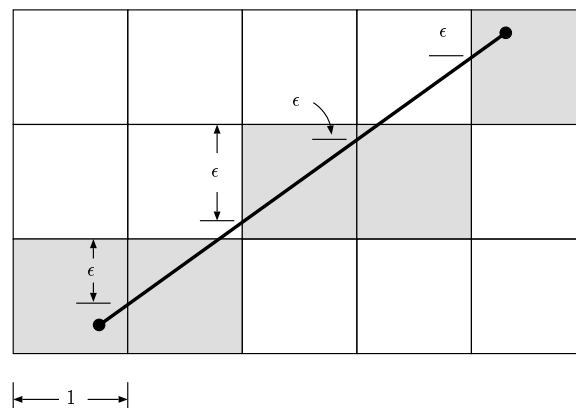
Obrázek 4.1: Naměřená hysterezní smyčka (vlevo) a její unifikace (vpravo)

musíme navíc tento vektor přetransformovat do obrázku, resp. matice o rozměrech 100×100 . To provedeme několika způsoby. První metoda M01 je založena na součinu vektoru prediktorů se sebou samým, tedy $X = \mathbf{x}^T \mathbf{x}$. Tudíž výsledný obrázek je tvořen maticí s prvky $(X)_{i,j} = x_i \cdot x_j$, kde $i, j \in \{1, \dots, 100\}$. Další metoda M02 je potom obdobně založena na průměrování těchto hodnot, tedy $(X)_{i,j} = \frac{1}{2}(x_i + x_j)$, kde $i, j \in \{1, \dots, 100\}$. Vzniklé matice z obou metod můžeme vidět na obr. 4.2, přičemž vzhledem k předešlé normalizaci jsou hodnoty opět mezi 0 a 1. Tyto obrázky, stejně jako všechny následující, jsou tedy jednobarevné a vzhledem k tomu, že následující metody nebudou zachovávat normalizaci, budou následně přeškálovány, aby jejich obor hodnot byl také od 0 do 1.



Obrázek 4.2: Transformace vektoru prediktorů do obrázku

Pro následující metody budeme potřebovat nástroj, který přímku, potažmo křivku jednoznačně aproximuje do obrázku s n -dimenzionálním rastrem. V našem případě použijeme tzv. Bresenhamův algoritmus, který slouží k vykreslení úsečky mezi dvěma body do rastrové mřížky, ale lze ho snadno zobecnit i pro obecné křivky. Pseudokód této metody vidíme v algoritmu 4.1. Tento princip vidíme i na obr. 4.3, kde je znázorněno, které pixely obrázku se vykreslí, resp. pro které prvky v matici se zvýší hodnota symbolizující průchozí úsečku. Můžeme vidět, že po



Obrázek 4.3: Schéma vykreslování rastrové přímky pro Bresenhamův algoritmus (převzato z [31]).

diskretizaci spojitého prostoru na požadovaný rastr se vykreslují zleva doprava ty pixely, kterými prochází přímka, přičemž chceme za účelem zachování co nejužšího formátu úsečky, aby se nad sebou nevyskytovali dva pixely. Místo aby algoritmus rozhodoval, který vektor vykreslit, pomocí souřadnice y (která se zvětší o $m = \Delta y / \Delta x$ pokaždé, když x -ová souřadnice vzroste o 1), udržuje algoritmus chybovou hranici ϵ , která představuje zápornou hodnotu vzdálenosti bodu, ve kterém úsečka prochází pixel, od horního okraje pixelu, ve kterém se zrovna nacházíme. Hodnota ϵ je nastavena nejdříve na počáteční hodnotu (viz algoritmus 4.1) a zvyšuje se o m pokaždé, když se souřadnice x zvýší o jedna. Pokud nastane, že ϵ je větší než nula, posuneme se algoritmu při vykreslování úsečky o jeden krok nahoru. V tom okamžiku zároveň aktualizujeme chybu ϵ odečtením hodnoty 1, aby představovala vzdálenost od pixelu s novou hodnotou souřadnice y .

Algoritmus 4.1: Pseudokód pro Bresenhamův algoritmus

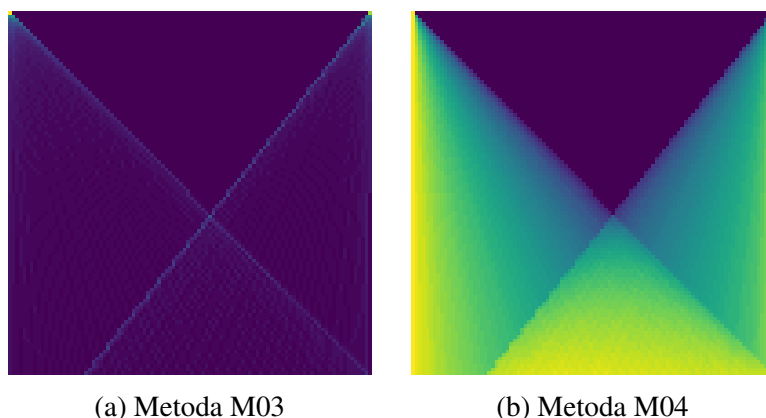
```

1 Hraniční body úsečky  $(x_1, y_1)$  a  $(x_2, y_2)$ , výsledný obrázek  $X = \mathbf{0}$ 
2 Necht'  $\Delta x = x_2 - x_1$ ,  $\Delta y = y_2 - y_1$ ,  $m = \frac{\Delta y}{\Delta x}$ ,  $i_1 = \lfloor x_1 \rfloor$ ,  $j = \lfloor y_1 \rfloor$ ,  $i_2 = \lfloor x_2 \rfloor$ ,
    $\epsilon = -\left(1 - (y_1 - j) - \frac{\Delta y(1 - (x_1 - i_1))}{\Delta x}\right)$ 
3 for  $i = i_1$  to  $i_2$  do
4    $X_{i,j} = X_{i,j} + 1$ 
5   if  $\epsilon \geq 0$  then
6      $j = j + 1$ 
7      $\epsilon = \epsilon - 1$ 
8    $i = i + 1$ 
9    $\epsilon = \epsilon + m$ 

```

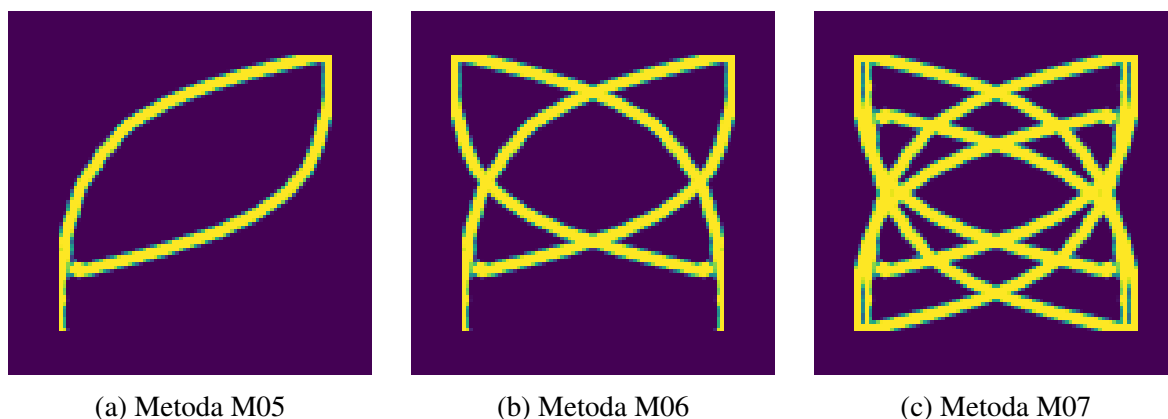
Další metody vytváření obrázku z vektoru prediktorů budou založeny na vynesení hodnot výstupního signálu na spodní hranu čtverce symbolizující náš výsledný obrázek a jejich propojení s horními vrcholy tohoto čtverce. Konkrétně pro metodu M03 rozdělíme vektor prediktorů na dvě části podle dosažené maximální hodnoty, které tím pádem budou reprezentovat spodní a horní větve hysterezní smyčky. Poté vyneseme hodnoty výstupního signálu ze spodní části hystereze na spodní hranu našeho čtvercového obrázku o velikosti jedna ve směru zleva doprava. Tyto body spojíme přímkou s levým horním rohem pomocí Bresenhamova algoritmu 4.1. Obdobně to uděláme s horní částí hysterezní křivky, s tím rozdílem, že hodnoty budeme vynášet zprava doleva a následně je spojíme s pravým horním rohem. Výsledný obrázek můžeme vidět na obr. 4.4a. Následující metoda M04 je s metodou M03 téměř totožná, s tím rozdílem, že kromě obarvení bodů, kterými prochází přímka, obarvíme také všechny body pod ní. Výsledek můžeme vidět na obr. 4.4b, přičemž tentokrát místo samotných úseček vidíme schodovitě se zvyšující barevné odstíny. Vzhledem k tomu že se několiknásobně překrývají úsečky, potažmo celé části plochy, jsou vzniklé obrázky přeškálovány, aby rozsah hodnot byl opět mezi 0 a 1.

Další metody jsou založeny na vytváření obrázku odpovídající grafu hysterezní smyčky ze zadaného vektoru prediktorů, přičemž pro zachování konzistence s předchozími metodami je obrázek tvořen tak, aby opět jeho rozměry byly 100×100 . A také pro reprezentaci přímky mezi jednotlivými body byl použit Bresenhamův algoritmus. Metoda M05 tedy spočívá v takto vykreslené hysterezní smyčce do obrázku. Dále metoda M06 vznikla stejně jako ta předchozí,



Obrázek 4.4: Transformace vektoru prediktorů do obrázku

kde navíc došlo k sfúzování se zrcadlovým překlopením sebe samého. Fúze byla provedena tak, že jsme vzali maximální hodnotu pro daný pixel, resp. prvek matice, z obou obrázků. Stejným způsobem fúzování budou vznikat obrázky i v dalších metodách. Dále tedy metoda M07 vznikla z M06, kde navíc došlo k vertikálnímu překlopení. Výsledky předchozích tří metod můžeme vidět na obrázku 4.5.

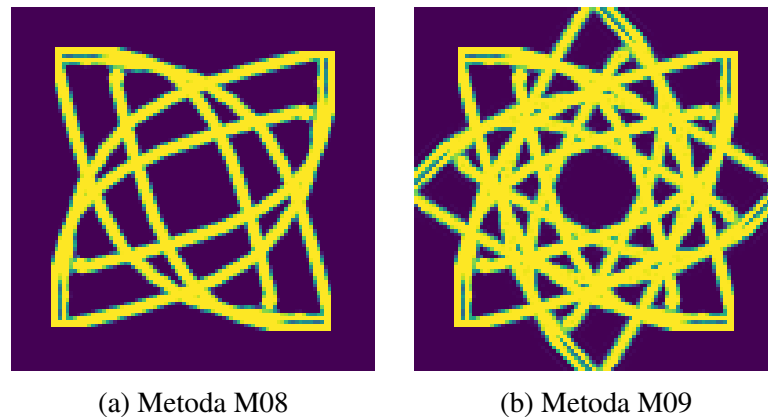


Obrázek 4.5: Transformace vektoru prediktorů do obrázku

Poslední metody namísto překlápění, jak tomu bylo v metodách M06 a M07, vznikly otáčením daného obrázku okolo jeho středu. Pro metodu M08 bylo použito otáčení o 90° , a tedy finální obrázek vznikl sfúzováním 4 obrázků. Na závěr při metodě M09 došlo k rotaci o 45° a na obrázku je tedy daná hysterezní smyčka osmkrát v různých polohách. Obě tyto metody vidíme na obrázku 4.6.

4.2 Datasetsy a proces trénování

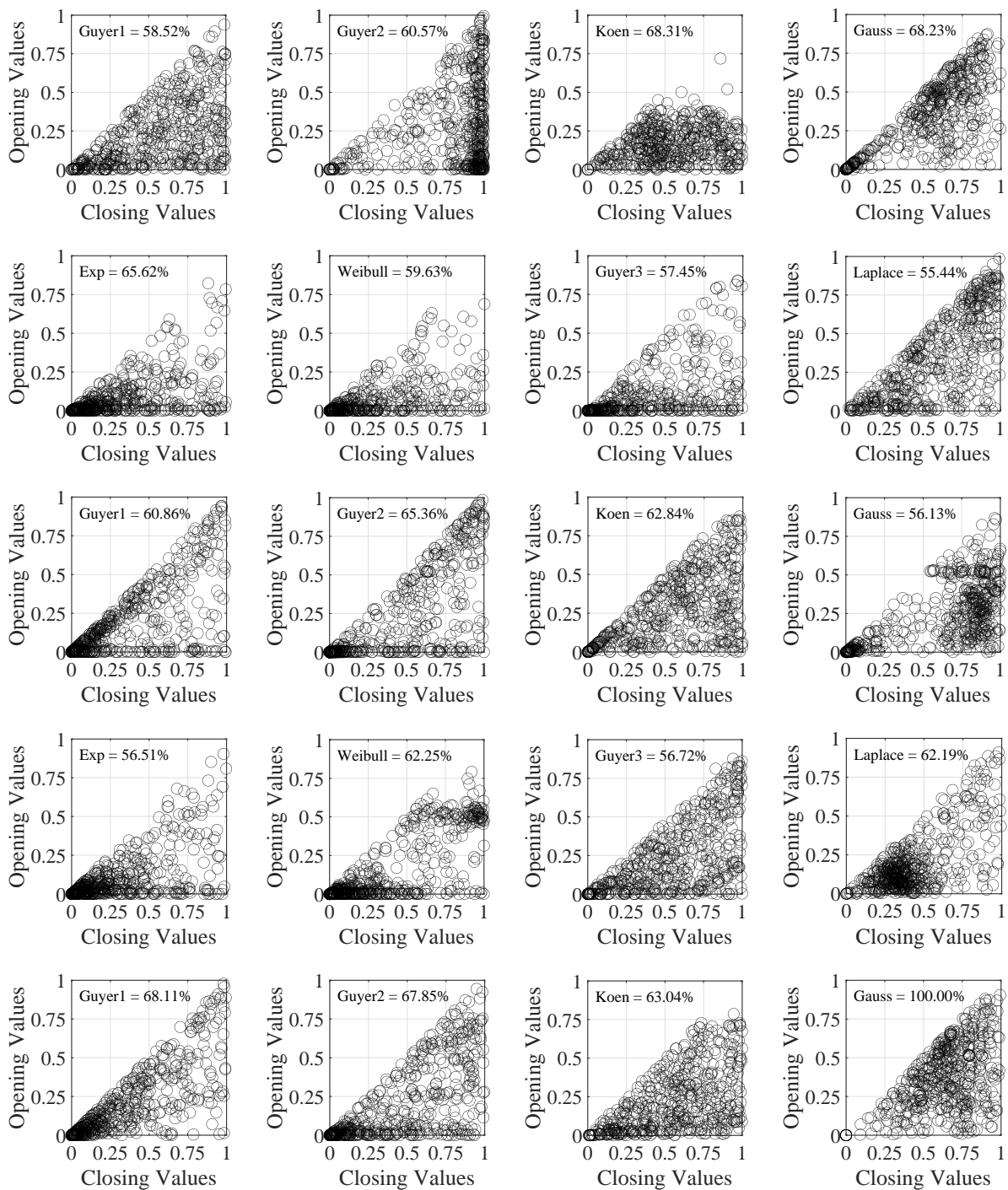
Pro účely trénování našich metod byl vytvořen dataset ze směsí distribučních funkcí představených v sekci 2.1, konkrétně Guyer1, Guyer2, Guyer3, Koen, dvourozměrné normální rozdělení, exponenciální rozdělení, Weibullovo rozdělení a Laplaceovo rozdělení. Tento dataset



