VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY

INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS

DIAGNOSTIKA STAVU STROJE/STROJNÍCH SOUČÁSTÍ POMOCI FUZZY MNOŽIN

DIAGNOSTICS OF THE STATE OF THE MACHINE/MACHINE COMPONENTS WITH THE HELP OF FUZZY SETS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — Bc. Roman Horák

AUTHOR

VEDOUCÍ PRÁCE — Ing. Tomáš Marada, Ph.D.

SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání diplomové práce

Ústav:	Ústav výrobních strojů, systémů a robotiky
Student:	**Bc. Roman Horák**
Studijní program:	Výrobní stroje, systémy a roboty
Studijní obor:	bez specializace
Vedoucí práce:	**Ing. Tomáš Marada, Ph.D.**
Akademický rok:	2022/23

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Diagnostika stavu stroje/strojních součástí pomoci fuzzy množin

Stručná charakteristika problematiky úkolu:

Fuzzy systémy vykazují schopnost predikce a jsou často využívány pro účely klasifikace. V případě klasifikace nám fuzzy systém poskytují nejenom informaci predikované příslušnosti ke třídě ale také míru, jakou k třídě náleží. Práce se bude zabývat tvorbou algoritmů pro diagnostiku stroje nebo strojních součástí pomocí metod umělé inteligence. Zdrojové datasety budou simulovaná data, reálné veřejné datasety a případně měření z praxe. Cílem bude otestovat a porovnat několik metod a vytvořit algoritmy pro možné nasazení v praxi. Programovacím prostředím bude Python nebo Matlab.

Cíle diplomové práce:

Rešerše v oblasti technické diagnostiky strojů/strojních součástí.
Popis použitelných metod.
Popsat a vytvořit algoritmy.
Porovnat výsledky z různých metod a různých datasetů.
Zdokumentovat vytvořené řešení.
Vlastní závěr a doporučení pro praxi.

Seznam doporučené literatury:

BLATA, Jan a Janusz JURASZEK. Metody technické diagnostiky: teorie a praxe = Metody diagnostyki technicznej : teorie a praktyka. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, 2013. ISBN 978-80-248-2997-5.

HELEBRANT, František. Technická diagnostika a spolehlivost - Provoz a údržba strojů. Ostrava: VŠB - Technická univerzita Ostrava, 2008. ISBN 978-80-248-1690-6.

VONDRÁK, Ivo. Umělá inteligence a neuronové sítě. 3. vyd. Ostrava: VŠB - Technická univerzita Ostrava, 2009. ISBN 978-80-248-1981-5.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2022/23

V Brně, dne

L. S.

doc. Ing. Petr Blecha, Ph.D. ředitel ústavu	doc. Ing. Jiří Hlinka, Ph.D. děkan fakulty

ABSTRAKT

Diplomová práce se zabývá využitím fuzzy logiky v oblasti technické diagnostiky. Práce je rozdělena na teoretickou a praktickou část. V teoretické části je popsána technická diagnostika, fuzzy logika a genetický algoritmus. Na teoretickou část navazuje část praktická, ve které je fuzzy logika odzkoušena na Iris datasetu a následně jsou nabyté znalosti aplikovány na technický dataset vyhodnocující poruchové stavy stroje. Na závěru kapitol obou datasetů jsou shrnuty a zhodnoceny výsledky. Poslední kapitola praktické části je věnována popisu vytvořených skriptů v softwaru Matlab 2022b. Součástí práce jsou přílohy, ve kterých jsou uloženy vytvořené FIS modely a napsané skripty.

ABSTRACT

Diploma thesis deals with the use of fuzzy logic in the field of technical diagnostics. The thesis is divided into theoretical and practical parts. The theoretical part describes technical diagnostics, fuzzy logic and genetic algorithm. The theoretical part is followed by a practical part in which fuzzy logic is tested on the Iris dataset and then the acquired knowledge is applied to a technical dataset evaluating machine fault conditions. At the end of the chapters of both datasets, the results are summarized and evaluated. The last chapter of the practical part is devoted to the description of the developed scripts in software Matlab 2022b. Part of the work are attachments in which the created FIS models and written scripts are stored.

KLÍČOVÁ SLOVA

Fuzzy logika, technická diagnostika, genetický algoritmus, Iris dataset, funkce příslušnosti, fitness funkce, Matlab, matice záměn

KEYWORDS

Fuzzy logic, technical diagnostics, genetic algorithm, Iris dataset, membership function, fitness function, Matlab, confusion matrix

BIBLIOGRAFICKÁ CITACE

HORÁK, Roman. *Diagnostika stavu stroje/strojních součástí pomoci fuzzy množin,* Brno, Vysoké učení technické v Brně, Fakulta strojního inženýrství. 2023, 71 s., Vedoucí diplomové práce Ing. Tomáš Marada, Ph.D.

PODĚKOVÁNÍ

Tímto bych chtěl poděkovat vedoucímu mé diplomové práce panu Ing. Tomáši Maradovi, Ph.D. za odborné rady, věcné připomínky a vedení. Dále děkuji svým rodičům za pět let trvající podporu při studiu.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Tomáše Marady, Ph.D. a s použitím literatury uvedené v seznamu.

V Brně dne 23.5.2023 ……

Roman Horák, Bc.

OBSAH

1 ÚVOD 15
2 TECHNICKÁ DIAGNOSTIKA 17
2.1 Diagnostické metody 17
2.1.1 Vibrodiagnostika 17
2.1.2 Tribodiagnostika 18
2.1.3 Termodiagnostika 18
2.1.4 Akustická diagnostika 18
2.1.5 Elektrodiagnostika 18
3 FUZZY LOGIKA 19
3.1 Fuzzy množiny 19
3.2 Fuzzifikace a funkce příslušnosti 19
3.3 Inferenční pravidla 21
3.4 Fuzzy modely 21
3.4.1 Mamdani 21
3.4.2 Sugeno 22
3.5 Deffuzifikace 23
3.5.1 Deffuzifikace Mamdani systému 23
3.5.2 Deffuzifikace Sugeno systému 24
4 GENETICKÝ ALGORITMUS 25
5 FUZZY LOGIKA A IRIS DATASET 27
5.1 Použitý software 27
5.2 Vykreslení parametrů kosatců do grafů 27
5.3 Nastavení systému 29
5.4 Nastavení funkcí příslušnosti 29
5.4.1 Volba tvarů a parametrů funkcí příslušnosti 30
5.5 Stanovení inferenčních pravidel 31
5.6 Vyhodnocení předpřipravených systémů 32
5.7 Trénink předpřipravených systémů 33
5.8 Konečný trénink systému 34
6 ROZBOR TECHNICKÉHO DATASETU 36
6.1 Informace o datasetu 36
6.1.1 Druhy simulovaných poruch 36
6.1.2 Použitá senzorika 37
6.1.3 Data v grafické podobě 37
6.2 Zpracování dat 38
6.3 Vykreslení dat 39
7 TVORBA A ZHODNOCENÍ FIS TECHNICKÉHO DATASETU 41
7.1 FIS systém s daty z druhého snímače 41
7.2 FIS systém s daty pouze prvního snímače 42
7.2.1 Trénink parametrů a pravidel s vyšším počtem generací a populací 43
7.2.2 Trénink parametrů a pravidel se sníženým počtem generací a populací 45
7.2.3 Trénink parametrů s vyšším počtem generací a populací 46
7.2.4 Trénink parametrů se sníženým počtem generací a populací 47
7.2.5 Trénink parametrů a pravidel pomocí K-fold křížové validace 48
7.3 Porovnání FIS systémů s klasifikačními metodami 50
8 POPIS POUŽITÝCH SKRIPTŮ 53
8.1 Skripty pro Iris dataset 53

8.1.1 Iris_grafy.mlx 53
8.1.2 Iris_evaluace.mlx 53
8.1.3 Iris_tunefis.mlx 53
8.2 Skripty pro EXsys-vibro dataset 54
8.2.1 EXsys_zpracovani_dat.mlx 54
8.2.2 EXsys_grafy.mlx 54
8.2.3 EXsys_tvorba_FIS.mlx 55
8.2.4 EXsys_evaluace.mlx 55
8.2.5 EXsys_trenink.mlx 56
8.2.6 EXsys_multi_trenink.mlx 56
8.2.7 EXsys_trenink_K_fold.mlx 56
8.2.8 fuzzyFitness.mlx 57
8.2.9 loadTabs.mlx 57
9 ZÁVĚR 59
10 SEZNAM POUŽITÝCH ZDROJŮ 61
11 SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK 63
11.1 Seznam zkratek 63
11.2 Seznam symbolů 63
11.3 Seznam tabulek 64
11.4 Seznam obrázků 64
12 SEZNAM PŘÍLOH 67
A VYTVOŘENÉ FIS SYSTÉMY 69
B SLOŽKA S DATY A SKRIPTY 71

1 ÚVOD

V posledních letech se stala technická diagnostika nedílnou součástí průmyslu. Cílem technické diagnostiky je předpovídat poruchy a změny stavů daných strojních zařízení či součástí, díky tomuto dochází ke zvyšování produktivity a s tím spojeného zvyšování zisku. Ke zjištění technického stavu stroje, či součásti je možno využít mnoho metod a tato práce se bude věnovat jedné z nich – fuzzy logice.

První kapitola teoretické části se zabývá popisem technické diagnostiky a jejich základních diagnostických metod. Jednou z těchto metod je vibrodiagnostika, která je nejčastěji využívaná diagnostická metoda, a právě ta je využita v praktické části práce. Další kapitola pojednává o fuzzy logice a popisuje její podstatné části k vytvoření fuzzy systému. Poslední kapitola teoretické části je věnována genetickému algoritmu, kde je stručně rozepsána jeho funkčnost.

Na teoretickou část navazuje část praktická, ve které je nejprve odzkoušeno sestavení fuzzy systémů na Iris datasetu, což je nejznámější dataset z oblasti strojového učení, a poté jsou získané vědomosti a výsledky aplikovány na technický dataset. Ve zmíněném technickém datasetu jsou data získávána ze dvojice akcelerometrů a celkem je sledováno pět stavů zařízení. Získaná data jsou nejprve zpracována, poté jsou určeny vhodné prediktory a následně vykresleny grafy. Ze zpracovaných dat je vytvořeno několik různých fuzzy systémů a všechny jsou poté zhodnoceny, a k závěru praktické části, i porovnány s klasifikačními metodami na stejném datasetu.

Po praktické části jsou popsány skripty, které byly napsány v softwaru Matlab 2022b a jsou využity pro tvorbu fuzzy systému Iris datasetu nebo technického datasetu s názvem EXsys-vibro.

2 TECHNICKÁ DIAGNOSTIKA

Technická diagnostika je specializovaný obor, který se zaměřuje na různé metody a postupy pro neinvazivní zjišťování technického stavu objektů. Tento obor poskytuje důležité a objektivní informace o provozním stavu strojů a jejich potřebách v oblasti údržby. V případě potenciální havárie může být diagnostikovaný stroj automaticky zastaven. Technická diagnostika se neustále modernizuje, využívá nové technologické prostředky a programové vybavení, stejně jako další měřící a monitorovací řídicí systémy. [1]

Diagnostika je prováděna za účelem: [2]

- bezpečnosti, spolehlivosti a efektivity stroje
- prodloužení životnosti stroje
- kontroly technického stavu a měření jednotlivých parametrů

2.1 Diagnostické metody

Diagnostické metody jsou procesy, při kterých se sbírají a analyzují údaje s cílem určit technický stav diagnostikovaného objektu. Metody lze obecně rozdělit následovně: [2]

- Subjektivní – jde o schopnost jednotlivých lidí a využití jejich smyslů k pozorování provozních projevů diagnostikovaného objektu a detekci odchylek od normálního stavu.
- Objektivní – je založena na přesném měření určité fyzikální veličiny, jejíž hodnota může sloužit jako indikátor technického stavu diagnostikovaného objektu.

Pro subjektivní diagnostiku lze využít několik smyslů: [2]

- Sluch – lze využít k detekci zvukových projevů diagnostikovaného objektu v slyšitelném rozsahu pomocí technického stetoskopu.
- Zrak – umožňuje sledovat vizuální projevy diagnostikovaného objektu, jako jsou unikátní provozní hmoty, barvy, drsnosti povrchu, změny vzhledu a přítomnost cizích těles. Může být podpořen použitím technického endoskopu nebo např. mikroskopu.
- Hmat – umožňuje detekovat chvění, vůli v uložení, teplotu a drsnost diagnostikovaného objektu.
- Čich – může být použit k detekci přítomnosti zapáchajících látek, přehřívajících izolací a třecích obložení.

Pro objektivní diagnostiku lze měřit a analyzovat: [2]

- Provozní parametry stroje (výkon, příkon, otáčky, rychlosti, tlaky aj.)
- Fyzikální veličiny a jejich průběhy (napětí, proud, průtok aj.)
- Kmitání strojů a jejich součástí (rychlost, zrychlení kmitů, amplitudu aj.)
- Produkty opotřebení v olejových náplních (množství a druh otěrových částic a nečistot, změna viskozity a chemické reakce, aj.)

2.1.1 Vibrodiagnostika

Jedná se o jednu z nejčastěji používaných metod pro diagnostiku technického stavu strojních zařízení. Je hojně využívána díky své schopnosti aplikace na různorodá strojní zařízení. Pro vyhodnocení stavu se využívá vibrační signál, který je následně zpracován a analyzován.

Měření a analýzu vibračního signálu provádíme pomocí rychlosti, zrychlení nebo výchylky vibrací [3].

2.1.2 Tribodiagnostika

Tribodiagnostika je založena na získávání informací z maziva strojních zařízení. Tato metoda se zaměřuje na dvě hlavní oblasti. První oblast se zabývá zjišťováním stavu degradace samotného maziva. Druhá oblast se zaměřuje na analýzu maziva s cílem určit celkové poškození a technický stav provozovaného zařízení. [3]

2.1.3 Termodiagnostika

Ve většině případů probíhá bezkontaktní měření povrchové teploty sledovaného objektu. Pro toto měření se používá termokamera, která dokáže zobrazit rozložení povrchové teploty snímaného objektu. Tento typ kamery snímá infračervené záření, ze kterého poté vypočítává povrchovou teplotu a následně vytváří viditelný obraz s rozložením povrchové teploty daného objektu. [2]

2.1.4 Akustická diagnostika

Tato metoda je založena na podobném principu jako vibrodiagnostika, na rozdíl od vibrodiagnostiky, akustická diagnostika sleduje projevy poruch na strojních zařízeních pomocí vyhodnocování akustických signálů. Hlukem se označuje jakýkoliv nežádoucí zvuk ve slyšitelném frekvenčním rozsahu mezi 20 Hz a 20 kHz. Pohybující se části strojů způsobují vibrace, které způsobují pohyb částic vzduchu a šíření akustického signálu, který je následně měřen, analyzován a vyhodnocen. [3]

2.1.5 Elektrodiagnostika

Elektrodiagnostika se zaměřuje na měření elektrických veličin (elektrického proudu, napětí, odporu apod.), sledování změn ve funkci elektrických zařízení a změny izolačních odporů elektrických vedení a objektů. Dělí se na diagnostiku silové elektrotechniky a řídicí elektroniky, kde v silové energetice se často sleduje stav izolace a odporů v instalaci. Elektrodiagnostika je neustále významně rozvíjená díky rychlému vývoji řídicí a výpočetní techniky. [1]

3 FUZZY LOGIKA

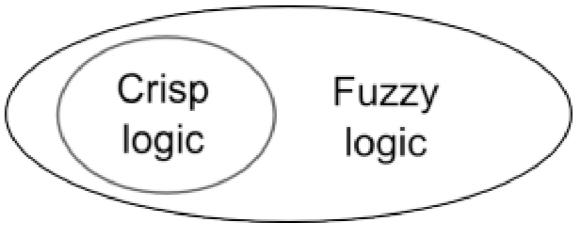
Teorie fuzzy množin byla představena Lotfi A. Zadehem v roce 1965 a od té doby se v této oblasti uskutečnilo mnoho vývojů. Nicméně, zdá se, že to byli právě Japonci, kteří plně využili potenciál fuzzy množin a tuto oblast komercializovali. Japonci získali více než 2000 patentů v oblastech průmyslu, spotřebních výrobcích, elektronických přístrojích, automobilech a dopravních prostředcích, monitorovacích systémech, a tak dále. [4]

Fuzzy logika se obecně používá (na rozdíl od binární logiky) k popisu stavů, které nemají přesně definované hranice. Pro konkrétní vstupní hodnotu mohou být dva nebo více stavů do jisté míry pravdivé. [7]

3.1 Fuzzy množiny

Fuzzy logika vychází z teorie fuzzy množin, která je podmnožinou klasické teorie množin, jak je ukazuje obr. 1. [6] Fuzzy (mlhavé) množiny jsou určeny k matematickému popisu vágních (nepřesných) pojmů. Na rozdíl od klasických množin, které přiřazují hodnotu 0 (false) nebo hodnotu 1 (true), jsou fuzzy množiny univerzálnější a umožňují přiřazovat hodnoty v rozmezí [0, 1]. Vyšší hodnoty znamenají větší míru příslušnosti k dané množině. Míra příslušnosti může být vyjádřena také v procentech. [8]

Fuzzy množiny jsou způsob, jakým lze určit, na kolik daný objekt vyhovuje vágnímu popisu. Uvažujme například větu „Adam je vysoký". Je tato věta pravdivá, pokud je Adam vysoký 180 cm? Velká většina lidí by raději odpověděla ve stylu „středně vysoký“ nebo „poměrně vysoký“, namísto odpovědi ano/ne – jak je tomu v binární logice. Teorie fuzzy množin považuje „vysoký“ za predikát a říká, že hodnota „vysoký“ je číslo v rozmezí 0 a 1. [5]



Obr. 1 Klasická teorie množin je součástí teorie fuzzy množin [6]

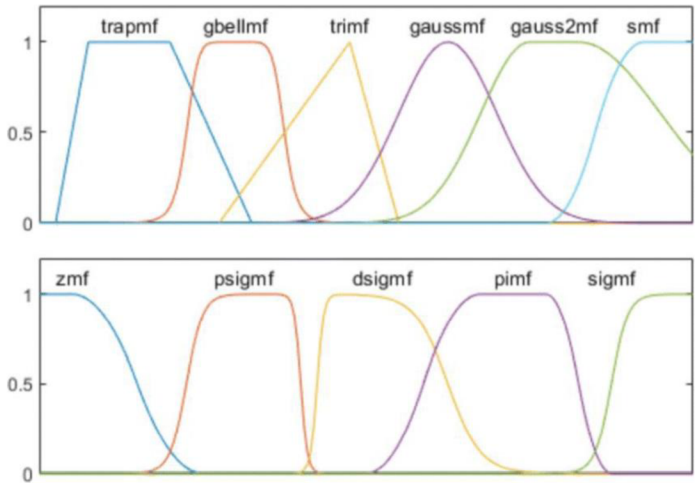
3.2 Fuzzifikace a funkce příslušnosti

Proces fuzzifikace spočívá v převedení jedné vstupní proměnné na několik fuzzy proměnných, přičemž počet těchto proměnných odpovídá počtu definovaných fuzzy množin pro daný vstup v systému. Každá z těchto fuzzy proměnných představuje míru příslušnosti dané vstupní hodnoty k příslušné množině. Tímto způsobem je určena pravděpodobnost, s jakou vstupní proměnná spadá do jednotlivých předem definovaných fuzzy množin. [7]

Na obr. 2 lze vidět tvary funkcí příslušnosti, které je možné použít v softwaru Matlab. V tab. 1 níže jsou pro tyto druhy uvedeny názvy, počty parametrů a rovnice výpočtů.

