



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 LOŽISEK S VYUŽITÍM STROJOVÉHO UČENÍ

BEARING DIAGNOSTICS USING MACHINE LEARNING

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. KRISTÝNA ZONYGOVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. DANIEL ZUTH, Ph.D.

BRNO 2022

Zadání diplomové práce

Ústav: Ústav výrobních strojů, systémů a robotiky
Studentka: **Bc. Kristýna Zonygová**
Studijní program: Kvalita, spolehlivost a bezpečnost
Studijní obor: bez specializace
Vedoucí práce: **Ing. Daniel Zuth, Ph.D.**
Akademický rok: 2021/22

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 ložisek s využitím strojového učení

Stručná charakteristika problematiky úkolu:

Práce se bude zabývat tvorbou algoritmů pro diagnostiku ložisek na základě metod umělé inteligence. Zdrojové datasey budou simulované data, reálné veřejné datasey a případně měření z praxe. Cílem bude otestovat a porovnat několik metod a vytvořit algoritmy pro možné nasazení v praxi. Programovacím prostředím bude Python.

Cíle diplomové práce:

Rešerše v oblasti technické diagnostiky ložisek.
Popis poruch ložisek a jejich projev v diagnostickém signálu.
Popis použitelných metod umělé inteligence.
Popsat a vytvořit algoritmy.
Porovnat výsledky z různých metod a různých dataseů.
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 2021/22

V Brně, dne

L. S.

doc. Ing. Petr Blecha, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Diplomová práce se zabývá použitím metod umělé inteligence za účelem klasifikace poruch ložiska. Jsou zde popsány klasifikační metody SVC (Support Vector Classification), KNN (K-Nearest Neighbors Classifier), RFC (Random Forest Classifier) a CNN (Convolutional Neural Network), které jsou testovány na datech vibračního signálu kuličkového ložiska ze dvou různých datasetů. Všechny metody dosahují poměrně dobrých výsledků klasifikace (od 94,1 % do 99,8 %). Součástí jsou také skripty v programovém prostředí Python, které využívají knihovny s volnou licencí. Ty poskytují možnost natrénování klasifikačních metod (SVC, KNN, RFC nebo CNN) na vlastních datech, nebo využití již natrénovaných modelů.

Summary

The Master's thesis deals with the use of artificial intelligence methods in order to classify bearing failures. The SVC (Support Vector Classification), KNN (K-Nearest Neighbors Classifier), RFC (Random Forest Classifier) and CNN (Convolutional Neural Network) classification methods are described and tested on ball-bearing vibration signals from two different datasets. All methods achieve quite well accuracy (from 94.1 % to 99.8 %). Scripts in the Python programming environment that use libraries with free-licenses are also included. They provide the possibility of training classification methods (SVC, KNN, RFC or CNN) on your own data, or the use of already trained models.

Klíčová slova

strojové učení, diagnostika ložisek, prediktory, Support Vector Classification, K-Nearest Neighbors Classifier, Random Forest Classifier, konvoluční neuronová síť

Keywords

machine learning, bearing diagnostics, predictors, Support Vector Classification, K-Nearest Neighbors Classifier, Random Forest Classifier, Convolutional Neural Network

ZONYGOVÁ, K. *Diagnostika ložisek s využitím strojového učení*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2022. 63 s. Vedoucí diplomové práce Ing. Daniel Zuth, Ph.D.

Prohlašuji, že předložená diplomová práce je původní a zpracovala jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušila autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

Bc. Kristýna Zonygová

Děkuji vedoucímu diplomové práce Ing. Danielu Zuthovi, Ph.D. za odborné vedení, podnětné rady a odbornou pomoc při zpracování mé práce.

Bc. Kristýna Zonygová

Obsah

1	Úvod	3
2	Technická diagnostika	4
2.1	Metody technické diagnostiky	4
2.1.1	Vibrodiagnostika	4
2.1.2	Tribodiagnostika	6
2.1.3	Termodiagnostika	6
2.1.4	Akustická diagnostika	6
3	Poruchy ložisek a jejich projev v diagnostickém signálu	7
3.1	Frekvence ložiskových závad (FLZ)	8
3.1.1	Frekvence závady vnitřního kroužku	8
3.1.2	Frekvence závady vnějšího kroužku	8
3.1.3	Frekvence závady klece ložiska	9
3.1.4	Frekvence závady valivých těles	9
4	Data a jejich předzpracování	10
4.1	Dataset A - Dataset Paderborn univerzity	10
4.2	Dataset B - Universidade Federal do Rio de Janeiro	11
4.3	Frekvence ložiskových závad (FLZ) použitých datasetů	12
4.3.1	FLZ - Dataset A	12
4.3.2	FLZ - Dataset B	13
5	Použité prediktory	14
5.1	Standardní směrodatná odchylka výběru (s)	14
5.2	Efektivní hodnota (RMS)	14
5.3	Činitel výkmitu ($crest\ faktor - CF$)	14
5.4	Parametr $K(t)$	15
5.5	Kurtosis metoda	15
5.6	Frekvence s největší amplitudou	16
5.7	Fourierova transformace (FT)	16
6	Metody umělé inteligence	18
7	Použité metody klasifikace dat	23
7.1	Support Vector Classification - klasifikace metodou podpůrných vektorů	23
7.2	K-Nearest Neighbors Classifier - klasifikace k-nejbližšími sousedy	23
7.3	Random Forest Classifier - klasifikace náhodným lesem	24
7.4	Optimalizace parametrů	24
7.5	Konvoluční neuronová síť - Convolutional Neural Network (CNN)	25
7.6	Matice záměn (Confusion matrix)	27

8	Aplikace metod a výsledky	28
8.1	Předzpracování dat a výpočet prediktorů - Dataset A	28
8.1.1	Vypočítané prediktory	28
8.1.2	Standardizace prediktorů a optimalizace parametrů	32
8.2	Výsledky metod klasifikace dat - Dataset A	32
8.2.1	Support Vector Classifier - metoda podpůrných vektorů	33
8.2.2	K-Neighbors Classifier - klasifikace podle k nejbližších sousedů . . .	34
8.2.3	Random Forrest Classifier - klasifikace metodou náhodného lesa . .	35
8.3	Konvoluční neuronová síť (CNN) - Dataset A	35
8.3.1	CNN - model 1	36
8.3.2	CNN - model 2	38
8.3.3	CNN - model 3	39
8.3.4	CNN - model 4	39
8.4	Předzpracování dat a výpočet prediktorů - Dataset B	40
8.4.1	Vypočítané prediktory	41
8.4.2	Standardizace prediktorů a optimalizace parametrů	44
8.5	Výsledky metod klasifikace dat - Dataset B	45
8.5.1	Support Vector Classifier - metoda podpůrných vektorů	46
8.5.2	K-Neighbors Classifier - klasifikace podle k nejbližších sousedů . . .	47
8.5.3	Random Forrest Classifier - klasifikace metodou náhodného lesa . .	48
8.6	Konvoluční neuronová síť (CNN) - Dataset B	49
8.6.1	CNN - model 1	49
8.6.2	CNN - model 2	50
8.6.3	CNN - model 3	51
8.6.4	CNN - model 4	51
8.7	Shrnutí výsledků	52
9	Závěr	53
10	Seznam použitých zkratk a symbolů	58
11	Seznam příloh	59
A	Příloha - seznam a popis skriptů	60
B	Příloha	63

1. Úvod

Umělá inteligence se začíná čím dál více prosazovat i v oblasti technické diagnostiky. Různé metody klasifikace jsou schopné po natrénování automaticky klasifikovat vkládaná data, což může značně usnadňovat práci a umožnit přenechání rutinních činností na práci stroje, která je mnohdy také rychlejší a nepodléhá chybám lidského faktoru. Nicméně je třeba mít na paměti, že predikce těchto metod nemusí být vždy správné a existuje určité procento chybné klasifikace dat.

V této práci budou představeny dva datasety vibrodiagnostiky různých poškození ložisek. Na nich budou trénovány vybrané klasifikační metody umělé inteligence. První část práce je zaměřena na metody technické diagnostiky. Mezi ty patří především vibrační diagnostika (analýza vibrací), tribodiagnostika (analýza maziva), termodiagnostika (analýza teplotního signálu) a akustická diagnostika (analýza zvukového signálu). V praxi bývá často využívanou diagnostikou vibrační diagnostika, která analyzuje strojní zařízení na základě jím vydávaných vibrací. Oblíbená je především pro širokou oblast svého použití, dostupnost vybavení nebo snadnou implementaci, která nenarušuje chod stroje. Dále jsou v práci rozvedeny možné poruchy ložisek a jak se projevují v diagnostickém signálu. Jsou zde také uvedeny rovnice pro frekvence ložiskových závad (frekvence závady vnitřního kroužku, vnějšího kroužku, klece ložiska a valivých těles). Následuje představení datasetů, které byly v práci použity a jejich frekvence ložiskových závad.

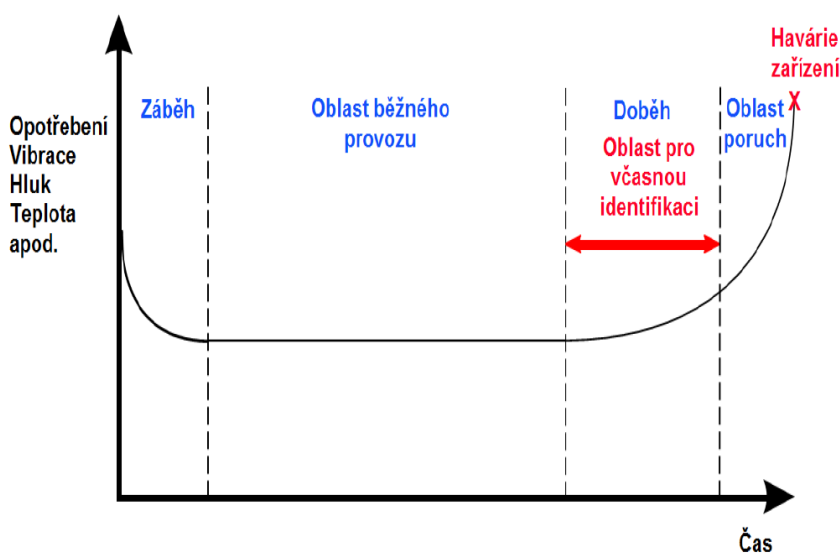
Za účelem klasifikace dat bylo nejprve nutné použítá data předzpracovat (vypočítat příznaky, které charakterizují data nebo stavy ložiska). Nejprve byly vybrány prediktory z časové oblasti - standardní směrodatná odchylka výběru, efektivní hodnota RMS , činitel výkmitu $crestfaktor$, parametr $K(t)$ a $Kurtosis$ faktor. Poté byla provedena Fourierova transformace, která převádí signál z časové do frekvenční oblasti. A jako další prediktor byla vybrána frekvence s největší amplitudou.

Další částí je představení metod umělé inteligence od jednoduchých grafů a stromů řešení, rozhodovacích stromů přes zmínku o expertních systémech, zpracování přirozeného jazyka, proces rozpoznávání až po genetické algoritmy, strojové učení a neuronové sítě. Následuje popis konkrétních použitých metod klasifikace, kterými jsou SVC (Support Vector Classification - klasifikace metodou podpurných vektorů), KNN (K-Nearest Neighbors Classifier - klasifikace k-nejbližšími sousedy), RFC (Random Forest Classifier - klasifikace náhodným lesem) a konvoluční neuronová síť (CNN - Convolutional Neural Network). V závěru kapitoly je popsána matice záměn, která přehledně zobrazuje úspěšnost klasifikace.

V praktické části jsou pro Dataset A i Dataset B představeny vypočítané prediktory a hledány parametry klasifikačních metod dosahující co nejlepších výsledků. Na závěr je pro tyto parametry udána přesnost klasifikace těmito metodami. Součástí práce jsou také skripty algoritmů, které jsou blíže popsány v příloze A.

2. Technická diagnostika

Pojem diagnóza se dříve spojoval převážně s medicínským oborem, dnes se používá také v technické oblasti. [1] Technická diagnostika jakožto samostatný obor slouží ke stanovování technického stavu objektu, kdy za pomoci měřicí techniky objektivně vyhodnocuje příznaky. K tomu využívá nedestruktivní metody bez nutnosti demontáže. [2] Včasná identifikace vznikajících poruch na zařízení (viz obrázek 2.1) zajišťuje kromě větší bezpečnosti a spolehlivosti provozu také plánování oprav s předstihem, čímž docílí šetření finančních prostředků. [3] Existuje velké množství metod technické diagnostiky, některé z těchto metod budou popsány níže. Nicméně žádná konkrétní metoda technické diagnostiky nezajistí za jakýchkoliv provozních podmínek a na jakémkoliv zařízení vždy správnou diagnostiku. Proto je vhodné využívat kombinaci více metod a zvyšovat tím pravděpodobnost zachycení nežádoucího stavu včas. [1]



Obrázek 2.1: Průběh opotřebení - vanová křivka [1].

2.1. Metody technické diagnostiky

K diagnostikování poruch ložisek je možné použít metody popsané níže. Nicméně v technické diagnostice patří mezi nejčteněji používané metody vibrodiagnostika. [2]

2.1.1. Vibrodiagnostika

Jedná se o diagnostiku pracující s vibracemi zařízení, které jsou snímány měřicí technikou v podobě rychlosti, zrychlení nebo výchylky. [1] Signál z vibrací se následně zpracuje a jeho analýzou může být odhalena počínající závada. [4]

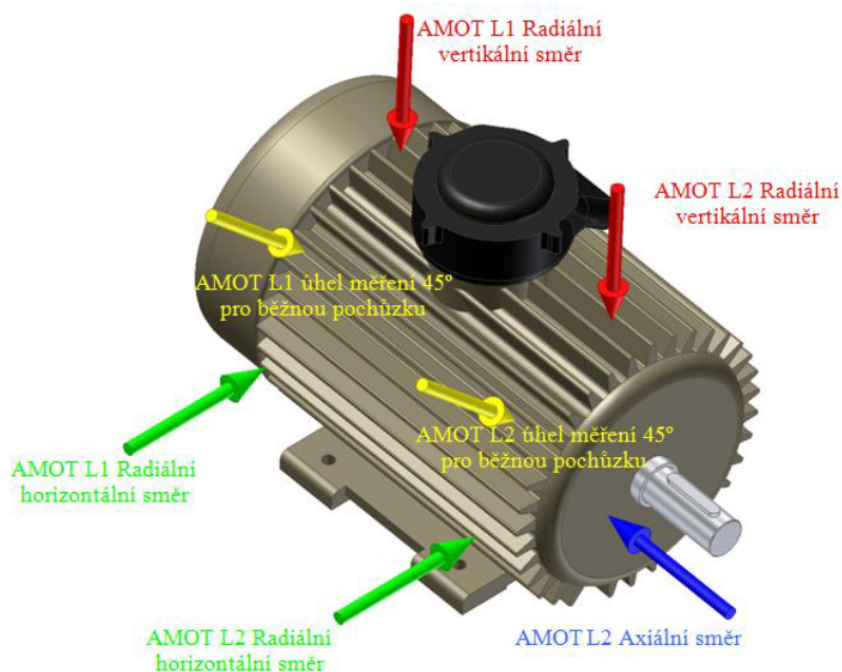
Mezi výhody vibrodiagnostiky patří:

- rozsáhlá oblast využití na různých strojních zařízeních (zejména rotačních strojích) [4]
- možnost měření vibrací za plného provozu stroje

- metoda má široký záběr identifikace poruch
- umožňuje včasnou detekci poruchy
- kromě indikace poruch poskytuje také doplňkové informace (hlučnost, životnost stroje, kvalita a přesnost výroby, dynamické zatížení konstrukce)
- snadná dostupnost přístrojové (hardware i software) výbavy
- rozměry snímačů a způsob jejich uchycení dovolují jednoduché měření signálu
- metoda je schopna konstantního sledování [2]

Vibrace se zaznamenávají snímači jež převádí mechanické kmitání na elektrický signál, který je dále zesilován a zpracován do digitální podoby. Nejrozšířenějšími snímači jsou akcelerometry (snímače zrychlení), které fungují na principu piezoelektrického jevu. Při působení síly na piezoelektrický krystal vzniká na jeho stěnách elektrický náboj a ten je úměrný zrychlení. [1]

Mezi důležité parametry snímače patří dynamický rozsah, frekvenční odezva, horní frekvenční limit, dolní prahová frekvence, rušení, rezonanční frekvence, citlivost, teplotní vliv a teplotní rozsah. Podstatné je také umístění snímačů, kdy je potřeba snímač umístit co nejbližší měřenému místu přímo na konstrukční část. V každém měřicím bodě se obvykle měří v radiálním, axiálním a vertikálním směru (viz obrázek 2.2). Významnou roli hrají i podmínky měření (otáčky stroje i zatížení) a způsob uchycení snímačů (šroubový spoj, přilepení lepidlem, magnetický spoj či ruční sonda). [1]



Obrázek 2.2: Umístění snímačů a směry měření [3].

Detailnější popis vibrodiagnostiky lze nalézt v normě ČSN ISO 13373: Monitorování stavu a diagnostika strojů - Monitorování stavu vibrací.

2.1. METODY TECHNICKÉ DIAGNOSTIKY

2.1.2. Tribodiagnostika

Při této metodě se informace získávají a analyzují ze vzorků maziva strojního zařízení. Na jedné straně dochází k pozorování stavu a degradace samotného maziva [1] (např. vlivem oxidace, vniknutím jiných látek či vody [4]). Dále je také možné na základě částic obsažených v mazivu zjišťovat technický stav (poškození) zařízení a toto místo lokalizovat. [4] Mezi nejčastější zkoumané parametry patří kromě obsažených nečistot také kinematická viskozita, pH hodnota a bod vzplanutí. [2]

2.1.3. Termodiagnostika

Mezi další možnosti určení stavu diagnostikovaného zařízení patří analýza teplotního signálu. [2] Měření je možné provádět buď kontaktními teploměry, nebo lze využít bezdotykové teploměry (zejména termovizní kamery), které především vedly k rozšíření této diagnostiky také do strojního průmyslu. [1] Avšak nevýhodou této metody je možnost zkreslení od okolního prostředí. [2]

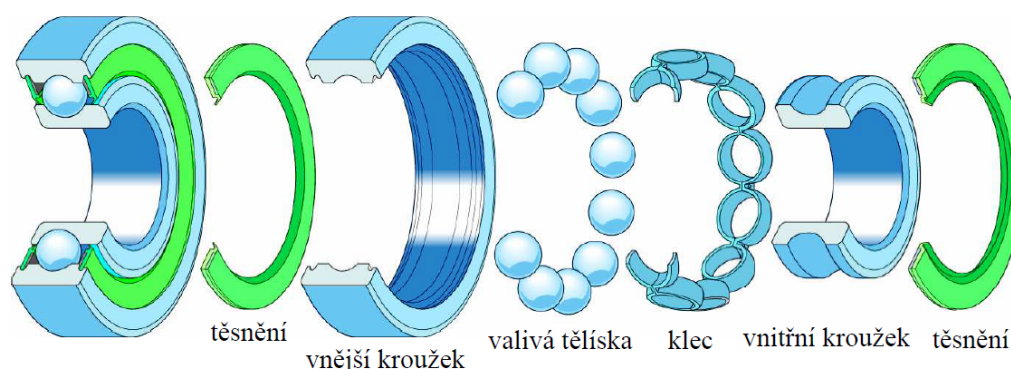
2.1.4. Akustická diagnostika

Akustická diagnostika je založena na podobném principu jako vibrodiagnostika. Vibrace (z provozu strojního zařízení či indikující poruchu) také způsobují chvění okolního vzduchu. [1] Tento akustický signál je tedy možné v podobě intenzity hluku zaznamenávat a dále analyzovat. [2] Při poruše (např. poškozená valivá ložiska [2]) je vydáván vysokofrekvenční signál, který predikuje poruchu a to i skrytou pod povrchem ve formě mikrotrhlin. [3] Mezi nevýhody akustické diagnostiky se řadí interference vln, která zkresluje výsledné hodnoty. Další problém nastává v případě více zdrojů hluku (ať už se jedná o zvuky z okolního prostředí nebo jiné části zařízení), kdy je komplikované tyto signály rozlišit. [2]

3. Poruchy ložisek a jejich projev v diagnostickém signálu

Ložiska zajišťují vzájemný pohyb součástí zařízení [2], snížení tření a přenos sil mezi rotujícími částmi a konstrukcí [4]. Dle principu (styku hřídele s ložiskem) se dělí na kluzná a valivá. [5] Jelikož analyzovaná data pocházejí z valivých ložisek, tak budou dále uvažována právě tato ložiska.

Valivé ložisko se zpravidla skládá z vnějšího kroužku, valivých tělísek (různých tvarů), klece, vnitřního kroužku a těsnění (viz obrázek 3.1). Otáčivý pohyb způsobuje tření, které vede k opotřebení. Mimo toto běžné opotřebení dochází také k poruchám ložisek a to z různých příčin. Velký vliv má mazivo, ať už se jedná o nedostatečné mazání, špatně zvolené mazivo, nečistoty v něm obsažené nebo další. Významný podíl nese rovněž nesprávná montáž (nesouosost, nevyváženost), přetížení a v neposlední řadě únava materiálu. [2]



Obrázek 3.1: Části valivého ložiska [4].