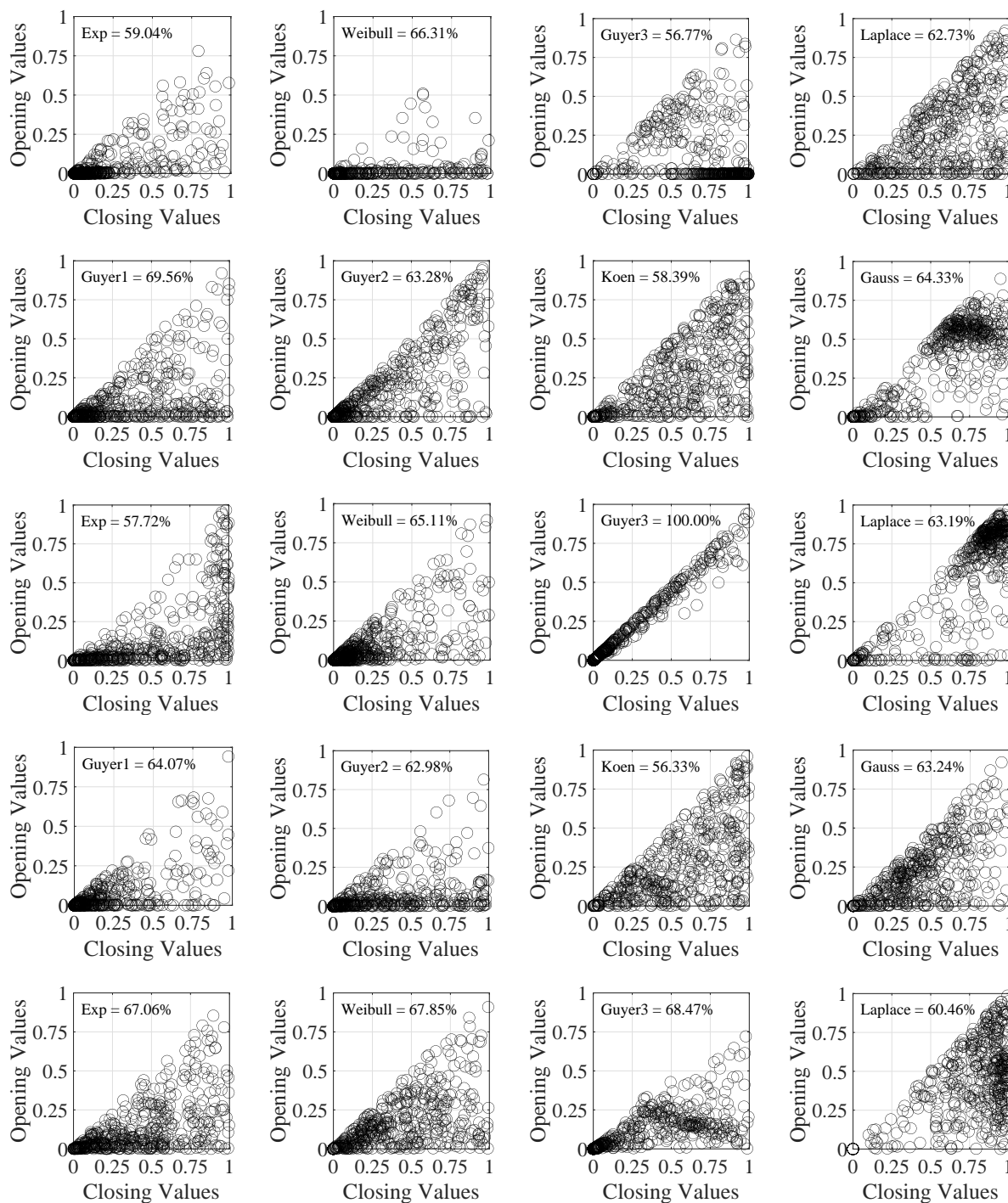
Obrázek 4.6: Transformace vektoru prediktorů do obrázku

byl vygenerován tak, že ve směsi je vždy jedna složka v dominantním zastoupení větším než 50% a ostatní složky tvoří šum, přičemž je zajištěno, aby byly postupně rovnoměrně použity všechny jednotlivé distribuce jako dominantní složky. Vygenerovány byly směsi jedné až osmi komponent. Konkrétně při generování vytvoříme náhodně vektory zastoupení ve směsi délek 1, 2 až 8. Součet prvků každého vektoru je roven jedné a vždy má jeden dominantní prvek větší než 0.5. Kvůli vyváženosti dat použijeme na pozici dominantní složky postupně každé z osmi distribucí, přičemž pro více než dvou- a více-komponentní směsi budou ostatní složky náhodně voleny opět z námi používaných pravděpodobnostních rozdělení. Jeden takový blok datasetu bude potom obsahovat $8 \times 8 = 64$ vzorků. Pro naše účely jsme vytvořili data, které obsahují 40 takovýchto bloků, přičemž pro každý z nich je náhodně vygenerován 8 sad parametrů příslušných pravděpodobnostních rozdělení. Celkem tedy vzniklo 20480 pravděpodobnostních směsí, pomocí kterých byly vytvořeny PM prostory, jejichž ukázkou vidíme na obr. 4.7. Na tyto PM prostory bylo aplikováno zafixované vstupní zatížení, které odpovídá zatížení z obr. 4.1 a které pochází z reálných zpracovávaných dat v práci [7] z testování stejného typu zeměřských tlumičů, které odpovídá vstupním zatížením, která používáme v této práci. Tím vznikly odpovídající hysterezní smyčky, které sloužili jako finální dataset, který byl dále předzpracován podle sekce 4.1. Tento dataset byl dále pro účely trénování vždy stejným způsobem rozdělen na trénovací, validační a testovací data v poměr 2 : 1 : 1.

Způsoby použití metod z kapitoly 3 jsou dva, jeden klasifikační a jeden regresní. Pro klasifikační přístup bude predikovaná proměnná distribuční funkce, která má v dané směsi dominantní postavení, tudíž budeme klasifikovat daný vstup do jedné z osmi tříd. Takový přístup bude mít za důsledek, že nám daná metoda vyhodnotí podle jakého jednoho pravděpodobnostního rozdělení máme danou hysterezní smyčku popsat, resp. jaké rozdělení hraje při jejím popisu dominantní roli. Pro směsi nám tento přístup však příliš nepomůže. Samozřejmě by bylo možné vytvořit větší dataset, kde by bylo více dominantních složek v podobném zastoupení ve směsi, avšak takový dataset pro všechny možnosti kombinací složek ve směsi by byl příliš velký a obtížný pro jeho vyvážené generování. Proto kromě klasifikačního přístupu použijeme i regresní, konkrétně multi-target regresi, ve které budeme predikovat procentuální zastoupení dané složky. Tedy pro daný vstup nám bude vráceno osm hodnot z intervalu $[0, 1]$. Aby toho bylo dosaženo, bude tato podmínka, kde je to možné, zahrnutá v architektuře (NN, CNN), případně použita



Obrázek 4.7: Výběr PM prostorů z testovacích dat a jejich dominantní komponenta



Obrázek 4.7: Výběr PM prostorů z testovacích dat a jejich dominantní komponenta (pokračování)

v post-zpracování (RF, XGBoost, . . .), a to vždy tak, že se záporné hodnoty přetvoří na nuly a součet koeficientů zastoupení komponent ve směsi, tedy naše predikované proměnné, bude normalizován na hodnotu jedna.

4.2.1 Trénování ensemble metod

Veškeré implementace metod strojového učení byly provedeny v programovacím jazyku Python. Začneme nejdříve s použitím stromových ensemble metod (viz sekce 3.2) pro klasifikaci. Tyto metody jsme nejprve optimalizovali pomocí procedury GridSearchCV s 3-fold křížovou validací vůči metrice accuracy na testovacích datech, přičemž tato funkce opět pochází z knihovny scikit-learn. Možné hodnoty jednotlivých hyperparametrů nalezneme v tabulkách 4.1 až 4.3. Konkrétně pro random forest (RF) jsme optimalizovali pro různé hodnoty pro počet stromů, maximální hloubku stromů, kritéria pro vyhodnocení kvality dělení, minimální počet vzorků pro dělení uzlu a minimum vzorků, které musí být v listu. Pro AdaBoost jsme optimalizovali hyperparametry počet stromů a rychlost učení (*learning rate*) a pro XGBoost potom počet stromů, maximální hloubku stromů, learning rate, minimální snížení ztrátové funkce pro další dělení listového uzlu (*min split loss*) a hodnoty L_1 a L_2 regularizace. Nejlepší hodnoty hyperparametrů pro RF jsou počet stromů = 100, maximální hloubka stromů = 50, kritérium = Gini, minimum vzorků v uzlu pro dělení = 2, minimum vzorků v listu = 2. Pro AdaBoost jsou počet stromů = 100, learning rate = 0.15. A pro XGBoost jsou počet stromů = 100, maximální hloubka stromů = 10, learning rate = 0.3, min split loss = 0, L_2 regularizace = 1.1 a L_1 regularizace = 0.

Hyperparametry	Možnosti
Počet stromů	{10, 50, 100 , 200, 500}
Maximální hloubka stromů	{5, 10, 20, 30, 50 }
Kritérium	{ Gini , Entropy}
Minimum vzorků v uzlu pro dělení	{ 2 , 5, 10}
Minimum vzorků v listu	{1, 2 , 4}

Tabulka 4.1: Prohledávané hyperparametry pro random forest klasifikaci (nejlepší parametry tučně)

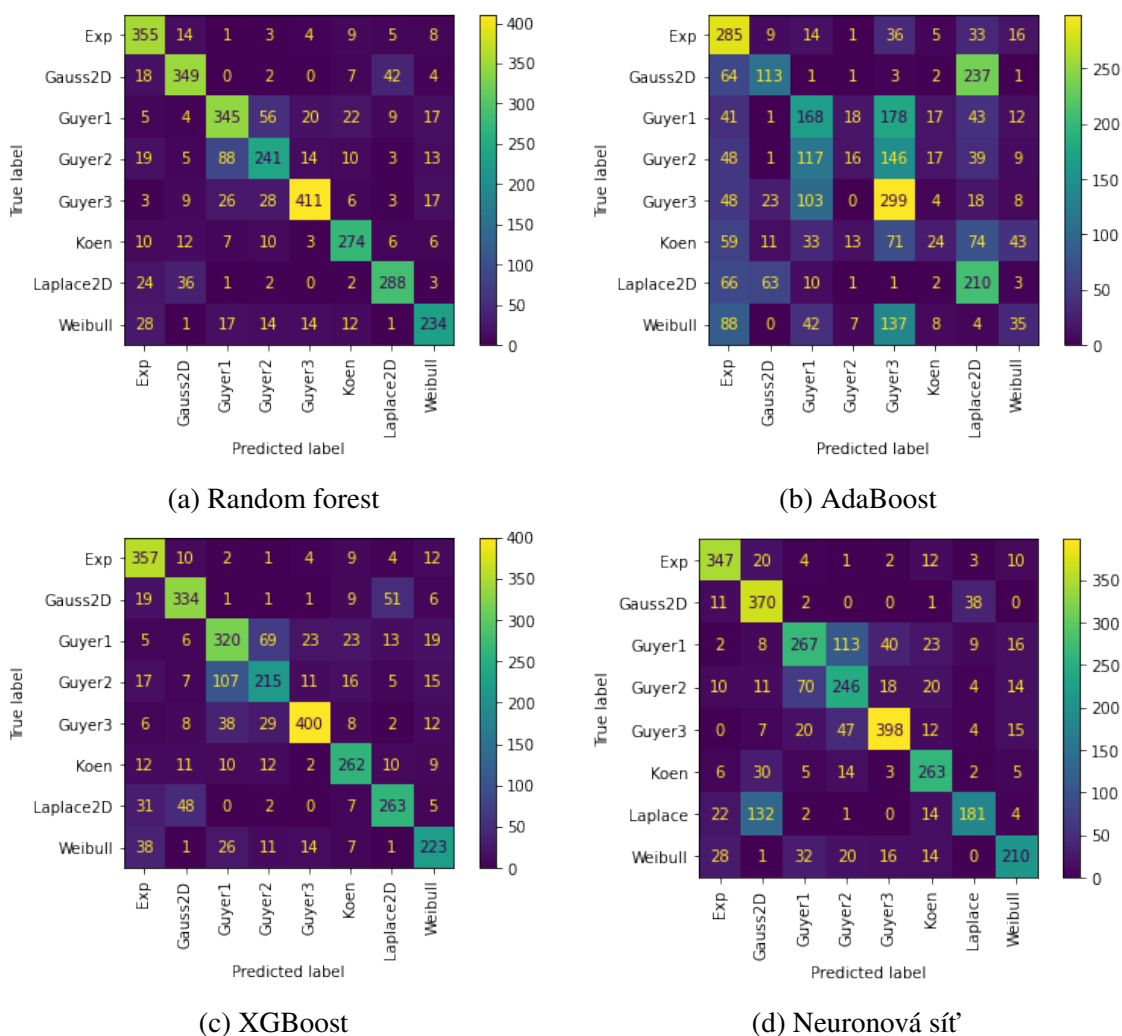
Hyper-parametry	Možnosti
Počet stromů	{10, 50, 100 , 200, 500}
Learning rate	{0.05, 0.1, 0.15 , 0.2, 0.3}

Tabulka 4.2: Prohledávané hyperparametry pro AdaBoost klasifikaci (nejlepší parametry tučně)

Kvalitu jednotlivých klasifikátorů vidíme jednak v matici záměn (*confusion matrix*) na obrázku 4.8 a dále pak v celkovém srovnání na testovacích datech v tabulce 4.8, kde vždy předpona w- značí weighted-precision, weighted-recall atd. Můžeme vidět, že kromě AdaBoostu mají RF a XGBoost dobré klasifikační schopnosti, kde největší problémy tvoří rozlišování mezi dvojicemi Guyer1 vs Guyer2, Gaussovské rozdělení vs Laplaceovo, a dále exponenciální vs Weibullovo, což lze očekávat vzhledem k relativní podobnosti těchto dvojic rozdělení.

Hyper-parametry	Možnosti
Počet stromů	{10, 50, 100 , 200, 500}
Maximální hloubka stromů	{5, 10 , 20, 30, 50}
Learning rate	{0.1, 0.2, 0.3 , 0.4}
Min split loss	{ 0 , 0.5, 1, 1.5}
L_2 regularizace	{1.0, 1.1 , 1.2}
L_1 regularizace	{ 0 , 0.1, 0.2}

Tabulka 4.3: Prohledávané hyperparametry pro XGBoost klasifikaci (nejlepší parametry tučně)



Obrázek 4.8: Matice záměn pro RF, AdaBoost, XGBoost a NN

4.2.2 Trénování NN a CNN metod

Klasifikace pomocí neuronových sítí byla implementována v softwarové knihovně Keras. Jednotlivé architektury těchto sítí byly opět optimalizovány, tentokrát vůči hodnotě ztrátové funkce na validačních datech, kterou konkrétně byla křížová entropie \mathcal{L}_{CE} (viz sekce 3.1). Tré-

nování bylo vždy nastaveno na 200 iterací, kde navíc byla zastavovací podmínka, pokud se posledních 15 iterací nezlepšila validační ztráta. Vzhledem k větší výpočetní náročnosti a také větší variabilitě architektury neuronových sítí bylo místo prohledávání všech možností (*grid search*) použito náhodné prohledávání (*random search*), konkrétně metoda `RandomSearch` pocházející z knihovny `KerasTuner`. Možné hodnoty hyperparametrů v architektuře nalezneme pro NN sítě v tabulce 4.4 a pro CNN v tabulce 4.5, kde vidíme vypsání jednotlivé možnosti, přičemž parametry týkající se vrstev (počet neuronů, počet filtru a velikost konvoluční masky) byly optimalizovány pro každou vrstvu zvlášť a hodnota learning rate je navíc volen podél celého zvoleného intervalu. Pro prohledávání nejlepší architektury bylo pro NN natrénováno 256 sítí a pro CNN 64 sítí. Ve všech architekturách je používána ReLU aktivační funkce a u konvolučních vrstev byl vždy volen half padding.

Hyper-parametry	Možnosti
Počet dense vrstev	{2, 3, 4, ..., 8}
Počet neuronů v každé dense vrstvě	{64, 80, 96, ..., 512}
Použití dropout po každé sudé dense vrstvě	{yes, no}
Dropout rate	{0.05, 0.1, 0.2, 0.25, 0.5}
Learning rate	[1 e-6, 1 e-2]

Tabulka 4.4: Prohledávané hyper-parametry pro NN

Hyper-parametry	Možnosti
Počet 2D konvolučních vrstev	{2, 3, 4, ..., 8}
Počet filtrů v každé konvoluční vrstvě	{2, 4, 8, 16, 32, 48, 64}
Velikost konvoluční masky v každé vrstvě	{{(3, 3), (4, 4), (5, 5), (6, 6)}
Použití dropout po každé sudé konvoluční vrstvě	{yes, no}
Dropout rate	{0.05, 0.1, 0.2, 0.25, 0.5}
Počet neuronů v dense vrstvě	{64, 80, 96, ..., 512}
Learning rate	[1 e-6, 1 e-2]

Tabulka 4.5: Prohledávané hyper-parametry pro CNN

Nejlepší nalezenou architekturu pro NN vidíme v tabulce 4.6, kde vždy za každou, kromě poslední, plně propojenou (dense) vrstvou je vrstva s aktivační ReLU funkcí a za poslední dense vrstvou je softmax aktivační funkce. Použitý learning rate byl 0.0005256. Výsledky na testovacích datech nalezneme opět v tabulce 4.8 a také na matici záměn 4.8d. Pro CNN byla architektura sítě optimalizována pro metodu transformace vstupních dat do obrázku pomocí metody M06, neboť při prvotním trénování dosahovala nejlepších výsledků. Tuto architekturu, kterou můžeme vidět v tabulce 4.7, jsme použili také při použití ostatních metod transformace prediktorů. V této architektuře je za každou konvoluční vrstvou a za vrstvou `dense_0` použita vrstva s ReLU aktivační funkcí a za vrstvou `dense_1` by použita softmax aktivační funkce. Použitý pooling byl ve všech vrstvách 2×2 a ve vrstvě `dropout_1` byla její hodnota 0.1, dále v `dropout_3` byla 0.5, v `dropout_5` byla 0.1 a ve vrstvě `dropout_6` byla 0.2. Použitý learning rate byl 0.0007184. Tuto architekturu jsme z časových důvodů použili i pro ostatní metody vytváření

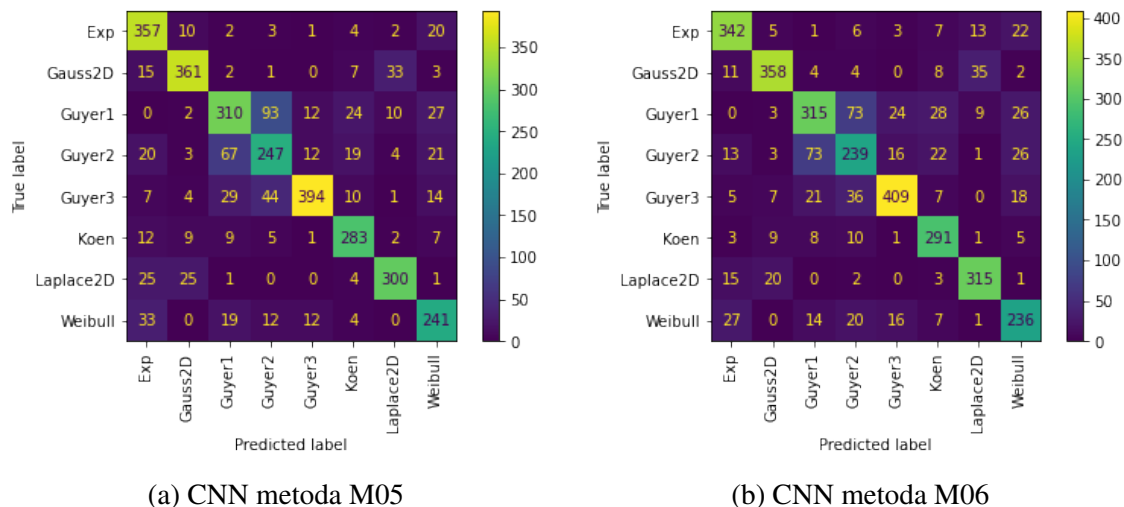
obrázků z prediktorů. Výsledky z testovacích dat na těchto konvolučních neuronových sítích jsou také zahrnuty v tabulce 4.8. Matice záměn, tentokrát pouze pro nejlepší dva výsledky konvolučních sítí, můžeme vidět na obrázku 4.9.