Tab. 1 Popis a vzorce tvarů funkcí příslušnosti z obr. 2 [10]

Označení	Název	Počet parametrů	Rovnice výpočtu
trapmf	Trapezoidal	4	$f(x;a,b,c,d) = \\ = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right)$
gbellmf	Generalized bell	3	$f(x;a,b,c) = \frac{1}{1+\left\|\frac{x-c}{a}\right\|^{2b}}$
trimf	Triangular	3	$f(x;a,b,c) = \\ = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$
gaussmf	Gaussian	2	$f(x;\sigma,c) = e^{\frac{-(x-c)^2}{2\sigma^2}}$
gauss2mf	Two-sided Gaussian	4	$f(x;\sigma_1,c_1) = e^{\frac{-(x-c_1)^2}{2\sigma_1^2}}$ $f(x;\sigma_2,c_2) = 1 - e^{\frac{-(x-c_2)^2}{2\sigma_2^2}}$
smf	S-shaped	2	$f(x;a,b) = \\ = \begin{cases} 0 & x \le a \\ 2\left(\frac{x-a}{b-a}\right)^2 & a \le x \le \frac{a+b}{2} \\ 1-2\left(\frac{x-b}{b-a}\right)^2 & \frac{a+b}{2} \le x \le b \\ 1 & x \ge b \end{cases}$
zmf	Z-shaped	2	$f(x;a,b) = \\ = \begin{cases} 1 & x \le a \\ 2\left(\frac{x-a}{b-a}\right)^2 & a \le x \le \frac{a+b}{2} \\ 1-2\left(\frac{x-b}{b-a}\right)^2 & \frac{a+b}{2} \le x \le b \\ 0 & x \ge b \end{cases}$
psigmf	Pi-shaped	4	$f(x;a_1,c_1,a_2,c_2) = \\ \frac{1}{1+e^{-a_1(x-c_1)}} \cdot \frac{1}{1+e^{-a_2(x-c_2)}}$
dsigmf	Difference of sigmoids	4	$f(x;a_1,c_1,a_2,c_2) = \\ \frac{1}{1+e^{-a_1(x-c_1)}} - \frac{1}{1+e^{-a_2(x-c_2)}}$
pimf	Product of sigmoids	4	$f(x;a,b,c,d) = \\ = \begin{cases} levá\ strana & smf(x;a,b) \\ pravá\ strana & zmf(x;c,d) \end{cases}$
sigmf	Sigmoid	2	$f(x;a_k,c_k) = \frac{1}{1+e^{-a_k(x-c_k)}}$



Obr. 2 Tvary funkcí příslušnosti v Matlabu [9]

3.3 Inferenční pravidla

Výhoda fuzzy logiky spočívá v možnosti stanovení lingvistických pravidel, která jsou lidskému uvažování blízká. [6]

Příklad pravidla v prostředí Matlabu:

If x is malé AND y is velké THEN z is střední

Kde „x“ a „y“ jsou vstupní proměnné a „malé“, „velké“ a „střední“ jsou funkce příslušnosti definované na jednotlivých proměnných. Výstupní proměnná je „z“. Operátor „AND“ se používá ke spojení jednotlivých pravidel (dále je počítáno s minimální hodnotou – sjednocení), další možný operátor je „OR“ (dále je počítáno s maximální hodnotou – průnik). [12]

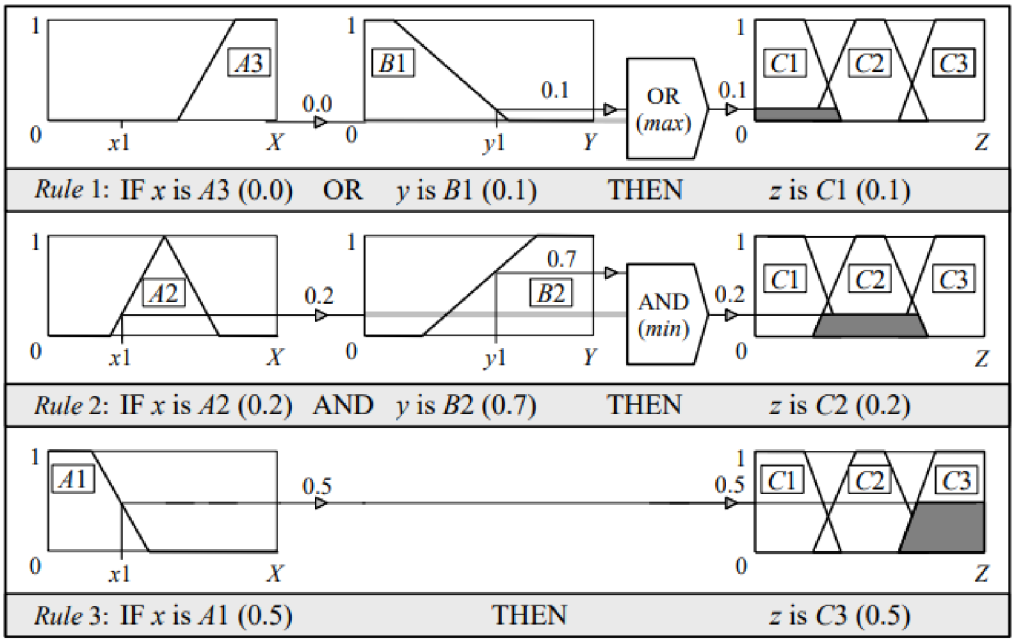
3.4 Fuzzy modely

Existují dva hlavní druhy Fuzzy modelů: Mamdani a Sugeno. Mamdani patří mezi první systémy založené na fuzzy logice. [11] Nejzásadnějším rozdílem, mezi danými modely, je způsob generování výstupu z fuzzy vstupů. Zatímco Mamdani používá techniku defuzzifikace fuzzy výstupu (používá funkce příslušnosti na výstupu), Sugeno používá k výpočtu výstupu vážený průměr (nepoužívá funkce příslušnosti na výstupu). Díky tomuto se Sugeno model stává výpočetně efektivnějším, ale přichází o interpretovatelnost výstupů. [14]

V následujících podkapitolách jsou uvedeny jednoduché příklady pro oba typy modelů.

3.4.1 Mamdani

Mandani proces se skládá se ze čtyř částí: fuzzifikace, vyhodnocení pravidel, agregace výstupů pravidel a deffuzifikace. [12] Na obr. 3 níže lze vidět příklad Mamdaniho metody.

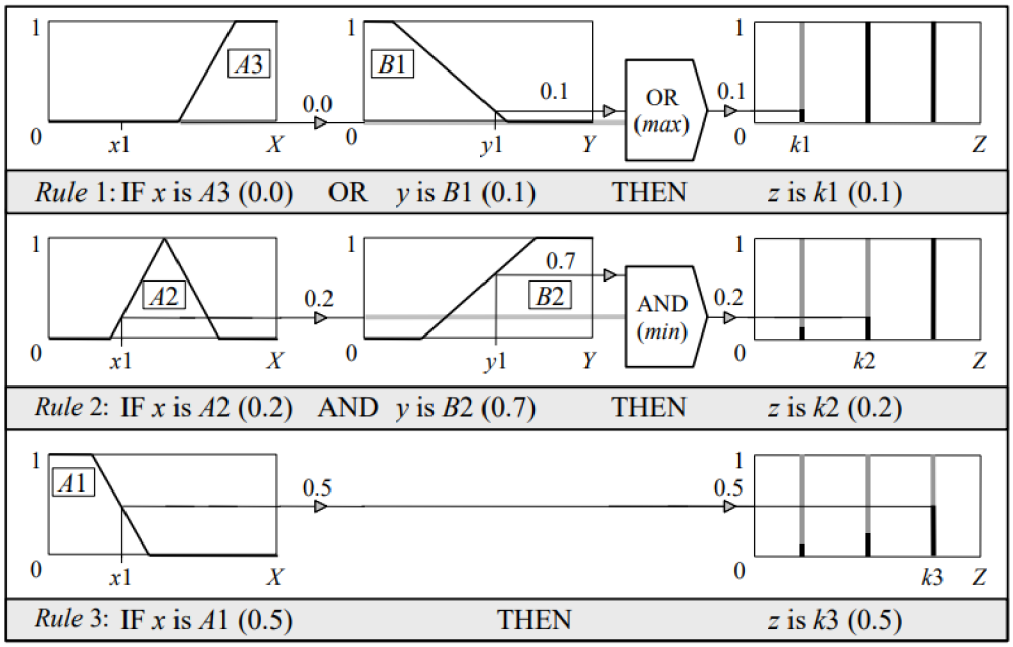


Obr. 3 Mamdani metoda [12]

Výstupem každého pravidla je fuzzy množina odvozená z výstupní funkce příslušnosti a fuzzy metody. Tyto výstupní fuzzy množiny jsou spojeny do jedné fuzzy množiny pomocí agregačního operátoru (AND nebo OR). Poté se pro výpočet konečné výstupní hodnoty defuzzifikuje výstupní fuzzy množina pomocí jedné z metod popsaných v části defuzzifikace (viz kap. 3.5). [13]

3.4.2 Sugeno

Sugeno proces je velmi podobný procesu Mamdani, rozdílný je pouze přístup k počítání výstupů.



Obr. 4 Sugeno metoda [12]

Na obr. 4 lze vidět, že každé pravidlo má vlastní výstup a celkový výstup systému je počítán pomocí váženého průměru. Tím pádem vzniká ušetřený čas, který je vyžadován v Mamdaniho modelu. [15]

Diagram fungování pravidel v Sugeno systému:

Každé pravidlo generuje dvě hodnoty (jak lze vidět na obr. 5): [13]

1) z_i – výstup pravidel, může být buď konstanta nebo lineární funkce.

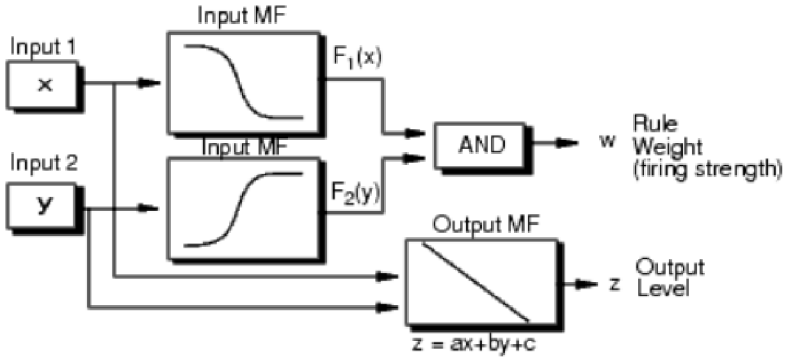
$$z_i = a_i x + b_i y + c_i \tag{1}$$

Kde x a y jsou vstupní hodnoty, a_i, b_i *a* c_i jsou konstantní koeficienty. Pro Sugeno systém nulového řádu je z_i konstanta ($a = b = 0$).

2) w_i – Hodnota odvozená z antecedentu pravidla, velikost hodnoty závisí na podmínkách daného pravidla.

$$w_i = \text{AndMethod}(F_1(\text{x}), F_2(\text{y})) \tag{2}$$

Kde F_1 *a* F_2 jsou funkce příslušnosti pro dané vstupní hodnoty.



Obr. 5 Diagram pravidel Sugeno systému [13]

3.5 Deffuzifikace

Účelem defuzzifikace je převést všechny závěry získané inferenčním mechanismem, který je vyjádřen ve fuzzy množinách, na konkrétní číselnou hodnotu. [7] Deffuzifikace Mamdani a Sugeno systému probíhá rozdílnými způsoby, ty jsou popsány v následujících podkapitolách.

3.5.1 Deffuzifikace Mamdani systému

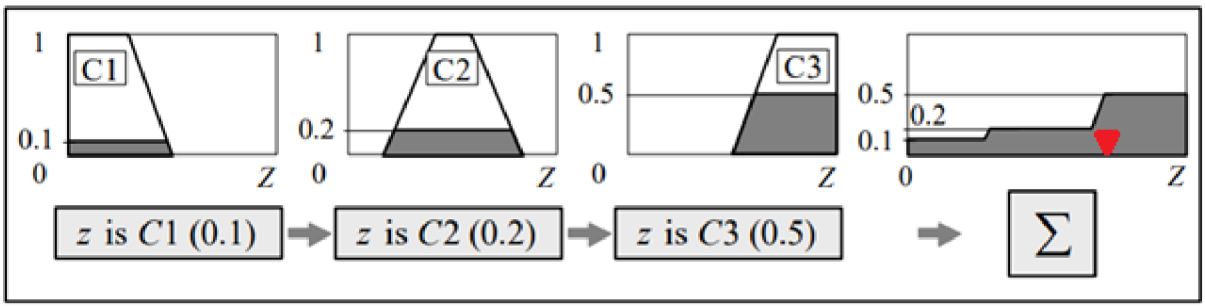
Díky Fuzzy Logic Toolboxu v Matlabu je umožněno použít až pět metod pro výpočet konkrétního čísla z fuzzy množiny: Centroid, Bisector, Middle of maximum, Smallest of maximum a Largest of maximum. Nejpoužívanější z nich je metoda Centroid, proto bude v práci popsána pouze tato metoda. [13]

Centroid

Centroidní defuzzifikace počítá těžiště bodu (centroidu) fuzzy množiny z plochy funkce příslušnosti vzhledem k ose x. [28]

$$Centroid = \frac{\sum_i u(x_i) \cdot x_i}{\sum_i u(x_i)} \tag{3}$$

Kde $u(x_i)$ je hodnota příslušnosti bodu x_i k diskurzu.



Obr. 6 Deffuzifikace Mamdani systému [12]

Z obr. 6 výše, který navazuje na obr. 3 (kap. 3.4.1) lze vidět spojení fuzzy množin z jednotlivých inferenčních pravidel s přibližným naznačením centroidu.

3.5.2 Deffuzifikace Sugeno systému

Proces defuzzifikace pro Sugenův systém je ve srovnání s Mamdaniho systémem výpočetně efektivnější, protože používá vážený průměr několika bodů, nikoli výpočet centroidu, či jiné metody, dvourozměrné oblasti.

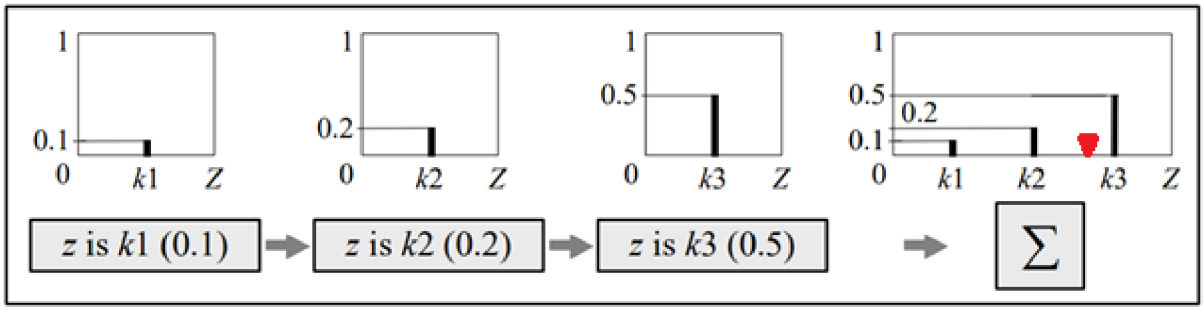
Každé pravidlo si lze představit jako definici polohy pohybujícího se prvku. To znamená, že výstupní hodnoty se mohou lineárně pohybovat ve výstupní části v závislosti na vstupních hodnotách. Hodnota pravidla pak určuje velikost prvku. [13]

Výpočet váženého průměru systému

Výstupem každého pravidla je vážený průměr, který je součinem w_i a z_i. [13]

$$WA = \frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i} \tag{4}$$

Kde N je počet pravidel, w_i a z_i jsou definovány v části o Sugeno metodě (kap. 6.4.2).



Obr. 7 Deffuzifikace Sugeno systému [12]

Z obr. 7 výše, který navazuje na obr. 4 (kap. 3.4.2) lze vidět spojení jednotlivých prvků inferenčních pravidel s přibližným označením váženého průměru.

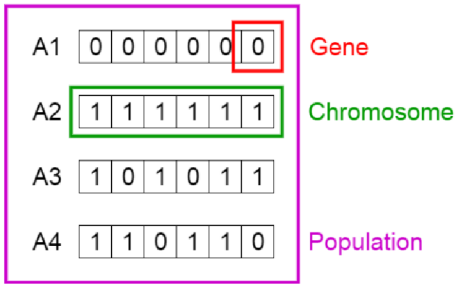
4 GENETICKÝ ALGORITMUS

U zrodu genetických algoritmů stál v 60. letech dvacátého století John Holland. Jeho myšlenkou bylo použít evoluční principy, založené na metodách optimalizace funkcí a umělé inteligenci, pro hledání řešení dané úlohy. [18]

Genetický algoritmus je počítačová simulace, ve které jsou jedinci náhodně vybráni, kombinováni, mutováni a následně pak odstraněni, nebo ponecháni v populaci na základě jejich kvality nebo vhodnosti (fitness hodnoty). Tento princip je inspirován přírodou – silnější jedinec vyhrává [20].

Základními pojmy v oblasti genetických algoritmů (schematicky zobrazené na obr. 8), které přenášejí informace a hodnotí vhodnosti, jsou: [18]

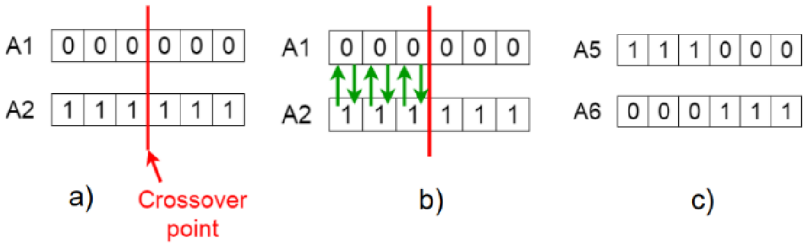
- Chromozóm – řetězec informací, který v sobě nese vlastnosti a chování každého jedince. Nejčastěji jde o řetězec nul a jedniček, kterým je zakódována pozice jedince v prostoru možných řešení.
- Gen – jednotlivé části chromozómů.
- Populace – skupina jedinců popsána svými chromozómy v rámci jedné generace.
- Fitness hodnota – číselná hodnota, která udává vhodnost každého jedince. Čím je hodnota nižší, tím je jedinec vhodnější.



Obr. 8 Schéma základních pojmů genetických algoritmů [21]

Algoritmus začíná pracovat s výchozí, náhodně zvolenou populací, kterou postupně zdokonaluje. Práce je ukončena, jakmile je dosaženo požadovaného minima fitness funkce, vyčerpání maximálního času nebo počtu generací. [19] Operace, které vedou k vytvoření nové populace jsou: [18]

- Selekce – v této fázi jsou vybráni nejschopnější jedinci a ti předávají své geny další generaci. Metod, jak tento krok provést je několik, nicméně ve výchozím nastavení Matlabu je použitá metoda ořezávání. [23] V této metodě jsou jedinci seřazeni dle fitness hodnoty, poté dle libovolného parametru rozděleni na dvě poloviny a z poloviny s vyššími hodnotami jsou náhodně či deterministicky vybráni rodiče.
- Křížení – po selekci následuje křížení, kde, stejně jako v přírodě, dochází k výměně části genetického kódu, tedy chromozómu. Pro každý pár je náhodně vybrán bod křížení, jak je zobrazeno na obrázku níže (obr. 9). V části a) je naznačen bod křížení, v části b) dochází k výměně genů mezi rodiči a v části c) jsou do populace přidáni další jedinci. [21]



Obr. 9 Schéma křížení genetického algoritmu [21]

- Mutace – jedná se o poslední operaci genetického algoritmu. U některých jedinců z nové generace jsou změněny náhodně změněny hodnoty z 0 na 1 a obráceně. Díky tomu se může v dané generaci objevit vlastnost, kterou žádný jedinec doposud neměl a nemohl ji tedy předat dále. Příklad mutace je uveden na obr. 10 níže.

Obr. 10 Příklad mutace genetického algoritmu

Existuje mnoho dalších druhů evolučních algoritmů, které fungují na podobném principu, jako genetický algoritmus (při tvorbě nových potomků zůstávají ti nejlepší). V Matlabu je možno při trénování systému použít další tři algoritmy:

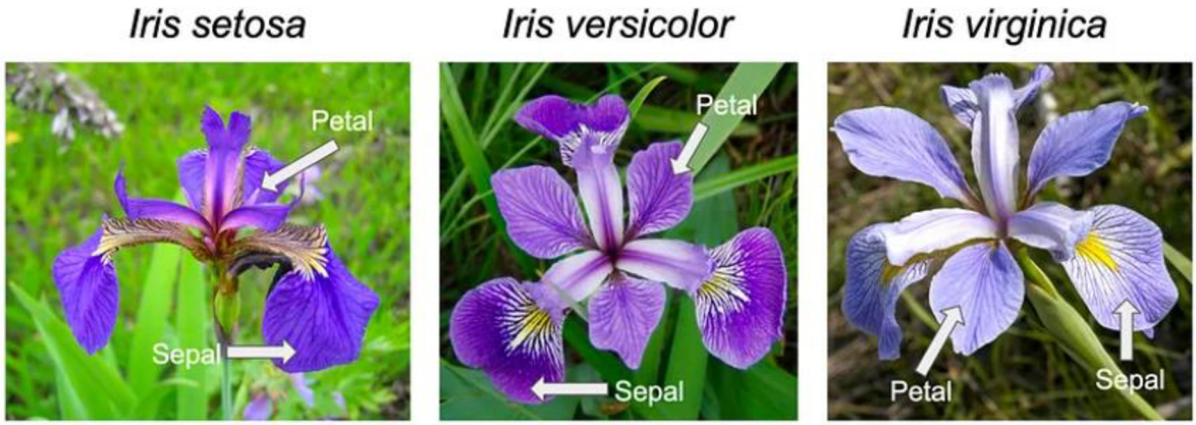
- Particle swarm – Optimalizace rojem částic.
- Pattern search – Vyhledávání vzoru.
- Simulated annealing algorithm – Simulované žíhání.