Životnost ložisek je omezena i za ideálních provozních podmínek v podobě vydrolování (tzv. pittingu - viz obrázek 3.2) vlivem kontaktní únavy materiálu. [2] Kromě obvyklého pittingu se mohou objevovat další druhy poškození - odlupování, škrábance, trhliny, vrypy, abnormální opotřebení, zadření, elektrické tavení nebo koroze. [6] Podrobnější informace lze také nalézt v normě ISO 15243:2017. [7]

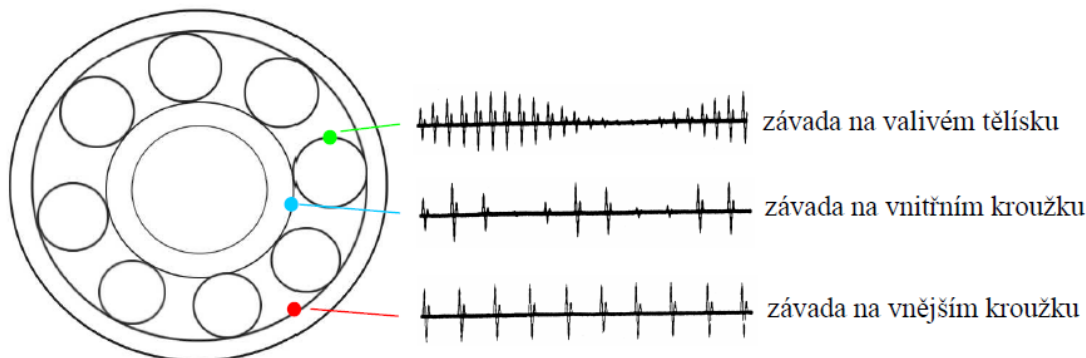


Obrázek 3.2: Pitting na valivém ložisku [2].

3.1. FREKVENCE LOŽISKOVÝCH ZÁVAD (FLZ)

3.1. Frekvence ložiskových závad (FLZ)

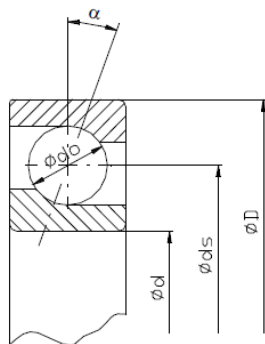
Vytvářející se poškození ložiska se při průchodu valivých prvků přes poškozenou část projevuje na jeho vibračním signále (viz obrázek 3.3). Frekvence generovaných rázových impulzů vibrací závisí na místě vzniku poškození, rychlosti otáčení a geometrii ložiska. [5]



Obrázek 3.3: Impulzy generované vadami ložiska [4].

Tyto ložiskové frekvence lze dopočítat pomocí níže uvedených vzorců a znalosti parametrů ložiska:

- n počet valivých tělísek
- f_r rotorová frekvence
- d_0 průměr valivých tělísek
- d_s střední průměr
- α kontaktní úhel



Obrázek 3.4: Parametry ložiska [5].

3.1.1. Frekvence závady vnitřního kroužku

$$f_{BPFI} = \frac{n}{2} \cdot f_r \cdot \left(1 + \frac{d_0}{d_s} \cos \alpha \right) \quad (3.1)$$

3.1.2. Frekvence závady vnějšího kroužku

$$f_{BPFO} = \frac{n}{2} \cdot f_r \cdot \left(1 - \frac{d_0}{d_s} \cos \alpha \right) \quad (3.2)$$

3.1.3. Frekvence závady klece ložiska

$$f_{FTF} = \frac{1}{2} \cdot f_r \cdot \left(1 - \frac{d_0}{d_s} \cos \alpha \right) \quad (3.3)$$

3.1.4. Frekvence závady valivých těles

$$f_{BSF} = \frac{d_s}{2d_0} \cdot f_r \cdot \left(1 - \left(\frac{d_0}{d_s} \cos \alpha \right)^2 \right) \quad (3.4)$$

Rovnice platí pro statický vnější kroužek. V situaci rotujícího vnějšího kroužku by se v rovnicích (kromě rovnice pro frekvenci závad valivých těles) zaměnila znaménka plus a minus. [2] Při zhoršování stavu ložiska se kromě těchto frekvencí začínají objevovat také postranní pásma posunutá o otáčkovou frekvenci nebo se jedná o násobek frekvence ložiskové závady (viz obrázek 3.5). [4]



Obrázek 3.5: Ukázkové frekvenční spektrum vadného ložiska v závěru jeho životnosti [2].

4. Data a jejich předzpracování

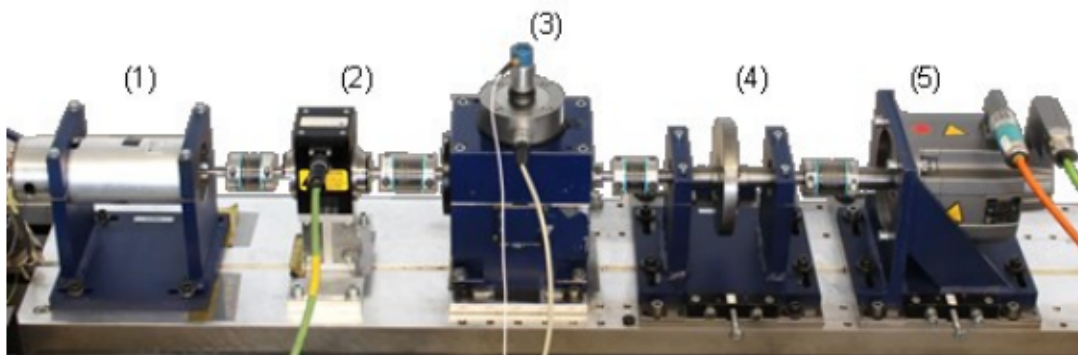
4.1. Dataset A - Dataset Paderborn univerzity

Data pocházejí z Paderborn univerzity v Německu, jejich použití je pro nekomerční akademické účely povoleno [8]. Jedná se o experimentální datasety monitorování stavu valivých ložisek založené na měření vibrací. Vibrace jsou zaznamenávány v podobě zrychlení na horním konci modulu valivých ložisek za použití piezoelektrického akcelerometru (model č. 336C04, PCB Piezotronics, Inc.) a zesilovače (typ 5015A, Kistler Group) s dolnoproputným filtrem 30 kHz. Signál je digitalizován a ukládán se vzorkovací frekvencí 64 kHz. Testovací souprava (viz obrázek 4.1) se skládá z elektrického motoru (1), hřídele pro měření točivého momentu (2), testovacího modulu valivých ložisek (3), setrvačníku (4) a zátěžového motoru (5).

Měření probíhalo na valivých ložiscích typu 6203, která byla vkládána do testovacího modulu. Dataset se skládá z 20 měření po 4 sekundách pro každé nastavení ložiska (rychlost otáčení 900-1500 rpm, radiální síla působící na zkušební ložisko 400-1000 N, zatěžovací moment 0,1-0,7 Nm). Teplota byla po celou dobu experimentu udržována mezi 45-50 °C. Pro účely klasifikace dat jsou ložiska rozdělena na:

- ložiska bez poškození (OK)
- poškození vnějšího kroužku (OR)
- poškození vnitřního kroužku (IR)
- kombinované poškození (KB)

Z toho se jednalo o ložiska bez poškození (6 ložisek), uměle poškozená ložiska (12 ložisek) a ložiska poškozená přirozeně při zrychlených testech životnosti (14 ložisek).



Obrázek 4.1: Testovací zařízení [8].

Data ke stažení a veškeré podrobnější informace ohledně dat je možné nalézt v publikaci a na stránkách univerzity [8]. V práci nebyla nedohledána jednotka měřených vibrací. Jelikož akcelerometry podobného typu obvykle měří v jednotkách g , je zde uvedena tato jednotka.

4.2. Dataset B - Universidade Federal do Rio de Janeiro

Původní dataset se skládá z 1951 vícerozměrných časových řad získaných senzory na simulátoru poruch SpectraQuest's Machinery Fault Simulator (MFS) Alignment-Balance-Vibration (ABVT). [9] Nicméně pro práci byla použita pouze měření 4 simulovaných stavů: normální funkce, poškození vnějšího kroužku ložiska, klece ložiska a kuličky ložiska. V tabulce 4.1 je uvedena specifikace měření a aparatury.

Tabulka 4.1: Specifikace

motor	1/4	CV DC
frekvenční rozsah	700-3600	rpm
hmotnost systému	22	kg
průměr osy	16	mm
délka osy	520	mm
rotor	15,24	cm
rozteč ložisek	390	mm
počet valivých tělísek	8	ks
průměr kuliček	0,7145	cm
průměr klece	2,8519	cm

Pro měření byly použity:

- 3 akcelerometry IMI Sensors (Model 601A01) měřící v axiálním, radiálním a tangenciálním směru. Citlivost senzorů je ($\pm 20\%$) 100 mV na g, frekvenční rozsah (± 3 dB) 0,27 – 10000 Hz a rozsah měření ± 50 g
- jeden triaxiální akcelerometr IMI Sensors (model 604B31), který měří v radiálním, axiálním a tangenciálním směru (se stejnými parametry jako výše zmíněné, kromě frekvenčního rozsahu (± 3 dB) 0,5 – 5000 Hz)
- Analogový tachometr Monarch Instrument MT-190
- Mikrofon Shure SM81 s frekvenčním rozsahem 20 – 20000 Hz
- Dva 4kanálové analogové moduly National Instruments NI 9234 se vzorkovací frekvencí 51,2 kHz

Každé měření probíhalo po dobu 5 sekund se vzorkovací frekvencí 50 kHz a to:

- 50 bez poškození ložiska (normal)
- 376 poškození klece ložiska (cage)
- 372 poškození vnějšího kroužku (outer)
- 323 poškození kuličky (ball)

Výrobce dodaná ložiska s poruchami (vnějšího a vnitřního kroužku, klece) byla postupně vkládána do polohy mezi rotor a motor (underhang position) a do vnější pozice, kdy je rotor uprostřed mezi motorem a ložiskem (overhang position). Pro zvýraznění chyb ložisek byla přidána závaží 6 g, 20 g a 35 g pro nevyváženost. [9]

4.3. Frekvence ložiskových závad (FLZ) použitých datasetů

4.3.1. FLZ - Dataset A

V případě dat z Paderborn univerzity, kde bylo použito kuličkové ložisko typu 6203 s parametry:

počet valivých tělísek	$n = 8$
průměr valivých tělísek	$d_0 = 6,75$ mm
střední průměr	$d_s = 28,5$ mm
kontaktní úhel	$\alpha = 0^\circ$

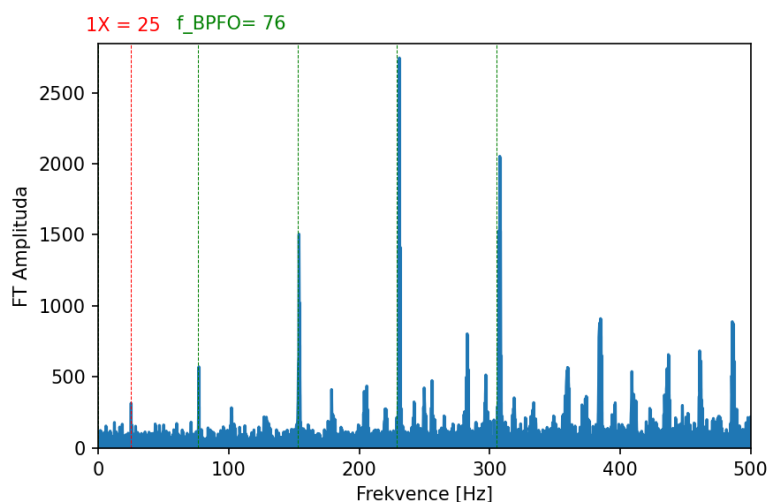
pak ložiskové frekvence podle rovnic 3.1 a 3.2 vycházejí

$$f_{BPBI} \approx 4,947 \cdot f_r, \quad (4.1)$$

$$f_{BPFO} \approx 3,053 \cdot f_r. \quad (4.2)$$

Poškození klece ložiska ani závada valivých těles nebyly v těchto datech klasifikovány.

Na obrázku 4.2 je graf frekvenční analýzy (viz kapitola 5.7) vibrací ložiska s poškozeným vnějším kroužkem (OR). Lze vidět, že první výraznější amplituda se nachází na první otáčkové frekvenci $1X \approx 25$ Hz. Další vysoké hodnoty pak odpovídají frekvenci závady vnějšího kroužku $f_{BPFO} \approx 76$ Hz a jejím násobkům.



Obrázek 4.2: Ukázka frekvenční analýzy vibrací ložiska s poškozeným vnějším kroužkem (OR). Červená čára ukazuje otáčkovou frekvenci $1X$, zelená čára značí frekvenci závady vnějšího kroužku f_{BPFO} a její násobky.

4.3.2. FLZ - Dataset B

V případě datasetu B z Universidade Federal do Rio de Janeiro, kde byla použita kuličková ložiska s parametry:

počet valivých tělísek	$n = 8$
průměr valivých tělísek	$d_0 = 7,145 \text{ mm}$
střední průměr	$d_s = 28,519 \text{ mm}$
kontaktní úhel	$\alpha = 0^\circ$

pak ložiskové frekvence podle rovnic 3.2 a 3.3 a 3.4 vycházejí

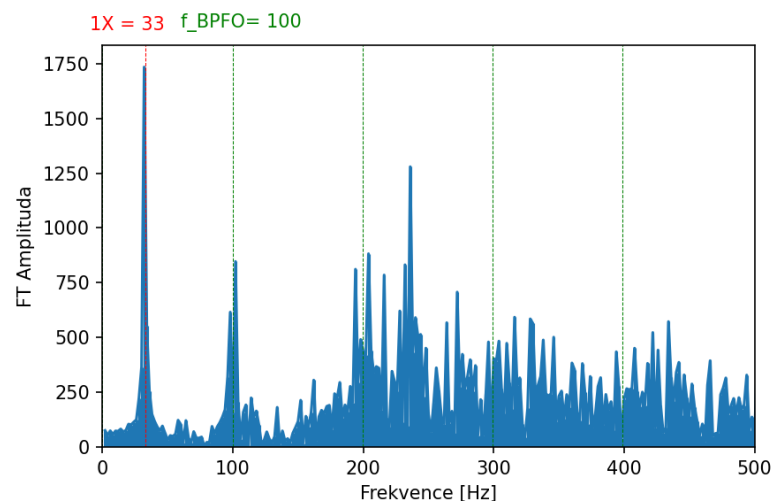
$$f_{BPBO} \approx 2,998 \cdot f_r, \quad (4.3)$$

$$f_{FTF} \approx 0,375 \cdot f_r. \quad (4.4)$$

$$f_{BSF} \approx 1,871 \cdot f_r. \quad (4.5)$$

Poškození vnitřního kroužku ložiska nebylo v těchto datech klasifikováno.

Na obrázku 4.3 je graf frekvenční analýzy (viz kapitola 5.7) vibrační ložiska s poškozeným vnějším kroužkem (outer). Lze vidět, že první výrazná amplituda se nachází na první otáčkové frekvenci $1X \approx 33 \text{ Hz}$. Další vysoká hodnota pak odpovídá frekvenci závady vnějšního kroužku $f_{BPFO} \approx 100 \text{ Hz}$.



Obrázek 4.3: Ukázka frekvenční analýzy vibrační ložiska s poškozeným vnějším kroužkem (outer). Červená čára ukazuje otáčkovou frekvenci $1X$, zelená čára značí frekvenci závady vnějšního kroužku f_{BPFO} a její násobky.

5. Použité prediktory

Pro klasifikační metody byly použity níže popsané prediktory (příznaky), které by měly charakterizovat testovaná data. Nejprve bylo zacíleno na časovou oblast, ze které pocházejí prediktory standardní směrodatná odchylka výběru s (viz kapitola 5.1), efektivní hodnota RMS (viz kapitola 5.2), činitel výkmitu *crestfaktor* (viz kapitola 5.3), parametr $K(t)$ (viz kapitola 5.4) a Kurtosis metoda (viz kapitola 5.5). Níže v textu je uvedena Fourierova transformace (viz kapitola 5.7) a z ní vycházející prediktor frekvence s největší amplitudou (viz kapitola 5.6). Tyto prediktory jsou poté využity pro klasifikaci metodami SVC (Support Vector Classifier) (viz kapitola 7.1), KNN (K - Nearest Neighbors Classifier) (viz kapitola 7.2) a RFC (Random Forest Classifier) (viz kapitola 7.3).

5.1. Standardní směrodatná odchylka výběru (s)

$$s = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^n |x - \bar{x}_i|^2}, \quad (5.1)$$

kde n je počet naměřených hodnot, x_i naměřené hodnoty a \bar{x} jejich aritmetický průměr.

5.2. Efektivní hodnota (RMS)

RMS (Root Mean Square) udává efektivní hodnotu měřené veličiny, která je často užívanou charakteristikou. Vyjadřuje průměrný výkon a přibližně se rovná 0,707 násobku amplitudy. [4] Úzce souvisí s energetickým obsahem veličiny a sleduje také časový průběh. [2] Výpočet je tvaru

$$x_{RMS}(= x_{ef}) = \sqrt{\frac{1}{T} \cdot \int_0^T x(t)^2 dt}, \quad (5.2)$$

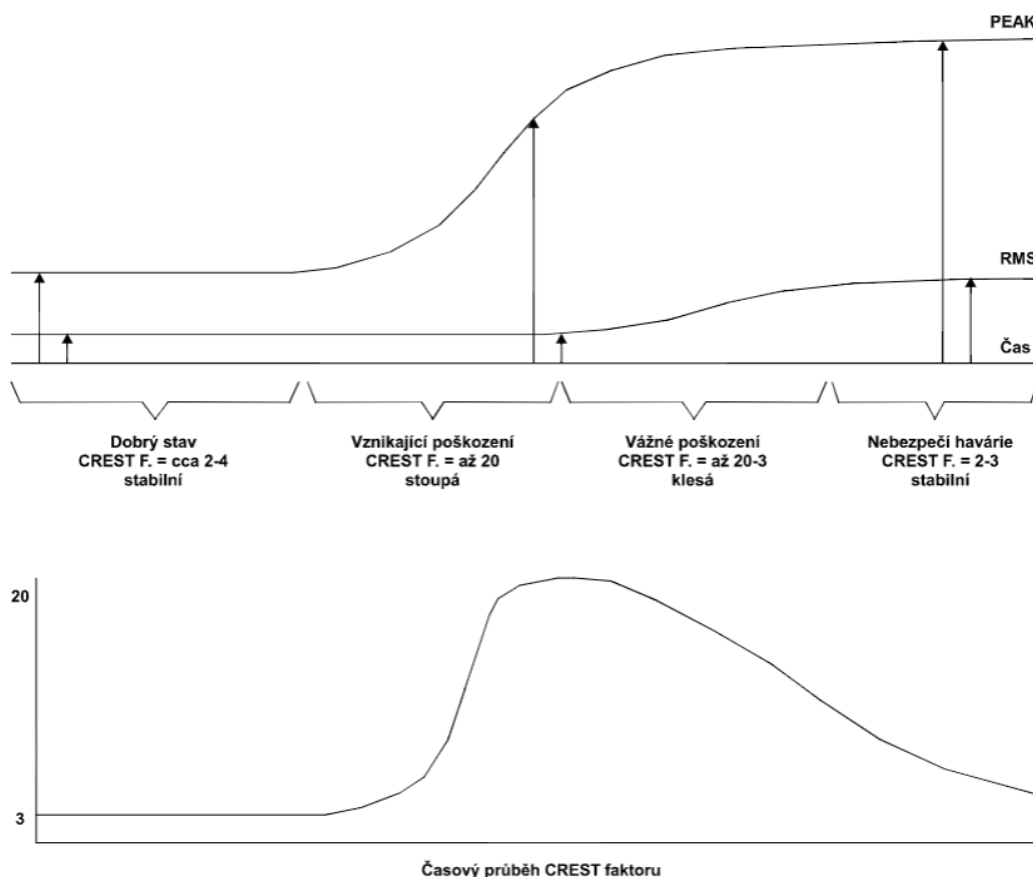
kde T je doba měření vyhodnocované veličiny. [4]

5.3. Činitel výkmitu (*crest faktor* - CF)

Pro kontrolu stavu ložisek se využívá také crest faktor CF , který je schopen rychle upozornit na počínající poškození, kdy se hodnota crest faktoru rychle zvyšuje. S rostoucím poškozením však tato hodnota opět klesá (viz obrázek 5.1) [1]. Jedná se o poměr

$$CF = \frac{x_a}{x_{ef}}, \quad (5.3)$$

kde x_a je špičková hodnota a x_{ef} efektivní hodnota amplitudy. [4]



Obrázek 5.1: Časový průběh crest faktoru, špičková hodnota (PEAK) a efektivní hodnota (RMS) v průběhu poškození ložiska [1].