Layer (type)	Output Shape	Param #
dense_0 (Dense)	(None, 320)	32320
dense_1 (Dense)	(None, 224)	71904
dense_3 (Dense)	(None, 128)	28800
dense_4 (Dense)	(None, 352)	45408
dense_5 (Dense)	(None, 112)	39536
dense_6 (Dense)	(None, 224)	25312
dropout (Dropout)	(None, 224)	0
dense_7 (Dense)	(None, 8)	1800
Total params: 245,080		
Trainable params: 245,080		
Non-trainable params: 0		

Tabulka 4.6: Nejlepší architektura NN sítě pro klasifikaci

Layer (type)	Output Shape	Param #
conv2d_0 (Conv2D)	(None, 100, 100, 16)	416
conv2d_1 (Conv2D)	(None, 96, 96, 32)	12832
max_pooling2d_1 (MaxPooling2D)	(None, 48, 48, 32)	0
dropout_1 (Dropout)	(None, 48, 48, 32)	0
conv2d_2 (Conv2D)	(None, 44, 44, 4)	3204
conv2d_3 (Conv2D)	(None, 39, 39, 64)	9280
max_pooling2d_3 (MaxPooling2D)	(None, 19, 19, 64)	0
dropout_3 (Dropout)	(None, 19, 19, 64)	0
conv2d_4 (Conv2D)	(None, 15, 15, 16)	25616
conv2d_5 (Conv2D)	(None, 13, 13, 64)	9280
max_pooling2d_5 (MaxPooling2D)	(None, 6, 6, 64)	0
dropout_5 (Dropout)	(None, 6, 6, 64)	0
flatten (Flatten)	(None, 2304)	0
dense_0 (Dense)	(None, 768)	1770240
dropout_6 (Dropout)	(None, 768)	0
dense_1 (Dense)	(None, 8)	6152
Total params: 1,837,020		
Trainable params: 1,837,020		
Non-trainable params: 0		

Tabulka 4.7: Nejlepší architektura CNN sítě pro klasifikaci pro metodu M06



Obrázek 4.9: Matice záměn pro nejlepší dvě metody CNN

Metoda	Accuracy	w-Precision	w-Recall	w-F1	w-AUC
Random forest	0.7803	0.7805	0.7803	0.7793	0.9709
AdaBoost	0.3594	0.3524	0.3594	0.3140	0.7623
XGBoost	0.7419	0.7430	0.7419	0.7409	0.9627
NN	0.7131	0.7201	0.7131	0.7098	0.9414
CNN M01	0.7050	0.6992	0.7050	0.7003	0.9386
CNN M02	0.7434	0.7450	0.7434	0.7422	0.9625
CNN M03	0.7131	0.7175	0.7131	0.7131	0.9412
CNN M04	0.7041	0.7161	0.7041	0.7059	0.9395
CNN M05	0.7791	0.7826	0.7791	0.7790	0.9698
CNN M06	0.7828	0.7832	0.7828	0.7821	0.9754
CNN M07	0.7669	0.7684	0.7669	0.7666	0.9643
CNN M08	0.7563	0.7552	0.7563	0.7555	0.9638
CNN M09	0.7391	0.7367	0.7391	0.7370	0.9613

Tabulka 4.8: Výsledné hodnoty metrik pro klasifikační metody na testovacích datech (nejlepší hodnota kritérií tučně, druhá nejlepší podtrženě)

4.2.3 Trénování regresní přístupu jednotlivých metod

Přístup pomocí klasifikačních metod má omezenou použitelnost, neboť nám dává informaci pouze o dominantní složce, a tudíž je možné ho dále použít pouze pokud bychom chtěli modelovat PM prostor pouze jednou pravděpodobnostní distribucí. Pokud bychom chtěli přímo z hysterezní smyčky získat informaci o vhodné směsi distribucí včetně zastoupení jednotlivých složek, budeme muset přejít k regresnímu modelování. V tomto případě máme k dispozici stejný dataset prediktorů, avšak jako příznaky použijeme zastoupení jednotlivých složek ve směsi, ze které vznikly. V našem případě se tedy bude jednat o multi-target regresi 8 hodnot, které mají zafixované pořadí, a navíc z definice směsí (definice 2.1.1) plyne že se bude jednat o nezáporné hodnoty, jejichž součet je roven jedné. Těchto vazebných podmínek dosáhneme uměle v rámci

post-zpracování dat, kdy budou případné hodnoty menší než nula, předefinovány na 0 a součet predikovaných proměnných bude přenormován na jedničku. Všechny metody budou optimalizovány vůči MSE (viz vzorec (3.6)). Začneme-li ensemble metodami, tak jsme opět stejným způsobem optimalizovali RF, AdaBoost a XGBoost v množině stejných hyperparametrů znázorněných v tabulkách 4.9 až 4.11. Pro náhodný les je multi-output regrese implementovaná přímo a pro zbylé dvě jsme použili metodu `MultiOutputRegressor` z knihovny `scikit-learn`. Jejich nejlepší hodnoty byly pro RF počet stromů = 10, maximální hloubka stromů = 20, kritérium = Gini, minimum vzorků v uzlu pro dělení = 2, minimum vzorků v listu = 1. Pro AdaBoost je optimální počet stromů = 10, learning rate = 0.05. Pro XGBoost jsou počet stromů = 100, maximální hloubka stromů = 10, learning rate = 0.4, min split loss = 0, L_2 regularizace = 1 a L_1 regularizace = 0. Hodnoty MSE a MAE metrik (viz sekce 3.1) na testovacích datech nalezneme pro srovnání pro všechny modely v tabulce 4.15, kde hodnoty těchto metrik jsou vyčísleny mezi skutečnými hodnotami parametrů zastoupení a jejich predikcí.

Hyperparametry	Možnosti
Počet stromů	{ 10 , 50, 100, 200, 500}
Maximální hloubka stromů	{5, 10, 20 , 30, 50}
Kritérium	{ Gini , Entropy}
Minimum vzorků v uzlu pro dělení	{ 2 , 5, 10}
Minimum vzorků v listu	{ 1 , 2, 4}

Tabulka 4.9: Prohledávané hyperparametry pro random forest regresi (nejlepší parametry tučně)

Hyper-parametry	Možnosti
Počet stromů	{ 10 , 50, 100, 200, 500}
Learning rate	{ 0.05 , 0.1, 0.15, 0.2, 0.3}

Tabulka 4.10: Prohledávané hyperparametry pro AdaBoost regresi (nejlepší parametry tučně)

Hyper-parametry	Možnosti
Počet stromů	{10, 50, 100 , 200, 500}
Maximální hloubka stromů	{5, 10 , 20, 30, 50}
Learning rate	{0.1, 0.2, 0.3, 0.4 }
Min split loss	{ 0 , 0.5, 1, 1.5}
L_2 regularizace	{ 1.0 , 1.1, 1.2}
L_1 regularizace	{ 0 , 0.1, 0.2}

Tabulka 4.11: Prohledávané hyperparametry pro XGBoost regresi (nejlepší parametry tučně)

Dále jsme použili pro regresi neuronové sítě, které byly stejným způsobem optimalizovány podle tabulky 4.4, a dále CNN sítě podle tabulky 4.5 pro metodu M06. Výsledné architektury vidíme pro NN v tabulce 4.12, kde opět za každou dense vrstvou kromě poslední byla použita vrstva s ReLU aktivační funkcí a learning rate byl 0.0003489. Výslednou CNN uvádíme v tabulce 4.13. Zde byla za každou 2D konvoluční vrstvou a za vrstvou `dense_0` použita vrstva

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 96)	9696
dense_1 (Dense)	(None, 432)	41904
dropout (Dropout)	(None, 432)	0
dense_2 (Dense)	(None, 320)	138560
dense_3 (Dense)	(None, 288)	92448
dense_4 (Dense)	(None, 416)	120224
dense_5 (Dense)	(None, 224)	93408
dropout_1 (Dropout)	(None, 224)	0
dense_6 (Dense)	(None, 1402)	315450
dense_7 (Dense)	(None, 8)	11224
Total params: 822,914		
Trainable params: 822,914		
Non-trainable params: 0		

Tabulka 4.12: Nejlepší architektura NN sítě pro multi-target regresi

Layer (type)	Output Shape	Param #
conv2d_0 (Conv2D)	(None, 100, 100, 32)	832
conv2d_1 (Conv2D)	(None, 98, 98, 48)	13872
max_pooling2d_1 (MaxPooling2D)	(None, 49, 49, 48)	0
dropout_1 (Dropout)	(None, 49, 49, 48)	0
conv2d_2 (Conv2D)	(None, 44, 44, 32)	55328
conv2d_3 (Conv2D)	(None, 42, 42, 32)	9248
max_pooling2d_3 (MaxPooling2D)	(None, 21, 21, 32)	0
dropout_3 (Dropout)	(None, 21, 21, 32)	0
conv2d_4 (Conv2D)	(None, 17, 17, 32)	25632
flatten (Flatten)	(None, 9248)	0
dense_0 (Dense)	(None, 1024)	9470976
dropout_4 (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 8)	8200
Total params: 9,584,088		
Trainable params: 9,584,088		
Non-trainable params: 0		

Tabulka 4.13: Nejlepší architektura CNN sítě pro multi-target regresi

s ReLU aktivační funkcí a použitý learning rate byl 0.0001464. Byl opět vždy použit pooling 2×2 a hodnota ve vrstvě dropout_1 byla 0.5, ve vrstvě dropout_3 byla 0.25 a ve vrstvě dropout_4 byla 0.5. Srovnání jednotlivých výsledků jsou pak k nalezení v tabulce 4.15. Jak vidíme, tentokrát přístup pomocí převedení prostoru prediktorů do dvourozměrné úlohy a použití konvolučních neuronových sítí vedlo k výrazně lepším výsledkům. Opět nejlepšího řešení dosáhla metoda M06.

4.2.4 Kombinovaný regresor

Nad těmito modely pro regresi jsme se navíc rozhodli vytvořit hypermodely za cílem získání ještě lepších predikcí. Nejdříve začneme pomocí voting metody (viz sekce 3.2) pro výstupy jednotlivých modelů. Konkrétně pro nadmodel HYP1 použijeme voting z nejlepších 4 regresorů, tedy CNN metod M06, M05, M07 a M08. Pro model HYP2 k těmto čtyřem navíc přidáme RF, XGBoost a klasické NN. A na závěr pro model HYP3 provedeme voting mezi všemi 13 základními metodami regrese. V tabulce 4.15 můžeme vidět, že všechny hypermodely dosáhly lepších výsledků než samostatné metody, přičemž nejlepší schopnosti měl model HYP2. Další přístup tvorby hypermodelu byl pomocí klasické neuronové sítě nad vybranými modely. Použijeme k tomu jednoduchou NN síť, která byla tvořena jednou vstupní a jednou výstupní dense vrstvou. Pro tyto modely jsme opět použili 4, 7 a 13 základních metod regrese, jak tomu bylo pro voting. Tyto modely jsme označili jako HYP4, HYP5 a HYP6. Architekturu sítě HYP4 vidíme v tabulce 4.14, přičemž pro zbylé dvě je prakticky stejná s rozdílem, že místo vstupního vektoru prediktorů $4 \times 8 = 32$ používají vstup o velikosti 56, resp. 104. Tyto modely byly natrénovány na stejných trénovacích a validačních datech se stejným learning rate, na jakém byla natrénována základní NN. Trénování bylo nastaveno opět na 200 iterací se stejnou zastavovací podmínkou, pokud se po 15 iteracích nezlepší hodnota validační ztrátové funkce. Výsledky na testovacích datech nalezneme opět v tabulce 4.15, přičemž výsledky již nejsou lepší než při hypermodelech vzniklých použitím votingu.

Layer (type)	Output Shape	Param #
dense_0 (Dense)	(None, 128)	4224
dense_1 (Dense)	(None, 8)	1032
Total params: 5,256		
Trainable params: 5,256		
Non-trainable params: 0		

Tabulka 4.14: Architektura modelu HYP4 pro multi-target regresi

Kvalitu jednotlivých klasifikátorů na závěr této sekce ještě otestujeme na alternativním umělém datasetu, který vznikl odlišným způsobem generování. Tentokrát náhodně generujeme směsi složených ze šesti až osmi složek, přičemž náhodně vybereme dvě různé distribuční směsi, kde každé přiřadíme náhodné zastoupení 30 až 45 procent. Zbylé složky jsou náhodně generovány zbylými distribucemi ve zbývajícím rozsahu zastoupení. Tímto způsobem jsme vygenerovali nová data, která jsou stejně velká jako původní testovací dataset. Tento nový dataset tedy nemá v každém vzorku jednu dominantní složku, ale dvě přibližně stejně zastoupené distribuce. Tímto způsobem chceme ukázat schopnosti dostatečné generalizace regresoru na odlišném typu dat, které nebyly k dispozici při procesu trénování. V tabulce 4.15 nalezneme v pravé části hodnoty metrik pro tato alternativní data. Můžeme vidět, že sice došlo ke zhoršení, avšak ne příliš dramatickému. Nejlepšího výsledku na těchto datech dosáhly kombinované regresory vytvořené ze všech dílčích metod, i když pro původní testovací dataset byly favorizovány hypermodely tvořené menším počtem regresorů. Zároveň pro model HYP3 došlo k nejnižšímu zhoršení daných metrik oproti testovacím datům. Tento výsledek naznačuje trend, že lepší schopnosti generalizace dosahujeme při zakomponování více dílčích modelů, i když

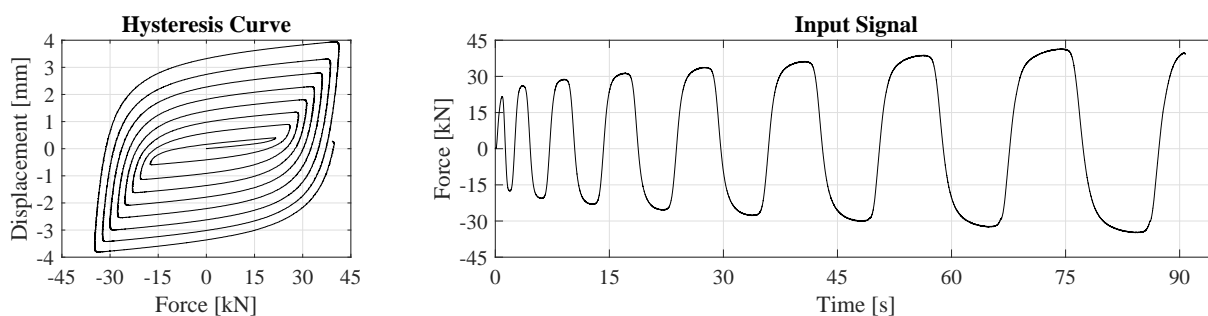
jednotlivě dosahují některé z nich výrazně horších predikcí. Proto na reálných datech budeme používat model HYP3.

Metoda	Test data		Alternative data	
	MSE	MAE	MSE	MAE
Random forest	0.02139	0.08654	0.02980	0.12834
AdaBoost	0.04343	0.14500	0.02662	0.13174
XGBoost	0.02172	0.09162	0.02768	0.12348
NN	0.02008	0.07867	0.03412	0.12559
CNN M01	0.02037	0.08336	0.02610	0.11479
CNN M02	0.02345	0.09480	0.02554	0.11652
CNN M03	0.02290	0.08829	0.02914	0.12226
CNN M04	0.02252	0.08916	0.02869	0.12184
CNN M05	0.01777	0.07352	0.03134	0.12724
CNN M06	0.01705	0.07465	0.02973	0.12559
CNN M07	0.01880	0.07770	0.03063	0.12780
CNN M08	0.01917	0.07734	0.03147	0.12952
CNN M09	0.02022	0.08336	0.03048	0.12962
HYP1	0.01688	<u>0.07400</u>	0.02877	0.12444
HYP2	0.01535	0.07464	<u>0.02568</u>	<u>0.12502</u>
HYP3	<u>0.01662</u>	0.08279	0.02337	0.12009
HYP4	0.01700	0.07320	0.02925	0.12520
HYP5	0.01716	0.07768	0.02936	0.13459
HYP6	0.01713	0.07852	0.02874	0.13339

Tabulka 4.15: Výsledné hodnoty metrik pro regresní metody na testovacích a alternativních datech (nejlepší hodnota kritéria tučně, druhá nejlepší podtrženě)