Nicméně v této práci bude použit pouze genetický algoritmus, jelikož dle literatury [24] dosahoval nejlepších výsledků.

5 FUZZY LOGIKA A IRIS DATASET

Tato část práce se věnuje rozebrání fuzzy logiky na nejznámějším datasetu z oblasti strojového učení – Iris datasetu, také známý jako Fisherův Iris dataset. Tento soubor dat proslavil Ronald Fisher v roce 1936, ve svém článku „The use of multiple measurements in taxonomic problems“ jako příklad lineární diskriminační analýzy.

Dataset obsahuje 50 vzorků od tří druhů kosatců – Iris Setosa, Iris Versicolor a Iris Virginica (viz obr. 11 níže) – celkem tedy 150 vzorků. U každého druhu byly měřeny délky a šířky okvětních a kališních lístků květin (sepal – kališní lístek, petal – okvětní lístek). [16]



Obr. 11 Fotografie Iris druhů [16]

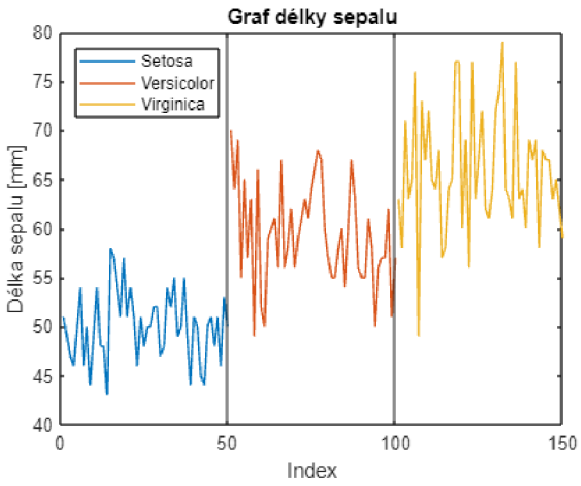
5.1 Použitý software

K vykreslování grafů a jednotlivým výpočtovým úkonům je použit software Matlab, konkrétně ve verzi R2022b, jednotlivé skripty jsou tvořeny v zabudovaném live editoru. Výhoda použití Matlabu spočívá v možnosti doinstalování Fuzzy Logic Toolboxu, ve kterém je zabudován Fuzzy Logic Designer, ten umožňuje navrhovat a simulovat fuzzy systémy v uživatelsky přívětivém prostředí.

5.2 Vykreslení parametrů kosatců do grafů

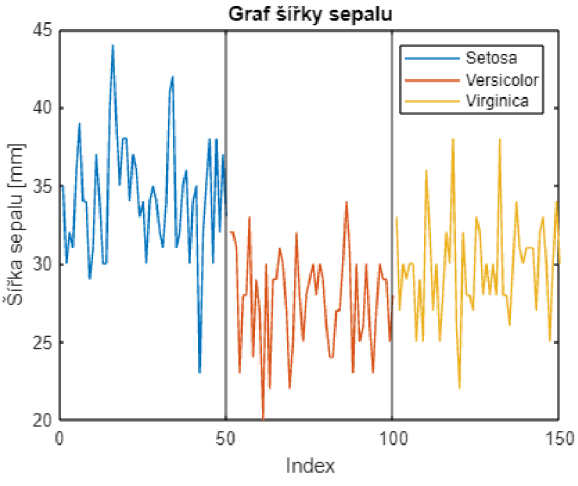
Pro snadnost orientace v Iris datasetu a zpřehlednění jednotlivých dat jsou vykresleny čtyři grafy. Na obr. 12a je vykreslen graf, na kterém jsou zobrazeny hodnoty délkových rozměrů sepalu (kališního lístku) pro všechny Iris druhy. Na pravé straně (obr. 12b) jsou zobrazeny šíře sepalů. Na obr. 13a a obr. 13b jsou zobrazeny grafy vykreslující rozměry délek a šířek, tentokrát pro petal (okvětní lístek).

Pod každým grafem lze vidět tabulku s minimálními, maximálními a průměrnými hodnotami jednotlivých lístků pro každý druh. Tyto hodnoty jsou využívány při nastavování parametrů jednotlivých funkcí příslušnosti, popsaných v následující podkapitole 5.4.1.



	Setosa	Versicolor	Virginica
Max. hodnota	58	70	79
Min. hodnota	43	49	49
Prům. hodnota	50.06	59.36	65.88

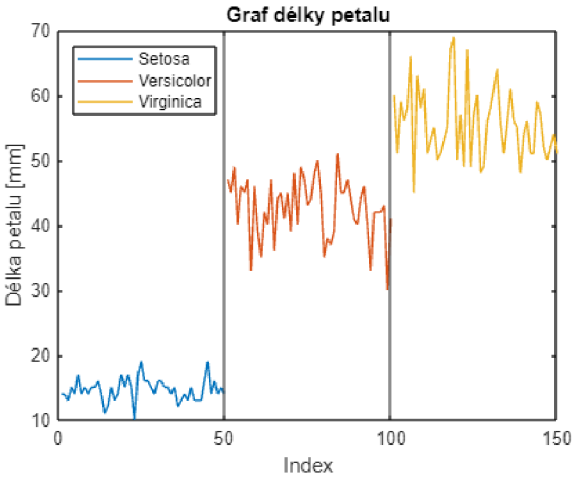
a)



	Setosa	Versicolor	Virginica
Max. hodnota	44	34	38
Min. hodnota	23	20	22
Prům. hodnota	34.28	27.7	29.74

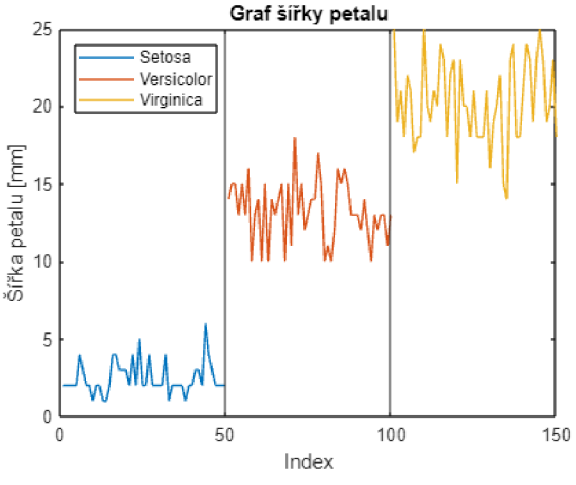
b)

Obr. 12 Grafy a tabulky: a) délek sepalu, b) šířek sepalu



	Setosa	Versicolor	Virginica
Max. hodnota	19	51	69
Min. hodnota	10	30	45
Prům. hodnota	14.62	42.6	55.52

a)



	Setosa	Versicolor	Virginica
Max. hodnota	6	18	25
Min. hodnota	1	10	14
Prům. hodnota	2.46	13.26	20.26

b)

Obr. 13 Grafy a tabulky: a) délek petalu, b) šířek petalu

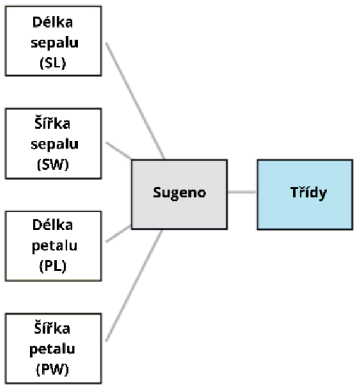
Z vykreslených grafů na obr. 12 a obr. 13 je možno již předem usoudit, že by neměl být problém v rozeznání Setosy od Versicolor nebo Virginicy, jelikož délky a šířky petalů jsou o poznání nižší. Lze tedy pro příklad říct, že pokud délky petalu budou nižší než 20 mm a šířky petalu nižší než 7 mm, tak se bude jednat o Iris druh Setosa. Hlavní problém, se kterým si bude muset daný model poradit je rozeznávání druhů Versicolor a Virginica.

5.3 Nastavení systému

Systém je potřeba předpřipravit pro následné trénovaní pomocí genetického modelu a fitness funkce (více v kapitole 5.7). Bude se tedy jednat o základní nastavení a nebude se zacházet příliš do hloubky.

Iris dataset obsahuje čtyři parametry (délky a šířky sepalu a petalu) a druh, tak se zde nabízí možnost zvolení systému se čtyřmi vstupy pro jednotlivé parametry a jeden výstup (schéma systému je zobrazeno na obr. 14). Ve výstupu bude pomocí váženého určen druh.

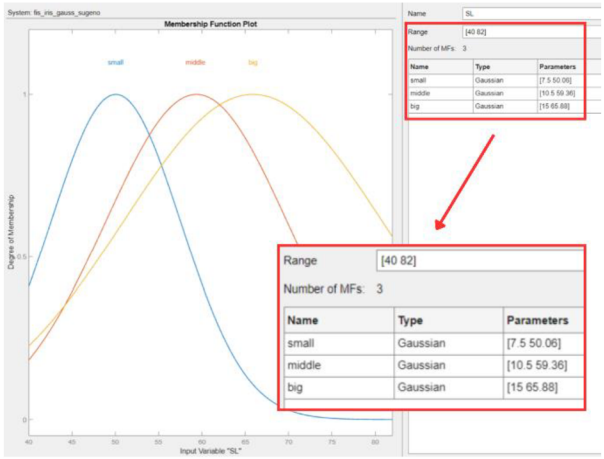
Model systému byl vybrán Sugeno, jelikož zde není zapotřebí interpretovat výstupy a má rychlejší vyhodnocování (viz kapitola 3.4.2). Výstupem systému budou tedy tři číselné hodnoty, kde každému druhu bude přiřazeno číslo (1 – Setosa, 2 – Versicolor, 3 – Virginica).



Obr. 14 Znázornění Fuzzy Iris systému

5.4 Nastavení funkcí příslušnosti

Pro každý ze čtyř vstupů, který byl v předchozí kapitole definován, je potřeba určit počet, tvar (typ), rozsah a parametry funkcí příslušnosti. Jelikož má systém tři výstupy, tak se pro jednoduchost nabízí možnost použití třech funkcí příslušnosti (malý, střední a velký) pro jednotlivé vstupy. Rozsah (Range) je určen na základě maximálních a minimálních hodnot z kapitoly 5.2, s tím, že minimální hodnota byla zmenšená o 3 a maximální hodnota zvýšená o 3. Níže na obr. 15 lze vidět příklad nastavení funkce příslušnosti pro délku sepalu. Typy a parametry jsou rozebrány v následující podkapitole.



Obr. 15 Zobrazení nastavení Gaussian funkce příslušnosti délky sepalu (SL)

5.4.1 Volba tvarů a parametrů funkcí příslušnosti

Mezi nejpoužívanější tvary funkcí příslušnosti patří: Triangular, Trapezoidal a Gaussian [17]. Pro větší možnost výběru byly do výběru zahrnuty ještě další dva tvary: Two-sided Gaussian a Generalized bell.

V následujících podnadpisech je znázorněno zvolení parametrů pro jednotlivé tvary funkcí příslušnosti. Hodnoty parametrů vycházely z vypočítaných hodnot v kapitole 5.2. Pro stručnost jsou uvedeny pouze parametry pro délku sepalu, nastavení ostatních vstupů (šířka sepalu, délka a šířka petalu) probíhalo obdobným způsobem.

Triangular

Tvar trojúhelníkové křivky je dán třemi parametry: středem, počáteční a koncovou hodnotou. Střed je zvolen jako průměrná hodnota dané funkce příslušnosti. Počáteční hodnota je minimální hodnota zmenšená o 2, v případě small (malý) funkce příslušnosti je hodnota -1000 (kvůli zvětšení obsahu). Koncová hodnota je maximální hodnota zvětšená o 2, v případě big (velký) funkce příslušnosti je hodnota 1000 (kvůli zvětšení obsahu). V následující tab. 2 je vypsáno nastavení parametrů Triangular křivek pro délku sepalu.

Tab. 2 Nastavení parametrů Triangular křivky pro délku sepalu

Name	Type	Parameters
small	Triangular	[-1000 50.06 60]
middle	Triangular	[47 59.36 72]
big	Triangular	[47 65.88 1000]

Trapezoidal

Tvar lichoběžníkové křivky je dán čtyřmi parametry: dvěma středy, počáteční a koncovou hodnotou. První střed je vypočítán jako součet minimální hodnoty s rozdílem maximální a minimální hodnoty poděleného třemi. Druhý střed je rozdíl maximální hodnoty a rozdílu maximální a minimální hodnoty podělené třemi. Počáteční a koncové hodnoty jsou určeny stejně jako u Triangular funkce příslušnosti. V následující tab. 3 je vypsáno nastavení parametrů Trapezoidal křivek pro délku sepalu.

Tab. 3 Nastavení parametrů Trapezoidal křivky pro délku sepalu

Name	Type	Parameters
small	Trapezoidal	[-1000 48 53 60]
middle	Trapezoidal	[47 56 63 72]
big	Trapezoidal	[47 59 69 1000]

Gaussian

Tvar Gaussovské křivky je dán dvěma parametry: středem a šířkou zvonu. Střed je zvolen jako průměrná hodnota dané funkce příslušnosti. Šířka zvonu je vypočítána jako rozdíl maximální a minimální hodnoty podělený dvěma. V následující tab. 4 je vypsáno nastavení parametrů Gaussian křivek pro délku sepalu.

Tab. 4 Nastavení parametrů Gaussian křivky pro délku sepalu

Name	Type	Parameters
small	Gaussian	[7.5 50.06]
middle	Gaussian	[10.5 59.36]
big	Gaussian	[15 65.88]

Two-sided Gaussian

Tvar oboustranné Gaussovské křivky je dán čtyřmi parametry: dvěma středy, šířkou funkce nalevo od středu a šířkou funkce napravo od středu. Středy jsou určeny stejně jako u Trapezoidal funkce příslušnosti. Šířka funkce nalevo je vypočítána jako rozdíl maximální a minimální hodnoty podělený třemi, v případě small (malý) funkce příslušnosti je hodnota 1000 (kvůli zvětšení obsahu). Šířka funkce napravo je vypočítána stejně, jenom s rozdílem, v případě big (velký) funkce příslušnosti, kdy je hodnota 1000 (kvůli zvětšení obsahu). V následující tab. 5 je vypsáno nastavení parametrů Two-sided Gaussian křivek pro délku sepalu.

Tab. 5 Nastavení parametrů Two-sided Gaussian křivky pro délku sepalu

Name	Type	Parameters
small	Two-sided Gaussian	[1000 48 5 53]
middle	Two-sided Gaussian	[7 56 7 63]
big	Two-sided Gaussian	[10 59 1000 69]

Generalized bell

Tvar zvonové křivky je dán třemi parametry: středem, tvarem a šířkou. Střed je zvolen jako průměrná hodnota dané funkce příslušnosti. Pro tvary byla zvolena neutrální hodnota 2. Šířka zvonu je vypočítána jako rozdíl maximální a minimální hodnoty podělený dvěma. V následující tab. 6 je vypsáno nastavení parametrů Generalized bell křivek pro délku sepalu.

Tab. 6 Nastavení parametrů Generalized bell křivky pro délku sepalu

Name	Type	Parameters
small	Generalized bell	[7.5 2 50.06]
middle	Generalized bell	[10.5 2 59.36]
big	Generalized bell	[15 2 65.88]

5.5 Stanovení inferenčních pravidel

Byly zvoleny celkem tři pravidla, kde každé pravidlo reprezentuje jeden druh. První pravidlo říká: Pokud je délka sepalu malá (small), šířka sepalu velká (big), délka petalu malá (small) a šířka petalu malá (small), tak se jedná o druh Setosa. Druhé a třetí pravidlo je tvořeno stejným způsobem, akorát s jinými hodnotami (viz obr. 16 níže). Váhy pravidel zůstaly v počátečním nastavení na hodnotě 1.

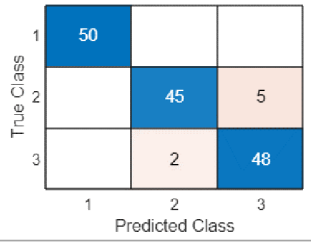
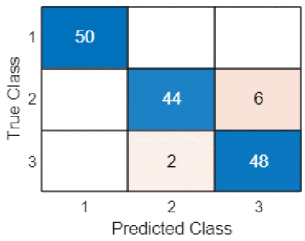
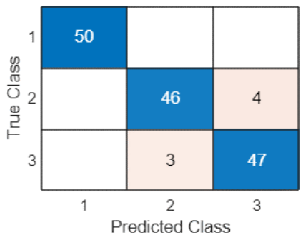
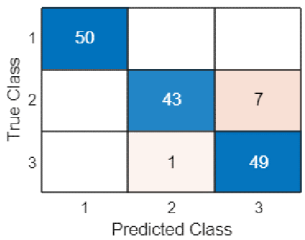
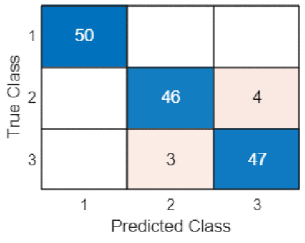
	Rule	Weight	Name
1	If SL is small and SW is big and PL is small and PW is small then Species is Setosa	1	rule1
2	If SL is middle and SW is small and PL is middle and PW is middle then Species is Versicolor	1	rule2
3	If SL is big and SW is middle and PL is big and PW is big then Species is Virginica	1	rule3

Obr. 16 Pravidla předpřipravených systémů

5.6 Vyhodnocení předpřipravených systémů

V následující tab. 7 je zobrazeno vyhodnocení pro každý předpřipravený systém. Vyhodnocován je počet chybně určených druhů, procentuální úspěšnost systému a zobrazení, které druhy byly špatně určeny pomocí matice záměn (confusion matrix). V matici záměn je pod číslem 1 uvedena Setosa, pod číslem 2 Versicolor a 3 Virginica.

Tab. 7 Porovnání systémů s různými tvary funkcí příslušnosti

Druh	Počet chybných	% úspěšnost	Matice záměn
Triangular	7	95.33	
Trapezoidal	8	94.67	
Gaussian	7	95.33	
Two-side Gaussian	8	94.67	
Generalized bell	7	95.33	

Na předběžném vyhodnocení lze vidět, že všechny systémy mají vysokou úspěšnost kolem 95 %. Jednotlivé matice záměn ukazují, že chybně určené třídy jsou pouze 2 a 3 (tedy druhy Versicolor a Virginica). Mezi jednotlivými systémy nelze zatím pozorovat výrazné rozdíly, ty se projeví více až v části trénování (kapitola 5.7).

5.7 Trénink předpřipravených systémů

Doladění systému probíhá na všech dostupných datech (150 vzorků), jelikož se jedná o malý počet vzorků, tak je nerozumné přerozdělovat data do trénovací a testovací skupiny. Jakým způsobem je trénink proveden je popsáno v části popisů skriptů (kapitola 8.1.3). Genetický algoritmus je nastaven na hodnotu 30 generací a 120 populací.