5.4. Parametr $K(t)$

Na podobném principu jako crest faktor je postaven parametr $K(t)$, který zohledňuje počáteční stav ložiska. S klesající hodnotou $K(t)$ roste míra poškození ložiska. Výpočet je tvaru

$$K(t) = \frac{a_{ef(0)} \cdot a_v(0)}{a_{ef(t)} \cdot a_v(t)}, \quad (5.4)$$

kde

$a_{ef(0)}$ je efektivní zrychlení v čase 0 (neznáme-li hodnotu, doporučuje se $10 \text{ m} \cdot \text{s}^{-2}$ [10]),

$a_v(0)$ je výkmit zrychlení v čase 0 (neznáme-li hodnotu, doporučuje se $40 \text{ m} \cdot \text{s}^{-2}$ [10]),

$a_{ef(t)}$ je efektivní hodnota zrychlení v čase t ,

$a_v(t)$ je hodnota výkmitu zrychlení v čase t . [5]

5.5. Kurtosis metoda

Metoda Kurtosis předpokládá normální rozdělení signálu nepoškozeného ložiska. V případě vzniku poškození se v signále začnou objevovat špičky signálu. Ty způsobují, že signál už nevykazuje normální (Gaussovo) rozdělení a s rostoucím poškozením začíná růst

5.6. FREKVENCE S NEJVĚTŠÍ AMPLITUDOU

také hodnota Kurtosis faktoru. Měření je obvykle rozděleno do 5 pásem v rozsahu mezi 2,5-80 kHz. Rovnice pro výpočet Kurtosis faktoru je

$$K = \frac{\int_{-\infty}^{\infty} (x - \bar{x})^4 \cdot P(x) dx}{\sigma^4}, \quad (5.5)$$

kde

x je amplituda signálu

\bar{x} střední hodnota

$P(x)$ pravděpodobnost x

σ standardní směrodatná odchylka vůči nulovému signálu. [1]

5.6. Frekvence s největší amplitudou

Po provedení Fourierovy transformace (viz kapitola 5.7) se vybere frekvence s největší amplitudou v rozsahu do 2000 Hz (kde se očekává projev ložiskových závad [4]).

5.7. Fourierova transformace (FT)

Analyzovat signály je výhodné nejen v časové oblasti. V technické diagnostice se hojně uplatňuje také analýza frekvencí diagnostikovaného systému. Na obrázku 5.2 můžeme vidět, jak se různé frekvence vibrací jednotlivých částí systému projevují v celkovém signálu a ve frekvenčním spektru. Pro převod z časové oblasti do oblasti frekvenční lze využít Fourierovu transformaci. [4] Periodickou funkci (signál) $x(t)$ v čase T lze zapsat jako nekonečnou Fourierovu řadu tvaru

$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cdot \cos(n\omega t) + b_n \cdot \sin(n\omega t)); \quad \omega = \frac{2\pi}{T}, \quad (5.6)$$

kde ω je základní frekvence a Fourierovy (spektrální) koeficienty lze vyjádřit jako

$$a_n = \frac{2}{T} \cdot \int_0^T x(t) \cdot \cos(n\omega t) dt, \quad b_n = \frac{2}{T} \cdot \int_0^T x(t) \cdot \sin(n\omega t) dt. \quad (5.7)$$

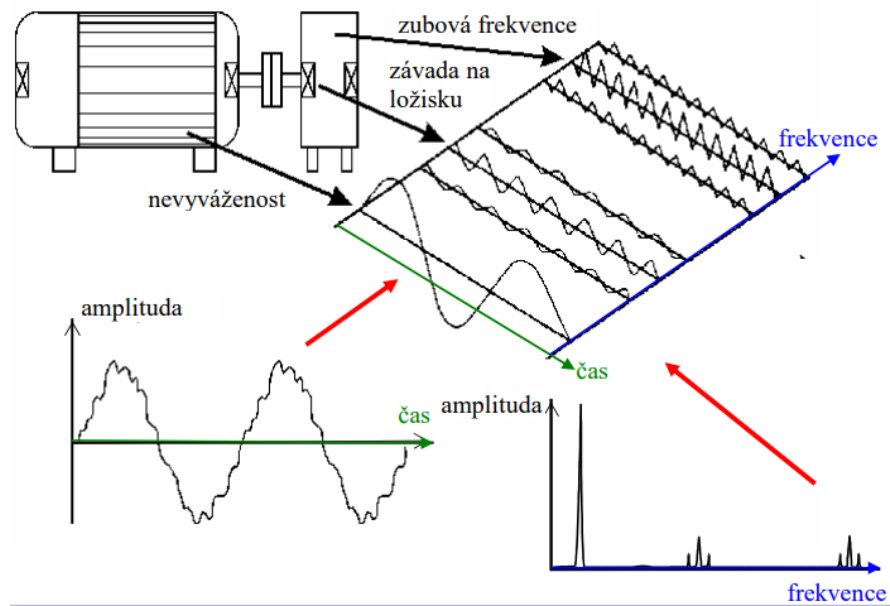
Diskrétní Fourierova transformace (DFT)

Jelikož naměřená digitální data jsou diskrétního charakteru, tak se funkce $x(t_k)$ zapisuje jako konečná Fourierova řada ve tvaru

$$x(t_k) = \frac{a_0}{2} + \sum_{n=1}^{\frac{N}{2}} \left(c_n \cdot \cos\left(\frac{2\pi n t_k}{T} + \phi_n\right) \right), \quad (5.8)$$

kde N značí počet naměřených diskrétních hodnot, t_k ($k = 1, \dots, N$) časový okamžik a spektrální koeficienty jsou ve tvaru amplitudy c_n a fáze ϕ_n

$$c_n = \sqrt{a_n^2 + b_n^2} \quad \phi_n = \arctan\left(-\frac{b_n}{a_n}\right). \quad (5.9)$$



Obrázek 5.2: Podstata frekvenční analýzy [4].

Použitý algoritmus RFFT - Real Fast Fourier transform

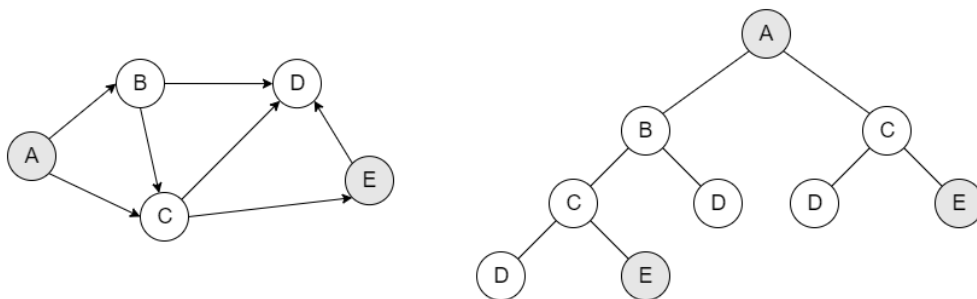
Při praktických výpočtech se využívá algoritmu rychlé Fourierovy transformace (FFT - Fast Fourier transform). Nicméně výše zmíněné Fourierovy transformace převádějí signál do komplexních čísel. Jelikož byla použita reálná data, není třeba využívat komplexní rovinu, postačí reálná část. Pro tuto aplikaci byly použity funkce `fft.rfft` z knihovny NumPy.

6. Metody umělé inteligence

Definicí umělé inteligence může být více, jednou z nich je definice Prof. Marvina Minského z MIT, který říká: "Umělá inteligence je věda o vytváření strojů využívající postupy, které prováděny člověkem, by vyžadovaly jeho inteligenci." [11]

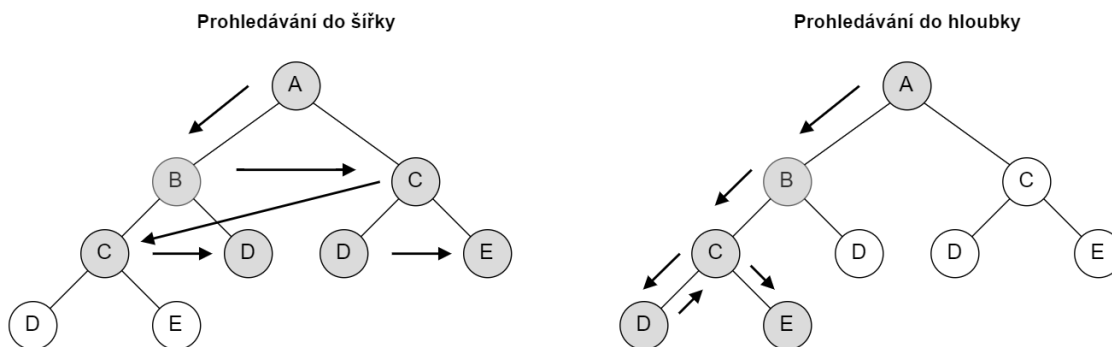
Grafy a stromy řešení

Graf je matematická struktura, která je dána množinou uzlů a množinou hran, kde hrana vede mezi dvěma uzly (může mít i svou váhu a orientaci). Takovým grafem můžeme znázornit definovaný problém, např. viz obrázek 6.1, kde máme počáteční stav a koncový stav (uzly), mezi kterými vedou orientované hrany. K řešení problému hledáme podle orientovaných hran cestu z počátečního stavu ke koncovému stavu. **Strom řešení** vychází z počátečního uzlu (*kořene*) a podle pravidel k tomuto uzlu postupuje expanzí do další úrovně vytvořením nových uzlů. *Listem stromu* (koncovým uzlem) pak nazýváme ten uzel, který nelze dále expandovat nebo splňuje zadaný cíl (určení třídy). [11]



Obrázek 6.1: Vlevo orientovaný graf, vpravo strom řešení, kde A je počáteční uzel (kořen), E je koncový uzel (list) [11].

K nalezení cesty mezi počátečním a koncovým uzlem lze aplikovat *slepé prohledávání* a to buď *do hloubky* nebo *do šířky* (viz obrázek 6.2). V případě prohledávání stromu řešení do hloubky se postupuje od kořene uzly nejvíce vlevo až po nejnižší uzel. Pokud poslední (listový) uzel není požadovaný (chybný listový uzel), pak se řídicí mechanismus posunuje o úroveň výše a prohledává další nejbližší větev vlevo (tzv. backtracking).



Obrázek 6.2: Slepé prohledávání do hloubky a do šířky [11].

V případě slepého prohledávání do šířky se nejprve od kořene prohledávají všechny uzly na dané úrovni, až poté se mechanismus posunuje na další (nižší) úroveň.

Metodou slepého prohledávání lze (za předpokladu dostatečného času a paměti počítače) nalézt požadované řešení. Nicméně proces je možné zefektivnit *heuristickým prohledáváním*, kdy použijeme nějakou lokální znalost či pravidlo pro daný uzel. Heuristickým prohledáváním je například *metoda největšího gradientu* nebo *metoda minimálních nákladů*. [11]

Rozhodovací stromy

Jedná se o graf s množinou uzlů a orientovaných hran (shora dolů), kde uzly představují testovací atribut nebo klasifikační třídu a hrany reprezentují hodnotu testovacího atributu. Z každého uzlu vychází takový počet hran, kolika hodnot nabývá testovací atribut. Počáteční uzel nazýváme (*kořen*), od něj vedou orientované hrany přes uzly až ke koncovému uzlu (*list stromu*), který nelze dále větvit (určuje tedy třídu). [12]

Klasifikace rozhodovacím stromem probíhá zadáním vektoru prediktorů, které mohou být (i současně) jak kvalitativní, tak kvantitativní. Při klasifikaci se postupuje od kořene a v každém neterminálním uzlu rozhodovacího stromu se postupně porovnávají zadané prediktory až dospějeme k listovému uzlu, který určuje klasifikační třídu. [13]

Expertní systémy

V komerční sféře často používanou metodou jsou expertní systémy, které nahrazují specialistu (experta) v dané oblasti a přinášejí řešení složitých úloh. To vyžaduje, aby systém obsahoval bázi znalostí, které jsou schopné popsat řešený problém a řídicí systém pak odůvodnit nalezené řešení. Jedná se tedy o podrobné znalosti úzké problematiky. [11]

Zpracování přirozeného jazyka

Další oblastí umělé inteligence je zpracování přirozeného jazyka, který zpracovává informace při komunikaci mezi lidmi s cílem vyvodit význam této komunikace. Jedním z užitečných přístupů při této úloze je studium a porozumění gramatice jazyka (což jsou definovaná pravidla určující správné formulování frází a klauzulí). [11]

Proces rozpoznávání

Metody zařazující předkládaná data do určitých tříd nazýváme procesem rozpoznávání. Nejdříve se rozpoznávanému objektu stanovují hodnoty charakteristik a vytvoří se jeho obraz (image). Ve druhé fázi už dochází ke klasifikaci obrazu a vlastnímu zařazení do příslušné třídy. Klasifikace může probíhat pomocí *příznakových vektorů* charakterizujících objekt nebo *strukturálním rozpoznáváním* s kvalitativními charakteristikami (řetězci symbolů primitivů, ze kterých se obraz skládá). Podmnožinou rozpoznávání je **počítačové vidění**, které se zaměřuje na vizuální informace (od rozpoznávání základních vzorů po složité analýzy obrazu). [11]

Strojové učení

Strojové učení na rozdíl od výše jmenovaných je schopno adaptovat se učním (neboli zefektivnit řešení stejné úlohy v dalších krocích). Dle stupně odvozování učícího se systému rozlišujeme různé metody učení:

- učení memorováním - jedná se o pouhé zaznamenávání bez dalšího odvozování (v podstatě databázový systém)
- učení přijímáním rad - učící se systém využívá nejjednodušší principy odvozování ze znalostí již získaných od učitele
- učení dedukcí - vychází z učení přijímáním rad s tím, že učící se systém je už schopen určitého vyvozování při zachování původního informačního obsahu
- učení analogií - systém vyhledává společné rysy s analogickou situací (o které má již znalosti) a podle této situace se přizpůsobí řešení nové situaci
- učení induktivní - učící se systém si znalosti vytváří teprve na základě zobecňování konkrétního případu [11]

Genetické algoritmy

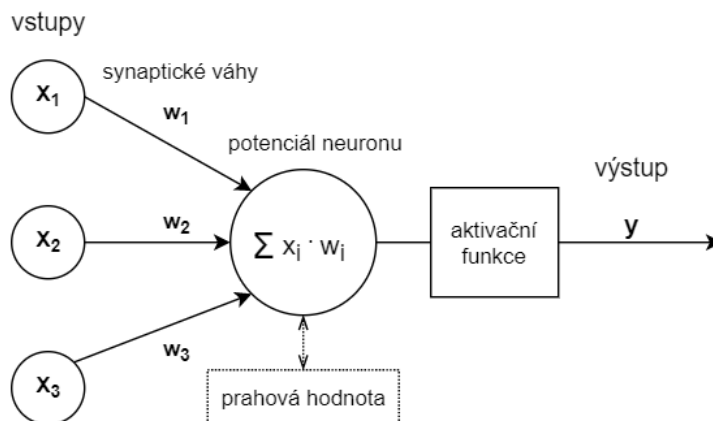
Genetické algoritmy jak už název napovídá vycházejí z principů genetiky. Jedná se o kopírování řetězců (chromozomů) se současnou výměnou jejich částí (křížením). Populaci se pak nazývá množina chromozomů, které mají určitou délku (počet genů o hodnotách obvykle 0 a 1) a hodnotu cílové funkce (tzv. fitness neboli vhodnost chromozomu). Algoritmus je tvořen reprodukcí (kopírováním chromozomů do nové populace, kdy chromozomy s vyšší hodnotou fitness funkce se kopírují s větší pravděpodobností), následuje křížení (kdy se náhodně vytvoří páry chromozomů a vymění se náhodný počet jejich genů). Z důvodů omezení ztráty výhodných genů na nevyhovujících chromozomech je poslední fází mutace algoritmu, kdy se (obvykle s malou pravděpodobností) náhodně mění hodnota genu. [11]

Neuronové sítě

Jedná se o algoritmy se snahou simulovat funkce lidského myšlení, kdy jsou znalosti ukládány ve formě síly vazeb mezi jednotlivými neurony. S rostoucím počtem případů se posilují vazby vedoucí ke správné klasifikaci a naopak se oslabují nesprávné vazby. Pro neuronové sítě je charakteristický proces učení, kdy sama neuronová síť si na základě trénovacích dat vytvoří pravidla, podle kterých transformuje vstupní data na data výstupní. Neuron sítě je stejně jako biologický vzor tvořen:

- dendrity - představují vstup dat
- tělo buňky - sčítá přijaté signály, které mohou vést k excitaci neuronu
- axonové vlákno - přenáší excitovaný signál dále do sítě
- synapse - zapojení mezi neurony, které předává výstupní signál z axonového vlákna na dendrity nového neuronu (s možností vedený signál zesílit nebo zeslabit).

Perceptron je model neuronu, který je v případě překročení jeho prahové hodnoty ϕ excitován a hodnota výstupního signálu y je rovna 1 (nebo je dána aktivační funkcí). K inhibici neuronu ($y = 0$) dochází v případě, že potenciál neuronu nepřekročí prahovou



Obrázek 6.3: Grafické znázornění modelu neuronu.

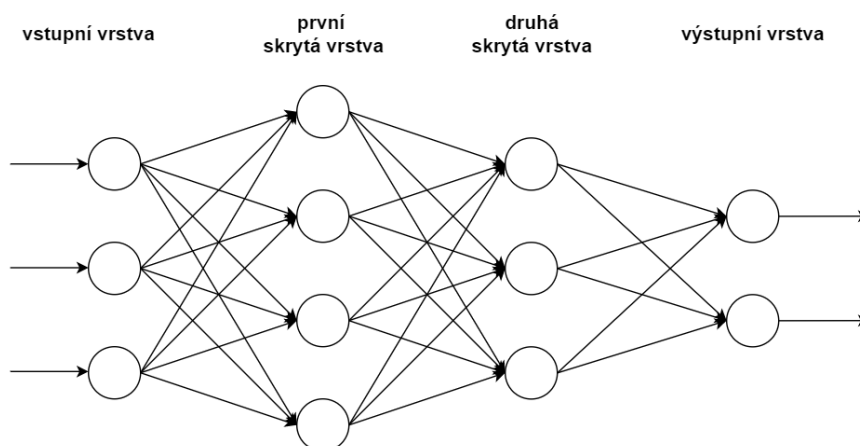
hranici. Potenciál neuronu se vypočítá jako vážený součet vstupních signálů (viz obrázek 6.3). To popisuje rovnice tvaru

$$y = \text{sgn} \left(\sum_{i=1}^n w_i x_i - \phi \right), \quad (6.1)$$

$$\begin{aligned} \text{sgn}(x) &= 0 & x \leq 0, \\ \text{sgn}(x) &= 1 & x > 0. \end{aligned}$$

Stanovení hodnot vah je klíčové pro správnou klasifikaci dat. K tomu se neuronové síti předkládá trénovací množina dat.

Neuronové síť mohou mít různé podoby. Rozšířené jsou **vícevrstvé síťe**, které jsou tvořeny minimálně 3 vrstvami - vstupní vrstvou, vnitřní skrytou vrstvou (nebo více skrytými vrstvami) a výstupní vrstvou. Tyto vrstvy jsou mezi sebou synapticky propojeny (viz obrázek 6.4).



Obrázek 6.4: Grafické znázornění modelu neuronové sítě.

Proces trénování probíhá tak, že:

- na vstupní vrstvu neuronové sítě přivedeme prvek trénovací množiny
- dochází k excitaci vstupní vrstvy a převodu excitovaného signálu na synapse další vrstvy

- zde je signál regulován dle synaptických vah (zesílen či zeslaben) a vypočítán potenciál neuronu
- podle potenciálu je neuron excitován (inhibován) na hodnotu jeho aktivační funkce
- signál postupuje k dalším vrstvám neuronové sítě až k výstupní vrstvě
- výstupní signál je porovnán se skutečným stavem trénovacího prvku (jejich rozdíl určuje chybu neuronové sítě)
- podle chyby neuronové sítě dochází k úpravám synaptických vah mezi vrstvami s cílem zmenšení této chyby při příštím průchodu sítí
- takto se pokračuje dál pro další prvky trénovací množiny

Obecně je hledáno minimum funkce chyby neuronové sítě E

$$E = \frac{1}{2} \sum_{i=1}^n (y_i - o_i), \quad (6.2)$$

kde y_i výstupní signál neuronové sítě,
 o_i požadovaný výstupní signál neuronové sítě,
 n počet prvků trénovací množiny

a následně se upravují synaptické váhy. [11]

7. Použité metody klasifikace dat

Ke klasifikaci byly vybrány (metody s učitelem) SVC (Support Vector Classifier) (viz kapitola 7.1), KNN (K - Nearest Neighbors Classifier) (viz kapitola 7.2) a RFC (Random Forest Classifier) (viz kapitola 7.3), které dosahovaly dobré úspěšnosti klasifikace při dřívějším testování (viz například v článku [14] věnovaném strojovému učení ve vibrodiagnostice s cílem určovat závady rotačního stroje).