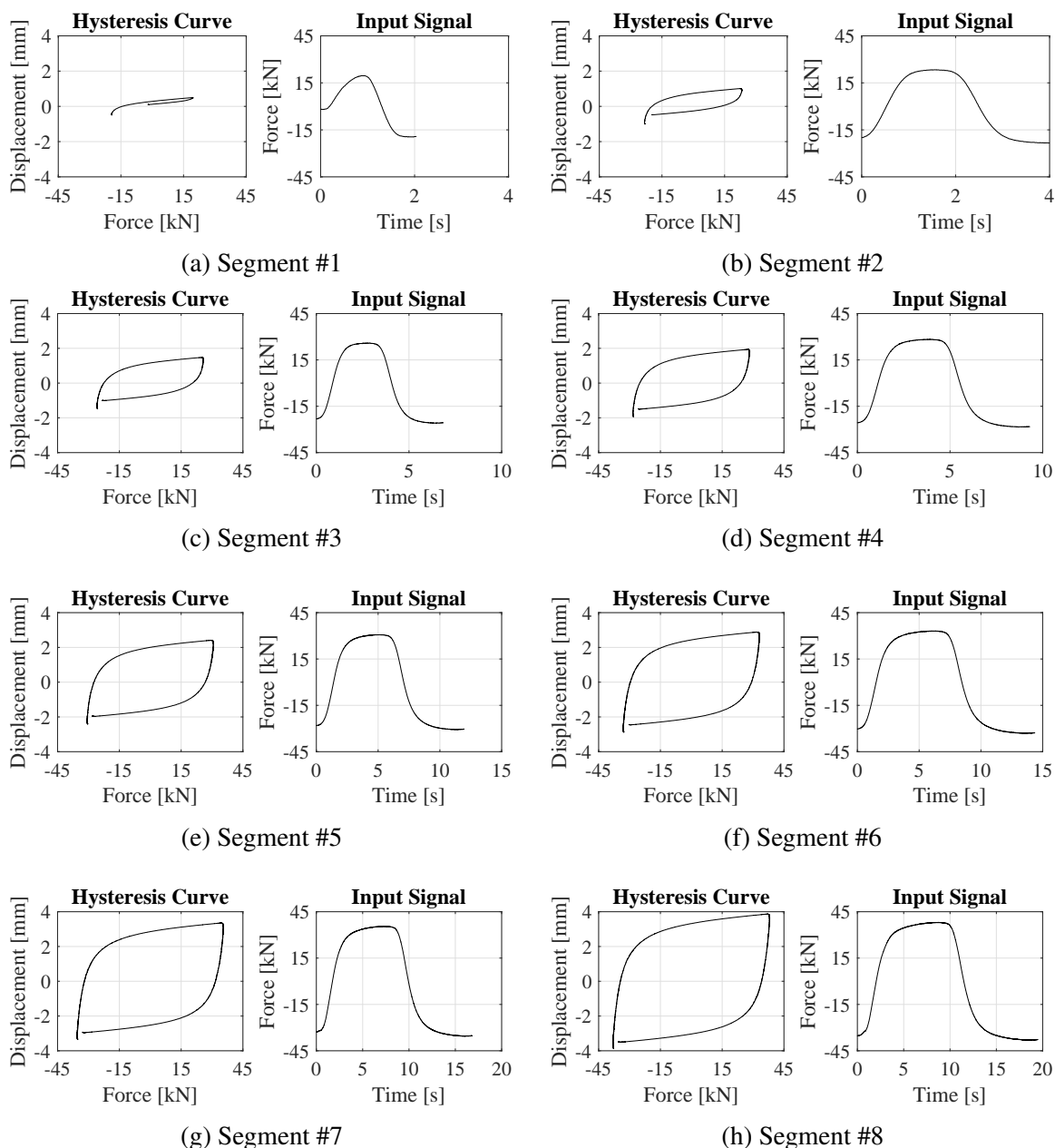
4.3 Vyhodnocení na reálných datech I

Po generovaných datech přejdeme k aplikaci na naměřená data z testování zemětřesných tlumičů, viz sekce 1.3. Jednalo se o cyklický test, při kterém bylo mechanickým zatížením si-



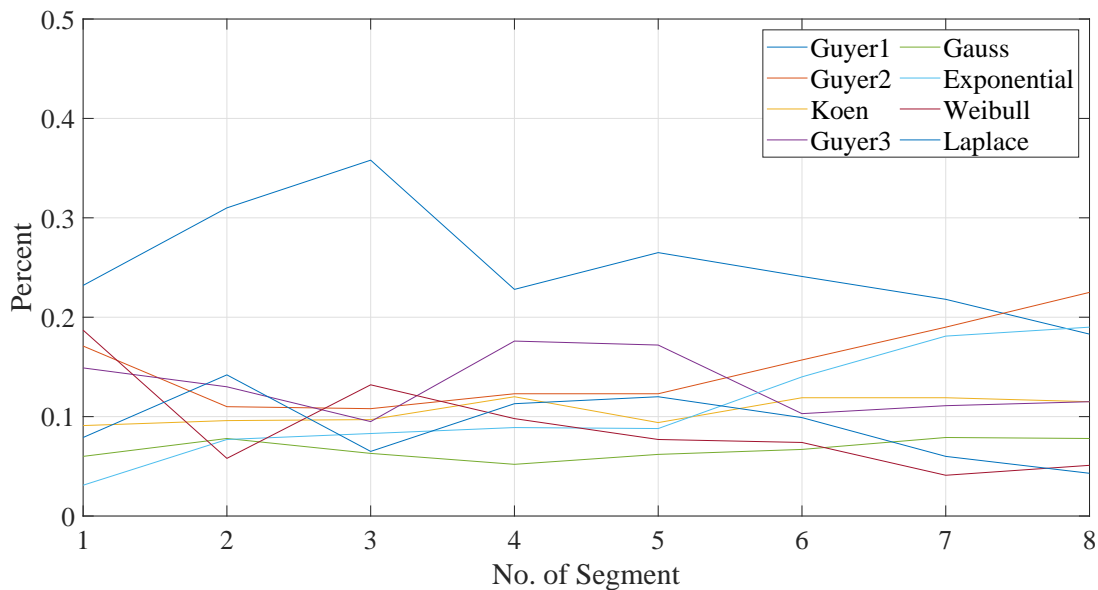
Obrázek 4.10: Naměřená data z testování ocelových zemětřesných tlumičů

mulujícím zemětřesení osmkrát za sebou namáhán daný tlumič. Informace o způsobu provedení experimentu nalezneme v [14]. Na obr. 4.10 můžeme vidět naměřenou hysterezní křivku a průběh odpovídajícího vstupního zatížení. Data jsme rozdělili podle jednotlivých cyklů, přičemž jednotlivé smyčky můžeme vidět na obrázku 4.11. Pomocí našeho finálního modelu HYP3 jsme predikovali zastoupení jednotlivých pravděpodobnostních rozdělení v distribuční směsi. Hodnoty zastoupení jednotlivých komponent pro všech osm smyček můžeme vidět na obrázku 4.12.



Obrázek 4.11: Separovaná data z testování ocelových zemětřesných tlumičů (pokračování)

Je patrné, že dominantní zastoupení napříč smyčkami má rozdělení Guyer1, a dále je vidět patrný nárůst zastoupení exponenciálního rozdělení a rozdělení Guyer2 u třech posledních hysterezních smyček, které reprezentují největší poškození tlumiče a je zde přítomna největší neelasticita. Jiný patrnější trend už napříč jednotlivými křivkami nenajdeme. Pro nalezení PM prostoru pro každý segment jsme se rozhodli použít vždy čtyři nejvíce zastoupená rozdělení se zachovanou proporcí v rámci směsi.



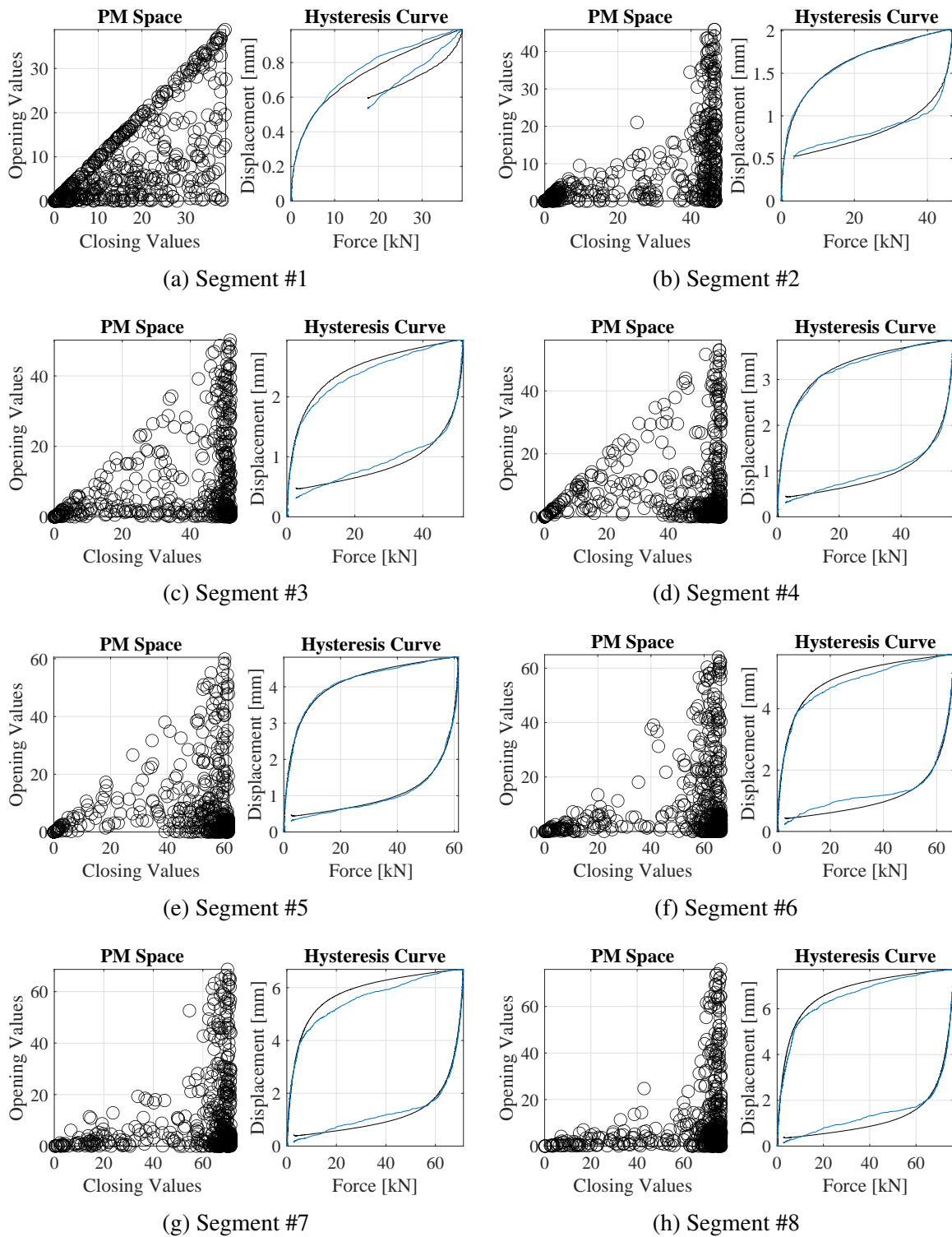
Obrázek 4.12: Zastoupení jednotlivých pravděpodobnostních rozdělení podle modelu HYP3

Vzhledem k tomu, že jsme pomocí dosavadního strojového postupu získali informaci, která pravděpodobnostní rozdělení použít včetně jejich zastoupení ve směsi, na optimalizaci pomocí heuristických algoritmů (viz sekce 2.3) zbyla předpověď pouze parametrů jednotlivých distribučních komponent. Nejlepší nalezené PM prostory nalezneme na obr. 4.13 a jejich popis pomocí parametrů jednotlivých distribucí vidíme v tabulce 4.16, přičemž α_1 až α_4 značí zastoupení komponent ve směsi (viz definice 2.1.1). H^* značí Hellingerovu divergenci (viz sekce 2.2), která byla přenormována počtem bodů, kterými je tvořena příslušná hysterezní smyčka, aby bylo možné navzájem porovnávat kvalitu odhadů.

Výsledky hysterezních smyček jsou po vizuální stránce uspokojivé a hodnota H^* se řádově pohybuje na hodnotách dosažených při optimalizaci na jiných datech v pracích [7] a [8], avšak je zde další prostor pro zlepšení, zejména pak u nejvíce rozevřených hysterezních smyček, které byly problematické i v předešlých pracích. Vzhledem k tomu, že výsledné směsi pro tři poslední segmenty jsou podle našeho finálního modelu tvořeny stejnými komponentami, rozhodli jsme pokusit o zlepšení těchto výsledků. Prvním pokusem o zlepšení bylo natrénování našeho finálního modelu HYP3 na simulovaných datech pro směsi právě těchto čtyř komponent tvořených pouze rozděleními, které nám vyšli pro poslední tři smyčky, tedy rozdělení Guyer1, Guyer2, Guyer3 a exponenciální rozdělení. Výsledné zastoupení a nalezené PM prostory, společně s odpovídajícími hysterezními křivkami nalezneme na obrázku 4.14 a v tabulce 4.17. Ve dvou případech #6 a #7 se výsledek z pohledu hodnoty H^* mírně zlepšil, ale pro poslední smyčku #8 je výsledek naopak horší v H^* .

Segment	Směs	Zastoupení	Parametry rozdělení
Segment #1 $H^* = 8.531 \cdot 10^{-4}$	Guyer1 Guyer2 Guyer3 Weibull	$\alpha_1 = \mathbf{0.315}$ $\alpha_2 = 0.232$ $\alpha_3 = 0.201$ $\alpha_4 = \underline{0.252}$	$\eta = 1.473$ $\kappa = 0.040$ $\eta = 1.318$ $\kappa = 2.919$ $\eta = 1.415$ $\kappa = 1.793$ $\lambda = 0.615$ $\kappa_1 = 0.268$ $\lambda_1 = 8.732$ $\kappa_2 = 0.941$ $\lambda_2 = 5.908$
Segment #2 $H^* = 6.571 \cdot 10^{-4}$	Guyer1 Guyer2 Guyer3 Laplace	$\alpha_1 = \mathbf{0.449}$ $\alpha_2 = 0.159$ $\alpha_3 = 0.188$ $\alpha_4 = \underline{0.204}$	$\eta = 0.044$ $\kappa = 1.863$ $\eta = 3.072$ $\kappa = 2.384$ $\eta = 0.628$ $\kappa = 1.256$ $\lambda = 0.558$ $\mu_1 = 0.037$ $\lambda_1 = 0.022$ $\mu_2 = 0.004$ $\lambda_2 = 0.997$
Segment #3 $H^* = 9.518 \cdot 10^{-4}$	Guyer1 Guyer2 Koen Weibull	$\alpha_1 = \mathbf{0.515}$ $\alpha_2 = 0.155$ $\alpha_3 = 0.140$ $\alpha_4 = \underline{0.190}$	$\eta = 0.041$ $\kappa = 2.159$ $\eta = 3.857$ $\kappa = 1.510$ $\kappa = 2.351$ $\lambda = 3.048$ $\kappa_1 = 0.364$ $\lambda_1 = 9.583$ $\kappa_2 = 0.613$ $\lambda_2 = 9.575$
Segment #4 $H^* = 5.446 \cdot 10^{-4}$	Guyer1 Guyer2 Koen Guyer3	$\alpha_1 = \mathbf{0.351}$ $\alpha_2 = 0.190$ $\alpha_3 = 0.186$ $\alpha_4 = 0.272$	$\eta = 0.024$ $\kappa = 2.709$ $\eta = 3.544$ $\kappa = 0.011$ $\kappa = 5.726$ $\lambda = 0.798$ $\eta = 0.097$ $\kappa = 1.563$ $\lambda = 0.218$
Segment #5 $H^* = 5.601 \cdot 10^{-4}$	Guyer1 Guyer2 Guyer3 Laplace	$\alpha_1 = \mathbf{0.389}$ $\alpha_2 = 0.182$ $\alpha_3 = \underline{0.253}$ $\alpha_4 = 0.176$	$\eta = 0.089$ $\kappa = 3.380$ $\eta = 2.921$ $\kappa = 0.835$ $\eta = 0.007$ $\kappa = 1.160$ $\lambda = 0.106$ $\mu_1 = 0.935$ $\lambda_1 = 0.531$ $\mu_2 = 0.100$ $\lambda_2 = 0.103$
Segment #6 $H^* = 5.166 \cdot 10^{-4}$	Guyer1 Guyer2 Guyer3 Exponential	$\alpha_1 = \mathbf{0.367}$ $\alpha_2 = \underline{0.240}$ $\alpha_3 = 0.182$ $\alpha_4 = 0.212$	$\eta = 0.107$ $\kappa = 2.773$ $\eta = 0.081$ $\kappa = 3.645$ $\eta = 0.011$ $\kappa = 1.127$ $\lambda = 0.131$ $\mu_1 = 0.209$ $\mu_2 = 0.224$
Segment #7 $H^* = 6.832 \cdot 10^{-4}$	Guyer1 Guyer2 Guyer3 Exponential	$\alpha_1 = \mathbf{0.308}$ $\alpha_2 = \underline{0.269}$ $\alpha_3 = 0.168$ $\alpha_4 = 0.255$	$\eta = 0.074$ $\kappa = 2.962$ $\eta = 0.053$ $\kappa = 2.685$ $\eta = 0.011$ $\kappa = 1.160$ $\lambda = 0.131$ $\mu_1 = 0.807$ $\mu_2 = 0.133$
Segment #8 $H^* = 7.280 \cdot 10^{-4}$	Guyer1 Guyer2 Guyer3 Exponential	$\alpha_1 = 0.257$ $\alpha_2 = \mathbf{0.316}$ $\alpha_3 = 0.161$ $\alpha_4 = \underline{0.267}$	$\eta = 0.045$ $\kappa = 3.192$ $\eta = 0.045$ $\kappa = 3.170$ $\eta = 0.038$ $\kappa = 1.134$ $\lambda = 0.143$ $\mu_1 = 0.880$ $\mu_2 = 0.101$

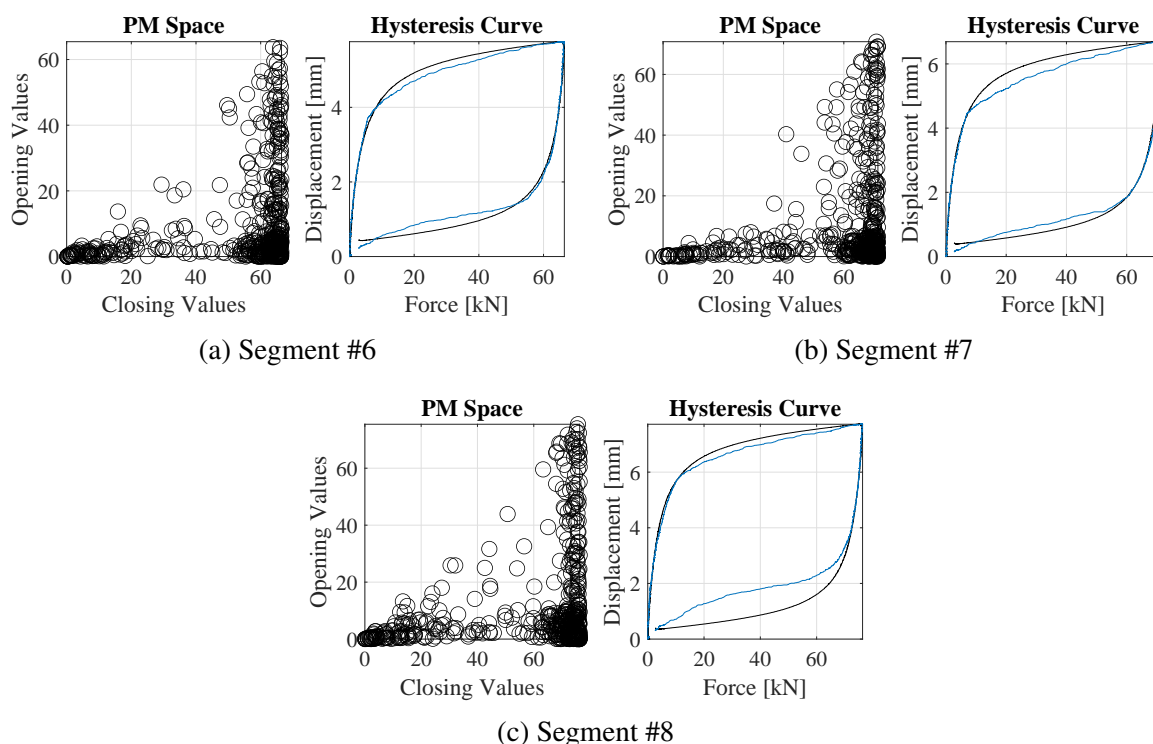
Tabulka 4.16: Rozdělení a parametry identifikovaných PM prostorů tlumiče při použití H^* metriky (největší zastoupení ve směsi tučně, druhé největší podtržené)