Vyhodnocován je opět počet chybných vzorků, procentuální úspěšnost systému a zobrazení, které druhy byly špatně určeny pomocí matice záměn (confusion matrix). Nově se vyhodnocuje i počet inferenčních pravidel systému, čas tréninku a počet proměnných, které byly trénovány. Počet pravidel je do hodnocení zahrnut, jelikož je nežádoucí mít systém, ve kterém bude vysoký počet pravidel (s rostoucím počtem pravidel klesá přehlednost systému). Čas tréninku je zohledňován kvůli zjištění, který systém trvá, jak dlouho vytrénovat a počet proměnných je dán druhem funkce příslušnosti. Výpočet počtu proměnných pro tento případ je dán následovně:

$$p_p = n_i * n_{mfs} * n_p + (n_i + n_o) \cdot n_{maxr} \tag{5}$$

Kde: n_i [-] - počet vstupů (inputů) systému

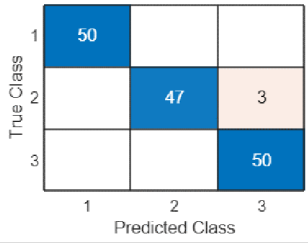
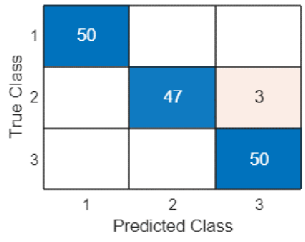
n_{mfs} [-] - počet funkcí příslušnosti v jednom vstupu

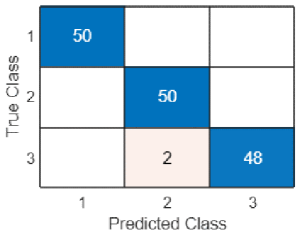
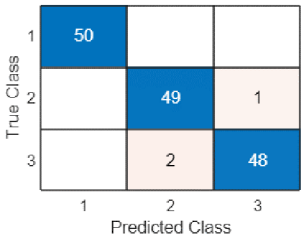
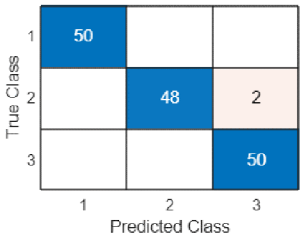
n_p [-] - počet parametrů daného druhu funkce příslušnosti (viz tab. 1 z kap. 3.2)

n_o [-] - počet výstupů (outputů) systému

n_{maxr} [-] - stanovený maximální počet pravidel systému

Tab. 8 Porovnání natrénovaných systémů s různými tvary funkcí příslušnosti

Druh	Počet chybných	% úspěšnost	Počet pravidel	Trvání / počet p_p	Matice záměn
Triangular	3	98	4	151.6 s / 81	
Trapezoidal	3	98	4	146.3 s / 93	

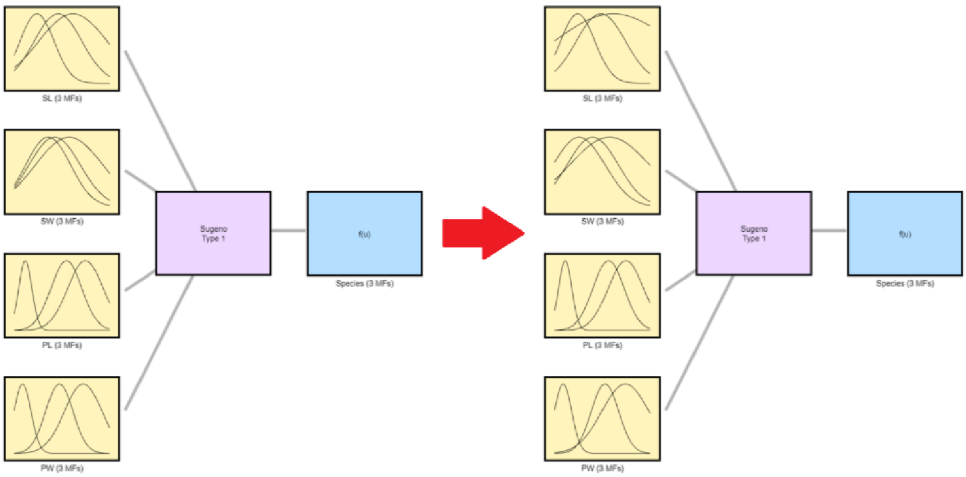
Gaussian	2	98.67	4	110.8 s / 69	
Two-side Gaussian	3	98	4	122.7 s / 93	
Generalized bell	2	98.67	4	110.6 s / 81	

Na porovnání lze vidět jediný znatelný rozdíl, a to rozdíl mezi dobou tréninku Triangular a Trapezoidal a dalších tří druhů. Vyšší časy u dvou zmíněných druhů jsou zapříčiněny vyšší šancí zvolení takových parametrů funkcí příslušnosti, do kterých nezapadne některý z trénovacích vzorků. To se poté projevuje vypisováním varovných zpráv, které zpomalují chod celého procesu. Z porovnání lze také vyčíst, že počet proměnných, který se mění s počtem parametrů funkce příslušnost, nemá vliv na čas tréninku.

Z tab. 8 lze vyčíst, že nejlépe dopadly druhy Gaussian a Generalized bell – počet špatně určených druhů je pouze dva, oproti ostatním, které špatně vyhodnotily tři druhy. Tyto dva druhy budou tedy přednostně používány v dalších postupech.

5.8 Konečný trénink systému

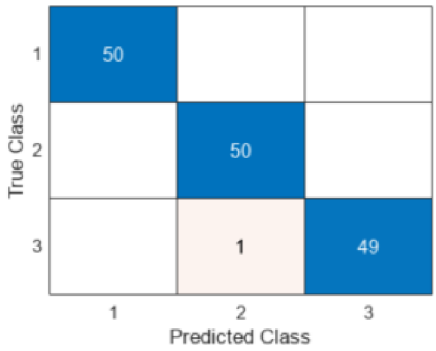
Pro konečnou fázi jsou vybrány dva druhy Gaussian a Generalized bell, kdy byl pro oba druhy puštěn trénink třikrát znovu, s tou změnou, že se zvýšil počet maximálních generací (z 30 na 60) a velikost populace (z 120 na 240). Systém s gaussianovými tvary funkcí příslušnosti byl vždy kolem čtyřicáté generace zastaven, z důvodu klesnutí průměrné fitness hodnoty pod toleranci, na jednom špatně určeném druhu a čtyřech pravidlech. Systém s Generalized bell tvarem funkcí příslušnosti se na tyto hodnoty dostal pouze jednou ze tří. Z toho důvodu je tvar funkce příslušnosti Gaussian zvolen jako nejvhodnější. Na obr. 17 je vidět změna tvarů funkcí příslušnosti před a po tréninku.



Obr. 17 Porovnání tvarů funkcí příslušnosti před (vlevo) a po (vpravo) tréninku

Na obr. 18 je zobrazena matice záměn (confusion matrix) konečného systému s Gaussian tvarem funkcí příslušnosti. Z matice lze vyčíst, že druh měl být zařazen do třídy 3 (Virginica) ale systém ho zařadil do třídy 2 (Versicolor). Z obr. 19 níže, na kterém je zobrazena tabulka vygenerovaná z Matlabu, lze vyčíst hodnoty špatně určeného druhu a jeho „System number“, což je číslo určené systémem. Kdyby toto číslo bylo vyšší, než 2.5, tak by byl druh zařazen do správné třídy.

Konečný systém má úspěšnost 99.33 % a pouze jeden špatně určený druh, což lze považovat za dobrý výsledek na to, že některé druhy mají velmi podobné parametry a člověk by podle jejich naměřených hodnot, v některých případech nevěděl, do jaké třídy je zařadit.



Obr. 18 Matice záměn konečného systému

	LS	WS	LP	WP	Class	System number
1	63	28	51	15	3	2.4724

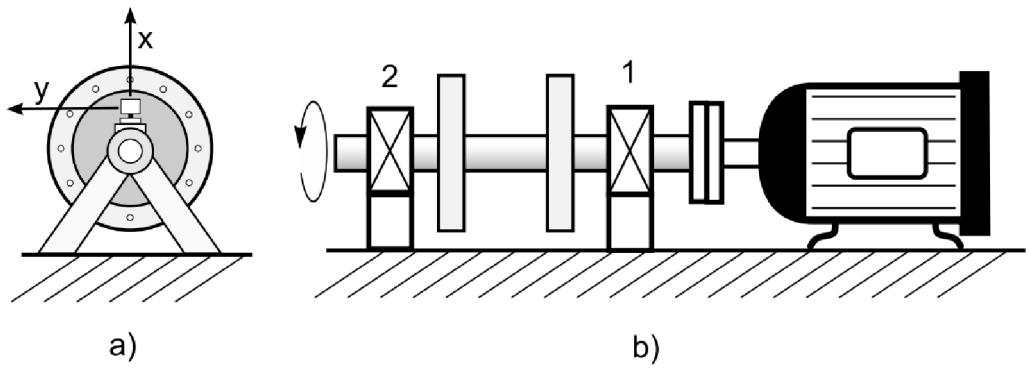
Obr. 19 Tabulka z Matlabu se špatně určeným druhem

6 ROZBOR TECHNICKÉHO DATASETU

Získané znalosti a zjištění z předešlé kapitoly 5 Fuzzy logika a Iris dataset jsou aplikovány na dataset z oblasti technické diagnostiky, konkrétně vibrodiagnostiky. Vybraný dataset, pojmenovaný EXsys-vibro, byl popsán a hodnocen pomocí klasifikačních metod v literatuře [25]. V této práci jsou porovnány výsledky dosažené ve zmíněné literatuře a výsledky dosažené díky použití fuzzy logiky.

6.1 Informace o datasetu

Data jsou získány z rotačního stroje zobrazeného na obr. 20 níže Stroj je určen pro simulační účely a nemá tak žádné využití v praxi. Nicméně je ideálním modelem pro simulování statické a dynamické nevývahy, které vznikají umístěním závaží do předpřipravených otvorů kotoučů. Tyto nevývahy jsou snímány dvojicí akcelerometrů umístěných na stojanech ložisek [25].



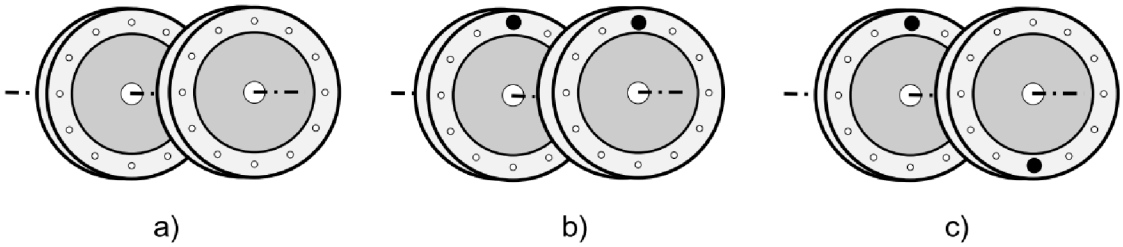
Obr. 20 Diagnostický model: a) zobrazení směrů os, b) pohled zepředu – zobrazení pozic snímačů [25]

6.1.1 Druhy simulovaných poruch

Na modelu z obr. 20 je simulována statická a dynamická nevývaha, kdy rozdíl mezi zmíněnými spočívá v odlišném umístění závaží do kotoučů. U statické nevývahy jsou závaží umístěna ve stejné poloze (viz obr. 21b) a u dynamické nevývahy jsou polohy závaží posunuty vzájemně o 180° (viz obr. 21c). V případě bezporuchového stavu nebyla do kotoučů umístěna žádná závaží (viz obr. 21a) [25].

Na rotačním stroji bylo simulováno celkem pět různých měření, kdy otáčky stroje byly pro všechna měření nastaveny na stejnou hodnotu 1190 ot/min. [25]

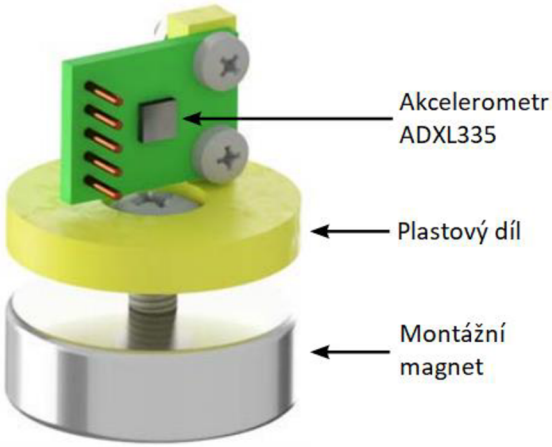
- **„OK“** – bezporuchový stav, kotouče nejsou osazeny závažím (obr. 21a)
- **„ST1“** – statická nevývaha první úrovně, kotouče jsou osazeny závažím o hmotnosti 2,5 g (obr. 21b)
- **„DYN1“** - dynamická nevývaha první úrovně, kotouče jsou osazeny závažím o hmotnosti 2,5 g (obr. 21c)
- **„ST2“** - statická nevývaha druhé úrovně, kotouče jsou osazeny závažím o hmotnosti 3,7 g (obr. 21b)
- **„DYN2“** - dynamická nevývaha druhé úrovně, kotouče jsou osazeny závažím o hmotnosti 3,7 g (obr. 21c)



Obr. 21 Schéma umístění závaží na kotoučích: a) bezporuchový stav, b) statická nevývaha, c) dynamická nevývaha [25]

6.1.2 Použitá senzorika

Naměřená data jsou získávána z dvojice akcelerometrů MEMS (MicroElectroMechanical Systems) ADXL335, které jsou umístěny na ložiskových domcích viz obr. 20b pozice 1 a 2. Jedná se o analogové akcelerometry jejichž výstupní signál je zpracováván měřící kartou DAQ NI USB-6009 s nastavenou vzorkovací frekvencí 2 kHz a 1000 zaznamenávanými vzorky. Snímač je připevněn pomocí šroubů k plastovému dílů a ten je přišroubován k montážnímu magnetu (viz obr. 22 níže). [25]

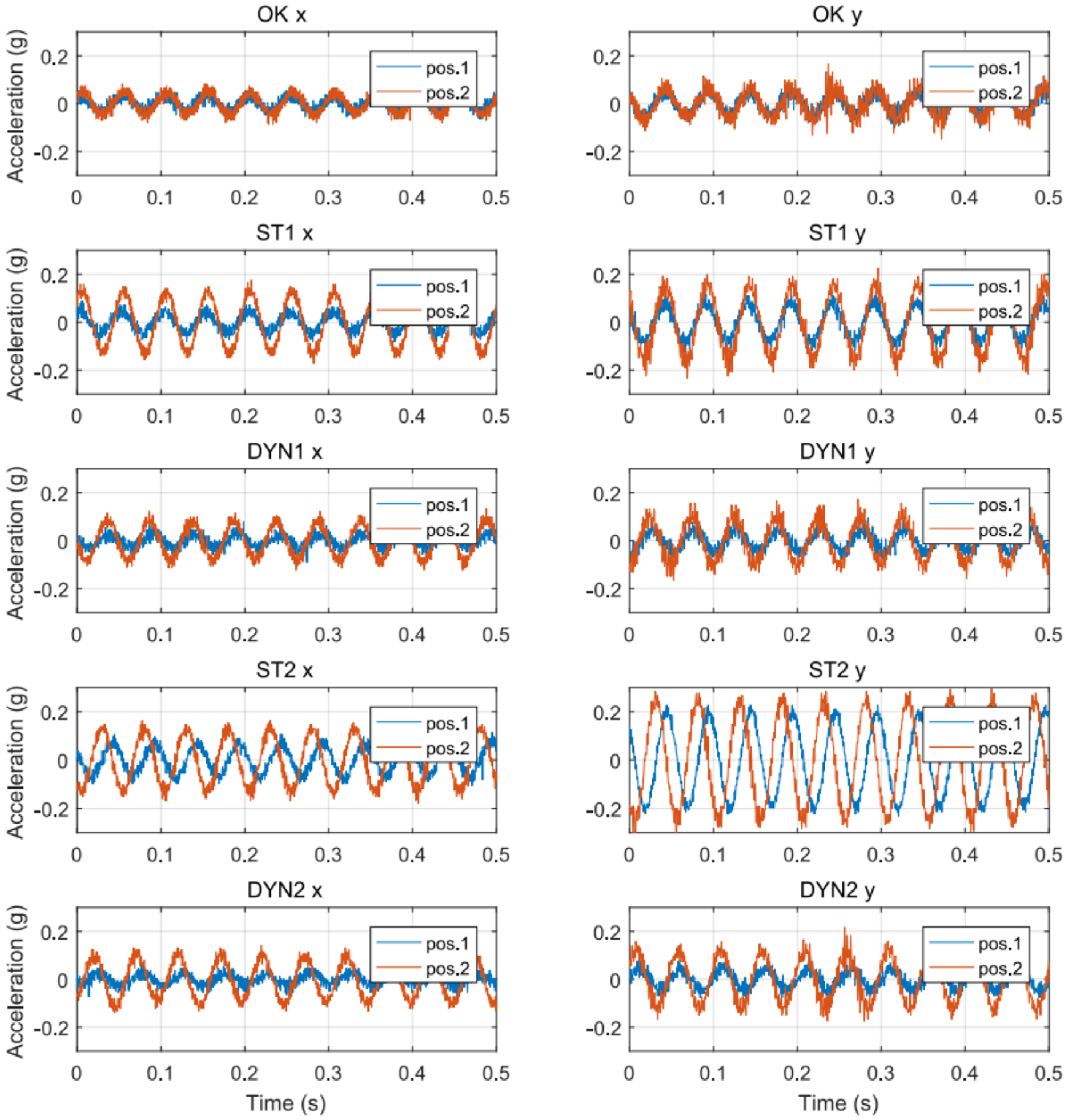


Obr. 22 Uchycení akcelerometru [25]

6.1.3 Data v grafické podobě

Na obr. 23 níže jsou zobrazeny grafy pro všechny možné stavy stroje a obě osy x a y. V každém jednom grafu jsou vykresleny dvě pozice (1 a 2) pro každý akcelerometr. Data jsou vykreslena z náhodně vybraných řádků ze záznamů. [25]

Tyto grafy slouží pouze pro názorné zobrazení, jak nezpracované (tedy až na normalizaci) data vypadají. V dalších částech jsou tyto data dále zpracovány.



Obr. 23 Vizualizace náhodných řádků z jednotlivých měření [25]

6.2 Zpracování dat

Zpracování dat je nedílnou součástí technické diagnostiky. Z naměřených dat je potřeba vyextrahovat reprezentační hodnoty, toho lze dosáhnout použitím vhodných prediktorů. Prediktory lze označit za číselné reprezentace nezpracovaných dat a existuje jich několik druhů. Výběr vhodných druhů je zásadním krokem pro vytvoření úspěšného modelu, je tedy zapotřebí mít znalosti, zkušenosti a přehled z tohoto odvětví. Prediktory by měly ideálně co nejvíce vystihovat daný problém diagnostiky.

Pro tuto práci byly zvoleny stejné prediktory jako v literatuře [25], jelikož jsou porovnávány výsledky dosažené pomocí klasifikačních metod a fuzzy logiky. Použitými prediktory tedy jsou:

- RMS – efektivní hodnota (Root Mean Square), popisuje průběh vibrací v čase
- STD – směrodatná odchylka (Standart Deviation), popisuje nevyváženost diagnostikovaného objektu
- PCA – analýza hlavních komponent (Principal Component Analysis), statistická metoda používaná k redukci dimenzionality dat
- Fázový posun FP – určuje zpoždění mezi signály, pomáhá určit rozdíly mezi statickou a dynamickou nevývahou

Zmíněné prediktory jsou aplikovány na všechny data, tedy až na fázový posun, ten porovnává pouze rozdíly mezi signálem z prvního a druhého snímače. Jelikož se předpokládá, že při statické a dynamické nevývaze dojde k různému fázovému posunu signálu mezi stojanem 1 a 2.

Vznikne tedy pět tabulek pro každou třídu: bezporuchový stav, statická nevývaha první a druhé úrovně a dynamická nevývaha první a druhé úrovně. V každé tabulce je poté 1000 řádků. Níže je zobrazena pro ukázku tab. 9 bezporuchového stavu s hodnotami z prvního řádku matlabovské tabulky.