7.1. Support Vector Classification - klasifikace metodou podpůrných vektorů

V principu se SVM (Support Vector Machines) metody ke klasifikaci do dvou tříd snaží najít nadrovinu prostoru tak, aby zadané trénovací prediktory ležely v opačných polovinách a měly od této nadroviny co největší vzdálenost. Tyto hraniční body vytyčují (podpůrné) vektory, podle kterých se určuje hledaná nadrovina. Čím větší mezera se mezi těmito krajními body (třídami) podaří nalézt, tím se zmenšuje pravděpodobnost chybné klasifikace. V případě, kdy data nejsou dokonale separovatelná pomocí nadroviny, lze využít jádrové transformace, která problém převádí do vyšší dimenze, kde už problém lineárně separovatelný je. Nevýhodou této metody je velká časová náročnost v případě velkého počtu dat. [15]

Parametry

V praktické části byla použita funkce `SVC()` (viz kapitola 8.2.1).

- `kernel` určuje jádrovou funkci (např. lineární, polynomiální, RBF (Radial Basis Function) nebo sigmoid).
- `C` je regularizační parametr, který dovoluje bodům ležet mimo své hranice. Čím je vyšší hodnota parametru `C`, tím umožňuje hranici být pružnější a správně klasifikovat všechna trénovací data.
- Parametr `gamma` určuje, jak velký vliv jednotlivá trénovací data mají.

Pro SVC metodu s jádrovou transformací RBF je důležité nastavit parametry `C` a `gamma`. [16]

7.2. K-Nearest Neighbors Classifier - klasifikace k-nejbližšími sousedy

Klasifikovaná data jsou zařazována do třídy podle toho, do kterých klasifikačních tříd patří (definovaný) počet k (dle vzdálenosti) nejbližších trénovacích dat. Volba vhodného parametru k závisí na předložených datech. Obecně platí, že čím je zvoleno větší k , tím je klasifikace méně náchylná na šum, ale také se klasifikační hranice stávají méně jednoznačné. Pro určení třídy se obvykle používá prostá převaha nejbližších sousedů, nicméně je možné přidat těmto sousedům váhy např. podle vzdálenosti od posuzovaného vzorku. [17]

7.3. Random Forest Classifier - klasifikace náhodným lesem

Skládá se z několika náhodných rozhodovacích stromů. Každý z těchto stromů je tvořen náhodným výběrem s vrácením (bootstrapový výběr) z trénovací sady nebo celou datovou sadou. Ke zlepšení přesnosti predikce a kontroly přetrénování se používá průměrování pravděpodobnostních odhadů jednotlivých klasifikátorů. [18]

Při rozdělování každého uzlu během konstrukce stromu je nejlepší rozdělení nalezeno buď ze všech vstupních příznaků nebo z jejich náhodné podmnožiny. Cílem zanášení této náhodnosti je snížit variabilitu odhadu (prakticky je snížení variability významné). Rozdílností rozhodovacích stromů může dojít k odlišným chybám predikce, kde zprůměrováním těchto predikcí se mohou některé chybné predikce eliminovat. [19]

Parametry

V praktické části byla použita funkce `RandomForestClassifier()` (viz kapitola 8.2.3).

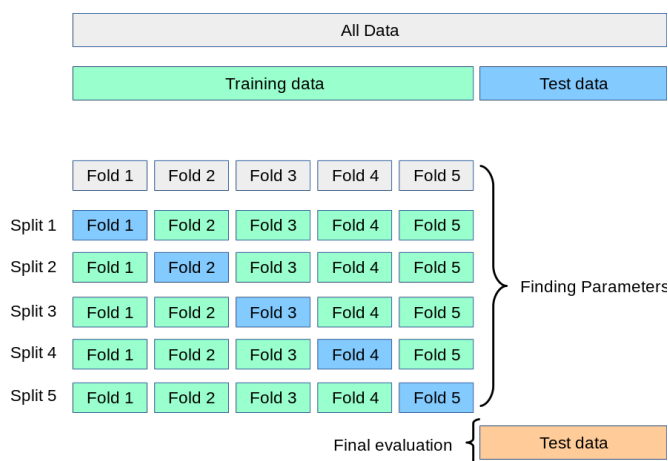
- parametr `n_estimators` udává počet rozhodovacích stromů v lese (obvykle čím více, tím lépe - do kritického počtu, po kterém už se výsledky příliš nezlepšují)
- parametr `criterion` udává funkci pro měření kvality rozdělení
- parametr `max_features` určuje velikost náhodného výběru podmnožiny příznaků [18]

Pro tuto metodu je důležité nastavit parametry `n_estimators` a `max_features`. [19]

7.4. Optimalizace parametrů

Každá výše zmíněná klasifikační metoda pracuje podle zadaných parametrů a může dosahovat různých výsledků. Jelikož předem není známo, které parametry budou nejlépe vyhovovat na použitá data, je potřeba otestovat různé parametry. K tomu lze využít funkci `GridSearchCV()` z knihovny `scikit-learn` z modulu `model_selection`. Ta prochází všechny zadané parametry navzájem a vyhodnotí, která kombinace ze zadaných parametrů dosahuje (podle požadované metriky, např. přesnosti) nejlepších výsledků. [20]

K ohodnocení úspěšnosti parametrů je použito křížové ověření (cross-validation), kdy se zadaná data (podle definovaného parametru `cv`) rozdělí na `cv` stejně velkých skupin dat. Jedna skupina je použita jako testovací, zbytek jako trénovací a vypočítá se úspěšnost. Toto se provede `cv`-krát, pokaždé s jinou skupinou jako testovací. Výsledné hodnocení je dáno průměrem průběžně počítaných úspěšností (viz obrázek 7.1). Cílem křížového ověřování je eliminovat riziko nevhodného rozdělení dat, kdy by model k trénování neměl všechny typy dat a zároveň šetří množství použitých dat. [21]



Obrázek 7.1: Rozdělení dat a křížová validace při optimalizaci parametru. [21]

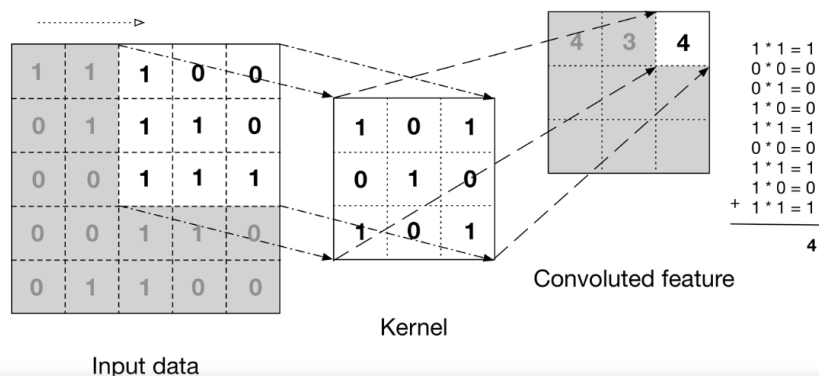
7.5. Konvoluční neuronová síť - Convolutional Neural Network (CNN)

Ke klasifikaci obrazových dat pomocí neuronových sítí je vhodné zvolit konvoluční vrstvy neuronové sítě. Ty dokáží snáze pracovat s velkým množstvím vstupních dat (v podobě obrázku) a redukovat je tak, aby se neztratily kritické vlastnosti pro správnou klasifikaci. [22]

Konvoluční neuronová síť se obvykle skládá z:

- konvolučních vrstev (convolutional layers)
- poolingových vrstev (pooling layers)
- plně propojené vrstvy (fully-connected layer) [23]

Principem **konvoluční vrstvy** je načítání vstupních obrazových dat po malých částech (např. okno velikosti 3 x 3), kde se provádí skalární součin (násobení prvků vahami okna a jejich součet). Tento filtr, který má stále stejně definované váhy se postupně posouvá po obrázku do všech pozic (obvykle s překryvem). Tím vzniká matice konvolovaných hodnot (aktivační mapa filtrů), která se posouvá do dalších vrstev sítě (viz obrázek 7.2). [22]



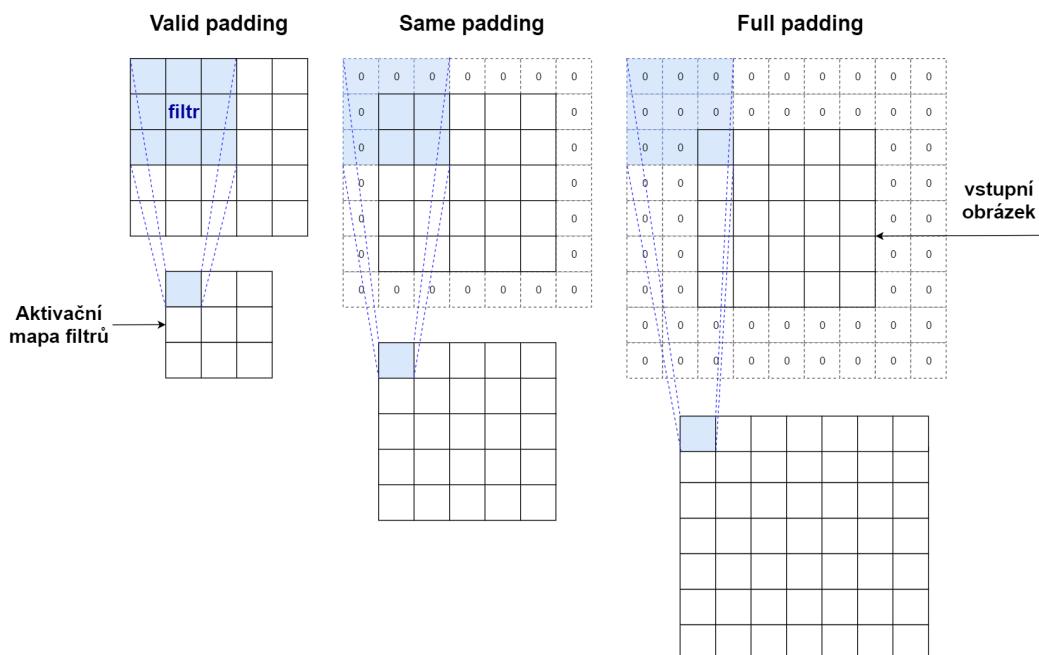
Obrázek 7.2: Znázornění aplikace filtru v konvoluční vrstvě. [23]

7.5. KONVOLUČNÍ NEURONOVÁ SÍŤ - CONVOLUTIONAL NEURAL NETWORK (CNN)

Na aktivační mapu filtrů může být aplikována aktivační funkce. Často se používá ReLU (Rectified linear unit), která zachovává kladné hodnoty a záporné hodnoty nastaví na 0, čímž do modelu vnáší nelinearitu. [24]

Parametry ovlivňující velikost výstupu konvoluční vrstvy:

1. počet filtrů (určuje hloubku výstupu)
2. krok filtru (počet pixelů, o které se filtr posune - obvykle je krok o jeden pixel, v případě většího kroku se velikost výstupu zmenšuje)
3. výplň (padding) - běžně se používá zero-padding, kdy se v pozicích filtru ležících mimo obrazovou matici, uvažuje hodnota 0 (viz obrázek 7.3)
 - valid padding - neboli také bez výplně, kdy se konvoluce vynechá v případě pozice filtru mimo matici, velikost výstupu je pak menší než velikost vstupní matice
 - same padding - doplní se 0 do pozic mimo hranice vstupu tak, aby výstup měl stejnou velikost jako vstupní matice
 - full padding - doplnění 0 mimo hranice vstupu tak, aby velikost výstupu byla větší než vstupu [24]



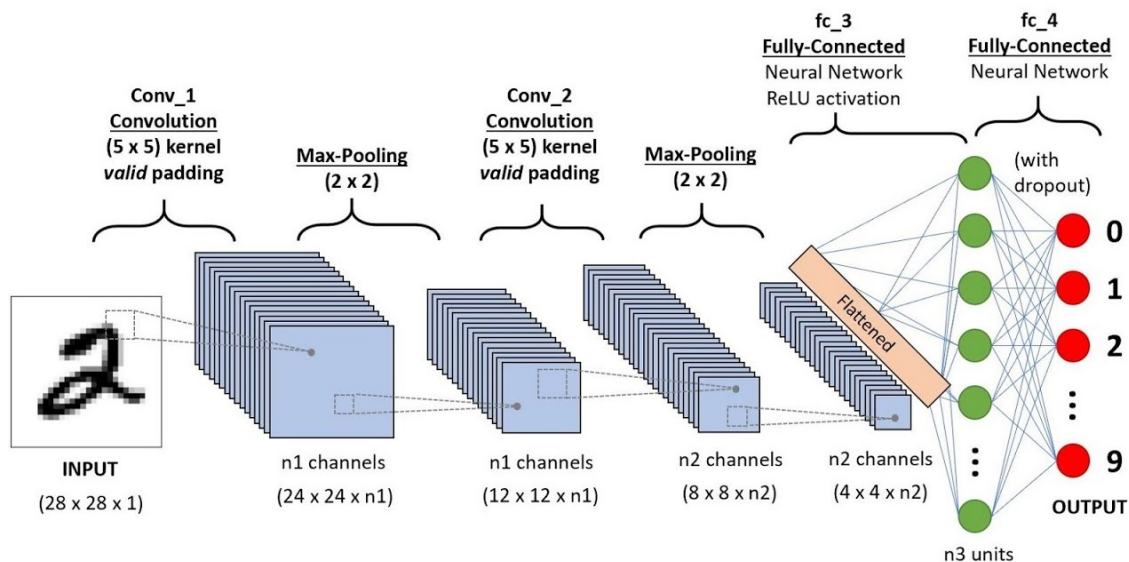
Obrázek 7.3: Ukázka valid padding, same padding a full padding.

Další užitečnou vrstvou jsou **pooling vrstvy**, které snižují velikost vstupních dat a také potlačují šum. Princip je podobný jako pro konvoluční vrstvu, kdy se použije okno (např. velikosti 2 x 2), které se posunuje po obrázku (tentokrát obvykle bez překryvu) a v případě Max Pooling vrstvy se vybere pouze nejvyšší hodnota z oblasti filtru (v případě Average Pooling vrstvy se ukládá průměrná hodnota z filtru). [22]

Těchto vrstev (konvolučních a poolingových) se zpravidla aplikuje více po sobě tak, že se střídají. První konvoluční vrstva sítě extrahuje především základní prvky obrazu (jako

jsou hrany), v dalších konvolučních vrstvách se identifikují složitější prvky (např. od rohů a kontur až po celé objekty). [23]

Na závěr je tento vícerozměrný výstup transformován do jednorozměrného (flattened) a vstupuje do plně propojené neuronové sítě. Ta na rozdíl od konvolučních vrstev zpravidla používá aktivační funkci `softmax`, která udává pravděpodobnost klasifikace (hodnotu mezi 0 a 1).



Obrázek 7.4: Znázornění, jak může vypadat struktura konvoluční sítě. [23]

7.6. Matice záměn (Confusion matrix)

Matice záměn (též chybová matice) se ve strojovém učení využívá k hodnocení úspěšnosti klasifikačních metod. Jedná se o matici, která na řádcích obsahuje hodnoty určené klasifikátorem a ve sloupcích skutečné hodnoty. Na diagonále pak nalezneme správně predikované případy, hodnoty mimo diagonálu značí chybnou předpověď klasifikační metody (například viz obrázek 8.9).

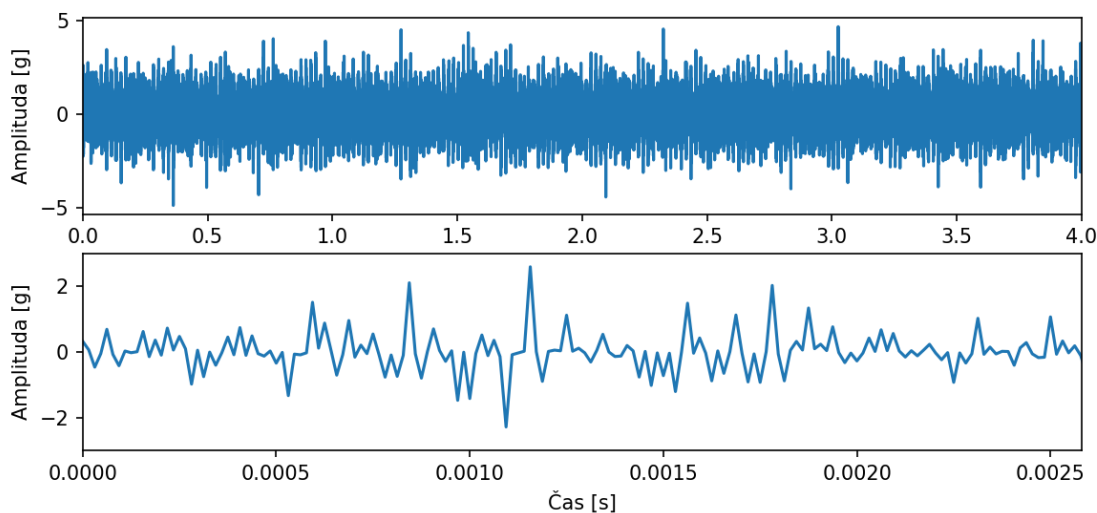
8. Aplikace metod a výsledky

Následující aplikace byly tvořeny v programovacím prostředí Python za použití bezplatných knihoven `scikit-learn` [25] pro strojové učení a `TensorFlow` [26] pro strojové učení a umělou inteligenci. Dále byly využity knihovny `SciPy` [27], `NumPy` [28], `pathlib` ze standardní knihovny Pythonu a pro vykreslování grafů `matplotlib` [29].

8.1. Předzpracování dat a výpočet prediktorů - Dataset A

Data byla ze souborů `.mat` načtena příkazem `loadmat` z knihovny `SciPy`. Z důvodu poškození dat nebyla použita složka dat KA08. Celkem bylo načteno 2480 měření po 4 sekundách, z toho:

- 480 měření ložiska bez poškození (OK)
- 880 měření ložiska s poruchou na vnějším kroužku (OR)
- 240 měření ložiska s kombinovaným poškozením (KB)
- 880 měření ložiska s poruchou na vnitřním kroužku (IR)



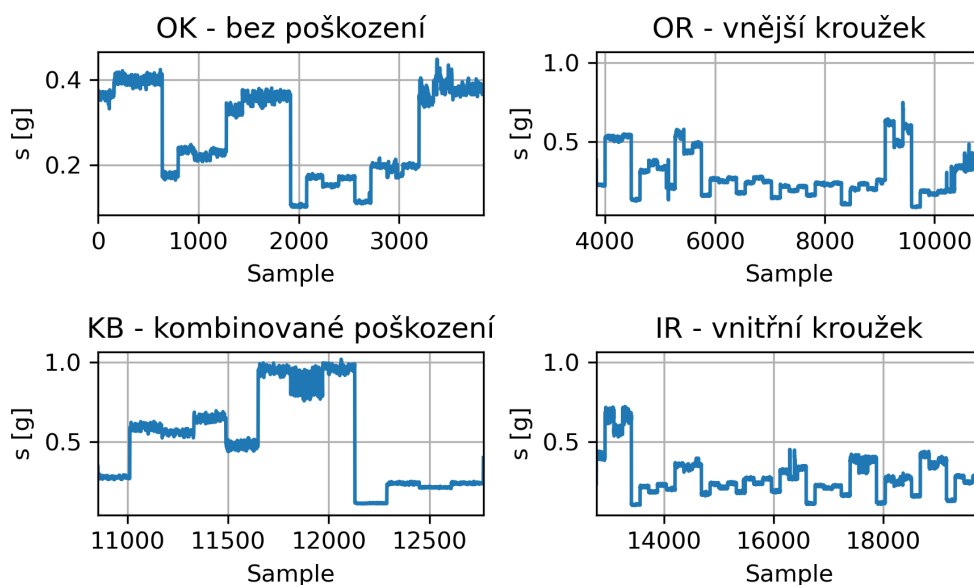
Obrázek 8.1: Ukázka vibrací ložiska bez poškození při $RPM = 900 \text{ min}^{-1}$.

Tato měření byla rozdělena na 0,5 sekundové úseky (celkem tedy 19790 úseků), kde pro každý tento úsek měření byly vypočítány prediktory (viz kapitola 5).