Obrázek 4.13: Nalezené PM prostory a k nim příslušné hysterezní křivky (modře) při použití H^* metriky

Segment	Směs	Zastoupení	Parametry rozdělení		
Segment #6 $H^* = 4.623 \cdot 10^{-4}$	Guyer1	$\alpha_1 = 0.224$	$\eta = 0.109$	$\kappa = 2.490$	$\lambda = 0.115$
	Guyer2	$\alpha_2 = \mathbf{0.316}$	$\eta = 0.094$	$\kappa = 3.792$	
	Guyer3	$\alpha_3 = \underline{0.234}$	$\eta = 0.013$	$\kappa = 1.110$	
	Exponential	$\alpha_4 = 0.226$	$\mu_1 = 0.278$	$\mu_2 = 0.467$	
Segment #7 $H^* = 5.739 \cdot 10^{-4}$	Guyer1	$\alpha_1 = 0.223$	$\eta = 0.026$	$\kappa = 3.269$	$\lambda = 0.103$
	Guyer2	$\alpha_2 = \mathbf{0.315}$	$\eta = 0.097$	$\kappa = 3.145$	
	Guyer3	$\alpha_3 = \underline{0.244}$	$\eta = 0.049$	$\kappa = 1.035$	
	Exponential	$\alpha_4 = 0.218$	$\mu_1 = 0.391$	$\mu_2 = 0.140$	
Segment #8 $H^* = 9.680 \cdot 10^{-4}$	Guyer1	$\alpha_1 = 0.230$	$\eta = 0.035$	$\kappa = 3.148$	$\lambda = 0.137$
	Guyer2	$\alpha_2 = 0.253$	$\eta = 0.042$	$\kappa = 3.032$	
	Guyer3	$\alpha_3 = \underline{0.258}$	$\eta = 0.138$	$\kappa = 1.179$	
	Exponential	$\alpha_4 = \mathbf{0.259}$	$\mu_1 = 0.292$	$\mu_2 = 0.265$	

Tabulka 4.17: Rozdělení a parametry identifikovaných PM prostorů tlumiče při použití H^* metriky po přetrénování směsi 4 nejlepších komponent (největší zastoupení ve směsi tučně, druhé největší podtržené)



Obrázek 4.14: Nalezené PM prostory a k nim příslušné hysterezní křivky (modře) při použití H^* metriky po přetrénování směsi 4 nejlepších komponent

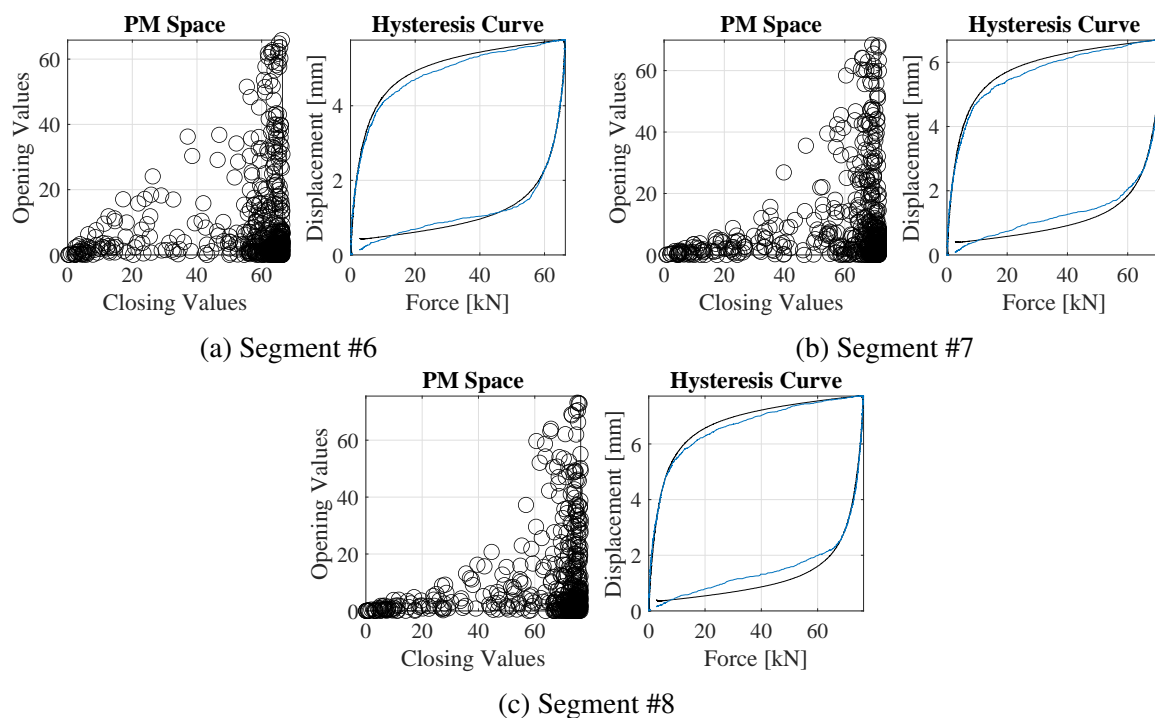
Další pokus o vylepšení původního výsledku je použití více komponent, a to ve smyslu směsi směr z definice 2.1.2. Tu aplikujeme tak, že na výsledné zastoupení komponent z původního modelu, která vidíme v tabulce 4.16, místo použití jedné sady parametrů pro danou

složku použijeme podsměs dvou stejných rozdělení, kde každé převezme poloviční zastoupení. V řeči definice 2.1.2 použijeme zastoupení z původního výsledku, ale místo hledání parametrů jedno rozdělení budeme hledat parametry pro dvě rozdělení stejného typu, na která vždy připadne polovina původního zastoupení. Výsledné PM prostory a příslušné hysterezní smyčky nalezneme na obrázku 4.15 a parametrický popis směsi potom v tabulce 4.18. Podle hodnot H^* jsme dosáhli celkově stabilnější výsledky. Avšak nového nejlepšího výsledku z hlediska H^* jsme dosáhli pouze pro poslední segment #8, což však vidíme z obrázku 4.15 bylo způsobeno vylepšením pouze horní části hysterezní smyčky.

Segment	Směs	Zastoupení	Parametry rozdělení		
Segment #6 $H^* = 5.466 \cdot 10^{-4}$	Guyer1	$\lambda_1 \cdot \alpha_1 = 0.183$	$\eta = 0.065$	$\kappa = 2.512$	
	Guyer1	$\lambda_1 \cdot \alpha_2 = 0.183$	$\eta = 0.071$	$\kappa = 2.957$	
	Guyer2	$\lambda_2 \cdot \alpha_1 = 0.120$	$\eta = 0.129$	$\kappa = 3.320$	
	Guyer2	$\lambda_2 \cdot \alpha_2 = 0.120$	$\eta = 0.032$	$\kappa = 4.538$	
	Guyer3	$\lambda_3 \cdot \alpha_1 = 0.091$	$\eta = 0.098$	$\kappa = 1.093$	$\lambda = 0.107$
	Guyer3	$\lambda_3 \cdot \alpha_2 = 0.091$	$\eta = 0.002$	$\kappa = 1.090$	$\lambda = 0.147$
	Exponential	$\lambda_4 \cdot \alpha_1 = 0.106$	$\mu_1 = 0.247$	$\mu_2 = 0.945$	
	Exponential	$\lambda_4 \cdot \alpha_2 = 0.106$	$\mu_1 = 0.612$	$\mu_2 = 0.139$	
Segment #7 $H^* = 5.983 \cdot 10^{-4}$	Guyer1	$\lambda_1 \cdot \alpha_1 = 0.154$	$\eta = 0.116$	$\kappa = 3.115$	
	Guyer1	$\lambda_1 \cdot \alpha_2 = 0.154$	$\eta = 0.026$	$\kappa = 3.269$	
	Guyer2	$\lambda_2 \cdot \alpha_1 = 0.134$	$\eta = 0.035$	$\kappa = 3.708$	
	Guyer2	$\lambda_2 \cdot \alpha_2 = 0.134$	$\eta = 0.058$	$\kappa = 3.940$	
	Guyer3	$\lambda_3 \cdot \alpha_1 = 0.084$	$\eta = 0.001$	$\kappa = 1.288$	$\lambda = 0.200$
	Guyer3	$\lambda_3 \cdot \alpha_2 = 0.084$	$\eta = 0.019$	$\kappa = 1.901$	$\lambda = 0.337$
	Exponential	$\lambda_4 \cdot \alpha_1 = 0.128$	$\mu_1 = 0.781$	$\mu_2 = 0.126$	
	Exponential	$\lambda_4 \cdot \alpha_2 = 0.128$	$\mu_1 = 0.867$	$\mu_2 = 0.165$	
Segment #8 $H^* = 5.882 \cdot 10^{-4}$	Guyer1	$\lambda_1 \cdot \alpha_1 = 0.128$	$\eta = 0.077$	$\kappa = 2.831$	
	Guyer1	$\lambda_1 \cdot \alpha_2 = 0.128$	$\eta = 0.024$	$\kappa = 3.641$	
	Guyer2	$\lambda_2 \cdot \alpha_1 = 0.158$	$\eta = 0.041$	$\kappa = 2.831$	
	Guyer2	$\lambda_2 \cdot \alpha_2 = 0.158$	$\eta = 0.106$	$\kappa = 4.527$	
	Guyer3	$\lambda_3 \cdot \alpha_1 = 0.080$	$\eta = 0.015$	$\kappa = 1.039$	$\lambda = 0.118$
	Guyer3	$\lambda_3 \cdot \alpha_2 = 0.080$	$\eta = 0.025$	$\kappa = 1.124$	$\lambda = 0.101$
	Exponential	$\lambda_4 \cdot \alpha_1 = 0.133$	$\mu_1 = 0.945$	$\mu_2 = 0.136$	
	Exponential	$\lambda_4 \cdot \alpha_2 = 0.133$	$\mu_1 = 0.994$	$\mu_2 = 0.101$	

Tabulka 4.18: Rozdělení a parametry identifikovaných PM prostorů tlumiče při použití H^* metriky a při použití směsi podsměsí

Posledním způsobem, jak zlepšit odhady posledních tří hysterezních smyček, je místo použití Hellingerovy metriky pro optimalizaci parametrů jednotlivých komponent směsi, použití L_2 metriky. Ačkoliv nelze srovnávat hodnoty L_2^* a H^* , po vizuální stránce se výsledky získané tímto přístupem zdají výrazně lepší, a proto jsme touto optimalizací, při které minimalizujeme L_2 metriku, odhadli i zbylé hysterezní smyčky. Jednotlivé segmenty můžeme vidět na obrázku 4.16 a zápis příslušných PM prostorů v tabulce 4.19. Až na případ smyčky #2 se zdají být po vizuální stránce výsledky lepší než při optimalizaci vůči Hellingerově divergenci. Srovnali jsme

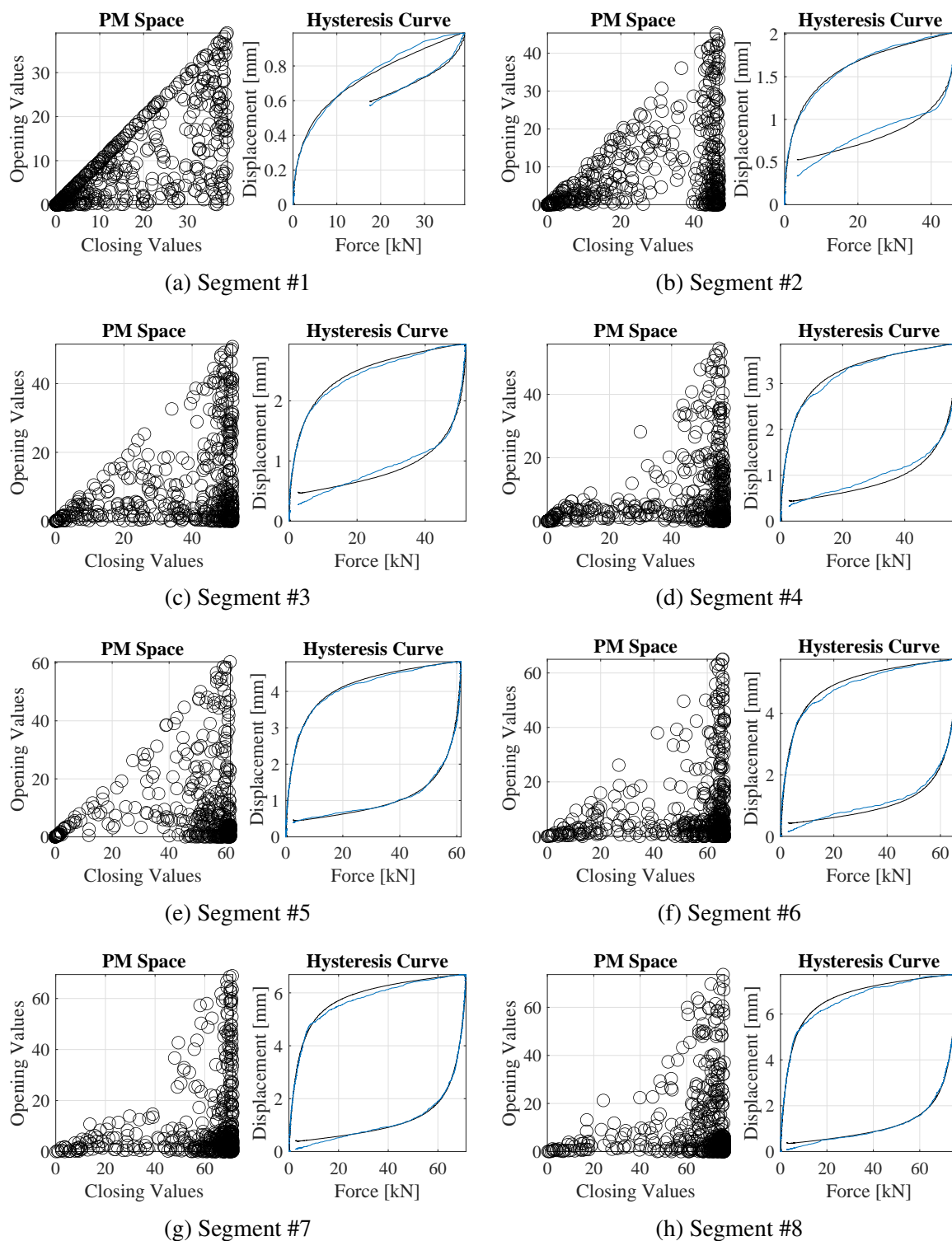


Obrázek 4.15: Nalezené PM prostory a k nim příslušné hysterezní křivky (modře) při použití H^* metriky a při použití směsi podsměsí

proto obě metriky pro všechny čtyři předchozí způsoby vůči oběma metrikám, což můžeme vidět v tabulce 4.20. Vzhledem k hodnotové nejednoznačnosti hodnocení nejlepšího výsledku na základě obou metrik, jsme se rozhodli na základě vizuálního porovnání získaných hysterezních smyček považovat většinově ten, který je nejlepší podle L_2 metriky se dvěma výjimkami. Tedy z původního natrénování finálního modelu na 8 distribucích, který nám dal výsledky zastoupení čtyř nejvýznamnějších komponent pro každou smyčku, jsme pro segmenty #2 a #4 získali parametry těchto pravděpodobnostních rozdělení optimalizací vůči Hellingerově divergenci a pro zbylé segmenty optimalizací vůči L_2 metrice.