Tab. 9 Zobrazení prvního řádku tabulky bezporuchového stavu

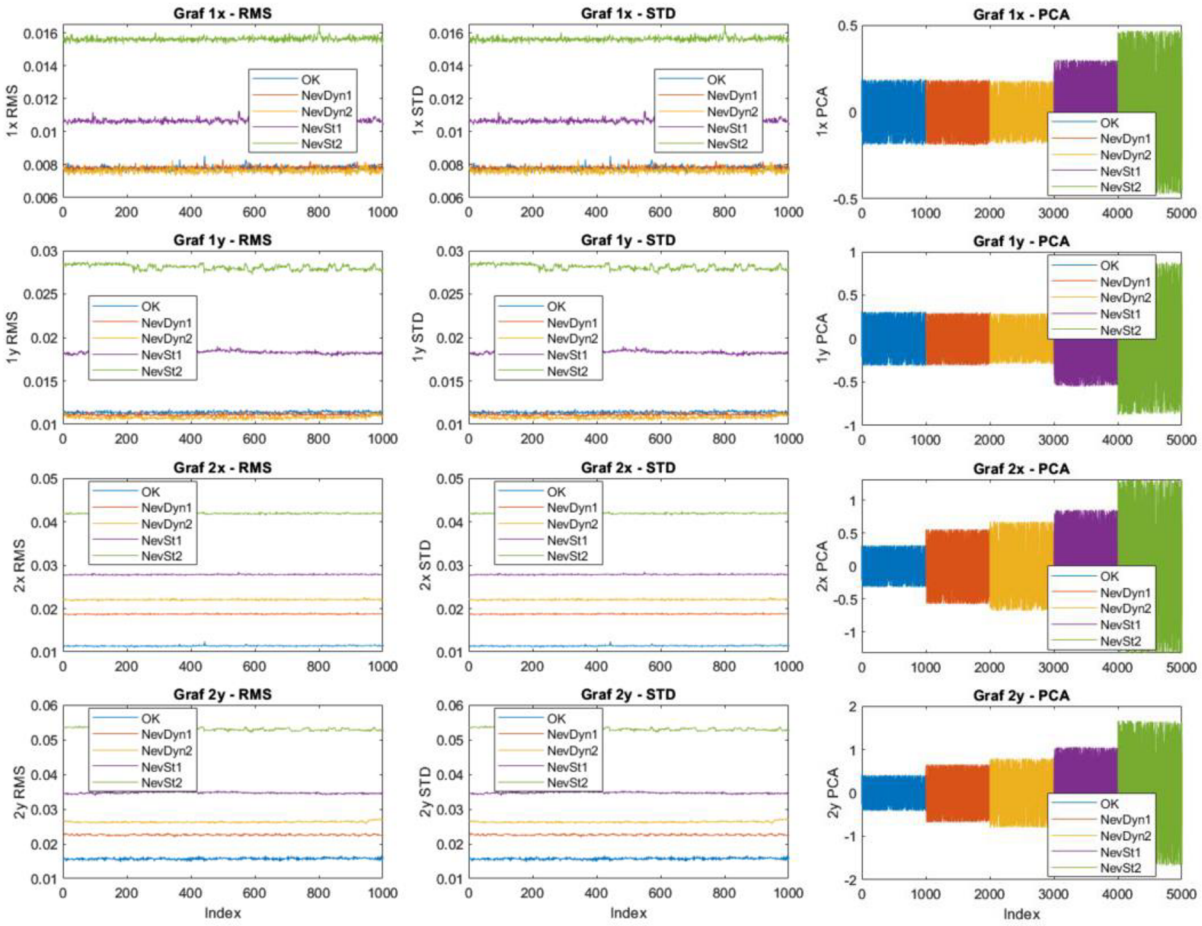
1x_OK_RMS	1x_OK_STD	1x_OK_PCA	1y_OK_RMS	1y_OK_STD	1y_OK_PCA	2x_OK_RMS	2x_OK_STD	2x_OK_PCA	2y_OK_RMS	2y_OK_STD	2y_OK_PCA	1_OK_FPx	1_OK_Fpy	Třída
0.0077	0.0077	0.1827	0.0115	0.0115	0.3109	0.0113	0.0113	0.3126	0.0158	0.0159	0.4098	4	5	1_OK

6.3 Vykreslení dat

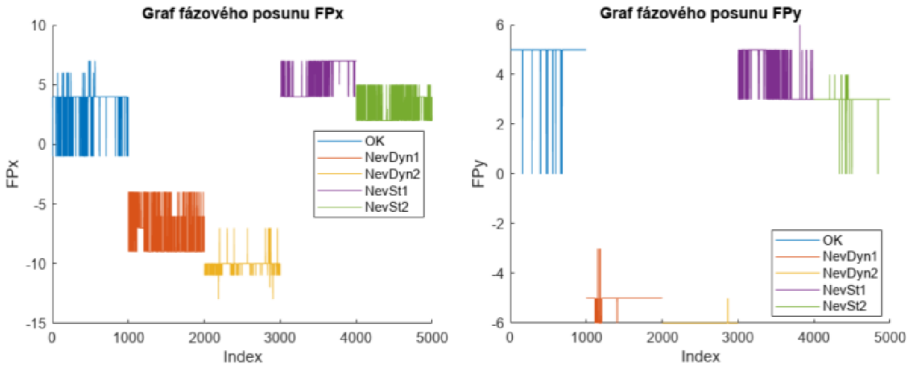
K důkladnějšímu prozkoumání datasetu jsou všechna data vykreslena do grafů. Grafy jsou vykresleny pro každý akcelerometr, dvě osy, určené prediktory a všechny možné stavy. Z připravených pěti tabulek (viz předchozí kapitola 6.2) jsou tedy vybrány data ze všech sloupců (kromě posledního).

Na každém grafu jsou vždy zobrazeny možné stavy, které jsou zobrazeny v legendách grafů:

- OK – bezporuchový stav (modrá barva)
- NevDyn1 – dynamická nevývaha první úrovně, použito závaží o hmotnosti 2,5 g (oranžová barva)
- NevDyn2 – dynamická nevývaha druhé úrovně, použito závaží o hmotnosti 3,7 g (žlutá barva)
- NevSt1 – statická nevývaha první úrovně, použito závaží o hmotnosti 2,5 g (fialová barva)
- NevSt2 – statická nevývaha druhé úrovně, použito závaží o hmotnosti 3,7 g (zelená barva)



Obr. 24 Zobrazení grafů pro jednotlivé stavy



Obr. 25 Grafy fázových posunů

Z vykreslených grafů na obr. 24 a obr. 25 lze vyčíst, že nebude problém s určeních jednotlivých druhů, jelikož v grafech s názvy: Graf 2x – RMS, Graf 2x – STD, Graf 2y – RMS, Graf 2y – STD nejsou hodnoty překryty a je jednoduché je rozeznat i pomocí jednoduché booleovské logiky. Díky druhému snímači je tedy jednoduché rozeznat rozdíly mezi jednotlivými stavy. Problém nastává ve chvíli, kdy se použije pouze na první snímač. Z grafů prvního snímače lze vyčíst, že bude problémové rozeznat rozdíly mezi třídami bezporuchového stavu a dynamických nevývah. Hodnoty statických nevývah se pohybují ve vyšších hodnotách, lze tak říct, že s rozeznáním těchto tříd by neměl být problém.

7 TVORBA A ZHODNOCENÍ FIS TECHNICKÉHO DATASETU

Systémy jsou tvořeny obdobným způsobem jako v kapitole 5 o Iris datasetu. Je vytvořeno několik systémů, které lze rozdělit do dvou hlavních nadkategorií: systém s daty prvního snímače a systém s daty druhého snímače. Rozdíly mezi těmito systémy jsou pouze v počtu vstupů (inputů) a na to navazujících lingvistických pravidel. FIS model je použit opět Sugeno a výstup (output) zůstává ve všech případech stejný, tvořený pěti třídami: OK, NevDy1, NevDyn2, NevSt1, NevSt2 (bezporuchový stav, dynamická nevývaha první a druhé úrovně, statická nevývaha první a druhé úrovně).

Tvorba jednotlivých systémů probíhá automatizovaným způsobem (oproti Iris datasetu), který je rozebrán v části popisu skriptů (kap. 8.2.3).

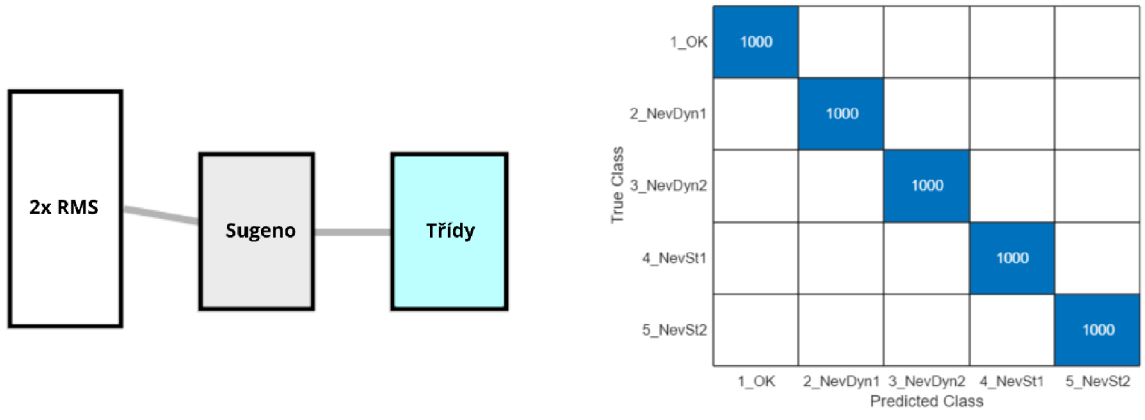
7.1 FIS systém s daty z druhého snímače

Z grafů v předešlé kapitole 6.3 je zřejmé, že není zapotřebí vytvářet složitý FIS systém, aby bylo dosaženo 100 % úspěšnosti systému. K dosažení 100 % úspěšnosti, tak stačí vytvořit jednoduchý systém s jedním vstupem s daty druhého snímače a vytvořených prediktorů RMS či STD.

Je tedy vytvořen systém s jedním vstupem, a to 2x RMS, což jsou data s RMS hodnotou druhého akcelerometru (x-ové složky). Tvar funkcí příslušnosti je použit Gaussian, který dosahoval nejlepších hodnot (viz kap. 5.8) a parametry jsou dány minimálními, průměrnými a maximálními hodnotami (více v kap. 8.2.3). Pravidla jsou tvořena způsobem zobrazeným na obr. 26 níže, kdy na grafu 2x – RMS z obr. 24 lze vidět, že třída OK má nejnižší hodnoty, a tak je v pravidlech zavedena jako malá (small). Další pravidla jsou tvořena stejným způsobem.

	Rule	Weight
1	If 2x RMS is small then Class is OK	1
2	If 2x RMS is mediumSmall then Class is NevDyn1	1
3	If 2x RMS is medium then Class is NevDyn2	1
4	If 2x RMS is mediumBig then Class is NevSt1	1
5	If 2x RMS is Big then Class is NevSt2	1

Obr. 26 Tabulka pravidel prvního FIS systému



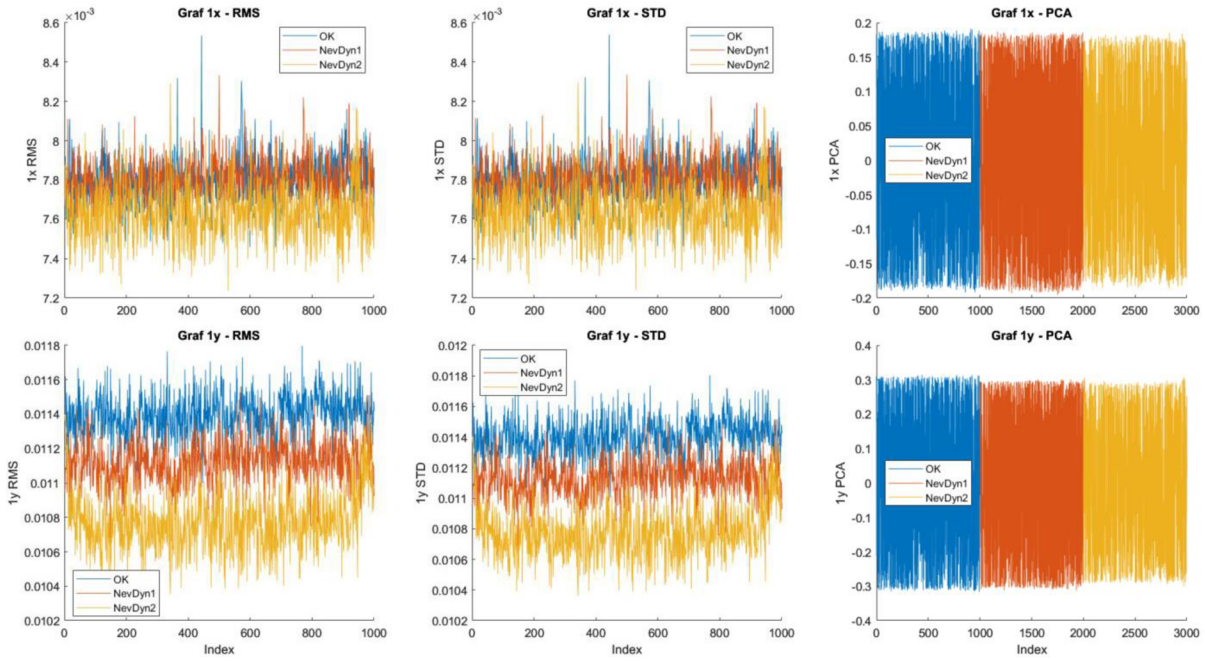
a) *b)*

Obr. 27 První FIS systém: a) schéma systému, b) matice záměn

Na obr. 27a je zobrazeno schéma systému s nastavenými Gaussian funkcemi příslušnosti a na obr. 27b vedle lze vidět matici záměn pro tento systém. Z matice záměn lze vyčíst, že všechny třídy byly správně zařazeny, což znamená, že systém má 100 % úspěšnost.

7.2 FIS systém s daty pouze prvního snímače

Složitějším úkolem, oproti předešlé podkapitole, je vytvořit systém s co nejvyšší úspěšnosti, při použití dat pouze prvního snímače, kde jsou první tři třídy (bezporuchový stav, dynamická nevývaha první a druhé úrovně) různě překryty. Na obr. 28 níže jsou tato překrytí znázorněny na detailnějších grafech. Zbylé dvě třídy (statická nevývaha první a druhé úrovně) nejsou s žádnými daty překryty a neměl by tak být problém je rozeznat.

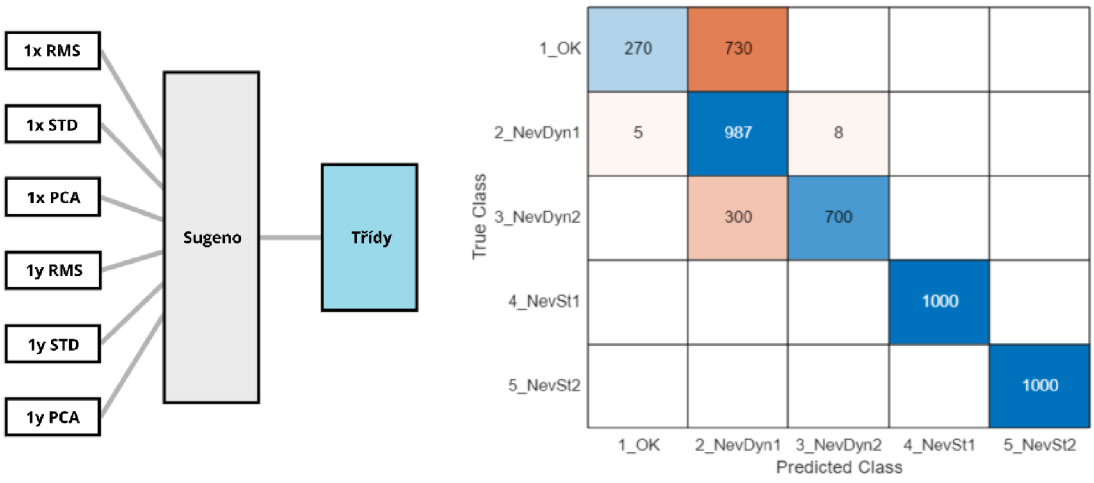


Obr. 28 Detailní zobrazení překrytí prvních třech tříd

Systém je tvořen šesti vstupy (viz obr. 30a), kde tvary funkcí příslušnosti jsou Gaussian a parametry jsou dány opět na základě minimálních, průměrných a maximálních hodnot (více v kap. 8.2.3). Pravidla systému (viz obr. 29 níže) jsou dána seřazením průměrných hodnot, od nejmenšího po největší, v jednotlivých vstupech a následným přiřazením ke každému vstupu (detailněji popsáno v kapitole 8.2.3).

	Rule	Weight
1	If 1x RMS is mediumSmall and 1x STD is mediumSmall and 1x PCA is Big and 1y RMS is medium and 1y STD is medium and 1y PCA is mediumBig then Class is OK	1
2	If 1x RMS is medium and 1x STD is medium and 1x PCA is mediumSmall and 1y RMS is mediumSmall and 1y STD is mediumSmall and 1y PCA is medium then Class is NevDyn1	1
3	If 1x RMS is small and 1x STD is small and 1x PCA is mediumBig and 1y RMS is small and 1y STD is small and 1y PCA is mediumSmall then Class is NevDyn2	1
4	If 1x RMS is mediumBig and 1x STD is mediumBig and 1x PCA is medium and 1y RMS is mediumBig and 1y STD is mediumBig and 1y PCA is small then Class is NevSt1	1
5	If 1x RMS is Big and 1x STD is Big and 1x PCA is small and 1y RMS is Big and 1y STD is Big and 1y PCA is Big then Class is NevSt2	1

Obr. 29 Tabulka pravidel FIS systému s prediktory prvního snímače



a) *b)*

Obr. 30 Druhý FIS systém: a) schéma systému, b) matice záměn

Oproti prvnímu FIS systému lze zde, na matici záměn (obr. 30b), vidět rozdíl v tom, že druhý systém má problém se zařazením tříd 1, 2 a 3, jelikož data mají podobné hodnoty (viz grafy na obr. 28). Úspěšnost systému činí 79,14 %, což nelze považovat za vysokou hodnotu, jelikož většina klasifikačních metod z literatury [25] dosáhla, na stejném datasetu, vyšších hodnot.

Za účelem dosažení vyšší úspěšnosti je předchozí vytvořený systém a další systémy trénovány. Trénink probíhá obdobným stylem, jako v kapitole 5.7 o Iris datasetu, kdy jsou opět trénovány parametry funkcí příslušnosti a lingvistická pravidla. Rozdíl oproti tréninku v kapitole o Iris datasetu je ten, že jsou data rozdělena na dvě části: tréninkovou a testovací sadu. Pokud by data nebyla rozdělena, tak by mohlo dojít k přeučení modelu, což by se projevovalo na zhoršené úspěšnosti na nových datech. Data jsou tedy rovnoměrně rozdělena do trénovací a testovací sady (2500 a 2500 vzorků).

V následujících podkapitolách je trénováno pět druhů systémů. U prvního druhu (kap. 7.2.1) jsou trénovány parametry funkcí příslušnosti a pravidla systému a genetický algoritmus má vyšší počty populací a generací. V případě druhého druhu (kap. 7.2.2) je pouze snížen počet generací a populací genetického algoritmu, jinak je trénink obdobný. U třetího druhu (kap. 7.2.3) jsou trénovány pouze parametry funkcí příslušnosti a pravidla jsou zanechána v původním nastavení. Čtvrtý druh (kap. 7.2.4) je obdoba třetího, s tím rozdílem, že je snížen počet generací a populací. Poslední pátý druh (kap. 7.2.5) je trénován odlišným způsobem, a to pomocí K-fold křížové validace.

7.2.1 Trénink parametrů a pravidel s vyšším počtem generací a populací

Trénování probíhá pomocí genetického algoritmu a vlastní fitness funkce, maximální počet generací je nastaven na hodnotu 100, počet populací na hodnotu 300 a jsou trénovány parametry funkcí příslušnosti a lingvistická pravidla.