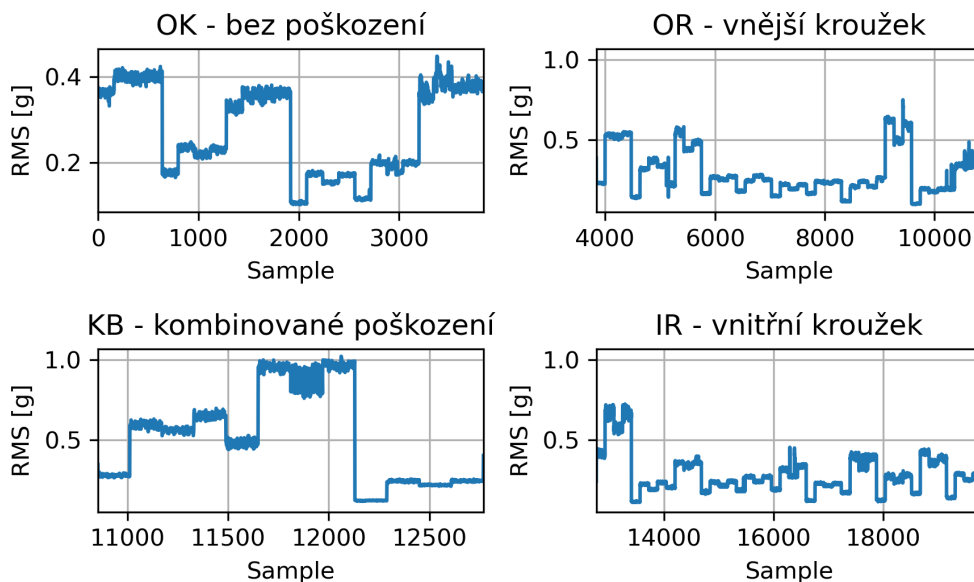
8.1.1. Vypočítané prediktory

Prediktory pro dataset A jsou zobrazeny v grafech na obrázcích 8.2, 8.3, 8.4, 8.5, 8.6 a 8.7.

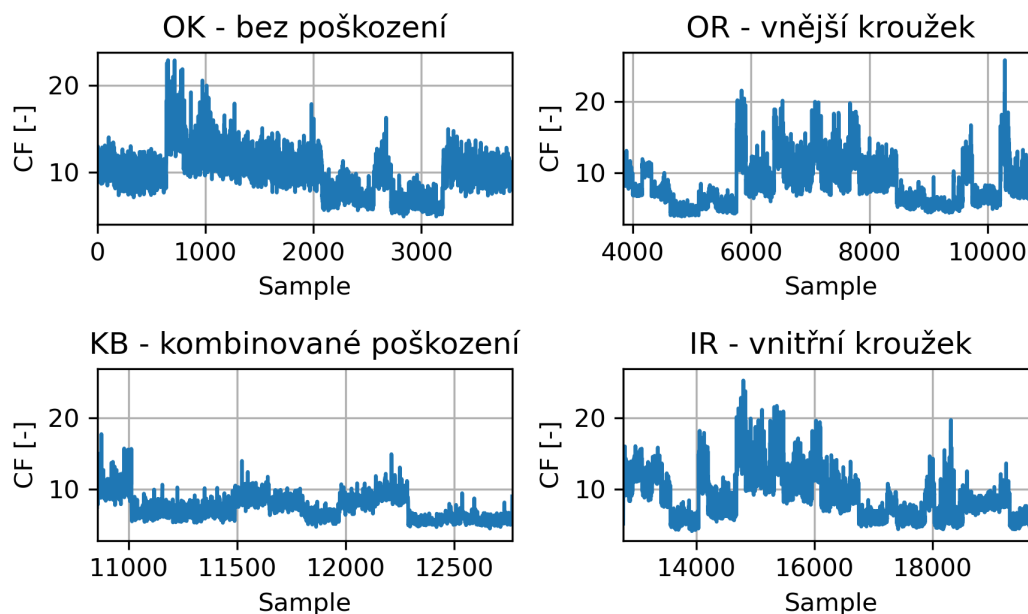
Standardní směrodatná odchylka výběru

Obrázek 8.2: Standardní směrodatná odchylka výběru s pro dataset A.

RMS - efektivní hodnota

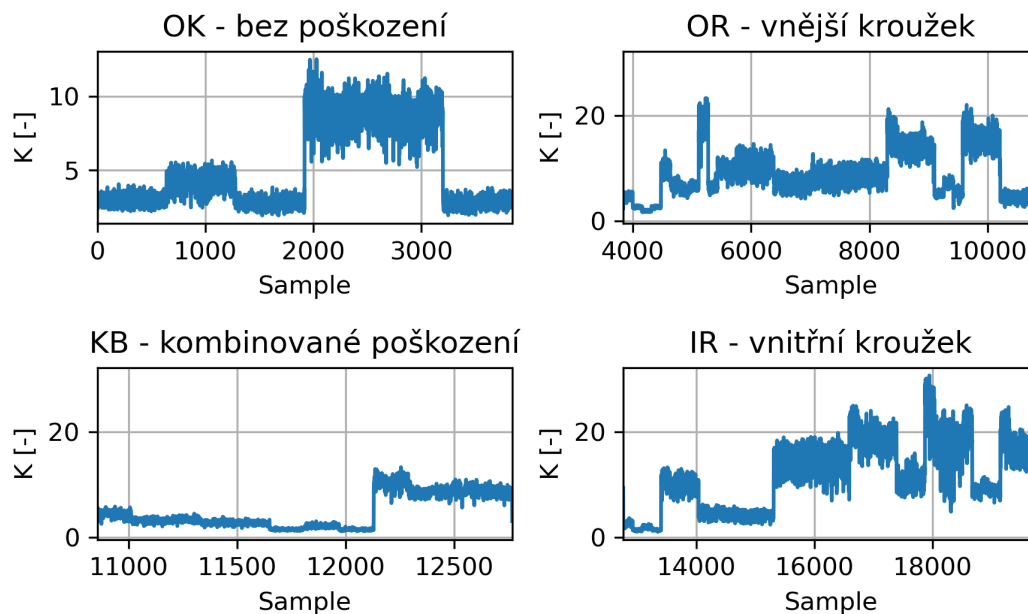
Obrázek 8.3: Efektivní hodnota RMS pro dataset A.

Crest faktor - činitel výkmitu



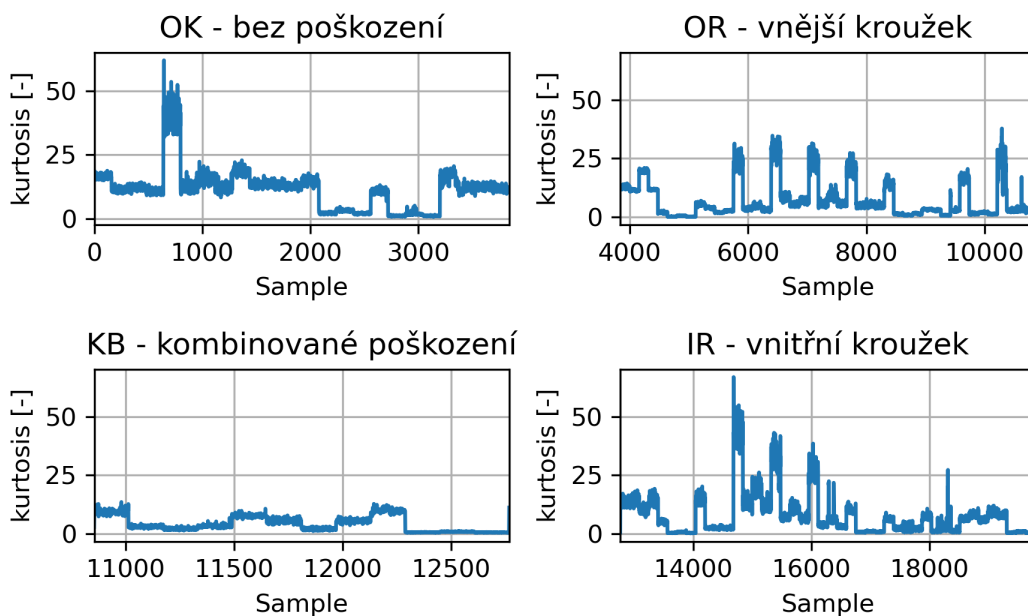
Obrázek 8.4: Činitel výkmitu *crestfaktor* pro dataset A.

K metoda



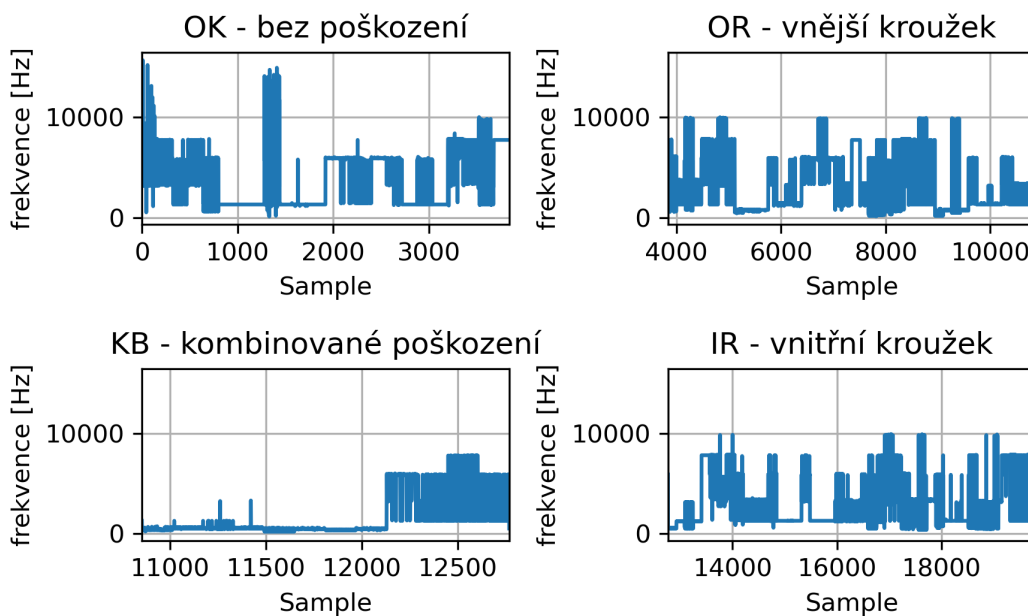
Obrázek 8.5: Parametr $K(t)$ pro dataset A.

Kurtosis faktor



Obrázek 8.6: Kurtosis faktor pro dataset A.

Frekvence maximální hodnoty z FT



Obrázek 8.7: Frekvence s největší amplitudou pro dataset A.

8.1.2. Standardizace prediktorů a optimalizace parametrů

Na vypočítané prediktory byla použita funkce `StandardScaler()` z knihovny `scikit-learn` modulu `preprocessing`. Ta od každé hodnoty odečte průměr a vydělí směrodatnou odchylkou předložených dat. [30] Poté byla data funkcí `train_test_split()` ze stejné knihovny (`scikit-learn` a modulu `preprocessing`) náhodně rozdělena na podmnožinu dat trénovacích (75 %) a testovacích (25 %). [31]

Podle kapitoly 7.4 byly funkcí `GridSearchCV()` s parametrem křížové validace `cv=5` otestovány na trénovacích datech parametry pro metodu SVC (Support Vector Classifier):

- `C = 1, 5, 10, 15, 20, 30, 40, 50, 75, 100, 200, 500, 1000, 2000, 3000`
- `kernel = linear, poly, rbf, sigmoid`
- `gamma = scale, auto`

Pro metodu KNN (K-Nearest Neighbors):

- `n_neighbors = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20`
- `weights = uniform, distance`
- `algorithm = auto, ball_tree, kd_tree, brute`

Pro metodu RFC (Random Forest Classifier):

- `n_estimators = 1, 5, 10, 30, 50, 100, 200, 500, 1000, 1500, 2000`
- `criterion = gini, entropy`

V tabulce 8.1 jsou uvedeny parametry, se kterými bylo dosaženo nejlepších výsledků (největší přesnosti).

Tabulka 8.1: Nejlepší dosažené výsledky optimalizace parametrů

				přesnost
SVC	<code>C = 2000</code>	<code>kernel = rbf</code>	<code>gamma = scale</code>	97,7 %
KNN	<code>n_neighbors = 1</code>	<code>weights = uniform</code>	<code>algorithm = auto</code>	96,5 %
RF	<code>n_estimators = 1500</code>	<code>criterion = gini</code>		97,6 %

8.2. Výsledky metod klasifikace dat - Dataset A

Pro realizaci metod je potřeba mít vektor X ve tvaru $(n_{\text{vzorků}}, n_{\text{příznaků}})$ obsahující trénovací vzorky a vektor y ve tvaru $(n_{\text{vzorků}})$, které nesou informaci o cílové hodnotě (klasifikační třídě) příslušného vzorku (viz obrázek 8.8).

Vektor X pro Dataset A obsahuje 19790 řádků. Na každém řádku je 6 prediktorů a to: standardní směrodatná odchylka výběru, RMS, crest faktor, parametr K , Kurtosis faktor a frekvence s největší amplitudou viz obrázek 8.8.

směrodatná odchylna	RMS	crest faktor	parametr K	Kurtosis faktor	frekvence s největší amplitudou
---------------------	-----	--------------	------------	-----------------	---------------------------------

Obrázek 8.8: Znárodnění jednoho řádku vektoru X pro Dataset A.

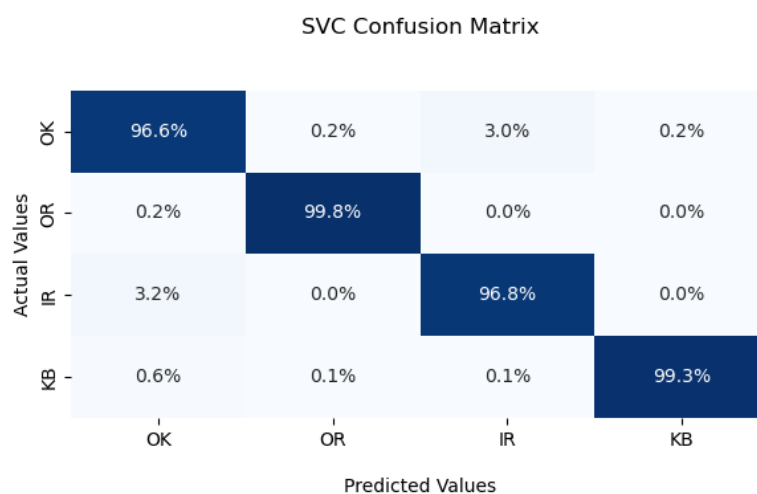
8.2.1. Support Vector Classifier - metoda podpůrných vektorů

Na testovací podmnožinu byla použita funkce z knihovny `scikit-learn` modulu `svm`:

```
SVC(decision_function_shape='ovo',kernel='rbf',C=2000,gamma='scale'),
```

- `decision_function_shape='ovo'` je schéma one-vs-one, které umožňuje klasifikaci do více tříd
- `kernel` určuje jádrovou funkci
- `C` je regularizační parametr, který dovoluje bodům ležet mimo své hranice
- Parametr `gamma` určuje, jak velký vliv jednotlivá trénovací data mají [16]
- ostatní parametry jsou zadány defaultně

Metoda při vyhodnocení úspěšnosti testovacími daty dosahovala přesnosti **97,9 %**, její matice záměn je zobrazena na obrázku 8.9.



Obrázek 8.9: Confusion matrix (matice záměn) metody SVC s přesností 97,9 % při použití parametrů: `kernel='rbf'`, `C=2000`, `gamma='scale'`.

8.2. VÝSLEDKY METOD KLASIFIKACE DAT - DATASET A

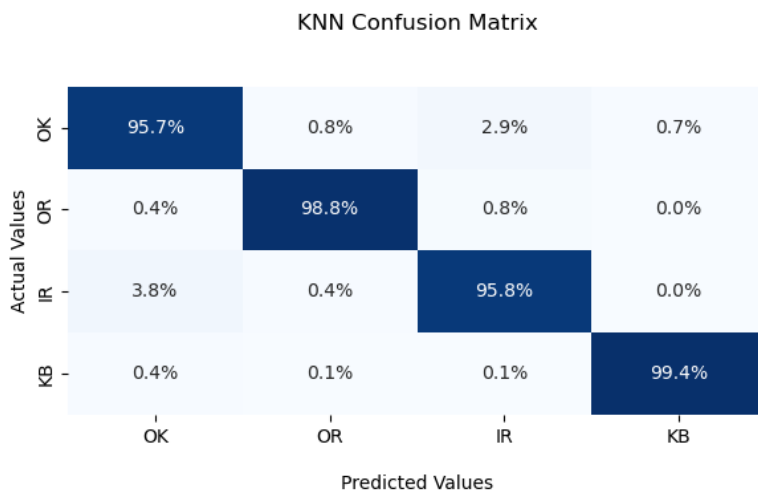
8.2.2. K-Neighbors Classifier - klasifikace podle k nejbližších sousedů

Na testovací podmnožinu byla použita funkce z knihovny `scikit-learn` modulu `neighbors`:

```
KNeighborsClassifier(n_neighbors=1, algorithm='auto', weights='uniform'),
```

- parametr `n_neighbors` definuje počet nejbližších sousedů, podle kterých bude určována klasifikační třída
- parametr `weights` určuje, zda budou mít někteří sousedi větší váhu při rozhodování ('`distance`'), nebo zda bude každému sousedovi přiřazena jednotná váha ('`uniform`')
- parametr '`auto`' vybírá nejvhodnější typ algoritmu (Brute Force, K-D Tree, Ball Tree) podle předložených dat. [32]
- ostatní parametry jsou zadány defaultně

Při vyhodnocení úspěšnosti pomocí testovacích dat metoda dosahovala přesnosti **97,4 %**, její matice záměn je zobrazena na obrázku 8.10.



Obrázek 8.10: Confusion matrix (matice záměn) metody KNN s přesností 97,4 % při použití parametrů: `n_neighbors=1`, `algorithm='auto'`, `weights='uniform'`.

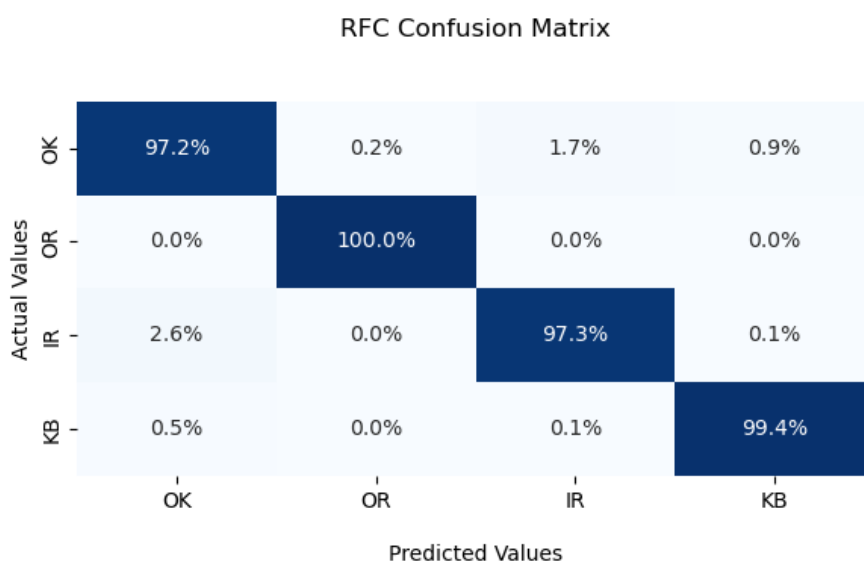
8.2.3. Random Forrest Classifier - klasifikace metodou náhodného lesa

Na testovací podmnožinu byla použita funkce z knihovny `scikit-learn` modulu `ensemble`:

```
RandomForestClassifier(n_estimators=1500,criterion='gini'),
```

- parametr `n_estimators` udává počet rozhodovacích stromů v lese
- `max_features` je defaultně nastaveno na `'sqrt'` jako odmocnina z počtu příznaků v datasetu
- parametr `criterion` udává funkci pro měření kvality rozdělení
- ostatní parametry jsou zadány defaultně

Metoda při vyhodnocení úspěšnosti testovacími daty dosahovala přesnosti **98,3 %**, její matice záměn je zobrazena na obrázku 8.11.

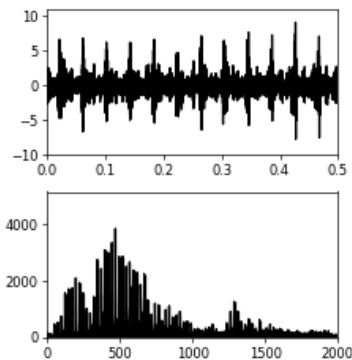


Obrázek 8.11: Confusion matrix (matice záměn) metody RFC s přesností 98,3 % při použití parametrů: `n_estimators=1500`, `max_features='sqrt'`, `criterion='gini'`.

8.3. Konvoluční neuronová síť (CNN) - Dataset A

Za účelem klasifikace dat pomocí konvoluční neuronové sítě byly vytvořeny obrázky grafů o velikosti (240 x 240), kde je v horní části zobrazen průběh vibrací v časové oblasti a v dolní části je jeho frekvenční analýza (viz obrázek 8.12).

8.3. KONVOLUČNÍ NEURONOVÁ SÍŤ (CNN) - DATASET A



Obrázek 8.12: Ukázka obrázku trénovacích dat pro konvoluční neuronovou síť - obrázek o velikosti (240 x 240). Nahoře časová oblast, dole frekvenční oblast kombinovaného poškození.