Segment	Směs	Zastoupení	Parametry rozdělení		
Segment #1 $L_2^* = 1.417 \cdot 10^{-3}$	Guyer1 Guyer2 Guyer3 Weibull	$\alpha_1 = \mathbf{0.315}$ $\alpha_2 = 0.232$ $\alpha_3 = 0.201$ $\alpha_4 = \underline{0.252}$	$\eta = 2.210$ $\eta = 2.379$ $\eta = 0.096$ $\kappa_1 = 0.175$ $\kappa_2 = 0.467$	$\kappa = 0.017$ $\kappa = 0.658$ $\kappa = 0.895$ $\lambda_1 = 7.638$ $\lambda_2 = 1.747$	$\lambda = 0.977$
Segment #2 $L_2^* = 1.548 \cdot 10^{-3}$	Guyer1 Guyer2 Guyer3 Laplace	$\alpha_1 = \mathbf{0.449}$ $\alpha_2 = 0.159$ $\alpha_3 = 0.188$ $\alpha_4 = \underline{0.204}$	$\eta = 0.042$ $\eta = 5.144$ $\eta = 2.415$ $\mu_1 = 0.108$ $\mu_2 = 0.934$	$\kappa = 2.803$ $\kappa = 0.231$ $\kappa = 1.016$ $\lambda_1 = 0.255$ $\lambda_2 = 0.891$	$\lambda = 0.374$
Segment #3 $L_2^* = 1.083 \cdot 10^{-3}$	Guyer1 Guyer2 Koen Weibull	$\alpha_1 = \mathbf{0.515}$ $\alpha_2 = 0.155$ $\alpha_3 = 0.140$ $\alpha_4 = \underline{0.190}$	$\eta = 0.045$ $\eta = 4.669$ $\kappa = 5.161$ $\kappa_1 = 0.651$ $\kappa_2 = 0.719$	$\kappa = 2.739$ $\kappa = 1.371$ $\lambda = 1.433$ $\lambda_1 = 5.180$ $\lambda_2 = 6.587$	
Segment #4 $L_2^* = 7.164 \cdot 10^{-4}$	Guyer1 Guyer2 Koen Guyer3	$\alpha_1 = \mathbf{0.351}$ $\alpha_2 = 0.190$ $\alpha_3 = 0.186$ $\alpha_4 = \underline{0.272}$	$\eta = 0.086$ $\eta = 3.790$ $\kappa = 8.401$ $\eta = 0.015$	$\kappa = 2.077$ $\kappa = 1.485$ $\lambda = 1.109$ $\kappa = 2.520$	$\lambda = 0.700$
Segment #5 $L_2^* = 5.580 \cdot 10^{-4}$	Guyer1 Guyer2 Guyer3 Laplace	$\alpha_1 = \mathbf{0.389}$ $\alpha_2 = 0.182$ $\alpha_3 = \underline{0.253}$ $\alpha_4 = 0.176$	$\eta = 0.101$ $\eta = 3.204$ $\eta = 0.008$ $\mu_1 = 0.996$ $\mu_2 = 0.106$	$\kappa = 3.723$ $\kappa = 0.062$ $\kappa = 1.233$ $\lambda_1 = 0.339$ $\lambda_2 = 0.101$	$\lambda = 0.115$
Segment #6 $L_2^* = 5.957 \cdot 10^{-4}$	Guyer1 Guyer2 Guyer3 Exponential	$\alpha_1 = \mathbf{0.367}$ $\alpha_2 = \underline{0.240}$ $\alpha_3 = 0.182$ $\alpha_4 = 0.212$	$\eta = 0.020$ $\eta = 0.124$ $\eta = 0.145$ $\mu_1 = 0.367$	$\kappa = 3.078$ $\kappa = 5.045$ $\kappa = 1.037$ $\mu_2 = 0.396$	$\lambda = 0.113$
Segment #7 $L_2^* = 4.643 \cdot 10^{-4}$	Guyer1 Guyer2 Guyer3 Exponential	$\alpha_1 = \mathbf{0.308}$ $\alpha_2 = \underline{0.269}$ $\alpha_3 = 0.168$ $\alpha_4 = 0.255$	$\eta = 0.017$ $\eta = 0.137$ $\eta = 0.097$ $\mu_1 = 0.476$	$\kappa = 2.840$ $\kappa = 3.670$ $\kappa = 1.123$ $\mu_2 = 0.203$	$\lambda = 0.149$
Segment #8 $L_2^* = 4.005 \cdot 10^{-4}$	Guyer1 Guyer2 Guyer3 Exponential	$\alpha_1 = 0.257$ $\alpha_2 = \mathbf{0.316}$ $\alpha_3 = 0.161$ $\alpha_4 = \underline{0.267}$	$\eta = 0.144$ $\eta = 0.070$ $\eta = 0.018$ $\mu_1 = 0.984$	$\kappa = 3.188$ $\kappa = 4.180$ $\kappa = 1.168$ $\mu_2 = 0.187$	$\lambda = 0.122$

Tabulka 4.19: Rozdělení a parametry identifikovaných PM prostorů tlumiče při použití L_2 metriky



Obrázek 4.16: Nalezené PM prostory a k nim příslušné hysterezní křivky (modře) při použití L_2 metriky

Segment	Div	Results A	Results B	Results C	Results D
Segment #1	H^*	$8.531 \cdot 10^{-4}$			$1.118 \cdot 10^{-3}$
	L_2^*	$2.705 \cdot 10^{-3}$			$1.417 \cdot 10^{-3}$
Segment #2	H^*	$6.571 \cdot 10^{-4}$			$1.588 \cdot 10^{-3}$
	L_2^*	$1.036 \cdot 10^{-3}$			$1.548 \cdot 10^{-3}$
Segment #3	H^*	$9.518 \cdot 10^{-4}$			$1.162 \cdot 10^{-3}$
	L_2^*	$1.168 \cdot 10^{-3}$			$1.083 \cdot 10^{-3}$
Segment #4	H^*	$5.446 \cdot 10^{-4}$			$8.210 \cdot 10^{-4}$
	L_2^*	$5.586 \cdot 10^{-4}$			$7.164 \cdot 10^{-4}$
Segment #5	H^*	$5.601 \cdot 10^{-4}$			$6.462 \cdot 10^{-4}$
	L_2^*	$5.974 \cdot 10^{-4}$			$5.580 \cdot 10^{-4}$
Segment #6	H^*	$5.166 \cdot 10^{-4}$	$4.623 \cdot 10^{-4}$	$5.466 \cdot 10^{-4}$	$7.157 \cdot 10^{-4}$
	L_2^*	$6.416 \cdot 10^{-4}$	$6.059 \cdot 10^{-4}$	$6.136 \cdot 10^{-4}$	$5.957 \cdot 10^{-4}$
Segment #7	H^*	$6.832 \cdot 10^{-4}$	$5.739 \cdot 10^{-4}$	$5.983 \cdot 10^{-4}$	$7.942 \cdot 10^{-4}$
	L_2^*	$7.284 \cdot 10^{-4}$	$6.284 \cdot 10^{-4}$	$6.760 \cdot 10^{-4}$	$4.643 \cdot 10^{-4}$
Segment #8	H^*	$7.280 \cdot 10^{-4}$	$9.680 \cdot 10^{-4}$	$5.882 \cdot 10^{-4}$	$9.257 \cdot 10^{-4}$
	L_2^*	$6.472 \cdot 10^{-4}$	$9.047 \cdot 10^{-4}$	$6.320 \cdot 10^{-4}$	$4.005 \cdot 10^{-4}$

Tabulka 4.20: Srovnání Hellingerovy (H) a L_2 metriky pro výsledné křivky optimalizované vůči H z natrénování na 8 distribucích (Results A), vůči H z natrénování na 4 distribucích (Results B), vůči H z natrénování na 8 distribucích s podsměsí (Results C) a vůči L_2 z natrénování na 8 distribucích (Results D) (nejlepší hodnota metriky tučně)

Kapitola 5

Jádrové odhady hustoty pravděpodobnosti

V poslední části si nejdříve představíme jádrové odhady hustoty pravděpodobnosti a poté rozebereme možnosti jejich aplikace na specifickou úlohu Preisachova trojúhelníku. Získaný jádrový odhad na PM prostoru nám poté bude sloužit pro vyhodnocení stavu poškození materiálu, resp. jak dobrých elastických vlastností materiál dosahuje. Představíme dále indexy elasticity, resp. poškození, založené na těchto specifických jádrových odhadech a vyhodnotíme je na testovacích datech, která jsme průběžně zpracovávali již v sekci 4.3. Začneme nejdříve definicí jádrového odhadu pro jednorozměrná data.

Definice 5.0.1. (Jádrový odhad hustoty pravděpodobnosti). Mějme X_1, \dots, X_n náhodný výběr o velikosti n ze spojitě náhodné veličiny X s distribucí $f(x)$. Dále necht' jádro $K(x) \geq 0$ je symetrickou funkcí, která splňuje

$$\int K(x) dx = 1, \quad \int x K(x) dx = 0, \quad \int x^2 K(x) dx > 0. \quad (5.1)$$

Pak $\forall x \in \mathbb{R}$ jádrový odhad hustoty pravděpodobnosti definujeme jako

$$\hat{f}(x; h) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) = \frac{1}{n} \sum_{i=1}^n K_h(x - X_i), \quad (5.2)$$

kde $h \in \mathbb{R}^+$ je vyhlazovací parametr a $K_h(x) = \frac{1}{h} K\left(\frac{x}{h}\right)$ je škálované jádro.

Na rozdíl od popisu rozložení hysteronů pomocí pravděpodobnostních rozdělení nepotřebujeme u jádrových odhadů znát žádné apriorní informace o jejich rozdělení, neboť se jedná o neparametrický odhad. Stačí nám pouze samotné PM prostory. Chyba takového odhadu $\hat{f}(x; h)$ vůči skutečné hodnotě $f(x)$ je zřejmě závislá na volbě vyhlazovacího parametru h . Pro vyčíslení této chyby v konkrétním bodě x můžeme použít střední kvadratickou chybu (MSE), kterou jsme představili již jako ztrátovou funkci v kapitole 3 a kterou můžeme dále jednoduše upravit na

$$\text{MSE}(\hat{f}(x; h)) = E(\hat{f}(x; h) - f(x))^2 = \underbrace{(E\hat{f}(x; h) - f(x))^2}_{\text{Bias}(\hat{f}(x; h))} + \text{Var}\hat{f}(x; h). \quad (5.3)$$

Místo bodové chyby je mnohem přínosnější uvažovat chybu na celém odhadu hustoty. Možným způsobem je střední integrovaná kvadratická chyba (*mean integrated square error*, MISE), kterou můžeme vyjádřit jako

$$\text{MISE}(\hat{f}(\cdot; h)) = \int \text{MSE}(\hat{f}(x; h)) dx = \int \text{Bias}^2(\hat{f}(x; h)) dx + \int \text{Var} \hat{f}(x; h) dx. \quad (5.4)$$

Pomocí jednoduchých úprav a Taylorova rozvoje můžeme tuto chybu odhadnout (viz [32]) do tvaru

$$\text{MISE}(\hat{f}(\cdot; h)) = \frac{1}{4} h^4 \mu_2^2(K) R(f'') + \frac{1}{nh} R(K) + o\left(h^4 + \frac{1}{nh}\right), \quad (5.5)$$

kde

$$R(K) = \int K^2(x) dx, \quad \mu_2(K) = \int x^2 K(x) dx, \quad R(f'') = \int (f''(x))^2 dx. \quad (5.6)$$

Funkce $\text{MISE}(\hat{f}(\cdot; h))$ nabývá minima vzhledem k proměnné h pro hodnotu

$$h_{\text{MISE}} = \left(\frac{R(K)}{\mu_2^2(K) R(f'') n} \right)^{1/5}. \quad (5.7)$$

Dosadíme-li tuto hodnotu do vztahu (5.5), pak

$$\text{MISE}(\hat{f}(\cdot; h)) = \frac{5}{4} (\mu_2^2(K) R^4(K) R(f''))^{1/5} n^{-4/5}, \quad (5.8)$$

z čehož plyne rychlost konvergence MISE pro nejlepší volbu vyhlazovacího parametru a dále $\text{MISE} = 0$ pro $n \rightarrow \infty$, a tedy lze při použití vyhlazovacího parametru $h = o(n^{-1/5})$ dosáhnout řádu konvergence $O(n^{-4/5})$. Kvalita jádrového tedy závisí na volbě jádra a vyhlazovacího parametru, ale z výrazu (5.7) vyplývá, že h_{MISE} závisí na hustotě, kterou odhadujeme. Upravíme-li (5.8) s použitím optimální hodnot vyhlazovacího parametru do tvaru

$$\text{MISE}(\hat{f}(\cdot; h)) = C(K) (R(f''))^{1/5} n^{-4/5}, \quad (5.9)$$

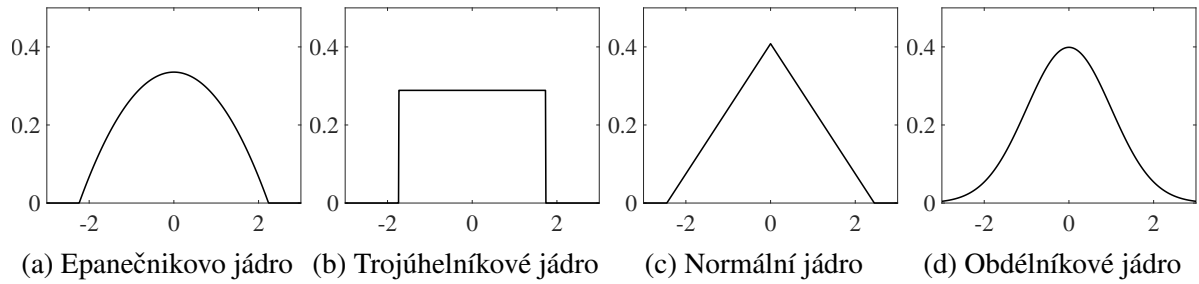
kde $C(K) = (\mu_2(K))^{2/5} (R(K))^{4/5}$. Jsou-li kromě $C(K)$ všechny zbylé proměnné konstantní, přejde minimalizace MISE na úlohu zvolení vhodného jádra. Omezme se na jádra, která jsou hustotami a také předpokládejme, že hodnota μ_2 je rovna jedné (pokud $\mu_2 \neq 1$, použití přeškálovaného jádra $\sqrt{\mu_2} K(\sqrt{\mu_2} x)$ neovlivní hodnotu $C(K)$). Minimalizace $C(K)$ potom dále přešla na úlohu minimalizace $\int K^2(x) dx$ s podmínkami $R(K) = \mu_2(K) = 1$. V jiném kontextu ukázali Hodges a Lehmann (viz [33]), že řešením je jádro

$$K_e(x) = \begin{cases} \frac{3}{4\sqrt{5}} \left(1 - \frac{1}{5}x^2\right), & -\sqrt{5} \leq x \leq \sqrt{5}, \\ 0, & \text{jinak,} \end{cases} \quad (5.10)$$

které se nazývá Epanečnikovo. Jeho tvar spolu s tvary dalších používaných jader nalezneme na obrázku 5.1 a jejich předpisy v tabulce 5.1 spolu s jejich efektivností v porovnání s Epanečnikovým jádrem. Efektivitu pro každé symetrické jádro K definujeme jako

$$\text{eff}(K) = (C(K_e)/C(K))^{5/4}. \quad (5.11)$$

Jednorozměrný případ odhadu hustoty pravděpodobnosti představený na začátku této kapitole můžeme zobecnit i do více rozměrů. Pro použití vícerozměrného jádrového odhadu však potřebujeme více vyhlazovacích parametrů, čímž je složitější jeho použití a v praxi se používá zejména pro dvourozměrný případ.

Obrázek 5.1: Tvary základních jádrových funkcí $K(x)$

Jádro	$\mathbf{K}(x)$	$\text{eff}(K)$
Epanečnikovo	$\begin{cases} \frac{3}{4\sqrt{5}} \left(1 - \frac{1}{5}x^2\right), & -\sqrt{5} \leq x \leq \sqrt{5} \\ 0, & \text{jinak} \end{cases}$	1
Trojúhelníkové	$\begin{cases} 1 - x , & x < 1, \\ 0, & \text{jinak} \end{cases}$	0.9859
Normální (Gaussovo)	$\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right)$	0.9512
Obdélníkové	$\begin{cases} \frac{1}{2}, & x < 1, \\ 0, & \text{jinak} \end{cases}$	0.9295

Tabulka 5.1: Přehled jádrových funkcí a jejich efektivity

Definice 5.0.2. (Vícerozměrný jádrový odhad hustoty pravděpodobnosti). Mějme $\mathbf{X}_1, \dots, \mathbf{X}_n$, kde $\mathbf{X}_i = (X_{i1}, \dots, X_{id})^\top$ je d -rozměrný vektor, náhodný výběr o velikosti n ze spojité náhodné veličiny $\mathbf{X} = (X_1, \dots, X_d)^\top$ s distribucí $f(\mathbf{x})$. Necht' dále jádro $K(\mathbf{x}) \geq 0$ splňuje

$$\int_{\mathbb{R}^d} \dots \int_{\mathbb{R}^d} K(\mathbf{x}) d\mathbf{x} = 1, \quad \int_{\mathbb{R}^d} \dots \int_{\mathbb{R}^d} \mathbf{x} K(\mathbf{x}) d\mathbf{x} = \mathbf{0}, \quad \int_{\mathbb{R}^d} \dots \int_{\mathbb{R}^d} \mathbf{x}\mathbf{x}^\top K(\mathbf{x}) d\mathbf{x} = \mu_2(K)\mathbf{I}, \quad (5.12)$$

kde \mathbf{I} je jednotková matice o rozměrech $d \times d$ a $\mu_2(K) > 0$ nezávisí na složce vektoru \mathbf{x} . Potom $\forall \mathbf{x} \in \mathbb{R}^d$ jádrový odhad d -rozměrné hustoty pravděpodobnosti definujeme jako

$$\hat{f}(\mathbf{x}; \mathbf{H}) = n^{-1} |\mathbf{H}|^{-1/2} \sum_{i=1}^n K(\mathbf{H}^{-1/2}(\mathbf{x} - \mathbf{X}_i)) = n^{-1} \sum_{i=1}^n K_{\mathbf{H}}(\mathbf{x} - \mathbf{X}_i), \quad (5.13)$$

kde \mathbf{H} je symetrická pozitivně definitní matice vyhlazovacích parametrů o rozměrech $d \times d$ a $K_{\mathbf{H}}(\mathbf{x}) = |\mathbf{H}|^{-1/2} K(\mathbf{H}^{-1/2}\mathbf{x})$ je škálované jádro. Pro speciální volbu matice $\mathbf{H} = h^2\mathbf{I}$, kde $h > 0$ a \mathbf{I} je jednotková matice o rozměrech $d \times d$, potom jádrový odhad lze psát ve tvaru

$$\hat{f}(\mathbf{x}; \mathbf{H}) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right). \quad (5.14)$$

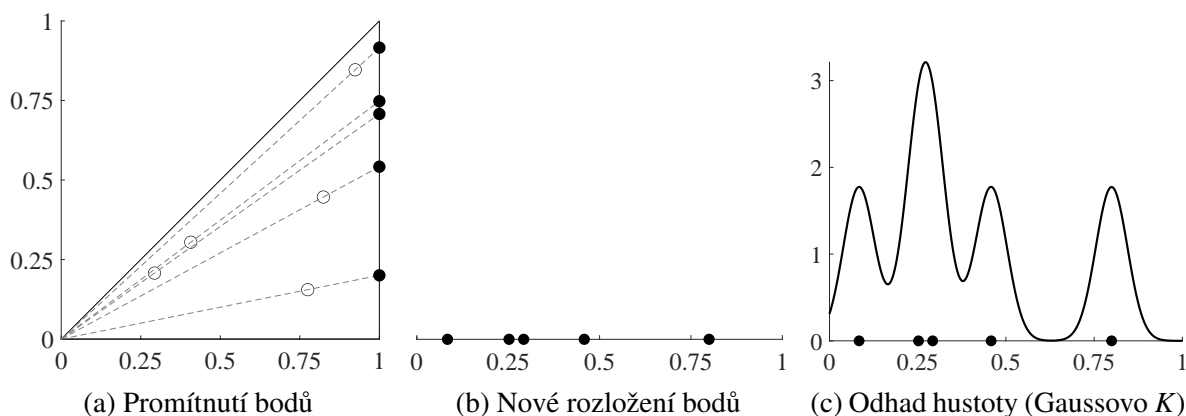
Jádro o více dimenzích K může být voleno jako součinnové jádro z jednorozměrných jader ve tvaru $K(\mathbf{x}) = \prod_{i=1}^d K(x_i)$, pokud je to pro danou aplikaci možné.