Trénink je proveden pro stejné tvary funkcí příslušnosti jako v kapitole 5.7, tedy Gaussian, Triangular, Trapezoidal, Two-sided Gaussian a Generalized bell. Pro každý uvedený tvar funkce příslušnosti je trénink proveden pětkrát a do tabulky je vždy zaznamenáváno:

- % úspěšnost tréninkových dat – udává procentuální úspěšnost ověřenou na tréninkové sadě
- Průměrný čas tréninku jedné generace – udává, jak dlouho trval průměrně přechod z jedné generace na druhou
- Počet generací – počet, po kterém byl trénink zastaven (max. počet 100)
- % úspěšnost testovacích dat – udává procentuální úspěšnost ověřenou na testovací sadě

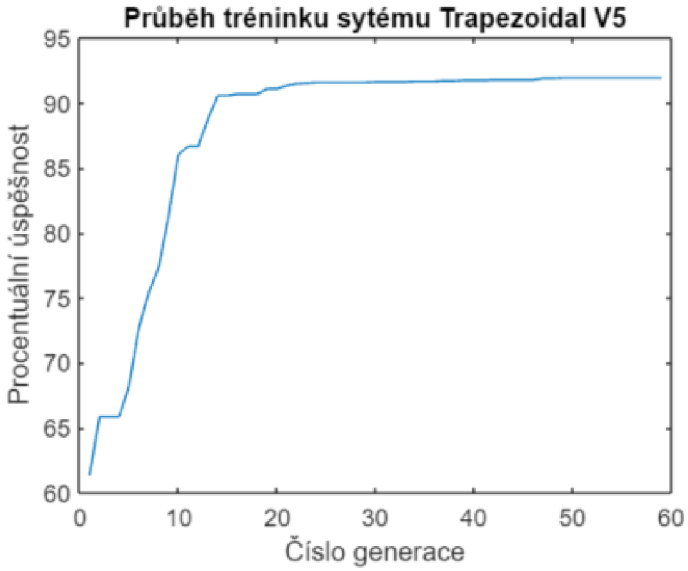
Tab. 10 Úspěšnosti tréninku parametrů a pravidel s vyšším počtem generací a populací

Druh a verze	% úspěšnost trén. dat	Průměrný čas trén. jedné generace [s]	Počet generací	% úspěšnost test. dat
Gaussian V1	90.72	18.72	20	88.76
Gaussian V2	91	15.91	56	88.92
Gaussian V3	91.2	15.49	44	89.48
Gaussian V4	90.8	15.99	37	88.84
Gaussian V5	91.28	16.91	32	89.4
Triangular V1	91.96	29.77	73	89.12
Triangular V2	91.36	45.82	34	89.2
Triangular V3	90.24	31.41	66	88.4
Triangular V4	88.64	35.57	59	86.84
Triangular V5	90.12	26.62	100	87.8
Trapezoidal V1	88.52	31.34	100	86.8
Trapezoidal V2	90.52	28.36	100	88.4
Trapezoidal V3	90.12	29.75	99	87.88
Trapezoidal V4	92.16	37.92	53	89.8
Trapezoidal V5	91.96	39,1	59	89.96
Two-sided Gaussian V1	91.16	16.92	35	89.28
Two-sided Gaussian V2	91.16	16.62	34	89.4
Two-sided Gaussian V3	91.16	16.61	31	89.4
Two-sided Gaussian V4	91.24	16.48	47	89.28
Two-sided Gaussian V5	91.16	17.01	28	89.32
Generalized bell V1	91.8	15.5	46	89.6
Generalized bell V2	91.6	15.38	36	89.72
Generalized bell V3	91.48	17.05	33	89.68
Generalized bell V4	91.4	16.06	29	89.76
Generalized bell V5	91.4	15.86	42	89.24
Průměr hodnot	90.97	22.63	51.72	88.97

Z předešlé tabulky (tab. 10) lze vyčíst, že systémy dosahovaly vysoké procentuální úspěšnosti na tréninkových datech (velká většina z nich více, než 90 %), nicméně, úspěšnost testovacích dat nepřekročila ani jednou hodnotu 90 % - průměrně se úspěšnost pohybovala o 2 % níže, než u tréninkových dat. To znamená, že došlo k přetrénování systému. Na obr. 31 níže, je zobrazen postup generací nejúspěšnějšího systému z tabulky (zeleně zaznačen), kde lze vidět, že již za čtrnáctou generací byl systém nad úspěšností 90 %, poté se zlepšoval malým krokem a na generaci 59 byl pozastaven z důvodu dosažení stanoveného maximálního počtu stejných generací.

U druhů Triangular a Trapezoidal jsou opět vyšší tréninkové časy z důvodů varovných zpráv (více v kapitole 5.7).

Na tab. 10 lze také vidět, že maximální stanovený počet generací 100, byl dosažen pouze třikrát a ve většině případů nedosáhl ani poloviny. Z tohoto důvodu a důvodu přeučení jsou v následujícím porovnání sníženy počty generací a populací.



Obr. 31 Průběh tréninku systému Trapezoidal V5

7.2.2 Trénink parametrů a pravidel se sníženým počtem generací a populací

Jedná se o stejný postup trénovaní, jako v předešlé kapitole, s tím rozdílem, že je snížen počet generací a populací. Nový počet generací je 60 a počet populací 200.

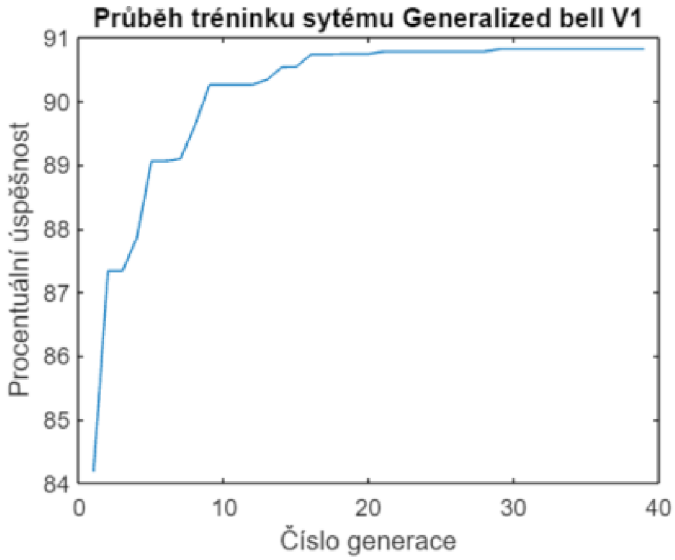
Tab. 11 Úspěšnosti tréninku parametrů a pravidel se sníženým počtem generací a populací

Druh a verze	% úspěšnost trén. dat	Průměrný čas trén. jedné generace [s]	Počet generací	% úspěšnost test. dat
Gaussian V1	90.48	8.75	50	89.64
Gaussian V2	90.52	8.67	36	89.88
Gaussian V3	90.08	8.58	27	89.72
Gaussian V4	90.44	8.75	38	89.84
Gaussian V5	89.84	8.57	26	89.56
Triangular V1	89.92	18.51	60	89.32
Triangular V2	90.36	18.37	60	90
Triangular V3	87.84	21.23	45	87.36
Triangular V4	90.24	17.62	60	89.68
Triangular V5	91.04	16.36	60	89.48
Trapezoidal V1	90.56	21.04	45	90.2
Trapezoidal V2	90.08	18.71	60	89.12
Trapezoidal V3	88.76	22.77	46	87.48
Trapezoidal V4	90.84	19.69	60	89.92
Trapezoidal V5	88.72	18.34	60	88.88
Two-sided Gaussian V1	90.56	8.77	46	90.04

Two-sided Gaussian V2	90.56	9.24	38	89.88
Two-sided Gaussian V3	90.64	8.81	43	89.68
Two-sided Gaussian V4	90.56	8.94	38	90.04
Two-sided Gaussian V5	90.48	8.93	39	89.88
Generalized bell V1	90.84	8.29	39	90.56
Generalized bell V2	90.88	8.59	31	90
Generalized bell V3	91.08	8.31	35	90.36
Generalized bell V4	90.44	8.46	32	89.84
Generalized bell V5	90.92	8.13	51	90.08
Průměr hodnot	90.27	12.9	45	89.62

Systémy se sníženým počtem generací a populací (z tab. 11) lze považovat za úspěšnější, oproti systémům z tab. 10, jelikož testovací sada dosahuje ve většině případů vyšších úspěšností. Průměrný rozdíl mezi trénovací a testovací sadou je v tomto případě pouhých 0,65 %, což je oproti předešlým 2 % slušné zlepšení. Jediné, v čem se systémy s nižším počtem generací a populací zhoršily je úspěšnost na trénovacích datech, což ale neznačí špatný výsledek. Pozitivní je také snížení času tréninku, kdy se průměrná hodnota snížila o cca 10 sekund na jednu generaci.

Nejúspěšnějším z tohoto porovnání se stal systém s tvarem funkcí příslušnosti Generalized bell – verze 1 (zeleně zaznačen). Dosáhl nejvyšší úspěšnosti na trénovacích datech – 90,56 % a úspěšnost tréninkových dat byla 90,84 %. Systém byl natrénován za zhruba 5,5 minuty a průběh jeho tréninku lze vidět na obr. 32 níže.



Obr. 32 Průběh tréninku systému Generalized bell V1

7.2.3 Trénink parametrů s vyšším počtem generací a populací

V předchozích případech byly v systémech trénovány, jak parametry funkcí příslušnosti, tak lingvistická pravidla. Pro co nejvyšší přehlednost systému, byly v tomto případě trénovány pouze parametry funkcí příslušnosti a pravidla tak zůstávají dle výchozího nastavení.

Jelikož v předchozí podkapitole vycházely nejúspěšněji systémy s tvarem funkcí příslušnosti Generalized bell, tak je tento trénink proveden pětkrát pro tento druh (viz tab. 12 níže). Genetický algoritmus je nastaven na hodnoty 60 generací a 200 populací.

Tab. 12 Úspěšnosti tréninku parametrů s vyšším počtem generací a populací

Druh a verze	% úspěšnost trén. dat	Průměrný čas trén. jedné generace [s]	Počet generací	% úspěšnost test. dat
Generalized bell V1	90.72	10.47	25	90.12
Generalized bell V2	91.2	10.19	47	90.36
Generalized bell V3	90.88	10.15	37	89.36
Generalized bell V4	91.16	10.45	34	90.04
Generalized bell V5	90.92	9.98	29	90.2

Z poklesu úspěšnosti testovacích sad lze usoudit, že systém byl přetrénován, proto je v další podkapitole snížen počet generací a populací.

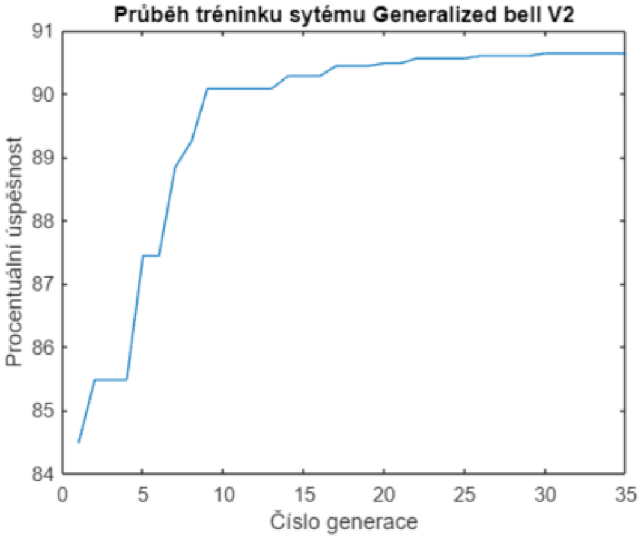
7.2.4 Trénink parametrů se sníženým počtem generací a populací

Trénink probíhá obdobným způsobem, jako v předešlé kapitole, rozdíl je pouze ve snížení počtu generací a populací. Nově je počet generací 35 a počet populací 100.

Tab. 13 Úspěšnosti tréninku parametrů se sníženým počtem generací a populací

Druh a verze	% úspěšnost trén. dat	Průměrný čas trén. jedné generace [s]	Počet generací	% úspěšnost test. dat
Generalized bell V1	90.8	5.38	35	90.16
Generalized bell V2	90.64	5.41	35	90.6
Generalized bell V3	90.48	5.40	25	90.6
Generalized bell V4	90.68	5.41	27	90.56
Generalized bell V5	90.48	5.51	34	90.52

Na tab. 13 lze vidět, že snížení počtu generací a populací přispělo k vyšší úspěšnosti testovacích dat, v některých případech je úspěšnost testovacích dat dokonce vyšší než úspěšnost dat trénovacích. Na obr. 33 níže je zobrazený průběh nejúspěšnějšího systému.



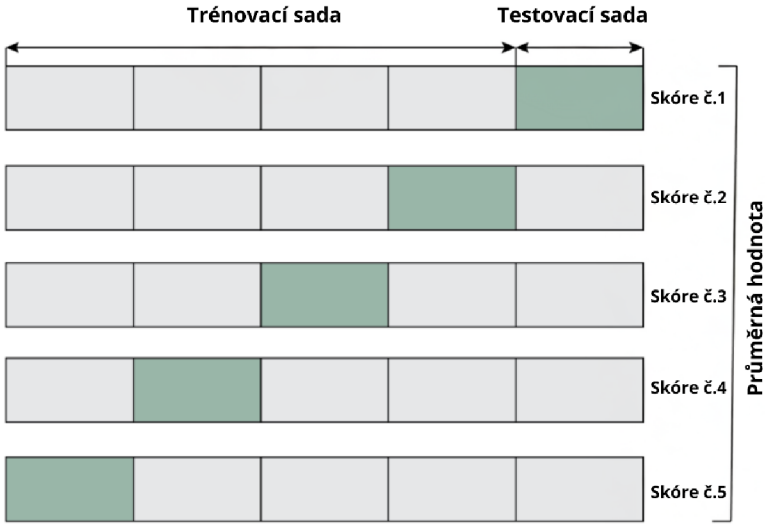
Obr. 33 Průběh tréninku systému Generalized bell V2

7.2.5 Trénink parametrů a pravidel pomocí K-fold křížové validace

Dalším možným způsobem, jak předejít přetrénovaní systému je použít K-fold křížovou validaci (K-fold cross-validation). V některých případech rozdělení dat do trénovacích a testovacích sad může dojít ke skrytí důležitých vzorků dat v testovací sadě, které se nedostanou k sadě tréninkové. Tento problém chytře řeší K-fold křížová validační metoda, při které jsou všechna data zahrnuta do tréninku, akorát v jiném časovém intervalu. Na obr. 34 je znázorněno schéma K-fold křížové validace pro k-fold 5. [26]

Obecný postup je následující:

- Rozdělení datasetu do k-foldů – nejprve je zapotřebí rozdělit dataset do k-foldů (sad), kdy každá z množin má stejnou velikost. Mezi nejčastěji používané hodnoty k-foldu patří hodnota 5 a 10. [27]
- Trénink každého k-foldu – poté je každý k-fold zvláště trénován za pomocí tréninkové sady a na konci je ohodnocen pomocí testovací sady.
- Vyhodnocení – po dokončení tréninku všech k-foldů dojde k vypočítání průměrné hodnoty ze všech testovacích sad. Tato hodnota značí celkové ohodnocení systému.



Obr. 34 Schéma K-fold křížové validace [26]

Celkem je trénováno deset systémů: pět pro počet k-foldů roven pěti a pět pro počet k-foldů roven deseti. To znamená, že nejprve je trénováno pět systémů, kde jsou data rozděleny na pět částí a poté je trénováno dalších pět systémů, kde jsou data rozděleny na deset částí. Počáteční nastavení systému je stejné jako v kapitole 7.2 – systém má šest vstupů a na ně navazující pravidla.

Trénink probíhá opět pomocí genetického algoritmu a vlastní fitness funkce, kde počet generací je nastaven na hodnotu 25 a počet populací 125. Počty jsou, oproti předchozím kapitolám sníženy, z důvodu mnohonásobného trénovaní systému s každým k-foldem zvláště.

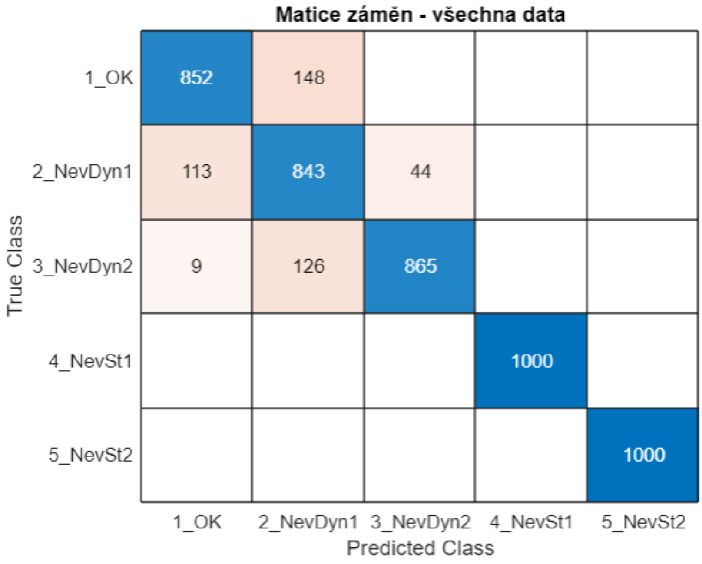
Po dokončení tréninku každého systému je zapsána jeho průměrná procentuální úspěšnost na tréninkových, testovacích a všech datech do tabulky. Z těchto hodnot je nejdůležitější průměrná úspěšnost na testovacích datech, jelikož k těmto datům neměl systém v danou chvíli přístup.

Tab. 14 Vyhodnocení procentuální úspěšnosti systému s K-fold křížovou validací

Počet k-foldů	Č. verze	Průměr. úspěšnost tréninkových dat	Průměr. úspěšnost testovacích dat	Průměr. úspěšnost všech dat
5	1	90.86	90.64	90.81
	2	90.82	90.60	90.78
	3	90.87	90.50	90.79
	4	90.32	90.16	90.29
	5	90.48	90.46	90.47
Průměrná hodnota		90.67	90.47	90.63
10	1	90.77	90.66	90.76
	2	90.69	90.66	90.69
	3	91.17	91.12	91.16
	4	90.70	90.58	90.68
	5	90.77	90.60	90.75
Průměrná hodnota		90.82	90.72	90.81

Jak je již zmíněno výše, v tab. 14 je nejdůležitější hodnota průměrná úspěšnost na testovacích datech, jelikož právě tato hodnota vypovídá nejvíce o úspěšnosti systému. Hodnota je totiž získána zprůměrováním úspěšností testovacích sad z každého k-foldu, ke kterým, v tu danou chvíli, neměl systém přístup. Na tabulce lze vidět, že lépe si vedly systémy, kde byl k-fold na hodnotě 10, což je způsobeno tím, že pro každý systém byl celkem dvojnásobný počet generací a populací oproti systémům s k-foldem 5.

Nejúspěšnějším je systém s deseti k-foldy – verze 3. Tento systém je také nejúspěšnější s porovnáním z předešlých podkapitol. Byla u něj dosažena průměrná úspěšnost na testovacích datech 91,12 %, na všech datech dohromady to poté dělalo 91,16 %. Na obr. 35 níže je zobrazena matice záměn všech dat pro tento systém, na které lze vidět, že byl opět problém s rozeznáním tříd 1, 2 a 3.



Obr. 35 Matice záměn nejúspěšnějšího systému s K-fold křížovou validací

7.3 Porovnání FIS systémů s klasifikačními metodami

V této kapitole jsou porovnávány vytvořené FIS systémy na datasetu EXsys-vibro s klasifikačními metodami aplikovanými na stejném datasetu, o kterých je článek [25]. Ve zmíněném článku je dataset hodnocen pomocí dvaceti různých klasifikačních metod zobrazených na obr. 36 níže. Obrázek popisuje procentuální úspěšnost systémů na testovacích sadách dat.

Data v článku jsou rozdělena stejným způsobem jako v kap. 6.2, polovina pro trénovací sadu a polovina pro testovací sadu, celkově je tedy 2500 záznamů pro obě sady (500 pro každou třídu).

Na obr. 36 jsou zobrazeny dva sloupce s úspěšností P1 a P2. U varianty P1 byly použity data pouze prvního snímače (stejně jako v kap. 7.2): 1x RMS, 1x STD, 1x PCA, 1y RMS, 1y STD a 1y PCA. U druhé varianty P2 byla použita všechna data – první snímač, druhý snímač a jejich fázové posuny.

		P1	P2
Decision trees:	Complex tree	85.5%	**100%**
	Medium tree	86.7%	**100%**
	Simple tree	88.0%	**100%**
Support vector machines:	**Linear SVM**	**90.2%**	**100%**
	Fine Gaussian SVM	87.8%	**100%**
	Medium Gaussian SVM	89.3%	**100%**
	Coarse Gaussian SVM	88.7%	**100%**
	Quadratic SVM	89.8%	**100%**
	Cubic SVM	88.2%	**100%**
K-Nearest neighbor classifiers:	Fine KNN	84.5%	**100%**
	Medium KNN	79.7%	**100%**
	Coarse KNN	64.3%	94.3%
	Cosine KNN	67.7%	**100%**
	Cubic KNN	78.3%	**100%**
	Weighted KNN	85.3%	**100%**
Ensemble classifiers:	Boosted trees	88.5%	22.8%
	RUSBoost trees	88.0%	22.8%
	Bagged trees	89.0%	**100%**
	Subspace KNN	72.0%	99.5%
	Subspace discriminant	89.8%	**100%**

Obr. 36 Porovnání úspěšnosti klasifikačních metod ze článku [25]

Na výsledcích klasifikačních metod lze vidět, že velká většina z nich neměla problém s rozeznáním tříd, pokud byla použita data z obou snímačů (sloupec P2). Nicméně i přesto se našly čtyři modely, které měly problém s rozeznáním i tak triviálního systému. Při použití dat pouze prvního snímače (sloupec P1) se hodnoty ve většině případů pohybují nad hodnotou 80 % a nejúspěšnějším je model Linear SVM s úspěšností 90,2 %. U této, nejúspěšnější metody, je v článku uvedena i úspěšnost na trénovací sadě – 91,08 %.