5.1 Aplikace jádrových odhadů na Preisachův trojúhelník

Aplikace jádrových odhadů na problematiku PM prostoru byla jedna z hlavních úloh práce [8], stejně jako porovnání jejich průběhů a vlastností na testovacích i reálných datech. V této části si proto představíme jednotlivé způsoby jádrových odhadů aplikovaných na Preisachův trojúhelník, které potom přímo použijeme jako nástroj pro výpočet indexů elasticity, resp. poškození.

Promítnutí PM prostoru na přepony

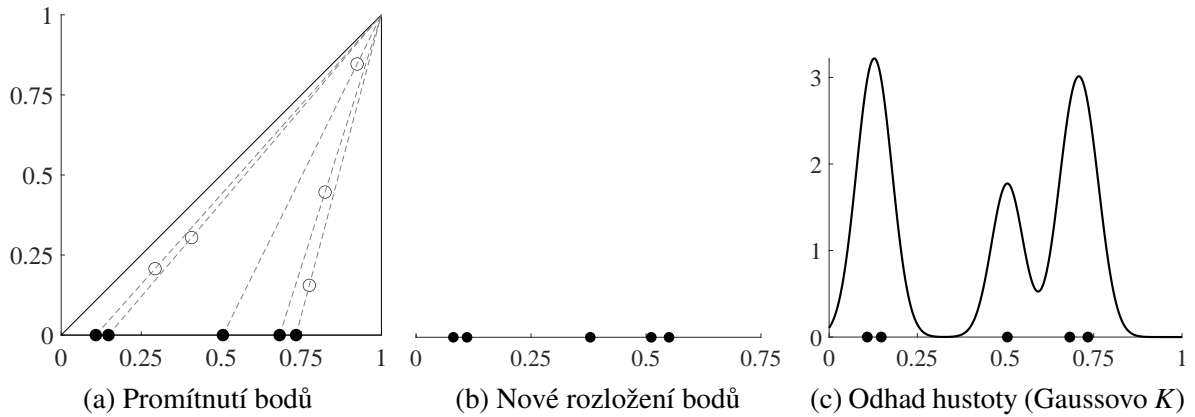
Prvním ze způsobů použijí jádrových odhadů hustoty je promítnutí jednotlivých hysterezních operátorů z Preisachova trojúhelníku na odvěsnu, ať už svislou (obr. 5.2a) nebo vodorovnou ((obr. 5.3a) prostřednictvím vrcholu trojúhelníku, který je protější vůči této odvěsně. Na takto nově rozložené body následně aplikujeme standardní jednorozměrný jádrový odhad představený dříve v této kapitole, který vždy probíhá směrem od přepony shora dolů, resp. zleva doprava, aby byl konzistentně vynášen průběh od dokonale elastických hysteronů k těm nejvíce poškozeným (obr. 5.2c a 5.3c). Dále je použita kombinace obou promítnutí na odvěsny (obr. 5.4a), kdy jádrový odhad následně probíhá shora dolů a plynule přechází zprava doleva (obr. 5.4b), přičemž tento odhad je dvakrát tak delší. Díky tomu zachycujeme detailnější strukturu PM prostoru pomocí dvou průmětů, aby nám neuniklo hromadění hysteronů podél jednoho z promítacích směrů.



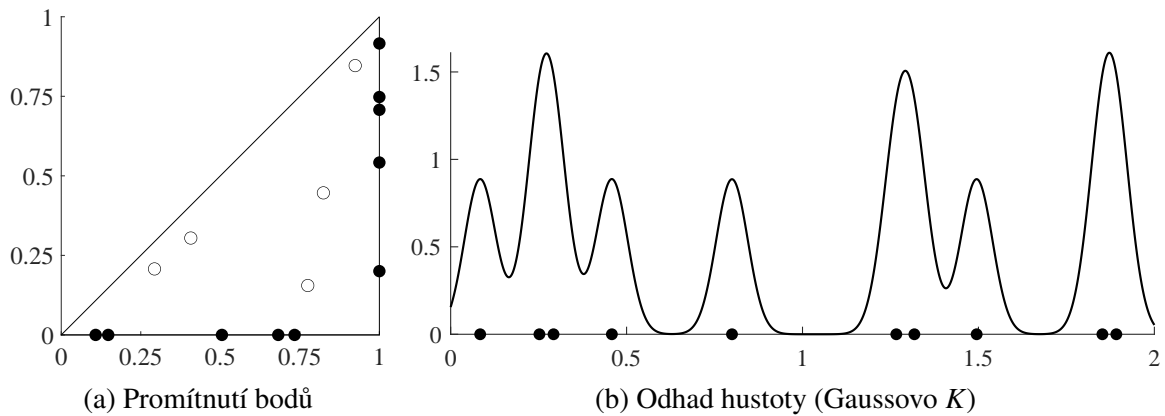
Obrázek 5.2: Jádrový odhad založený na promítnutí PM prostoru na svislou odvěsnu

Jádro s trojúhelníkovým nosičem

Další přístupy spočívají v přímém dvourozměrném jádrovém odhadu na PM prostor pomocí jader se specifickým tvarem nosiče a s daným způsobem projíždění PM prostoru. Jednak se jedná o odhad pomocí jádra s nosičem ve tvaru trojúhelníku. Tento nosič má vždy s Preisachovým trojúhelníkem totožný levý dolní vrchol a k němu protější strana leží na přímce, kterou opět udává svislá odvěsna Preisachova trojúhelníku. Strana nosiče na svislé ose má potom délku dvojnásobku vyhlazovacího parametru h (obr. 5.5a). Tento jádrový odhad také probíhá



Obrázek 5.3: Jádrový odhad založený na promítnutí PM prostoru na vodorovnou odvěsnu



Obrázek 5.4: Jádrový odhad založený na promítnutí PM prostoru na obě odvěsny

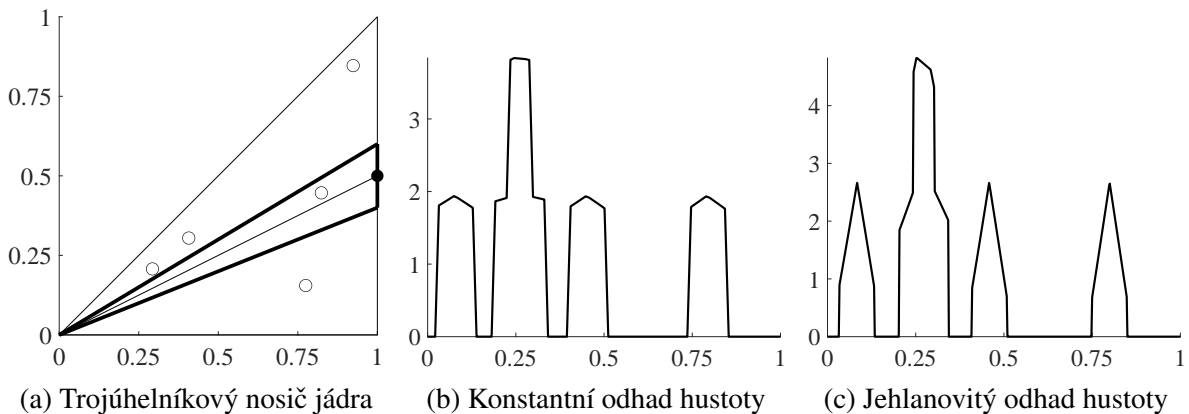
shora dolů, aby byla opět zachycena snižující se elasticita hysteronů v PM prostoru. Nad tímto nosičem je jednak vystavěno konstantní jádro, které zohledňuje množství hysteronů uvnitř nosiče, neboli odhad tvaru

$$\hat{f}(x; h) = \frac{1}{const} \sum_{i=1}^n \frac{1}{s(x; h)} \mathbb{I}_{s(x; h)}(X_i), \quad (5.15)$$

kde n je počet hysteronů, $s(x; h)$ je obsah nosiče jádra závislého na proměnné x a vyhlazovacím parametru h , $const$ je normovací konstanta, aby $\hat{f}(x; h)$ byla hustota pravděpodobnosti, a $\mathbb{I}_{s(x; h)}$ je charakteristická funkce pro nosič $s(x; h)$. Tento odhad vidíme na obrázku 5.5b. Druhé jádro pak tvoří jehlan nad tímto trojúhelníkovým nosičem s vrcholem v bodě x o výšce 1. Přejdeme-li k matematickému zápisu tohoto jádra, tak odhad hustoty je vyjádřen pomocí předpisu

$$\hat{f}(x; h) = \frac{1}{const} \sum_{i=1}^n \frac{1 - d_i(x)}{s(x; h)} \mathbb{I}_{s(x; h)}(X_i), \quad (5.16)$$

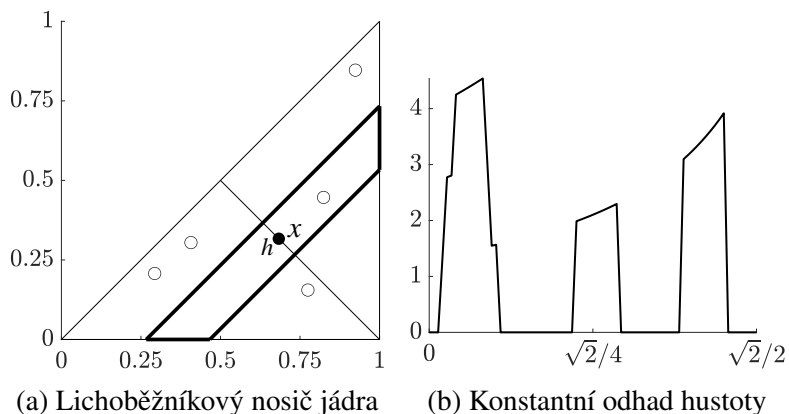
kde n je počet hysteronů, $s(x; h)$ je obsah nosiče jádra závislého na proměnné x a vyhlazovacím parametru h , $const$ je normovací konstanta, aby $\hat{f}(x; h)$ byla hustota pravděpodobnosti. Zde

Obrázek 5.5: Jádrový odhad založený na 2D jádru K s trojúhelníkovým nosičem

$\mathbb{I}_{s(x;h)}$ je charakteristická funkce pro nosič $s(x;h)$. Dále $d_i(x)$ udává kolmou vzdálenost hysteronu X_i od vodící přímky dané počátkem soustavy souřadnic a bodem x normovaná velikostí odvěsny PM trojúhelníku. Stejně jako pro případ promítnutí hysteronů byl použit i převrácený odhad, kde nosič má vrchol v pravém horním rohu a protější strana se pohybuje po vodorovné ose. A také kombinace těchto dvou směrů spojených za sebou do dvojnásobně dlouhého odhadu pro přesnější 2D informaci o rozložení hysteronů v PM prostoru.

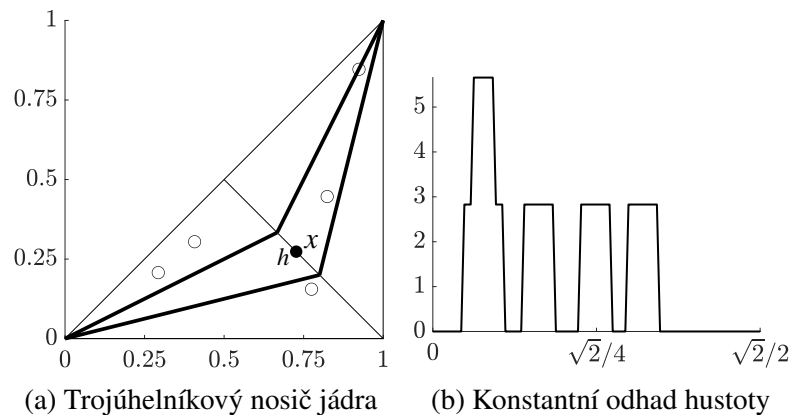
Jádro s lichoběžníkovým nosičem

Další metoda je opět založena na jádru se speciálním nosičem, a navíc zde odpadá problematika s volbou směru projíždění. Jedná se o použití jádra s nosičem ve tvaru lichoběžníku, který se posouvá od elastické diagonály tak, že jeho základny jsou s ní stále rovnoběžné (obr. 5.6a). Za „souřadnici“ tohoto odhadu bereme střed lichoběžníku, který se posouvá po těžnici PM trojúhelníku a je tudíž $\sqrt{2}/2$ -krát kratší než předchozí odhady, kde se „souřadnice“ pohybovala po odvěsně PM trojúhelníku. Nad tímto nosičem vystavíme opět konstantní jádro, které vede na formálně stejný zápis odhadu rovnicí (5.15) jako tomu bylo u jádra s trojúhelníkovým nosičem. Na obrázku 5.6b vidíme průběh tohoto jádrového odhadu.

Obrázek 5.6: Jádrový odhad založený na 2D jádru K s lichoběžníkovým nosičem

Jádro s křídlovým nosičem

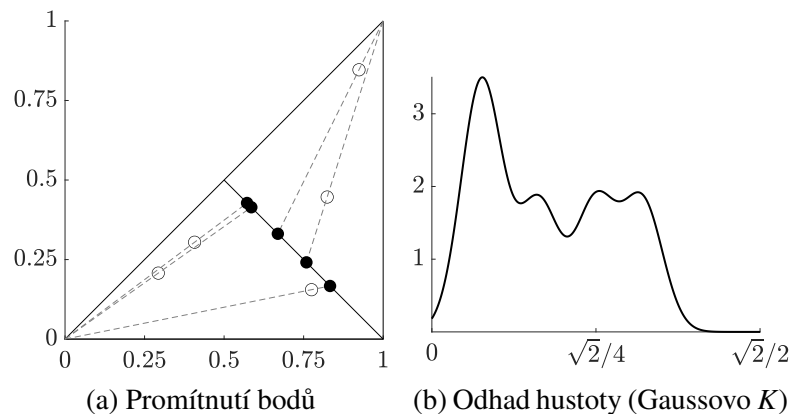
Další navržený odhad je založen na čtyřúhelníkovém nosiči, jehož dva vrcholy jsou pevně v počátku a v pravém horním rohu a zbylé dva se posouvají po těžnici vzdálené od sebe o vyhlazovací parametr h , jak můžeme vidět na obrázku 5.7a. Matematický zápis jádrového odhadu opět odpovídá rovnici (5.15).



Obrázek 5.7: Jádrový odhad založený na 2D jádru K s křídlovým nosičem

Promítnutí PM prostoru na výšku trojúhelníku

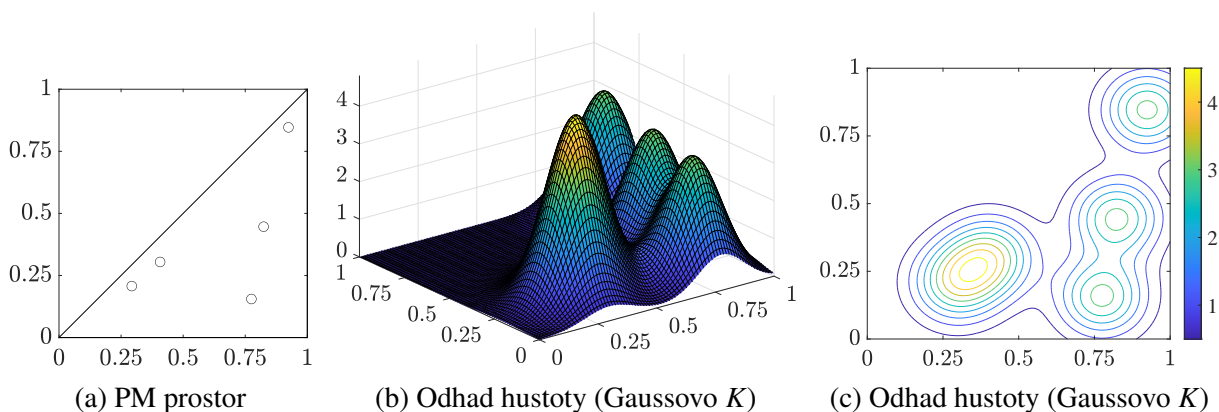
Vrátíme-li se k přístupu promítání hysteronů do jednodimenzionální úlohy, tak další způsob spočívá na promítání na výšku, která přísluší k přeponě PM trojúhelníku. A to tak, že tato výška rozdělí trojúhelník na dvě poloviny a body, co se nacházejí v polovině pod výškou, jsou promítány prostřednictvím levého spodního rohu PM trojúhelníku a body v horní polovině jsou na výšku promítány pomocí pravého horní rohu PM trojúhelníku (obr. 5.8a). Na tyto nově rozložené body použijeme standardní Gaussovské jádro (obr. 5.8b), kde jádrový odhad probíhá po výšce trojúhelníku opět od přepony k pravému dolnímu rohu, aby byl zachycen průběh od dokonale elastických bodů po ty nedokonale elastické.



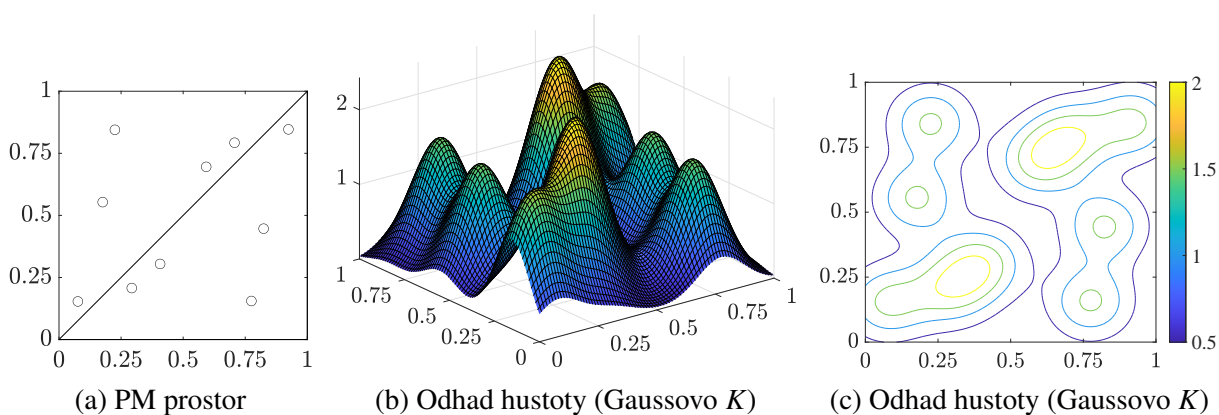
Obrázek 5.8: Jádrový odhad založený na promítnutí PM prostoru na výšku

2D jádrový odhad

Posledním představeným způsobem jádrového odhadu je použití dvourozměrného jádrového odhadu přímo na PM prostor, jak je možné vidět na obrázku 5.9b. Pro lepší vlastnosti při následném výpočtu divergencí je upraveno rozložení hysteronů tak, že je překlopen původní PM prostor pomocí bodové souměrnosti, konkrétně pomocí středu přepony tak, že celkem získáme dvakrát více bodů rozložených ve čtverci (obr. 5.10a).



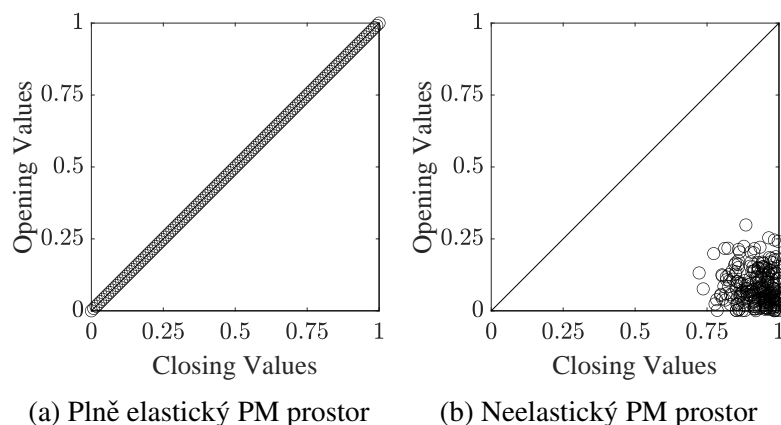
Obrázek 5.9: Jádrový odhad založený na přímém 2D jádrovém odhadu



Obrázek 5.10: Jádrový odhad založený na přímém 2D jádrovém odhadu

5.2 Index elasticity

Na základě testování jednotlivých metod jádrových odhadů v práci [8], se osvědčilo použití indexů elasticity IE , založených na vhodné divergenční míře (viz sekce 2.2) jádrového odhadu daného PM prostoru vůči dokonale elastickému PM prostoru, který má všechny hysterony na

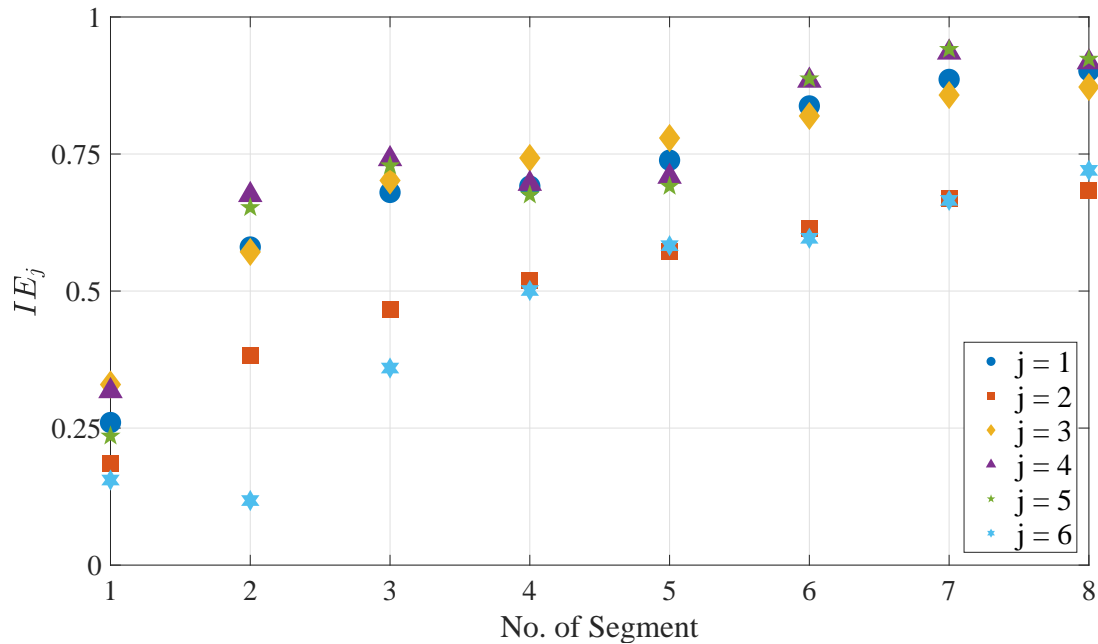


Obrázek 5.11: Referenční dokonale elastický a neelastický PM prostor

přeponě (obr. 5.11a). Nejlepších výsledků pro porovnání jádrových odhadů se dosáhlo použitím LeCamovy divergence, a proto ji použijeme i v této práci. Stejně tak za účelem normování hodnot navrženého indexu elasticity mezi 0 a 1, vydělíme danou divergenci nejvyšší empiricky dosaženou hodnotou divergence, získanou z jádrového odhadu PM prostoru na obr. 5.11b vůči jádrovému odhadu našeho referenčního dokonale elastického PM prostoru. Konkrétně pro jádrové odhady volme $IE_j = LC/LC_{max}$, kde LC je LeCamova divergence mezi odhadem zkoumaného PM prostoru vůči referenčnímu PM prostoru a LC_{max} je LeCamova divergence mezi odhady referenčního PM prostoru a normovacího PM prostoru, pro který nabývá divergence nejvyšší hodnoty. Pro jednotlivé indexy elasticity poté promítáme na obě odvěsny a používáme normální Gaussovské jádro ($j = 1$), jádro s trojúhelníkovým nosičem a hranolovitým jádrem přejíždějícím po obou odvěsnách ($j = 2$), jádro s lichoběžníkovým nosičem a konstantním jádrem ($j = 3$), jádro s křídlovým nosičem a konstantním jádrem ($j = 4$), dále promítáme na výšku PM trojúhelníku a používáme normální Gaussovské jádro ($j = 5$) a nakonec dvourozměrný jádrový odhad s překlopením PM prostoru ($j = 6$).

5.3 Vyhodnocení na reálných datech II

Na závěr této kapitoly aplikujeme navržené indexy elasticity na data z testování kovových zemětřesných tlumičů, která byla průběžně zpracovávána již v sekci 4.3. Výsledné hodnoty indexů IE_1 až IE_6 nalezneme na obrázku 5.12 a v tabulce 5.2. Nejvýraznější rozdíl mezi IE indexy nastává v segmentu #2, kde se jednotlivé metody výpočtu indexů elasticity nejvíce odlišují. Většinou indexu klesá citlivost pro nejpoškozenější segmenty, kde indexy IE_4 a IE_5 se dokonce již nechovají monotónně. Naopak, index IE_6 má horší citlivost pro první nejméně poškozené segmenty, ale na zbytku dat dosahuje dobrých rozlišovacích schopností. V indexu IE_4 také nastává pokles v segmentu #4. Dle obrázku 5.12 také vidíme, že počínaje segmentem #2 IE_5 kopíruje průběh IE_4 a počínaje segmentem #4 pak IE_6 kopíruje průběh IE_2 . Jako nejpoužitelnější se jeví pro naše data na celkovém průběhu indexy IE_1 a IE_2 a zbylé potom případně na mírně poškozených nebo naopak velmi poškozených segmentech.

Obrázek 5.12: Hodnoty indexů IE_1 až IE_6 pro data z testování zemětřesných tlumičů

	IE_1	IE_2	IE_3	IE_4	IE_5	IE_6
Seg #1	0.2601	0.1857	0.3295	0.3175	0.2356	0.1552
Seg #2	0.5803	0.3829	0.5714	0.6753	0.6525	0.1178
Seg #3	0.6799	0.4668	0.7017	0.7408	0.7288	0.3595
Seg #4	0.6914	0.5193	0.7428	0.6957	0.6755	0.5010
Seg #5	0.7387	0.5714	0.7792	0.7086	0.6913	0.5828
Seg #6	0.8377	0.6139	0.8194	0.8839	0.8873	0.5962
Seg #7	0.8861	0.6687	0.8576	0.9352	0.9411	0.6654
Seg #8	0.9018	0.6842	0.8723	0.9175	0.9232	0.7202

Tabulka 5.2: Hodnoty indexů IE_1 až IE_6 pro data z testování zemětřesných tlumičů

Závěr

V této práci jsme se zabývali Preisach-Mayergoyzovým (PM) popisem hystereze a vyhodnocení elastických vlastností materiálů pomocí tohoto popisu. Nejdříve jsme si proto představili samotnou hysterezi, její vlastnosti a dále PM model hystereze. Dále byly představeny pravděpodobnostní rozdělení pro popis tzv. Preisachova modelu, který charakterizuje hysterezní chování daného materiálu. Parametry těchto rozdělení byly optimalizovány pomocí heuristických algoritmů vzhledem k vhodné metrice mezi naměřenou hysterezní křivkou a hysterezní křivkou vzniklou z PM modelu.

Hlavním cílem této práce bylo použití metod strojového učení, kterými byly stromové metody (random forest, XGBoost, ...) a hluboké i konvoluční neuronové sítě pro klasifikaci vhodného rozdělení, případně směsi distribucí, a regresi zastoupení jejich jednotlivých komponent z naměřených hysterezních smyček. Také byly představeny nové metody pro konvoluční neuronové sítě, kdy byly z jednorozměrného vektoru prediktorů vyráběny dvourozměrné obrázky. Nejlepších výsledků dosáhl přístup pomocí konvolučních neuronových sítí, kde do obrázku byla vykreslena hysterezní smyčka se svým překlopením. Zatímco pro klasifikaci dominantního pravděpodobnostního rozdělení v PM prostoru byla konvoluční neuronová síť pouze nepatrně lepší než náhodný les, při regresi zastoupení jednotlivých komponent distribuční směsi byl rozdíl oproti základním metodám již výraznější. Kombinace všech představených metod do meta-modelu poté přinesla další mírné zlepšení, ale zároveň větší robustnost pro alternativní data. Takto natrénované metody výrazně urychlily hledání příslušného Preisachova prostoru, pro který jsou pak pomocí heuristických metod hledány pouze parametry distribucí v konkrétní směsi. Pro heuristické algoritmy se ukázalo ve většině případů dosažení lepších výsledků při použití optimalizace vůči L_2 metrice.

Na závěr byly představeny metody pro popis nalezených PM prostorů pomocí specifických jádrových odhadů hustoty pravděpodobnosti, které jsou upraveny přímo na problematiku rozložení hysteronů v PM trojúhelníku. Tyto jádrové odhady pak posloužily pro navržení indexu elasticity, resp. poškození materiálu. Nejslibnějších výsledků dosáhl index založený na promítání PM prostoru na odvěsny Preisachova trojúhelníku a následném použití jednorozměrného jádrového odhadu a dále potom index založený na jádrovém odhadu s trojúhelníkovým nosičem. Všechny zmíněné metody byly následně použity na reálná data testování kovových tlumičů, které pohlcují mechanické deformace za účelem ochrany budov před zemětřesením.

Literatura

- [1] I. D. Mayergoyz. *Mathematical models of hysteresis and their applications*. Academic Press - Elsevier, 2003.
- [2] G. Celeux, S. Frühwirth-Schnatter, Ch. Robert. *Model Selection for Mixture Models - Perspectives and Strategies*. Handbook of Mixture Analysis, CRC Press, 2018; arXiv:1812.09885.
- [3] G. Malsiner-Wallia, S. Frühwirth-Schnatter, B. Grün. *Identifying, Mixtures of Mixtures Using Bayesian Estimation*. Journal of computational and graphical statistics, Vol. 26(2), 285-295, 2017.
- [4] E. Sgouritsa, D. Janzing, J. Peters, B. Schoelkopf. *Identifying Finite Mixtures of Nonparametric Product Distributions and Causal Inference of Confounders*. Max Planck Institute for Intelligent Systems, Tübingen, Germany, 2013; arXiv:1309.6860.
- [5] O. Shojaei, M. Asadi, M. Finkelstein. *On Some Properties of α -Mixtures*. Metrika, Vol. 84, 1213-1240, 2021.
- [6] E. Suarez, A. Roldán, et. al. *Entropy analysis for damage quantification of hysteretic dampers used as seismic protection of buildings*. Applied Sciences, Vol. 7(6), 628, 2017.
- [7] E. Dolejš. *Odhadování struktury hysterezních materiálů prostřednictvím Preisach-Mayergoyzova prostoru*. Bakalářská práce, FJFI, ČVUT, 2020.
- [8] E. Dolejš. *Statistické a strojové rozpoznávání mechanických vlastností materiálů v PM prostoru*. Výzkumný úkol, FJFI, ČVUT, 2021.
- [9] J. A. Ewing. *On hysteresis in the relation of strain to stress*. Report of the British Association for the Advancement of Science, 502-504, 1889.
- [10] F. Preisach. *Über die magnetische Nachwirkung*. Zeitschrift für Physik, Vol. 94, 277–302, 1935.
- [11] M. A. Krasnoselskii, A. V. Pokrovskii. *Systems with hysteresis*. Berlin: Springer-Verlag, 1989.
- [12] M. Ismail, F. Ikhrouane, J. Rodellar. *The Hysteresis Bouc-Wen Model, a Survey*. Archives of Computational Methods in Engineering, Vol. 16(2), 161-188, 2009.

- [13] F. Šolc. *Modelování hystereze v mechatronických systémech*. Automatizace, Vol. 50(1), 18-24, 2007.
- [14] C. Abarkane. *Monitoring through acoustics techniques of RC earthquake-resistant structures subjected to dynamic loads on shaking table*. Disertační práce, Technical School of Building Engineering, University of Granada, 2019.
- [15] K. R. McCall, R. A. Guyer. *Equation of state and wave propagation in hysteretic nonlinear elastic materials*. Journal of Geophysical Research, Vol. 90(B12), 1994.
- [16] J.-P. Marin, K. Mengersen, C. P. Robert. *Bayesian Modelling and Inference on Mixtures of Distributions*. Handbook of Statistics, Vol. 25, 459-507, 2005.
- [17] I. Vajda. *Information-Theoretic Methods in Statistics*. Výzkumná zpráva, ÚTIA AV ČR, 1995.
- [18] R. V. Rao. *Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems*. International Journal of Industrial Engineering Computations, Vol. 7(1), 19-34, 2016.
- [19] N. N. Son, C. V. Kien, H. P. H. Anh. *Parameters identification of Bouc–Wen hysteresis model for piezoelectric actuators using hybrid adaptive differential evolution and Jaya algorithm*. Engineering Applications of Artificial Intelligence, Vol. 87, 2020.
- [20] R. Storn, K. Price. *Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces*. Journal of Global Optimization, Vol. 11(4), 341-359, 1997.
- [21] S. Kirkpatrick, C. D. Gelatt Jr., M. P. Vecchi. *Optimization by Simulated Annealing*. Science 220(4598), 671-680, 1983.
- [22] K. Hladká. *Optimalizace klasifikačních technik pro produkty srážek těžkých jader*. Výzkumný úkol, FJFI, ČVUT, 2020.
- [23] M. Grandini, E. Bagli, G. Visani. *Metrics for Multi-Class Classification: an Overview*. CRIF S.p.A, Bologna, Italy, 2020; arXiv:2008.05756.
- [24] L. Breiman. *Random forests*. Machine Learning 45, 5-32, 2001.
- [25] L. Breiman. *Bagging predictors*. Machine Learning 24, 123-140, 1996.
- [26] E. K. Ampomah, Z. Qin, G. Nyame. *Evaluation of Tree-Based Ensemble Machine Learning Models in Predicting Stock Price Direction of Movement*. Information 11(6), 332, 2020.
- [27] Y. Freund, R. E. Schapire. *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*. Journal of Computer and System Sciences 55(1), 119-139, 1997.

- [28] T. Chen, C. Guestrin. *XGBoost: A Scalable Tree Boosting System*. KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016; arXiv:1603.02754.
- [29] J. Berner, P. Grohs, G. Kutyniok, P. Petersen. *The Modern Mathematics of Deep Learning*. Chapter of the forthcoming book *Theory of Deep Learning* by Cambridge University Press, 2021; arXiv:2105.04026
- [30] S. Ruder. *An overview of gradient descent optimization algorithms*. Insight Centre for Data Analytics, NUI Galway, Aylie Ltd., Dublin, 2017; arXiv:1609.04747
- [31] K. I. Joy. *Bresenham's Algorithm* On-Line Computer Graphics Notes, Visualization and Graphics Research Group, Department of Computer Science, University of California, Davis, 2016.
- [32] B. W. Silverman. *Density estimation for statistics and data analysis*. New York: Chapman & Hall, 1998.
- [33] J. L. Hodges, E. L. Lehmann. *The efficiency of some non-parametric competitors of the t-test*. Annals of Mathematical Statistics, Vol 27, 324-335, 1956.