V následující tab. 15 je srovnání třech nejúspěšnějších klasifikačních systémů s patnácti vytvořenými FIS systémy. Za FIS systémy jsou vybrány tři nejúspěšnější z tabulek: tab. 10, tab. 11, tab. 12, tab. 13, tab. 14 uvedených v předešlých kapitolách. Každý z těchto pěti zástupců

skupin, byl tvořen jiným způsobem. Srovnávány jsou procentuální úspěšnosti trénovacích sad, testovacích sad a všech dat. Procentuální úspěšnost na všech datech u systémů, u kterých není uvedena, byla spočítána jako průměrná hodnota úspěšnosti tréninkové a testovací sady.

Tab. 15 Srovnání FIS systémů a klasifikačních metod

Druh	Proc. úspěšnost – tréninková sada	Proc. úspěšnost – testovací sada	Proc. úspěšnost – všechna data
1. Klasifikační metody			
Linear SVM	91.08	90.2	90.64
Quadratic SVM	-	89.8	-
Subpace discriminant	-	89.8	-
2. FIS – parametry, pravidla, 100 generací a 300 populací (viz tab. 10)			
Trapezoidal V5	91.96	89.96	90.96
Trapezoidal V4	92.16	89.8	90.98
Generalized bell V4	91.4	89.76	90.58
3. FIS – parametry, pravidla, 60 generací a 200 populací (viz tab. 11)			
Generalized bell V1	90.84	90.56	90.7
Generalized bell V3	91.08	90.36	90.72
Trapezoidal V1	90.56	90.2	90.38
4. FIS – parametry, 60 generací a 200 populací (viz tab. 12)			
Generalized bell V2	91.2	90.36	90.78
Generalized bell V5	90.92	90.2	90.56
Generalized bell V1	90.72	90.12	90.42
5. FIS – parametry, 35 generací a 100 populací (viz tab. 13)			
Generalized bell V2	90.64	90.6	90.62
Generalized bell V3	90.48	90.6	90.54
Generalized bell V4	90.68	90.56	90.62
6. FIS systémy s K-fold křížovou validací (viz tab. 14)			
K-fold 10 V3	91.17	91.12	91.16
K-fold 10 V1	90.77	90.66	90.76
K-fold 10 V2	90.69	90.66	90.69

Na srovnání lze vidět, že většina FIS systémů si vedla lépe než klasifikační metody. Hůře si vedly akorát přetrénované systémy, tedy ty s genetickým algoritmem tvořeným sto generacemi a tři sta populacemi, kde u nejúspěšnějšího ze systémů byla hodnota úspěšnosti na testovacích sadách nižší o zhruba 0,2 %. Počet generací a populací byl v následujících systémech snížen a dosáhly tak i lepších hodnot než klasifikační metody. Mezi nejúspěšnější patří systémy tvořeny pomocí K-fold křížové validace, které dosahují vysoké průměrné úspěšnosti. U nejúspěšnějšího je hodnota vyšší skoro o jedno procento oproti nejúspěšnější klasifikační metodě.

Srovnání dokazuje, že i FIS systémy mohou být velice efektivní ve strojovém učení, v některých případech dokonce i efektivnější než klasifikační metody. Za velikou výhodu těchto systémů lze považovat jejich pochopitelnost, zejména u čtvrtých a pátých systémů z tab. 15, kde byly trénovány pouze parametry funkcí příslušnosti a systém si tak zachoval ještě větší pochopitelnost a přehlednost.

8 POPIS POUŽITÝCH SKRIPTŮ

V této kapitole jsou slovně popsány všechny použité skripty vytvořené v softwaru Matlab 2022b v Live editoru. Stručnější popis lze naleznout i v jednotlivých skriptech, kde jsou okomentovány jejich hlavní části.

8.1 Skripty pro Iris dataset

V následujících podkapitolách jsou popsány skripty, které byly použity pro realizaci části této práce s názvem Fuzzy logika a Iris dataset. Každá následující podkapitola je pojmenována podle názvu daného skriptu, který lze naleznout v přílohách. Pořadí spuštění jednotlivých skriptů je dáno pořadím následujících podkapitol (pro spuštění skriptů Iris_evaluace.mlx a Iris_tunefis.mlx je potřeba nahrát vybraný FIS model z přílohy A).

8.1.1 Iris_grafy.mlx

Iris dataset je součástí matlabu, tudíž stačí jenom pomocí funkce load() soubor dat nahrát do workspace – tak je tomu učiněno na začátku skriptu. Poté jsou z jednotlivých tříd vybrány délky sepalu (kališních lístků) a petalu (okvětních lístků) a následně jsou vykresleny grafy pro délky a šířky sepalu a petalu pro jednotlivé třídy.

8.1.2 Iris_evaluace.mlx

Jednoduchý skript, ve kterém dochází k nahrání ručně vytvořeného FIS modelu a poté jeho ohodnocení pomocí matice záměn. Pod maticí záměn je vypsána procentuální úspěšnost systému a níže je vygenerována tabulka se špatně určenými třídami. V tabulce jsou uvedeny rozměry lístků, správná třída a hodnota, která byla daným rozměrům přiřazena systémem.

8.1.3 Iris_tunefis.mlx

Ze začátku skriptu dochází k načtení datasetu a nahrání zvoleného FIS systému. Dále jsou vybrány parametry, které se budou systémem ladit – jsou vybrány pouze vstupní parametry funkcí příslušností a pravidla. K ladění lze vybrat i výstupní parametry funkcí příslušnosti, nicméně, ty jsou nastaveny na pevné hodnoty (1, 2, 3) a bylo by spíše na škodu je jakkoliv měnit. V dalších krocích jsou nastaveny parametry genetického algoritmu: počet generací, počet populací, maximální počet stejných generací, maximální počet pravidel a typ optimalizace na „learning“ – učení, což umožňuje systému vytvářet nová pravidla. Poté je spuštěn trénink systému pomocí vlastní fitness funkce, která ohodnocuje každou populaci. Ve fitness funkci jsou zpočátku vypnuty varovné hlášky matlabu, které se vypisují, pokud pro dané Iris parametry nebylo stanoveno žádné pravidlo, tím je docíleno rychlejšího a přehlednějšího chodu tréninku.

Ve fitness funkci jsou hodnoceny:

- Varovné zprávy – pokud je vypsána varovná hláška, že pro dané Iris parametry nebylo stanoveno žádné pravidlo, tak je přičtena vysoká hodnota (výchozí hodnota – 100)
- Chybné třídy – je spočítáno kolik tříd bylo špatně určeno a tato hodnota je přičtena k celkové fitness
- Počet pravidel – cílem je mít systém s co nejnižším počtem pravidel (kvůli přehlednosti), tudíž čím vyšší počet pravidel, tím je vyšší hodnota přičtena k fitness

- Chybné rozdíly – u chybně určených tříd je spočítán rozdíl mezi správně a chybně určenou třídou, čím vyšší tento rozdíl je, tím vyšší hodnota je přičtena k fitness. Toto hodnocení přispívá k vyšší úspěšnosti systému.

Po dokončení tréninku je spuštěn skript z předchozí podkapitoly „Iris_evaluace.mlx“, díky kterému dojde k ohodnocení natrénovaného systému. V poslední fázi dochází k uložení natrénovaného systému do aktuální složky.

8.2 Skripty pro EXsys-vibro dataset

V následujících podkapitolách jsou popsány skripty, které byly použity pro realizaci části, která se věnovala aplikaci fuzzy logiky na technický dataset (kap. 6 a 7). Každá následující podkapitola je pojmenována podle názvu daného skriptu, který lze naleznout v přílohách. Pořadí spuštění jednotlivých skriptů je dáno pořadím následujících podkapitol.

8.2.1 EXsys_zpracovani_dat.mlx

Skript slouží ke zpracování „raw“ dat ze dvou akcelerometrů. Celkem je k dispozici dvacet textových souborů – každý akcelerometr má dvě složky (x a y) a dohromady je pět tříd (bezporuchový stav, dynamická nevývaha první a druhé úrovně, statická nevývaha první a druhé úrovně). V každém textovém souboru je tisíc řádků a tisíc sloupců záznamů.

Ve skriptu dochází nejprve k načtení všech těchto souborů ze složky, uložení do jedné matice a následně jsou pro každý řádek jednotlivých textových souborů vytvořeny prediktory RMS, STD a PCA. Poté je k prediktorům přidán ještě fázový posun mezi prvním a druhým akcelerometrem. Dále dochází k vytvoření a uložení pěti tabulek (pět tříd) s patnácti sloupci (viz kap. 6) do složky „data“ v aktuálním adresáři.

Skript je připraven pro jednoduché přidání dalšího prediktoru, kdy stačí pouze odkomentovat řádky 38 a 61. Na těchto řádcích je vzorově připraven prediktor MEAN, v případě záměny prediktoru či přidání dalšího, musí být dodržena struktura matice. Poté budou skripty, které dále navazují, fungovat správným způsobem.

8.2.2 EXsys_grafy.mlx

Navazujícím na předchozí, je skript tvořící grafy zvolených prediktorů jednotlivých tříd, do kterého jsou ze začátku nahrány předchozí tabulky pomocí vlastní funkce „loadTabs“. Do funkce stačí zadat začínající znaky názvu tabulek a při jejím spuštění je nahráno pět tabulek ze složky data do „workspace“. Na výběr je ze dvou druhů grafů:

1) Hodnoty jednotlivých tříd jsou vykresleny na stejném číselném rozmezí osy x (1-1000).
2) Hodnoty jednotlivých tříd jsou vykresleny na různém číselném rozmezí osy x: bezporuchový stav (1-1000), dynamická nevývaha první úrovně (1001-2000) a tak dále.

Grafy jednotlivých prediktorů akcelerometrů a jejich složek jsou zobrazeny v subplotu, což je jedno velké okno, ve kterém jsou vedle sebe vykresleny grafy. Grafy jsou tak zobrazeny vedle sebe, či pod sebou, a předchází se tak zbytečnému vyskakování jednotlivých oken. Pouze grafy fázových posunů jsou zobrazeny zvláště.

V konečné části je vytvořena tabulka minimálních, maximálních a průměrných hodnot jednotlivých prediktorů a tříd.

8.2.3 EXsys_tvorba_FIS.mlx

Skript slouží k automatické tvorbě FIS systému pomocí dostupných dat. Zpočátku jsou opět nahrány vytvořené tabulky pomocí funkce „loadTabs“. Dále je možné si, pomocí číselného zadání, vybrat, ze kterých sloupců těchto tabulek má být systém vytvořen, tedy z jakých snímačů, složek či prediktorů. Z vybraných sloupců jsou poté, pro jednotlivé třídy, získány minimální, průměrné a maximální hodnoty, na kterých je systém vystavěn. Dále je na výběr z pěti tvarů funkcí příslušnosti: Gaussian, Tringular, Trapezoidal, Two-sided Gaussian a Generalized bell. Zvolení tvaru funkce příslušnosti je opět pomocí číselného zadání do proměnné (hodnoty 1–5). V systému je pak nastaveno několik následujících hodnot:

- Počet vstupů (inputů) – je dán počtem vybraných sloupců.
- Rozptyl (range) – pro jednotlivé vstupy (inputy), tedy vybrané sloupce, je zadán rozptyl, který je určen minimální a maximální hodnotou zvětšenou či zmenšenou o 5 %.
- Tvar funkce příslušnosti – zmíněné tvary funkcí příslušnosti z předchozích vět jsou vždy dány číselnými parametry, určení těchto parametrů je stejné jako v kap. 5.4.1 (pomocí minimálních, průměrných a maximálních hodnot). Jediný rozdíl je u krajních parametrů tvarů Triangular, Trapezoidal a Two-sided Gaussian, ty jsou v tomto případě, namísto vysoké hodnoty (1000 nebo -1000), dány zvětšením (zmenšením) minimální nebo maximální hodnoty o 10 %.
- Výstup (output) – každé třídě je přiřazeno číslo jedna až pět: 1 – bezporuchový stav, 2 – dynamická nevývaha první úrovně, a tak dále.
- Seřazení dle průměrné hodnoty – v každém jednom vstupu (inputu) jsou seřazeny třídy podle průměrné hodnoty (od nejmenší po největší). Nejmenší hodnotě je poté přiřazeno pojmenování „small“, druhé nejmenší „medium small“, střední „medium“ druhé největší „medium big“ a největší „big“. Tyto pojmenování jsou dále používány v lingvistických pravidlech.
- Pojmenování vstupů (inputů) – po seřazení všech vstupů a tříd následuje pojmenování jednotlivých vstupů (inputů) z důvodu přehlednosti systému.
- Pojmenování výstupu (outputu) – ve všech případech stejná hodnota: „Class“.
- Přidání lingvistických pravidel – zpočátku byla všechna pravidla odstraněna a bylo zamezeno tvorbě nových pravidel, kvůli zrychlení tvorby systému. Počet pravidel je roven počtu tříd, tedy pěti. Pravidla jsou přidávána pomocí číselné řády závislé na průměrném seřazení tříd v jednotlivých vstupech (inputech). Jelikož jsou třídy v jednotlivých vstupech (inputech) seřazeny z předchozích kroků, tak stačí pouze jednoduchý for cyklus s čísly daných tříd.

Ke konci dochází k uložení vytvořeného FIS systému do aktuální složky pod názvem „fis_Exsys_X“. Kde X je číslo podle vybraného tvaru funkce příslušnosti ze začátku skriptu.

8.2.4 EXsys_evaluace.mlx

Skript sloužící k ověření úspěšnosti vytvořených systémů (natrénovaných i nenatrénovaných). Ze začátku je zapotřebí načíst tři soubory: tabulky, ze kterých budou brána data k hodnocení, vytvořenou matici s čísly sloupců k hodnocení a vytvořený FIS model. Z nahraných tabulek a vybraných sloupců jsou všechna data naskládána do jedné matice, ve které jsou data v jednotlivých sloupcích seřazena podle zařazení do tříd jedna až pět. Do této matice je poté vložen sloupec s čísly jednotlivých tříd. Ke konci dochází k vykreslení matice záměn s dobře a špatně zařazenými třídami a vypsání tabulky špatně určených hodnot. V té jsou uvedeny

hodnoty jednotlivých prediktorů, jejich správná třída a hodnota, se kterou byla data špatně zařazena.

8.2.5 EXsys_trenink.mlx

Tento skript slouží k trénování systémů vytvořených v předešlých podkapitolách. Trénování probíhá pomocí genetického algoritmu a vlastní fitness funkce, která je popsána v kapitole 8.2.8. Zpočátku je zapotřebí nahrát FIS systém, který má být trénován a tabulky a sloupce, ze kterých se berou data potřebná k trénování. Dále lze pomocí proměnné, do které se zadává procentuální počet tréninkových dat, rozdělit data na trénovací a testovací sadu (výchozí hodnota 50 %) a zadat počet generací a populací genetického algoritmu. Poté jsou data zamíchána a rozdělena do tréninkových a testovacích tabulek, které jsou následně uloženy do složky data v aktuálním adresáři. Z těchto tabulek je vytvořena matice s potřebnými sloupci a třídami, která je následně nahrána do genetického algoritmu.

Další část skriptu se věnuje nastavení genetického algoritmu. Nejprve jsou načteny proměnné vstupů (inputů) a pravidel, která mají být trénována. Poté jsou nahrány zvolené počty generací a populací ze začátku skriptu, nastaven maximální počet stejných generací a maximální počet pravidel. Následuje trénování systému pomocí funkce „tunefis“, ve které je každá populace a generace ohodnocena dle vlastní fitness funkce. Po dokončení tréninku je spočítán a vypsán průměrný čas tréninku jedné generace a zobrazeny matice záměn pro trénovací a testovací sadu. Pod každou maticí záměn je také uvedena procentuální úspěšnost systému na tréninkové a testovací sadě. Ke konci skriptu dochází k uložení natrénovaného modelu do aktuální složky.

8.2.6 EXsys_multi_trenink.mlx

Jedná se o obdobu předchozího skriptu EXsys_trenink.mlx, s tím rozdílem, že tento skript slouží pro trénink více systémů najednou (postupně, bez zásahu uživatele) – konkrétně dvaceti pěti systémů (pět systémů pro každý tvar funkce příslušnosti: Gaussian, Triangular, Trapezoidal, Two-sided Gaussian a Generalized bell). Pro možnost spuštění tohoto skriptu je zapotřebí vytvořit, pomocí skriptu EXsys_tvorba_FIS, pět systémů, které mají být trénovány. Tyto soubory je potřeba přesunout do složky, která je vybrána ze začátku skriptu, výchozí název složky je „multi_trenink“. Do této složky jsou také ukládány jednotlivé natrénované systémy. Poté vše probíhá způsobem popsaným v předchozím skriptu, až do doby vyhodnocování systémů. V tomto případě jsou hodnocení systémů zaznamenávány do matice, ze které je ke konci skriptu vytvořena tabulka.

8.2.7 EXsys_trenink_K_fold.mlx

Tento skript slouží k trénování FIS systému pomocí K-fold křížové validace (více o K-fold křížové validaci v kap. 7.2.5), kdy je trénováno pět systémů pro k-fold pět a pět systémů pro k-fold deset. Trénování probíhá opět pomocí genetického algoritmu a vlastní fitness funkce. Zpočátku je zapotřebí zadat název složky, do které mají být vytrénované systémy ukládány, nahrát FIS systém a nahrát tabulky a sloupce, ze kterých se berou data potřebná k trénování. Opět lze zvolit počet generací a populací (výchozí nastavení: počet generací 25, počet populací 125). Z nahraných tabulek a sloupců je vytvořena matice se všemi daty a výsledky. Poté dochází k rozdělení dat na k-foldy, tedy tréninkové a testovací sady a následný trénink pomocí genetického algoritmu a vlastní fitness funkce, stejně jako ve skriptu EXsys_trenink.mlx. U každého natrénovaného foldu jsou zhodnoceny úspěšnosti trénovacích a testovacích sad a všech dat. Po dokončení tréninku všech foldů jsou tyto hodnoty zprůměrovány, zapsány do matice a

dochází k uložení modelu do zvoleného adresáře. Tento postup je desetkrát (pět a pět k-foldů) opakován a po dokončení dojde k vytvoření tabulky obsahující hodnoty úspěšností.

8.2.8 fuzzyFitness.mlx

V tomto skriptu je fitness funkce, využívána předešlými skripty. Cílem fitness funkce je dosáhnout co nejnižší hodnot, čím je hodnota nižší, tím je systém úspěšnější. Ze začátku funkce jsou nastaveny čtyři proměnné, díky kterým lze nastavovat výslednou hodnotu fitness. Před evaluací jsou vypnuty varovné zprávy, jelikož zpomalují a znepřehledňují chod tréninku.

Kritéria hodnocení fitness:

- Varovná zpráva – pokud je vypsána varovná zpráva, že pro dané prediktory není určeno žádné pravidlo, tak je k fitness přičtena vysoká hodnota (ve výchozím nastavení 1000).
- Špatně zařazená třída – počet špatně zařazených tříd je promítnut do výsledné hodnoty (výchozí nastavení 1)
- Počet pravidel – čím vyšší počet pravidel, tím nižší přehlednost systému (výchozí nastavení 0.1)
- Počet bezporuchových tříd – čím vyšší počet bezporuchových tříd, tím vyšší fitness (výchozí hodnota 0).

Část fitness s počtem bezporuchových tříd nemá ve výchozím nastavení vliv na fitness hodnotu, kvůli dosažení co nejlepších výsledků. Nicméně je možnost tuto hodnotu zvýšit a docílit tím tak méně špatně určených bezporuchových tříd.

8.2.9 loadTabs.mlx

Funkce vytvořená pro nahrávání vytvořených tabulek ve skriptu EXsys_zpracovani_dat.mlx. Ze složky „data“ je nahráno pět tabulek, začínajících na zadané znaky, do workspace. Pokud je ve složce „data“ více než pět složek začínajících na zadané znaky, tak dochází k vypsání erroru a daný skript dále nepokračuje. Tato funkce je využívána ve všech skriptech, kromě prvního v pořadí (v tom jsou nahrávané tabulky vytvořeny).

9 ZÁVĚR

Diplomová práce se věnuje aplikaci fuzzy logiky na dataset, jak z oblasti technické, tak i netechnické. Práce je rozdělena na dvě části, teoretickou a praktickou. V části teoretické je provedena rešerše a v praktické části jsou tyto a další znalosti aplikovány.

Začátek teoretické části se věnuje technické diagnostice, jejímu popisu a obeznámení s diagnostickými metodami. První z uvedených metod je vibrodiagnostika, což je metoda, která se využívá v praktické části této práce. Na tuto kapitolu navazuje část věnující se popisu fuzzy logiky, kde jsou popsány hlavní části při tvorbě fuzzy systémů. Poslední kapitola teoretické části se věnuje genetickému algoritmu, který byl vybrán jako nejvhodnější z využívaných algoritmů při trénování systémů na datech.

Na část teoretickou navazuje praktická, ve které je nejprve fuzzy logika aplikována na Iris dataset, což je nejpoužívanější soubor dat z oblasti strojového učení. Dataset obsahuje celkem 150 záznamů třech druhů kosatců, kde každý z nich je rovnoměrně zastoupen. Z těchto záznamů bylo vytvořeno několik fuzzy systémů, u kterých byl ještě prostor pro zlepšení, a tak byly dále ještě trénovány pomocí genetického algoritmu. Nejúspěšnější ze systémů dosáhl vysoké úspěšnosti 99,33 %, což znamená, že pouze jeden druh kosatce byl špatně zařazen. Získané znalosti z tvorby fuzzy systémů, na základě dat kosatců, byly aplikovány na technický dataset, konkrétně dataset, s názvem EXsys-vibro, tvořený daty ze dvou akcelerometrů. Data z těchto snímačů byly zpracovány za pomocí čtyřech prediktorů a následně vykresleny do grafů, kde v každém z nich bylo zobrazeno pět možných stavu stroje. Již z vykreslených grafů bylo možno vidět, že pokud se použijí data z obou akcelerometrů, tak nebude problém s rozeznáním jednotlivých stavů. Proto byl vytvořen jeden jednoduchý systém se 100 % úspěšností a práce se spíše věnuje rozeznání stavů na základě dat pouze jednoho akcelerometru. Bylo tedy vytvořeno několik systémů s různými přístupy k učení, zde je potřeba podotknout, že systémy jsou tvořeny automaticky na základě vypracovaného skriptu, na rozdíl od předešlé části o Iris datasetu. Jako nejúspěšnější systémy se jevily systémy, kde byla použita K-fold křížová validace – nejúspěšnější dosáhl průměrné úspěšnosti 91,12 % na tréninkových datech. K závěru praktické části práce jsou dosažené výsledky porovnány s klasifikačními metodami aplikovanými na stejný dataset, z tohoto porovnání vyšla velká většina vytvořených systémů s vyšší úspěšností než systémy tvořené pomocí klasifikačních metod.

Tato diplomová práce a její dosažené výsledky dokazují, že i fuzzy logiku lze využít v oblasti strojového učení. Její aplikace je jednoduchá a ve vytvořených systémech se dá jednoduše orientovat, zejména v systémech, kde se optimalizovaly pouze parametry funkcí příslušnosti (kap. 7.2.3 a 7.2.4). Práce může tedy sloužit jako návod či inspirace pro praxi.

10 SEZNAM POUŽITÝCH ZDROJŮ

[1] TOMEH, Elias. *Technická diagnostika: vibrační diagnostika strojů a zařízení*. Liberec: Technická univerzita v Liberci. ISBN 978-80-7494-174-0.

[2] LEGÁT, Václav. *Management a inženýrství údržby*. Druhé doplněné vydání. Praha: Kamil Mařík – Professional Publishing, 2016, 622 stran, IV strany obrazových příloh: ilustrace (některé barevné). ISBN 978-80-7431-163-5.

[3] BLATA, Jan a Janusz JURASZEK. *Metody technické diagnostiky: teorie a praxe = Metody diagnostyki technicznej: teorie a praktyka*. 1. vyd. Ostrava: Vysoká škola báňská – Technická univerzita Ostrava, 2013. ISBN 978-80-248-2997-5.

[4] S. RAJASEKARAN a V. P. G.A. *Neural Network, Fuzzy Logic, and Genetic Algorithms - Synthesis and Application*. Prentice Hall. 2005.

[5] RUSSELL, Stuart a Peter NORVIG. *Artificial intelligence: a modern approach*. 3rd ed. Upper Saddle River: Prentice Hall, 2009. ISBN 978-0-13-604259-4.

[6] DERNONCOURT, Franck. *Introduction to fuzzy logic* [online]. 2013 [cit. 2023-03-21]. Dostupné z: https://www.researchgate.net/publication/267041266_Introduction_to_fuzzy_logic

[7] HÁJEK, Petr. *Fuzzy logika* [online]. 2011 [cit. 2023-03-21]. Dostupné z: https://moodle.sspbrno.cz/mod/resource/view.php?id=2132

[8] J. KLIR, George a Bo YUAN. *Fuzzy sets and fuzzy logic: theory and applications*. Pearson College Div, 1995. ISBN 0-13-101171-5.

[9] KOKKINOS, Konstantinos, Vayos KARAYANNIS, Evangelia LAKIOTI a Konstantinos MOUSTAKAS. *Exploring social determinants of municipal solid waste management: survey processing with fuzzy logic and self-organized maps*. Environmental Science and Pollution Research [online]. 2019, 26(35), 35288-35304 [cit. 2023-03-22]. ISSN 0944-1344. Dostupné z: doi:10.1007/s11356-019-05506-2

[10] *Fuzzy Membership Functions* [online]. [cit. 2023-03-22]. Dostupné z: https://functionbay.com/documentation/onlinehelp/default.htm#!Documents/fuzzymembershipfunctions.html

[11] BLEJ, Mohammed a Mostafa AZIZI. *Comparison of Mamdani-type and Sugeno-type fuzzy inference systems for fuzzy real time scheduling* [online]. 2016 [cit. 2023-03-22]. Dostupné z: https://www.researchgate.net/publication/311795100_Comparison_of_Mamdani-type_and_Sugeno-type_fuzzy_inference_systems_for_fuzzy_real_time_scheduling

[12] *Fuzzy Logic Control* [online]. [cit. 2023-03-23]. Dostupné z: https://nitsri.ac.in/Department/Computer%20Science%20&%20Engineering/FuzzyInference.pdf

[13] *Mamdani and Sugeno Fuzzy Inference Systems* [online]. [cit. 2023-03-23]. Dostupné z: https://www.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html

[14] KAUR, Arshdeep a Amrit KAUR. *Comparison of Mamdani-Type and Sugeno-Type Fuzzy Inference Systems for Air Conditioning System* [online]. [cit. 2023-04-03]. Dostupné z: https://www.ijsce.org/wp-content/uploads/papers/v2i2/B0612042212.pdf

[15] *Sugeno Fuzzy Model* [online]. [cit. 2023-04-03]. Dostupné z: https://researchhubs.com/post/engineering/fuzzy-system/takagi-sugeno-fuzzy-model.html

[16] CUI, Yong. *The Iris Dataset — A Little Bit of History and Biology* [online]. 2020 [cit. 2023-04-04]. Dostupné z: https://towardsdatascience.com/the-iris-dataset-a-little-bit-of-history-and-biology-fb4812f5a7b5

[17] KARAMI, A., G. H. ROSHANI, A. SALEHIZADEH a E. NAZEMI. *The Fuzzy Logic Application in Volume Fractions Prediction of the Annular Three-Phase Flows*. Journal of Nondestructive Evaluation [online]. 2017, 36(2) [cit. 2023-04-05]. ISSN 0195-9298. Dostupné z: doi:10.1007/s10921-017-0415-7

[18] *Genetické algoritmy* [online]. [cit. 2023-04-16]. Dostupné z: https://poli.cs.vsb.cz/edu/isy/down/ga.pdf

[19] VOLNÁ, Eva. *Evoluční algoritmy a neuronové sítě* [online]. 2012 [cit. 2023-04-16]. Dostupné z: https://web.osu.cz/~Volna/Evolucni_algoritmy_a_neuronove_site.pdf

[20] KEPRT, Aleš a Václav SNÁŠEL. *Genetické algoritmy pro redukci dimenze a analýzu binárních dat* [online]. 2005 [cit. 2023-04-16]. Dostupné z: https://www.researchgate.net/publication/306079634_Geneticke_algoritmy_pro_redukci_dimenze_a_analyzu_binarnich_dat

[21] ZUTH, Daniel. *Evoluční algoritmy* [Přednášky]. Brno: VUT FSI, 2023

[22] *Evoluční algoritmy* [online]. [cit. 2023-04-16]. Dostupné z: https://sorry.vse.cz/~berka/docs/izi456/kap_5.5.pdf

[23] *Genetic Algorithm Options* [online]. [cit. 2023-04-16]. Dostupné z: https://www.mathworks.com/help/gads/genetic-algorithm-options.html

[24] A. ROJAS, Juan, Helbert E. ESPITIA a Lilian A. BEJARANO. *Design and Optimization of a Fuzzy Logic System for Academic Performance Prediction* [online]. 2021 [cit. 2023-04-17]. Dostupné z: https://www.mdpi.com/2073-8994/13/1/133

[25] ZUTH, Daniel a Tomáš MARADA. *Using Artificial Intelligence to Determine the Type of Rotary Machine Fault*. Mendel [online]. Institute of Automation and Computer Science, Brno University of Technology, 2018, 24(2), 49–54 [cit. 2023-04-18]. ISSN 2571-3701. Dostupné z: doi:10.13164/mendel.2018.2.049

[26] ZVORNICANIN, Enes. *How to Use K-Fold Cross-Validation in a Neural Network?* [online]. 2023 [cit. 2023-04-18]. Dostupné z: https://www.baeldung.com/cs/k-fold-cross-validation

[27] BROWNLEE, Jason. *How to Configure k-Fold Cross-Validation* [online]. 2020 [cit. 2023-04-18]. Dostupné z: https://machinelearningmastery.com/how-to-configure-k-fold-cross-validation/

[28] *Defuzzification Methods* [online]. [cit. 2023-04-25]. Dostupné z: https://www.mathworks.com/help/fuzzy/defuzzification-methods.html

11 SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK

11.1 Seznam zkratek

SL	Délka sepalu (Sepal Length)
SW	Šířka sepalu (Sepal Width)
PL	Délka petalu (Petal Length)
PW	Šířka petalu (Petal Width)
RMS	Efektivní hodnota (Root Mean Square)
STD	Výběrová směrodatná odchylka (Standart Deviation)
PCA	Analýza hlavních komponent (Prncipal Component Analysis)
FP	Fázový posun
OK	Bezporuchový stav
NevDyn1	Dynamická nevývaha první úrovně
NevDyn2	Dynamická nevývaha druhé úrovně
NevSt1	Statická nevývaha první úrovně
NevSt2	Statická nevývaha druhé úrovně
FIS	Fuzzy inferenční systém
V1-5	Číslo verze (1 až 5)

11.2 Seznam symbolů

a_i	[-]	Koeficient Sugeno systému
b_i	[-]	Koeficient Sugeno systému
c_i	[-]	Koeficient Sugeno systému
F_1	[-]	Funkce příslušnosti pro daný vstup
F_2	[-]	Funkce příslušnosti pro daný vstup
N	[-]	Počet pravidel
n_i	[-]	Počet vstupů systému
n_{maxr}	[-]	Maximální stanovený počet pravidel systému
n_{mfs}	[-]	Počet funkcí příslušnosti v jednom vstupu
n_o	[-]	Počet výstupů
n_p	[-]	Počet parametrů funkce příslušnosti
p_p	[-]	Počet pravidel
$u(x_i)$	[-]	Hodnota příslušnosti
WA	[-]	Vážený průměr
w_i	[-]	Hodnota odvozená z antecedentu pravidla

x	[-]	Vstupní hodnota Sugeno systému
x_i	[-]	Bod diskurzu
y	[-]	Vstupní hodnota Sugeno systému
z_i	[-]	Výstup pravidel Sugeno systému

11.3 Seznam tabulek

Tab. 1 Popis a vzorce tvarů funkcí příslušnosti z obr. 2 [10] 20
Tab. 2 Nastavení parametrů Triangular křivky pro délku sepalu 30
Tab. 3 Nastavení parametrů Trapezoidal křivky pro délku sepalu 30
Tab. 4 Nastavení parametrů Gaussian křivky pro délku sepalu 31
Tab. 5 Nastavení parametrů Two-sided Gaussian křivky pro délku sepalu 31
Tab. 6 Nastavení parametrů Generalized bell křivky pro délku sepalu 31
Tab. 7 Porovnání systémů s různými tvary funkcí příslušnosti 32
Tab. 8 Porovnání natrénovaných systémů s různými tvary funkcí příslušnosti 33
Tab. 9 Zobrazení prvního řádku tabulky bezporuchového stavu 39
Tab. 10 Úspěšnosti tréninku parametrů a pravidel s vyšším počtem generací a populací 44
Tab. 11 Úspěšnosti tréninku parametrů a pravidel se sníženým počtem generací a populací 45
Tab. 12 Úspěšnosti tréninku parametrů s vyšším počtem generací a populací 47
Tab. 13 Úspěšnosti tréninku parametrů se sníženým počtem generací a populací 47
Tab. 14 Vyhodnocení procentuální úspěšnosti systému s K-fold křížovou validací 49
Tab. 15 Srovnání FIS systémů a klasifikačních metod 51

11.4 Seznam obrázků

Obr. 1 Klasická teorie množin je součástí teorie fuzzy množin [6] 19
Obr. 2 Tvary funkcí příslušnosti v Matlabu [9] 21
Obr. 3 Mamdani metoda [12] 22
Obr. 4 Sugeno metoda [12] 22
Obr. 5 Diagram pravidel Sugeno systému [13] 23
Obr. 6 Deffuzifikace Mamdani systému [12] 24
Obr. 7 Deffuzifikace Sugeno systému [12] 24
Obr. 8 Schéma základních pojmů genetických algoritmů [21] 25
Obr. 9 Schéma křížení genetického algoritmu [21] 26
Obr. 10 Příklad mutace genetického algoritmu 26
Obr. 11 Fotografie Iris druhů [16] 27
Obr. 12 Grafy a tabulky: a) délek sepalu, b) šířek sepalu 28
Obr. 13 Grafy a tabulky: a) délek petalu, b) šířek petalu 28
Obr. 14 Znázornění Fuzzy Iris systému 29
Obr. 15 Zobrazení nastavení Gaussian funkce příslušnosti délky sepalu (SL) 29
Obr. 16 Pravidla předpřipravených systémů 32
Obr. 17 Porovnání tvarů funkcí příslušnosti před (vlevo) a po (vpravo) tréninku 35
Obr. 18 Matice záměn konečného systému 35
Obr. 19 Tabulka z Matlabu se špatně určeným druhem 35

Obr. 20 Diagnostický model: a) zobrazení směrů os, b) pohled zepředu – zobrazení pozic snímačů [25] 36
Obr. 21 Schéma umístění závaží na kotoučích: a) bezporuchový stav, b) statická nevývaha, c) dynamická nevývaha [25] 37
Obr. 22 Uchycení akcelerometru [25] 37
Obr. 23 Vizualizace náhodných řádků z jednotlivých měření [25] 38
Obr. 24 Zobrazení grafů pro jednotlivé stavy 40
Obr. 25 Grafy fázových posunů 40
Obr. 26 Tabulka pravidel prvního FIS systému 41
Obr. 27 První FIS systém: a) schéma systému, b) matice záměn 41
Obr. 28 Detailní zobrazení překrytí prvních třech tříd 42
Obr. 29 Tabulka pravidel FIS systému s prediktory prvního snímače 42
Obr. 30 Druhý FIS systém: a) schéma systému, b) matice záměn 43
Obr. 31 Průběh tréninku systému Trapezoidal V5 45
Obr. 32 Průběh tréninku systému Generalized bell V1 46
Obr. 33 Průběh tréninku systému Generalized bell V2 47
Obr. 34 Schéma K-fold křížové validace [26] 48
Obr. 35 Matice záměn nejúspěšnějšího systému s K-fold křížovou validací 49
Obr. 36 Porovnání úspěšnosti klasifikačních metod ze článku [25] 50

12 SEZNAM PŘÍLOH

Příloha A Vytvořené FIS systémy

Příloha B Složka s daty a skripty

A VYTVOŘENÉ FIS SYSTÉMY

Příloha obsahuje složku s nejúspěšnějšími FIS systémy (hierarchie složky je zobrazena níže). Za Iris dataset je přiložen jeden nejúspěšnější systém a pět systému s různými tvary funkcí příslušnosti v základním nastavení. Za technický dataset je přiloženo celkem 16 systémů – jeden z kapitoly 7.1, kdy byly použita data z druhého snímače a 15 systémů z tabulky celkového srovnání (kapitola 7.3).

Příloha A-Vytvořené FIS systémy

- → EXsys-vibro
  - → Úspěšnosti tréninku parametrů a pravidel s vyšším počtem generací a populací
    - → fis_Exsys_trained_Generalized bell_V4.fis
    - → fis_Exsys_trained_Trapezoidal_V4.fis
    - → fis_Exsys_trained_Trapezoidal_V5.fis
  - → Úspěšnosti tréninku parametrů a pravidel se sníženým počtem generací a populací
    - → fis_Exsys_trained_Generalized bell_V1.fis
    - → fis_Exsys_trained_Generalized bell_V3.fis
    - → fis_Exsys_trained_Trapezoidal_V1.fis
  - → Úspěšnosti tréninku parametrů s vyšším počtem generací a populací
    - → fis_Exsys_trained_Generalized bell_V1.fis
    - → fis_Exsys_trained_Generalized bell_V2.fis
    - → fis_Exsys_trained_Generalized bell_V5.fis
  - → Úspěšnosti tréninku parametrů se sníženým počtem generací a populací
    - → fis_Exsys_trained_Generalized bell_V2.fis
    - → fis_Exsys_trained_Generalized bell_V3.fis
    - → fis_Exsys_trained_Generalized bell_V4.fis
  - → Vyhodnocení procentuální úspěšnosti systému s K-fold křížovou validací
    - → fis_Exsys_trained_k10_V1.fis
    - → fis_Exsys_trained_k10_V2.fis
    - → fis_Exsys_trained_k10_V3.fis
  - → fis_Exsys_druhySnimac.fis
- → Iris
  - → fis_Iris_Gaussian.fis
  - → fis_Iris_Gaussian_trained.fis
  - → fis_Iris_Generalized bell.fis
  - → fis_Iris_Trapezoidal.fis
  - → fis_Iris_Triangular.fis
  - → fis_Iris_Two_sided Gaussian.fis

B SLOŽKA S DATY A SKRIPTY

Obsahem přílohy je složka, ve které jsou dvě podsložky: Iris a EXsys-vibro. V podsložce EXsys-vibro jsou další tři podsložky: data, k_fold a multi_trenink, v podsložce data jsou textové soubory s naměřenými daty a další dvě podsložky jsou pouze pracovní (slouží k ukládání natrénovaných systémů). Dále jsou v podsložce EXsys-vibro vypracované skripty a funkce (popsány v kap. 8.2). V podložce Iris se nachází tři skripty (popsány v kap. 8.1).

Příloha B-Složka s daty a skripty

- → EXsys-vibro
  - → data
    - → 1x_nevyvahaDyn-1.txt
    - → 1x_nevyvahaDyn-2.txt
    - → 1x_nevyvahaSt-1.txt
    - → 1x_nevyvahaSt-2.txt
    - → 1x_OK.txt
    - → 1y_nevyvahaDyn-1.txt
    - → 1y_nevyvahaDyn-2.txt
    - → 1y_nevyvahaSt-1.txt
    - → 1y_nevyvahaSt-2.txt
    - → 1y_OK.txt
    - → 2x_nevyvahaDyn-1.txt
    - → 2x_nevyvahaDyn-2.txt
    - → 2x_nevyvahaSt-1.txt
    - → 2x_nevyvahaSt-2.txt
    - → 2x_OK.txt
    - → 2y_nevyvahaDyn-1.txt
    - → 2y_nevyvahaDyn-2.txt
    - → 2y_nevyvahaSt-1.txt
    - → 2y_nevyvahaSt-2.txt
    - → 2y_OK.txt
  - → k_fold
  - → multi_trenink
  - → EXsys_evaluace.mlx
  - → EXsys_grafy.mlx
  - → EXsys_multi_trenink.mlx
  - → EXsys_trenink.mlx
  - → EXsys_trenink_K_fold.mlx
  - → EXsys_tvorba_FIS.mlx
  - → EXsys_zpracovani_dat.mlx
  - → fuzzyFitness.mlx
  - → loadTabs.mlx
- → Iris
  - → Iris_evaluace.mlx
  - → Iris_grafy.mlx
  - → Iris_tunefis.mlx