Příprava dat

- do proměnné `X_img` jsou postupně načítány obrázky jako odstíny šedé (grayscale)
- do `y` jsou načteny cílové hodnoty (stavy ložiska) v číselné podobě 0 až 3
- funkcí `to_categorical` z knihovny `tensorflow.keras.utils` jsou do proměnné `y_one_hot` cílové hodnoty `y` upraveny do podoby vektoru o $(n-1)$ nulách a jedničce, jejíž pozice určuje stav (cílovou hodnotu), kde n je počet klasifikačních tříd
- načtená obrazová data `X_img` a upravené cílové hodnoty `y_one_hot` jsou náhodně rozděleny na trénovací data (75 %) `X_train` a testovací data (25 %) `y_train` funkcí `train_test_split()` z knihovny `sklearn.model_selection`

Postup trénování a testování sítě

- jako první se obrazová data normalizují (převědou z hodnot 0-255 na hodnoty mezi 0 a 1) pro snazší práci s daty
- zadá se tvar vstupních dat (výška obrázku, šířka obrázku, počet barevných kanálů)
- určí se architektura sítě (viz modely sítí 8.3.1, 8.3.2, 8.3.3 a 8.3.4), kde počet prvků výstupní vrstvy se rovná počtu klasifikačních tříd

8.3.1. CNN - model 1

Model konvoluční sítě je tvořen:

- 3 konvolučními vrstvami s filtrem velikosti (3x3)
- následované poolingovými vrstvami s filtrem velikosti (2x2)
- ve všech případech se jedná o same padding
- aktivační funkce jsou *ReLU* pro první konvoluční vrstvy a *softmax* pro výstupní vrstvu sítě

- následuje plně propojená síť o velikosti 64 prvků
- počet prvků výstupní vrstvy se rovná počtu klasifikačních tříd (4)

Kód skriptu

```
model = Sequential([
layers.Rescaling(1./255, input_shape=(img_size, img_size, 1)),
layers.Conv2D(64, (3,3), padding='same', activation='relu'),
layers.MaxPooling2D(pool_size=(2,2)),
layers.Conv2D(64, (3,3), padding='same', activation='relu'),
layers.MaxPooling2D(pool_size=(2,2)),
layers.Conv2D(64, (3,3), padding='same', activation='relu'),
layers.MaxPooling2D(pool_size=(2,2)),
layers.Flatten(),
layers.Dense(64),
layers.Dense(num_classes, activation='softmax') ])
```

Kompilace sítě s parametry:

- běžně používaný optimalizátor 'adam' [33]
- ztrátová funkce loss 'categorical_crossentropy' [34]
- jako metrika byla zvolena přesnost 'accuracy'

```
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

Trénování modelu

- vstupy jsou trénovací data `X_img_train` a jejich cílové hodnoty `y_img_train`
- počet epoch (iterací) se rovná 10
- 20 % z trénovacích dat slouží k validaci modelu po skončení každé epochy

```
model.fit(X_img_train,y_img_train, epochs=10, validation_split=0.2)
```

Průběh trénování sítě

Na obrázku 8.13 je zobrazen průběh přesnosti trénovacích a validačních dat pro jednotlivé epochy tréninku.

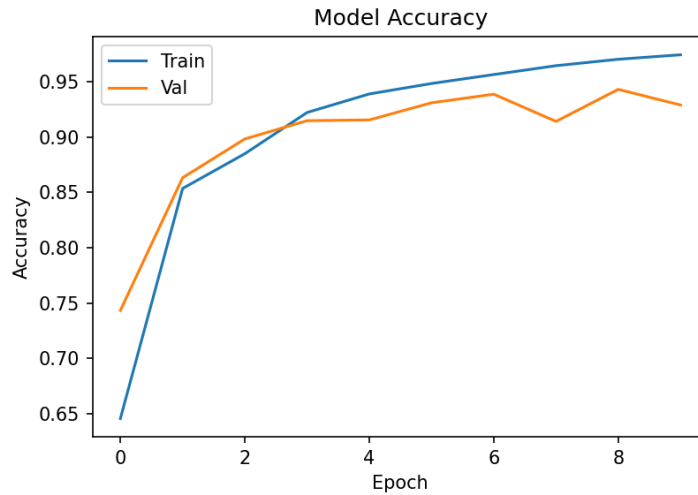
Hodnocení modelu

Pro hodnocení modelu jsou použita testovací data.

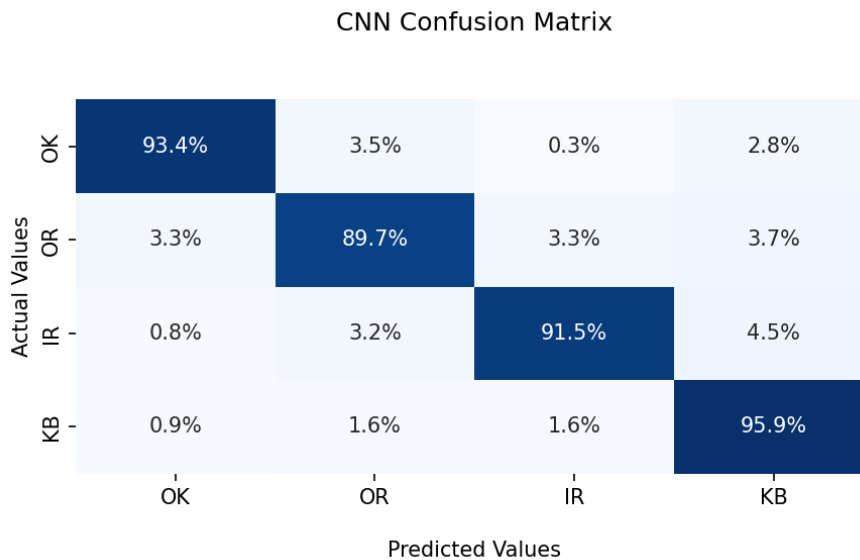
```
model.evaluate(X_img_test,y_img_test_one_hot)[1]
```

Tento model dosahuje přesnosti **92,7 %**, jeho matice záměn je znázorněna na obrázku 8.14

8.3. KONVOLUČNÍ NEURONOVÁ SÍŤ (CNN) - DATASET A



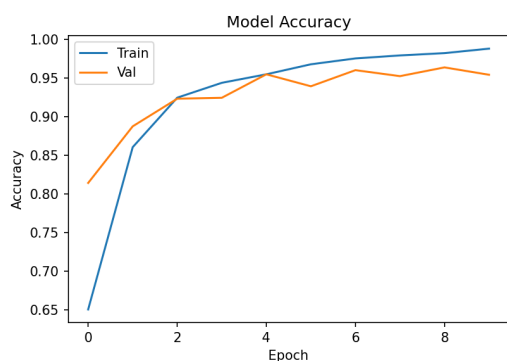
Obrázek 8.13: Graf přesnosti trénovacích a validačních dat pro jednotlivé epochy tréninku.



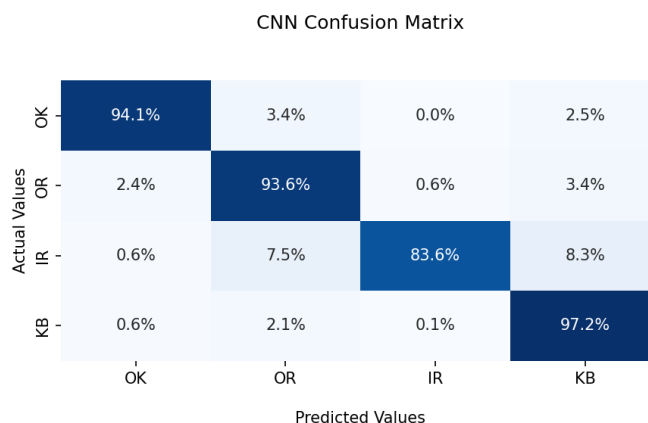
Obrázek 8.14: Confusion matrix (matice záměn) konvoluční neuronové sítě modelu 1 s přesností 92,7 %.

8.3.2. CNN - model 2

Tento model je obdobný modelu 1 (viz kapitola 8.3.1). Liší se aktivační funkcí ve výstupní vrstvě, která je pro tento model 'sigmoid'. Model dosahuje přesnosti **94,1 %**, graf průběhu trénování a matice záměn jsou na obrázcích 8.15 a 8.16.



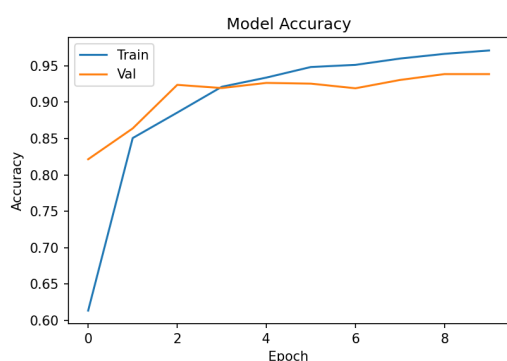
Obrázek 8.15: Graf přesnosti trénovačích a validačních dat.



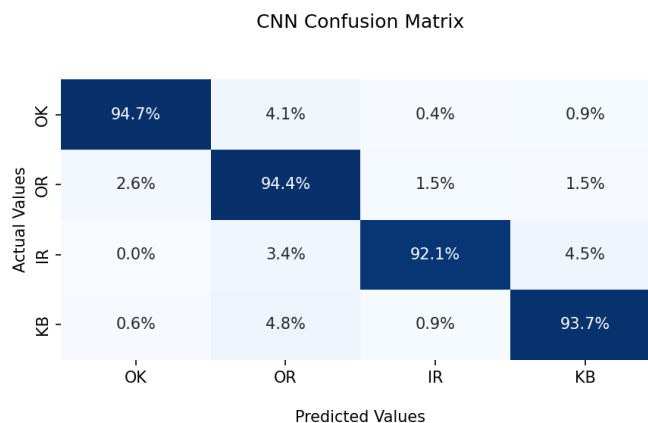
Obrázek 8.16: Confusion matrix s přesností 94,1 %.

8.3.3. CNN - model 3

Model je obdobný modelu 1 a 2 (viz kapitoly 8.3.1 a 8.3.2). Aktivační funkce byla ponechána jako v modelu 2 'sigmoid', akorát počet prvků plně propojené vrstvy (`Dense()`) se rovná 128. Model dosahuje přesnosti **94,0 %**, graf průběhu trénování a matice záměn jsou na obrázcích 8.17 a 8.18.



Obrázek 8.17: Graf přesnosti trénovačích a validačních dat.

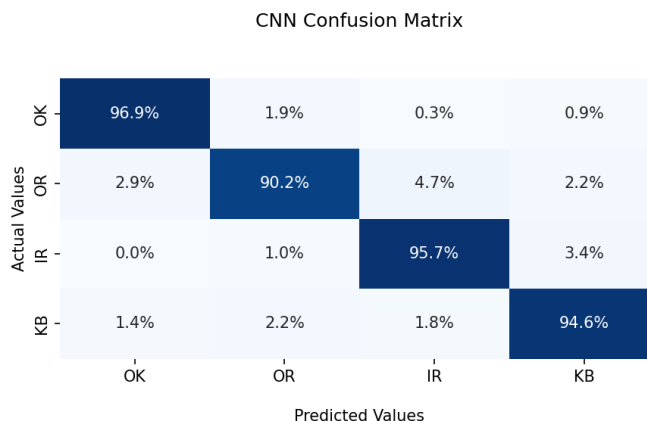
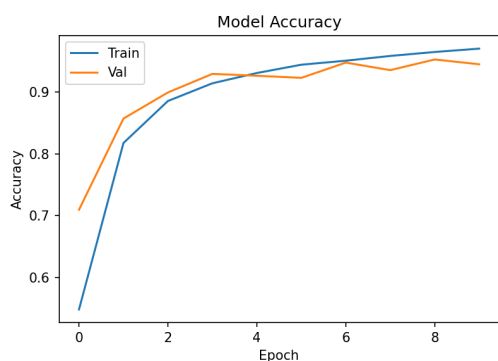


Obrázek 8.18: Confusion matrix s přesností 94,0 %.

8.3.4. CNN - model 4

Čtvrtý model je obdobný modelu 2 (viz kapitola 8.3.2). Aktivační funkce ve výstupní vrstvě je pro tento model 'sigmoid' a počet prvků plně propojené vrstvy (`Dense()`) se rovná 64. Změnou oproti předchozím modelům je filtr konvolučních vrstev, které byly všechny nastaveny na velikost (5x5). Model dosahuje přesnosti **93,6 %**, graf průběhu trénování a matice záměn jsou na obrázcích 8.19 a 8.20.

8.4. PŘEDZPRACOVÁNÍ DAT A VÝPOČET PREDIKTORŮ - DATASET B

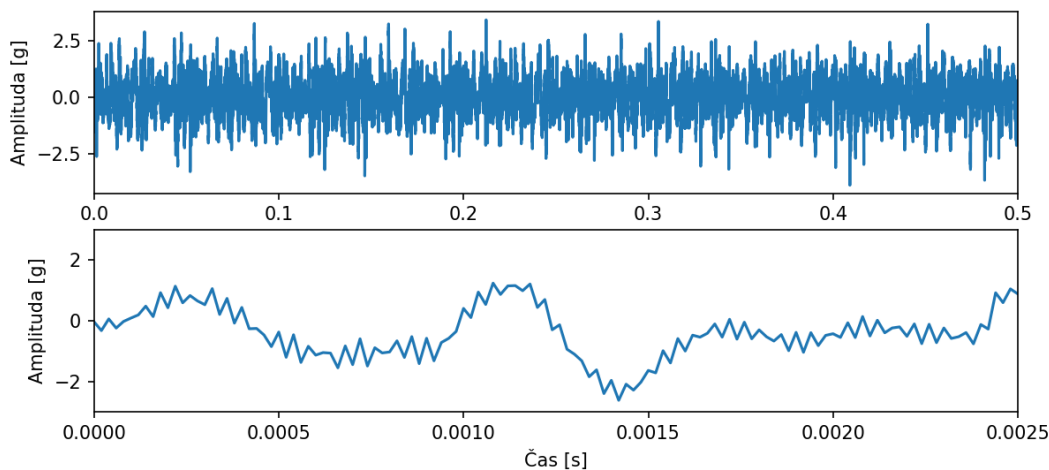


Obrázek 8.19: Graf přesnosti trénova-
cích a validačních dat.

Obrázek 8.20: Confusion matrix s přesností
93,6 %.

8.4. Předzpracování dat a výpočet prediktorů - Dataset B

Data byla ze souborů .csv načtena příkazem `reader` modulu `csv` ze standardní knihovny Pythonu. Celkem bylo načteno 1120 měření po 5 sekundách pro obě pozice (underhang a overhang) a každý směr (axiální, radiální, tangenciální).



Obrázek 8.21: Ukázka Vibrací ložiska v axiálním směru s poškozením kuličky při rychlosti otáčení $RPM = 846 \text{ min}^{-1}$.

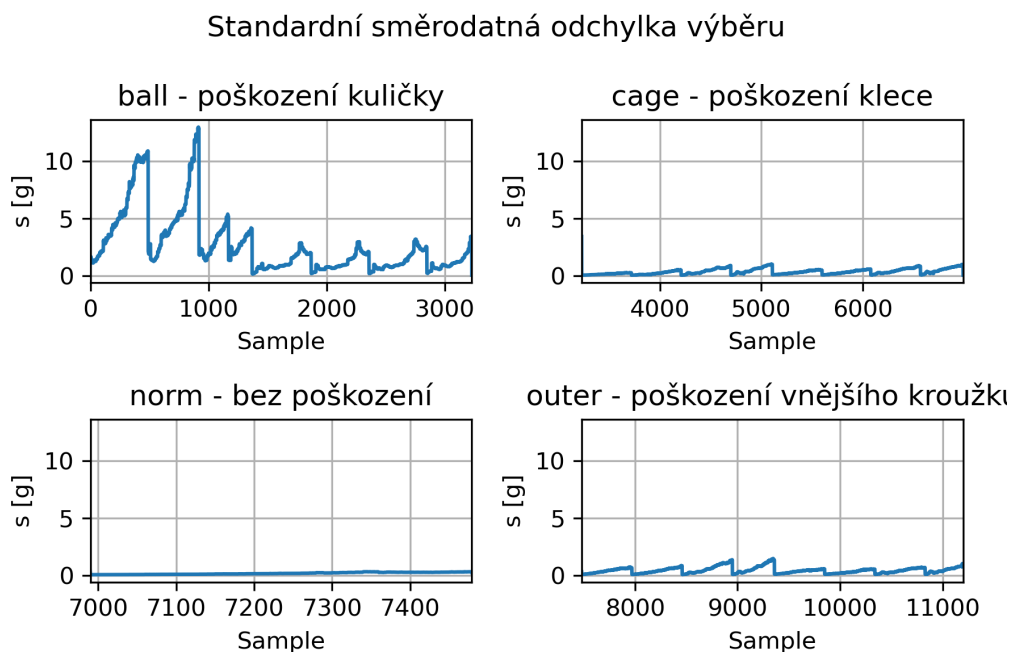
Tato měření byla rozdělena na 0,5 sekundové úseky (celkem tedy 11200 úseků), z toho:

- 3230 úseků ložiska s poškozením kuličky (ball)
- 3720 úseků ložiska s poruchou na vnějším kroužku (outer)
- 490 úseků ložiska bez poškození (norm)
- 3760 úseků ložiska s poruchou klece (cage)

Pro každý tento úsek měření byly vypočítány prediktory (viz kapitola 5).

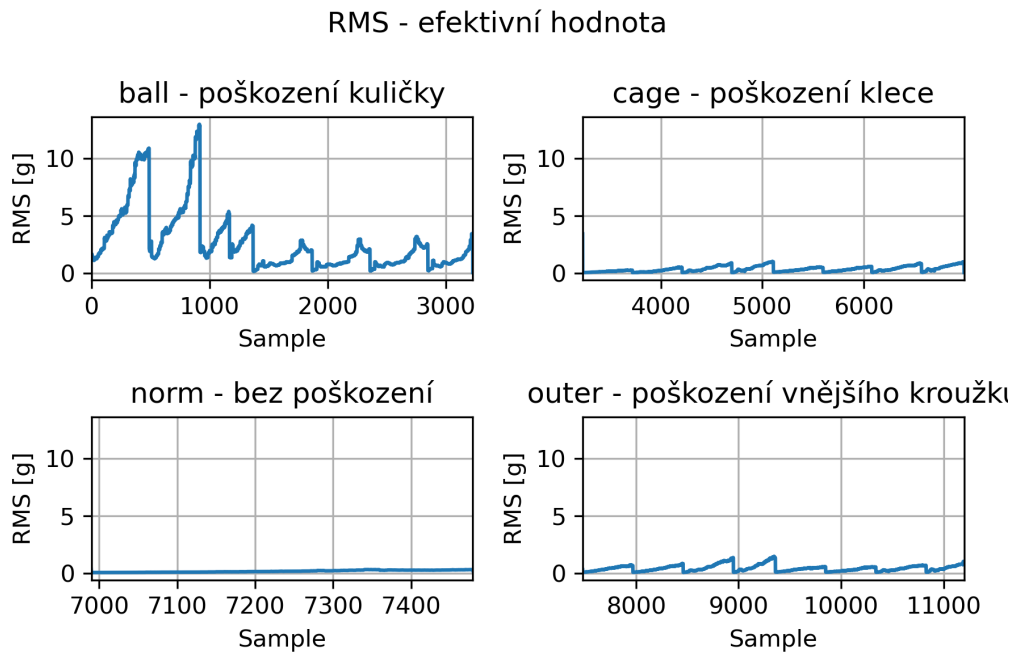
8.4.1. Vypočítané prediktory

Prediktory datasetu B pro radiální směr v pozici underhang jsou zobrazeny v grafech na obrázcích 8.22, 8.23, 8.24, 8.25, 8.26 a 8.27. Prediktory pro další směry je možné nalézt v příloze B.

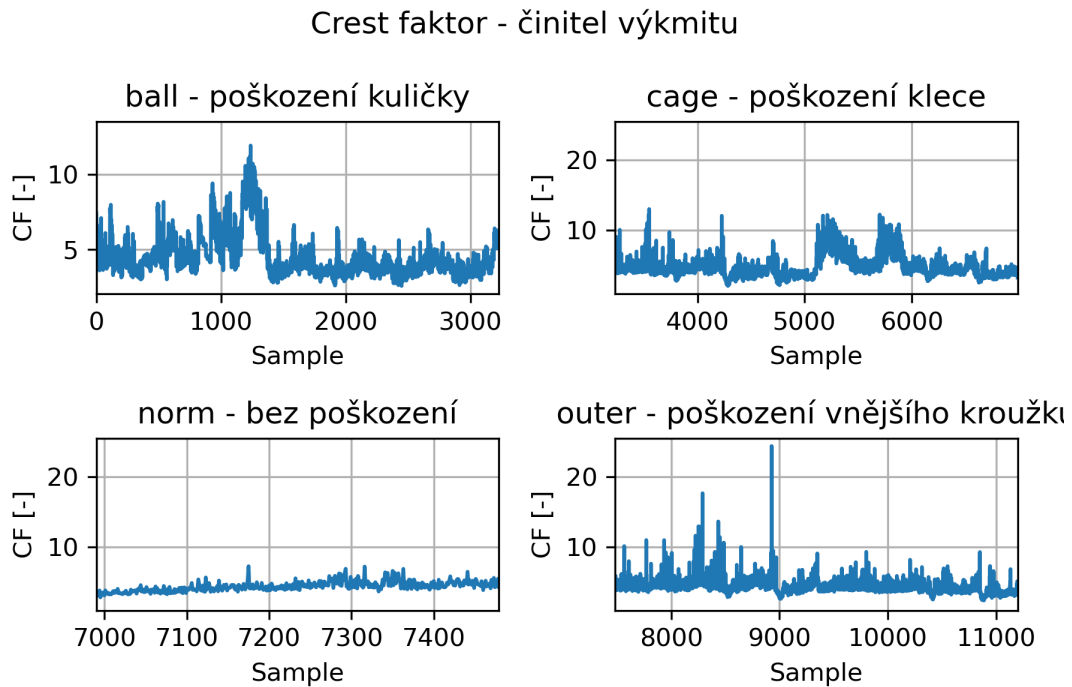


Obrázek 8.22: Standardní směrodatná odchylka výběru s pro radiální směr v pozici underhang datasetu B.

8.4. PŘEDZPRACOVÁNÍ DAT A VÝPOČET PREDIKTORŮ - DATASET B

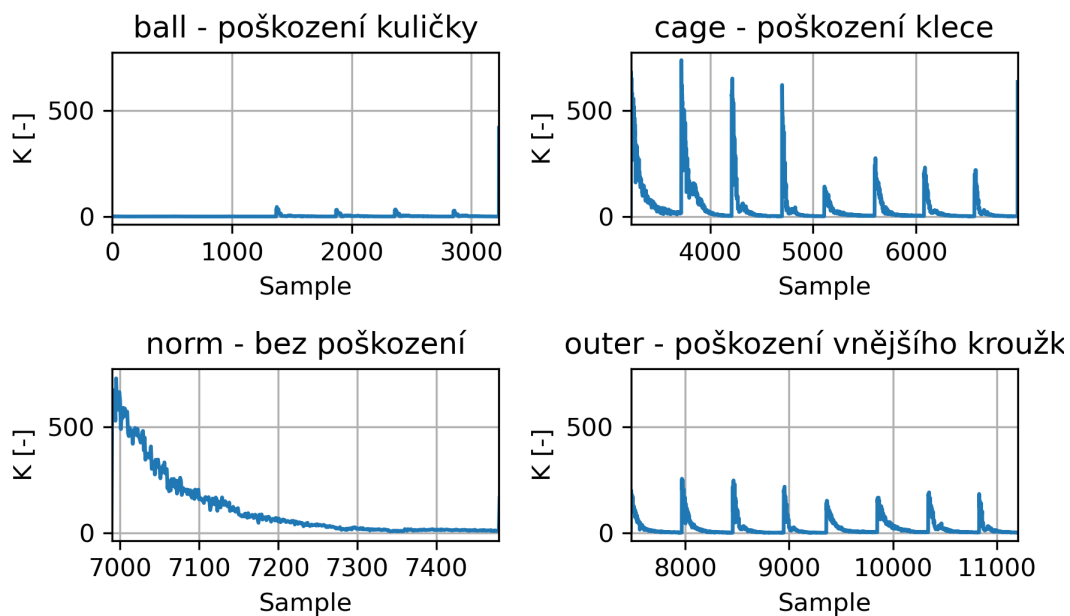


Obrázek 8.23: Efektivní hodnota RMS pro radiální směr v pozici underhang datasetu B.

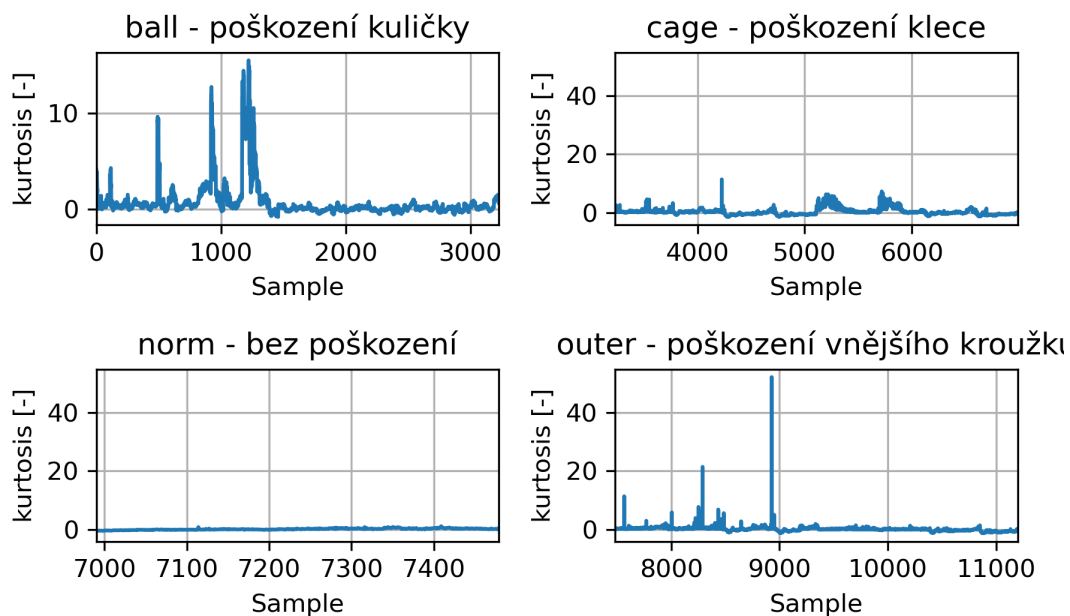


Obrázek 8.24: Činitel výkmitu $crestfaktor$ pro radiální směr v pozici underhang datasetu B.

K metoda

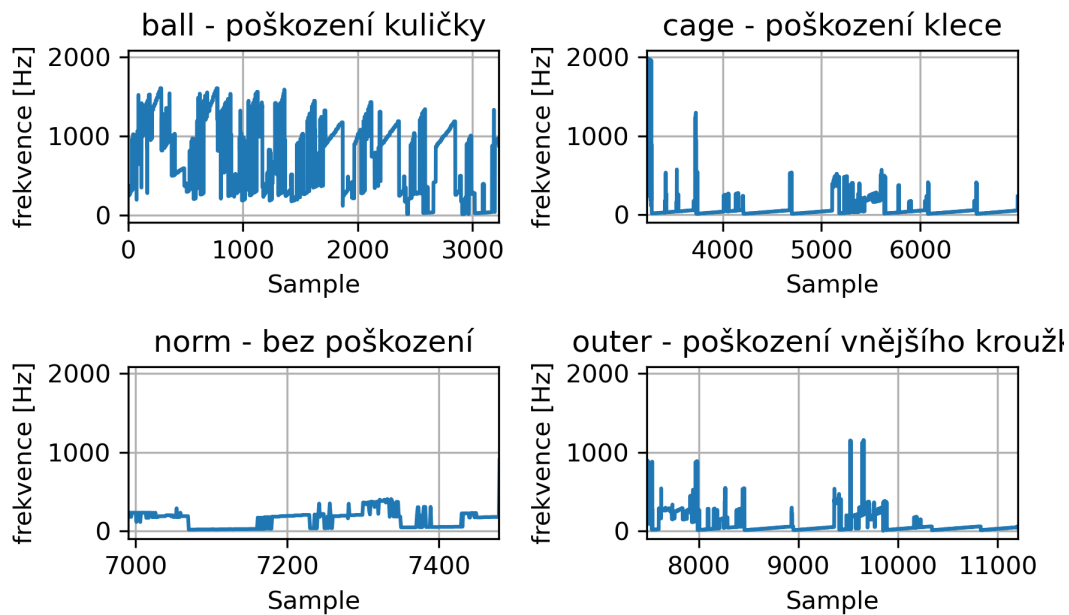
Obrázek 8.25: Parametr $K(t)$ pro radiální směr v pozici underhang datasetu B.

Kurtosis faktor



Obrázek 8.26: Kurtosis faktor pro radiální směr v pozici underhang datasetu B.

Frekvence maximální hodnoty z FT



Obrázek 8.27: Frekvence s největší amplitudou pro radiální směr v pozici underhang datasetu B.

8.4.2. Standardizace prediktorů a optimalizace parametrů

Na vypočítané prediktory byla použita funkce `StandardScaler()` z knihovny `scikit-learn` modulu `preprocessing`. Ta od každé hodnoty odečte průměr a vydělí směrodatnou odchylkou předložených dat. [30]

Poté byla data funkcí `train_test_split()` ze stejné knihovny (`scikit-learn` a modulu `preprocessing`) náhodně rozdělena na podmnožinu dat trénovacích (75 %) a testovacích (25 %). [31]

Podle kapitoly 7.4 byly funkcí `GridSearchCV()` s parametrem křížové validace `cv=5` otestovány na trénovacích datech parametry pro metodu SVC (Support Vector Classifier):

- `C = 1, 5, 10, 15, 20, 30, 40, 50, 75, 100, 200, 500, 1000`
- `kernel = linear, poly, rbf, sigmoid`
- `gamma = scale, auto`

Pro metodu KNN (K-Nearest Neighbors):

- `n_neighbors = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20`
- `weights = uniform, distance`
- `algorithm = auto, ball_tree, kd_tree, brute`

Pro metodu RFC (Random Forest Classifier):

- `n_estimators` = 1, 5, 10, 30, 50, 100, 200, 500, 1000, 1500, 2000
- `criterion` = gini, entropy

V tabulce 8.2 jsou uvedeny parametry, se kterými bylo dosaženo nejlepších výsledků (největší přesnosti).

Tabulka 8.2: Nejlepší dosažené výsledky optimalizace parametrů

				přesnost
SVC	<code>C = 200</code>	<code>kernel = rbf</code>	<code>gamma = auto</code>	99,7 %
KNN	<code>n_neighbors = 4</code>	<code>weights = distance</code>	<code>algorithm = auto</code>	99,6 %
RF	<code>n_estimators = 500</code>	<code>criterion = gini</code>		99,8 %

8.5. Výsledky metod klasifikace dat - Dataset B

Pro realizaci metod je potřeba mít vektor X ve tvaru (`n_vzorků, n_příznaků`) obsahující trénovací vzorky a vektor y ve tvaru (`n_vzorků`), které nesou informaci o cílové hodnotě (klasifikační třídě) příslušného vzorku.

Vektor X pro Dataset B obsahuje 11 200 řádků. Na každém řádku je 36 prediktorů a to (standardní směrodatná odchylka výběru, RMS, crest faktor, parametr K , Kurtosis faktor a frekvence s největší amplitudou) pro každý směr (radiální, axiální, tangenciální) a pozici (overhang a underhang) viz obrázek 8.28.

1	směrodatná odchylka	RMS	crest faktor	parametr K	Kurtosis faktor	frekvence s největší amplitudou	2	3	4	5	6

- 1: axiální směr, pozice underhang
- 2: axiální směr, pozice overhang
- 3: radiální směr, pozice underhang
- 4: radiální směr, pozice overhang
- 5: tangenciální směr, pozice underhang
- 6: tangenciální směr, pozice overhang

Obrázek 8.28: Znázornění jednoho řádku vektoru X pro Dataset B.

8.5. VÝSLEDKY METOD KLASIFIKACE DAT - DATASET B

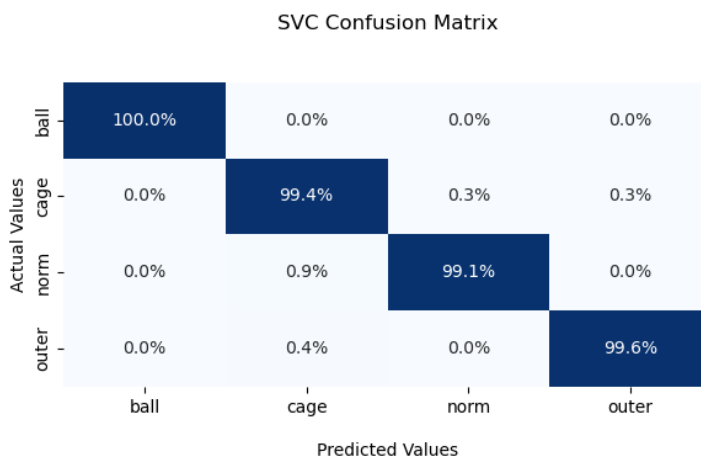
8.5.1. Support Vector Classifier - metoda podpůrných vektorů

Analogicky jako při Datasetu A byla na testovací podmnožinu použita funkce z knihovny `scikit-learn` modulu `svm`:

```
SVC(decision_function_shape='ovo',kernel='rbf',C=200,gamma='auto'),
```

- `decision_function_shape='ovo'` je schéma one-vs-one, které umožňuje klasifikaci do více tříd
- `kernel` určuje jádrovou funkci
- `C` je regularizační parametr, který dovoluje bodům ležet mimo své hranice
- Parametr `gamma` určuje, jak velký vliv jednotlivá trénovací data mají [16]
- ostatní parametry jsou zadány defaultně

Metoda při vyhodnocení úspěšnosti testovacími daty dosahovala přesnosti **99,6 %**, její matice záměn je zobrazena na obrázku 8.29.



Obrázek 8.29: Confusion matrix (matice záměn) metody SVC s přesností 99,6 % při použití parametrů: `kernel='rbf'`, `C=200`, `gamma='auto'`.

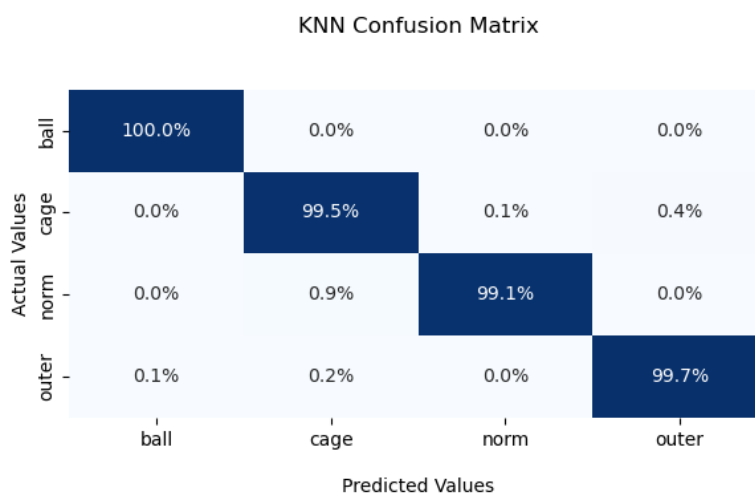
8.5.2. K-Neighbors Classifier - klasifikace podle k nejbližších sousedů

Na testovací podmnožinu byla použita funkce z knihovny `scikit-learn` modulu `neighbors`:

```
KNeighborsClassifier(n_neighbors=4, algorithm='auto', weights='distance'),
```

- parametr `n_neighbors` definuje počet nejbližších sousedů, podle kterých bude určována klasifikační třída
- parametr `weights` určuje, zda budou mít někteří sousedi větší váhu při rozhodování ('distance'), nebo zda bude každému sousedovi přiřazena jednotná váha ('uniform')
- parametr 'auto' vybírá nejvhodnější typ algoritmu (Brute Force, K-D Tree, Ball Tree) podle předložených dat. [32]
- ostatní parametry jsou zadány defaultně

Při vyhodnocení úspěšnosti pomocí testovacích dat metoda dosahovala přesnosti **99,7 %**, její matice záměn je zobrazena na obrázku 8.30.



Obrázek 8.30: Confusion matrix (matice záměn) metody KNN s přesností 99,7 % při použití parametrů: `n_neighbors=4`, `algorithm='auto'`, `weights='distance'`.

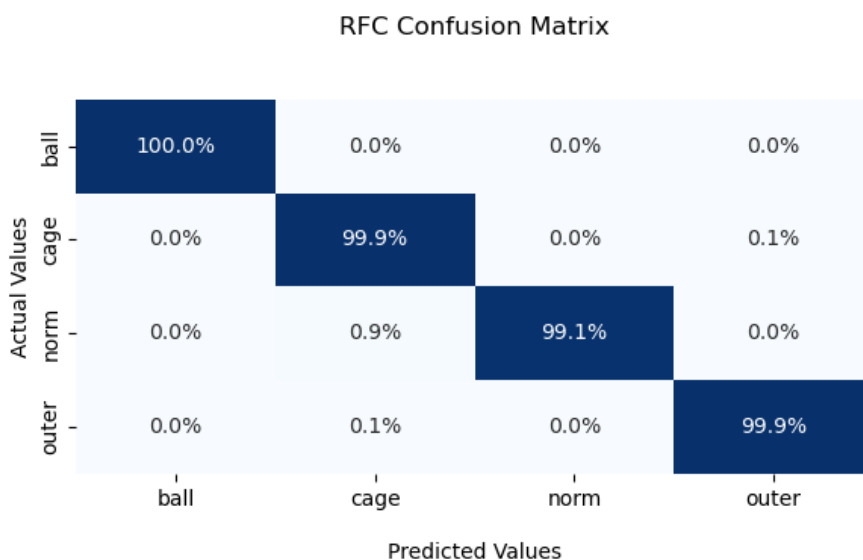
8.5.3. Random Forrest Classifier - klasifikace metodou náhodného lesa

Na testovací podmnožinu byla použita funkce z knihovny `scikit-learn` modulu `ensemble`:

```
RandomForestClassifier(n_estimators=500,criterion='gini'),
```

- parametr `n_estimators` udává počet rozhodovacích stromů v lese
- `max_features` je defaultně nastaveno na `'sqrt'` jako odmocnina z počtu příznaků v datasetu
- parametr `criterion` udává funkci pro měření kvality rozdělení
- ostatní parametry jsou zadány defaultně

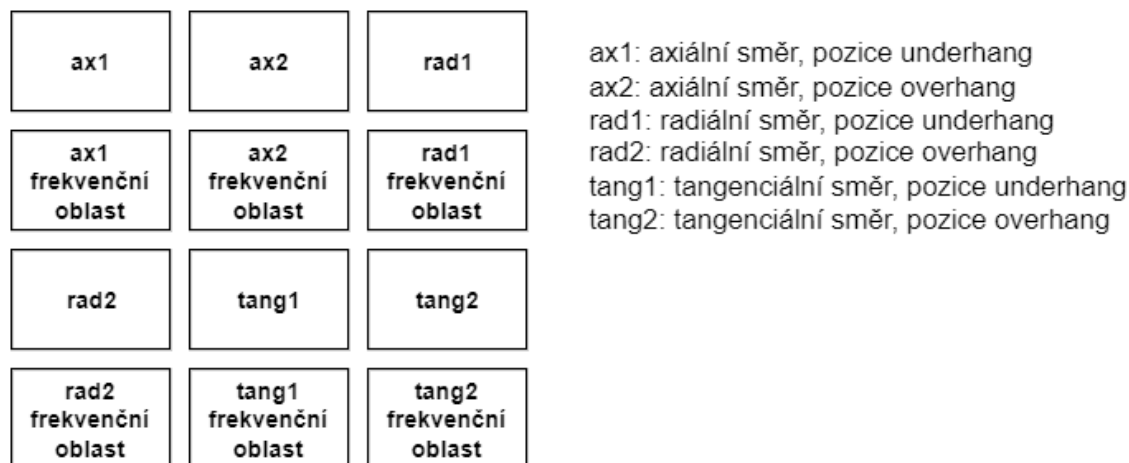
Metoda při vyhodnocení úspěšnosti testovacími daty dosahovala přesnosti **99,9 %**, její matice záměn je zobrazena na obrázku 8.31.



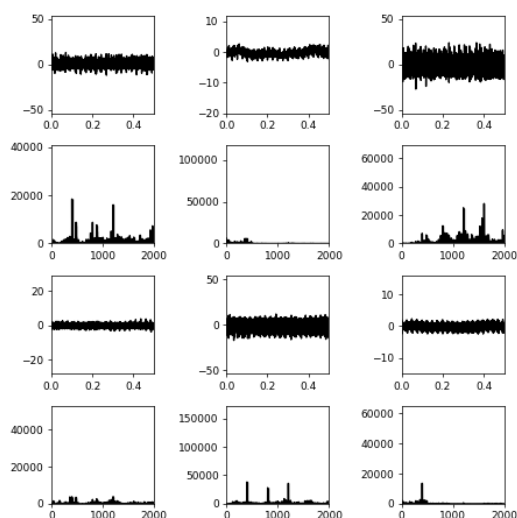
Obrázek 8.31: Confusion matrix (matice záměn) metody RFC s přesností 99,9 % při použití parametrů: `n_estimators=500`, `max_features='sqrt'`, `criterion='gini'`.

8.6. Konvoluční neuronová síť (CNN) - Dataset B

Za účelem klasifikace dat pomocí konvoluční neuronové sítě byly vytvořeny obrázky grafů o velikosti (315 x 315), kde je v horní části zobrazen průběh vibrací v časové oblasti a v dolní části je jeho frekvenční analýza pro všechny směry i pozice (viz obrázek 8.32 a 8.33). Příprava, postup trénování a testování konvolučních sítí je shodný jako pro Dataset A.



Obrázek 8.32: Ukázka struktury obrázků trénovacích dat pro konvoluční neuronovou síť - nahoře časová oblast, dole frekvenční oblast pro všechny směry a pozice.



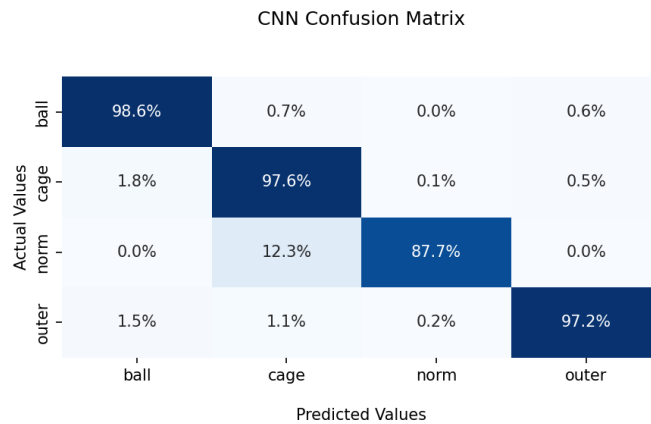
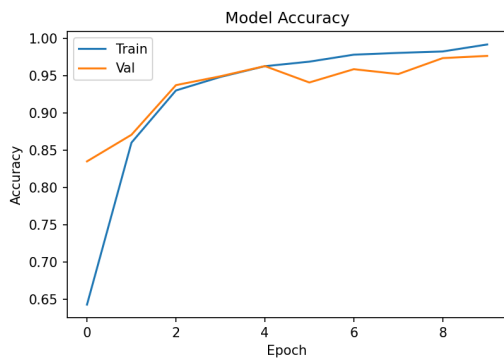
Obrázek 8.33: Ukázka obrázku trénovacích dat pro konvoluční neuronovou síť - obrázek o velikosti (315 x 315). Nahoře časová oblast, dole frekvenční oblast poškození kuličky ložiska pro všechny směry a pozice.

8.6.1. CNN - model 1

Model konvoluční sítě je shodný s modelem pro Dataset A (viz kapitola 8.3.1). Na obrázku 8.34 je zobrazen průběh přesnosti trénovacích a validačních dat pro jednotlivé epochy

8.6. KONVOLUČNÍ NEURONOVÁ SÍŤ (CNN) - DATASET B

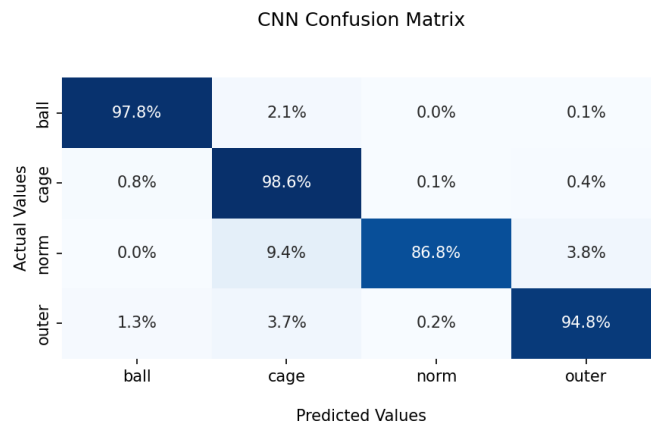
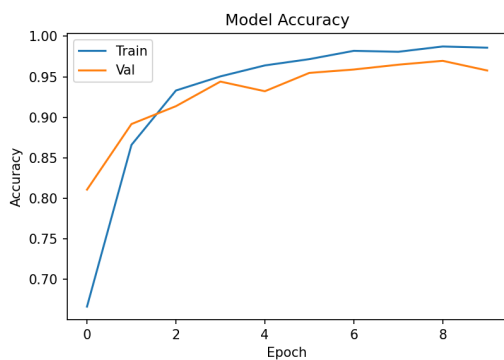
tréninku. Tento model dosahuje přesnosti **97,4 %**, jeho matice záměn je znázorněna na obrázku 8.35.



Obrázek 8.34: Graf přesnosti trénovacích a validačních dat. Obrázek 8.35: Confusion matrix s přesností 97,4 %.

8.6.2. CNN - model 2

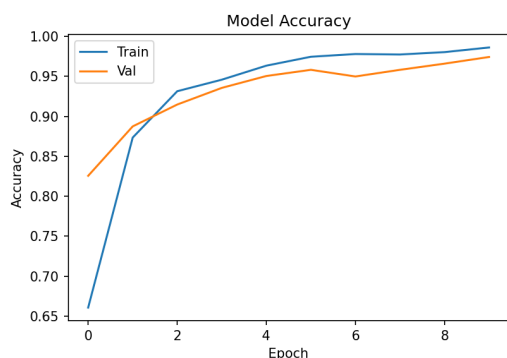
Tento model má shodnou strukturu jako model 2 pro Dataset A (viz kapitola 8.3.2). Model dosahuje přesnosti **97,0 %**, graf průběhu trénování a matice záměn jsou na obrázcích 8.36 a 8.37.



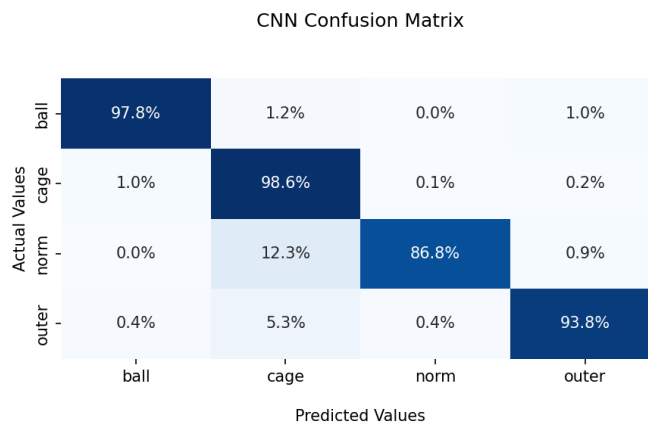
Obrázek 8.36: Graf přesnosti trénovacích a validačních dat. Obrázek 8.37: Confusion matrix s přesností 97,0 %.

8.6.3. CNN - model 3

Model má shodnou strukturu jako model 3 pro Dataset A (viz kapitola 8.3.3). Model dosahuje přesnosti **97,1 %**, graf průběhu trénování a matice záměn jsou na obrázcích 8.38 a 8.39.



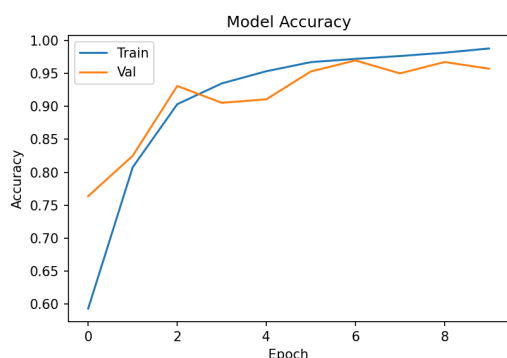
Obrázek 8.38: Graf přesnosti trénovacích a validačních dat.



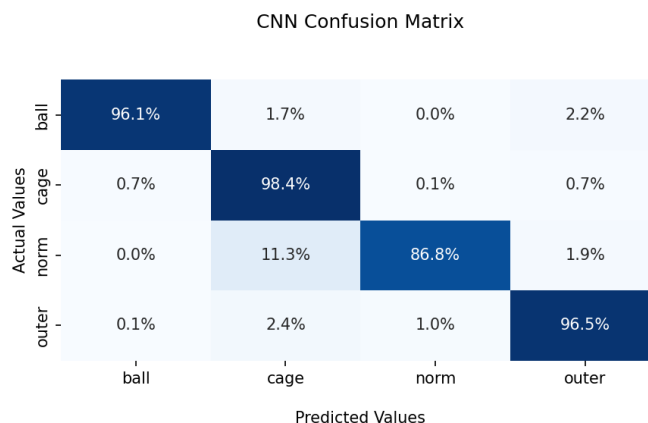
Obrázek 8.39: Confusion matrix s přesností 97,1 %.

8.6.4. CNN - model 4

Čtvrtý model má shodnou strukturu jako model 4 pro Dataset A (viz kapitola 8.3.4). Model dosahuje přesnosti **96,6 %**, graf průběhu trénování a matice záměn jsou na obrázcích 8.40 a 8.41.



Obrázek 8.40: Graf přesnosti trénovacích a validačních dat.



Obrázek 8.41: Confusion matrix s přesností 96,6 %.

8.7. SHRNU TÍ VÝSLEDKŮ

8.7. Shrnutí výsledků

V tabulce 8.3 (pro Dataset A) a tabulce 8.4 (pro Dataset B) jsou uvedeny nejlepší dosažené výsledky přesnosti klasifikačních metod otestovaných na trénovacích datech pro dané parametry.

Tabulka 8.3: Nejlepší dosažené výsledky klasifikačních metod na testovacích datech pro Dataset A s parametry

				přesnost
SVC	$C = 2000$	kernel = rbf	gamma = scale	97,9 %
KNN	n_neighbors = 1	weights = uniform	algorithm = auto	97,4 %
RFC	n_estimators = 1500	criterion = gini		98,3 %
CNN	model 2			94,1 %

Tabulka 8.4: Nejlepší dosažené výsledky klasifikačních metod na testovacích datech pro Dataset B s parametry

				přesnost
SVC	$C = 200$	kernel = rbf	gamma = auto	99,6 %
KNN	n_neighbors = 4	weights = distance	algorithm = auto	99,7 %
RFC	n_estimators = 500	criterion = gini		99,9 %
CNN	model 1			97,4 %

9. Závěr

Cílem práce je představit vybrané metody umělé inteligence, které dokáží určit poruchy ložisek z diagnostického signálu (konkrétně se v práci jedná o vibrační signál) a poskytnout algoritmy pro možné nasazení v praxi.

Na začátku práce jsou uvedeny metody technické diagnostiky. Práce se dále zaměřuje na vibrodiagnostiku, protože pokrývá velkou část závad na rotačních strojích včetně poruch ložisek a zároveň jsou datasety s měřením vibrací snadno dostupné. Dále jsou uvedeny poruchy ložisek a jejich projev v diagnostickém signálu, kde jsou uvedeny pro kuličková ložiska rovnice výpočtu frekvencí ložiskových závad (vnitřního kroužku, vnějšího kroužku, klece ložiska a valivých těles).

Následuje popis použitých datasetů. Dataset A využívá kuličkové ložisko a měří vibrační signál 4 stavů poškození ložiska (bez poškození, poškození vnějšího kroužku, poškození vnitřního kroužku a kombinovaného poškození). V případě Datasetu B se jedná o vibrační signál kuličkového ložiska pro tři směry a dvě pozice umístění ložiska. Zde byly monitorovány 4 stavy ložiska (bez poškození, poškození vnějšího kroužku, poškození klece a poškození kuličky ložiska). Pro tyto datasety jsou vypočítány jejich frekvence ložiskových závad, kdy v případě Datasetu A vycházejí frekvence závady vnitřního kroužku $f_{BPBI} \approx 4,947 \cdot f_r$ a frekvence závady vnějšího kroužku $f_{BPFO} \approx 3,053 \cdot f_r$. V případě Datasetu B jsou frekvence závady vnějšího kroužku $f_{BPBO} \approx 2,998 \cdot f_r$, frekvence závady klece ložiska $f_{FTF} \approx 0,375 \cdot f_r$ a frekvence závady valivého tělesa $f_{BSF} \approx 1,871 \cdot f_r$.

V další kapitole jsou popsány vybrané prediktory, které charakterizují data nebo stavy ložiska. Z časové oblasti to jsou: standardní směrodatná odchylka výběru s , efektivní hodnota RMS , činitel výkmitu *crestfaktor*, parametr $K(t)$ a *Kurtosis* faktor. Z frekvenční oblasti pomocí Fourierovy transformace (která převádí signál z časové do frekvenční oblasti) byla jako další prediktor zvolena frekvence s největší amplitudou. Dalšími možnými prediktory by mohly být špičková hodnota nebo z frekvenční analýzy energie spektra, popřípadě další statistické ukazatele jako rozpětí, rozptyl, variační koeficient nebo výběrová míra šikmosti.

Součástí práce je také uvedení metod umělé inteligence a použité metody klasifikace. Těmi jsou SVC (Support Vector Classification - klasifikace metodou podpurných vektorů), KNN (K-Nearest Neighbors Classifier - klasifikace k-nejbližšími sousedy), RFC (Random Forest Classifier - klasifikace náhodným lesem) a CNN (Convolutional Neural Network). U těchto metod byly testovány různé kombinace parametrů pro dosažení co nejlepších výsledků. Ke každému modelu je přidána matice záměn, která přehledně zobrazuje jeho úspěšnost klasifikace. Výhodou klasifikačních metod (SVC, KNN a RFC) je jejich snadná implementace a dosahování dobrých výsledků klasifikace, je ale potřeba data předzpracovat a charakterizovat v podobě prediktorů. V případě CNN postačí obrázky grafů průběhu vibračního signálu a graf jeho frekvenční analýzy bez nutnosti volit prediktory, jsou však náročnější na výběr vhodné architektury sítě a proces trénování (z výpočetního hlediska).

Pro Dataset A všechny metody dosahovaly úspěšnosti od 94,1 % (v případě CNN) do 97,9 % (u metody SVC). Pro Dataset B metody dosahovaly lepší úspěšnosti predikce dat od 97,4 % (v případě CNN) do 99,9 % (u metody RFC). Lze tedy říci, že přesnost klasifikace Datasetu B byla zhruba o 2-3 % úspěšnější než pro Dataset A. Také zvolený model konvolučních neuronových sítí dosahoval v obou případech nižší přesnosti predikce dat (zhruba o 2-3 %) než klasifikační metody s využitím prediktorů. Pro zlepšení predikce CNN by bylo možné zvolit větší počet epoch trénování sítě (jelikož model s každou epochou

mírně zlepšoval svou přesnost) či jinou architekturu. Výsledky také ukazují, že dataset s více směry měření vibrací (Dataset B) dosahoval lepších výsledků predikce a v případě metody RFC dosahoval téměř 100% úspěšnosti.

Z charakteru klasifikačních metod lze odtušit, že podobné principy by bylo možné aplikovat i na jiná diagnostická data. Je potřeba mít pouze dostatečné množství dat a vhodně zvolit parametry klasifikačních metod.

Na závěr byly vytvořeny skripty, ve kterých je možné si natrénovat klasifikační metody SVC, KNN, RFC i CNN na vlastních datech. Příložený jsou také skripty, které vytvářejí predikce vkládaných dat na základě již natrénovaného modelu. Ten je možné vložit vlastní nebo využít modely jež jsou natrénované na použité datasety (modely RFC a CNN nejsou příložený z důvodu jejich velikosti).

Literatura

- [1] 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.
- [2] TOMEH, Elias. Technická diagnostika: vibrační diagnostika strojů a zařízení. Liberec: Technická univerzita v Liberci. ISBN 978-80-7494-174-0.
- [3] BLATA, Jan. Metody technické diagnostiky: učební text předmětu "Technická diagnostika". Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, 2010. ISBN 978-80-248-2735-3.
- [4] BILOŠ, Jan a Alena BILOŠOVÁ. Aplikovaný mechanik jako součást týmů konstruktérů a vývojářů: část vibrační diagnostika. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, Fakulta strojní, 2012. ISBN 978-80-248-2755-1.
- [5] NĚMEČEK, Pavel a Elias TOMEH. Vibrační diagnostika základních závad strojů. Liberec: Technická univerzita v Liberci, 2010.
- [6] Analýza poškození ložisek. Coroll [online]. Hronov [cit. 2022-04-23]. Dostupné z: <https://www.coroll.cz/analyza-poskozeni-lozisek.html>
- [7] ISO 15243:2017. Rolling bearings — Damage and failures — Terms, characteristics and causes. 2. Switzerland, 2017.
- [8] LESSMEIER, Christian, James Kuria KIMOTHO, Detmar ZIMMER a Walter SEXTRO. Condition Monitoring of Bearing Damage in Electromechanical Drive Systems by Using Motor Current Signals of Electric Motors: Benchmark Data Set for Data-Driven Classification [online]. European Conference of the Prognostics and Health Management Society, Bilbao (Spain), 2016 [cit. 2022-03-30]. Dostupné z: <http://mb.uni-paderborn.de/kat/datacenter.ChairofDesignandDriveTechnology, FacultyofMechanicalEngineering,UniversityPaderborn>.
- [9] Machinery Fault Database: Signals, Multimedia, and Telecommunications Laboratory [online]. Universidade Federal do Rio de Janeiro, 2021 [cit. 2022-05-07]. Dostupné z: http://www02.smt.ufrj.br/~offshore/mfs/page_01.html
- [10] Měření vibrací ve vibrodiagnostice. Automa [online]. 2010(1), 32-36 [cit. 2022-04-19]. Dostupné z: https://www.automa.cz/cz/casopis-clanky/mereni-vibraci-ve-vibrodiagnostice-2010_01_40375_5744/
- [11] VONDRÁK, Ivo. Umělá inteligence a neuronové sítě. 3. vyd. Ostrava: VŠB - Technická univerzita Ostrava, 2009. ISBN 978-80-248-1981-5.
- [12] MACHOVÁ, Kristína. Strojové učenie: princípy a algoritmy. Košice: Elfa, 2002. ISBN 80-89066-51-8.
- [13] Klaschka, Jan ; Kotrč, Emil. Klasifikační a regresní lesy. Praha : JČMF, 2004. ISBN 80-7015-972-3. In Robust 2004. Sborník prací 13.letní školy JČMF s. 177-184. [Robust 2004. Letní škola JČMF /13./, Třešť, 07.06.2004-11.06.2004, CZ].

LITERATURA

- [14] ZUTH, Daniel. Ukázka využití strojového učení ve vibrodiagnostice. Řízení & údržba průmyslového podniku [online]. 2018, 11(4), TD14-TD16 [cit. 2022-03-08]. ISSN 1803-4535. Dostupné z: <https://www.vseoprmyslu.cz/knihovna/rizeni-a-udrzba/rizeni-a-udrzba-prumysloveho-podniku-4-2018.html>
- [15] Support Vector Machines. Scikit-learn: Machine Learning in Python [online]. c2007-2022 [cit. 2022-05-11]. Dostupné z: <https://scikit-learn.org/stable/modules/svm.html#svm-mathematical-formulation>
- [16] Sklearn.svm.SVC. Scikit-learn: Machine Learning in Python [online]. c2007-2022 [cit. 2022-05-11]. Dostupné z: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC>
- [17] Nearest Neighbors. Scikit-learn: Machine Learning in Python [online]. c2007-2022 [cit. 2022-05-10]. Dostupné z: <https://scikit-learn.org/stable/modules/neighbors.html#classification>
- [18] Sklearn.ensemble.RandomForestClassifier. Scikit-learn: Machine Learning in Python [online]. c2007-2022 [cit. 2022-05-11]. Dostupné z: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
- [19] Ensemble methods. Scikit-learn: Machine Learning in Python [online]. c2007-2022 [cit. 2022-05-11]. Dostupné z: <https://scikit-learn.org/stable/modules/ensemble.html#forest>
- [20] Sklearn.model_selection.GridSearchCV. Scikit-learn [online]. c2007-2022 [cit. 2022-05-13]. Dostupné z: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
- [21] Cross-validation: evaluating estimator performance. Scikit-learn [online]. c2007-2022 [cit. 2022-05-14]. Dostupné z: https://scikit-learn.org/stable/modules/cross_validation.html#k-fold
- [22] Neuronové sítě - konvoluční sítě a zpracování obrazu. Martin Pilát [online]. [cit. 2022-05-16]. Dostupné z: <https://martinpilat.com/cs/prirodou-inspirovane-algoritmy/neuronove-site-konvolucni-site-zpracovani-obrazu>
- [23] Introduction to Convolutional Neural Networks (CNN). Analytics Vidhya [online]. c2013-2022 [cit. 2022-05-15]. Dostupné z: https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/?fbclid=IwAR27eMxDW17ypW1a2eK107ABgbYNp6sBbIHDt17Nq8jf40kSRpeQlxGbv0#h2_2
- [24] Convolutional Neural Networks. IBM [online]. [cit. 2022-05-16]. Dostupné z: <https://www.ibm.com/cloud/learn/convolutional-neural-networks?fbclid=IwAR3SPdVRV5aYjXa1-FmQNUG30ET6HK5HaD1A3mzyS5yGCxYSaXYplJK1X50>
- [25] Scikit-learn: Machine Learning in Python [online]. c2007-2022 [cit. 2022-05-18]. Dostupné z: <https://scikit-learn.org/stable/index.html>

- [26] TensorFlow: An end-to-end open source machine learning platform [online]. [cit. 2022-05-11]. Dostupné z: <https://www.tensorflow.org/>
- [27] SciPy: Fundamental algorithms for scientific computing in Python [online]. c2022 [cit. 2022-05-18]. Dostupné z: <https://scipy.org/>
- [28] NumPy: The fundamental package for scientific computing with Python [online]. c2022 [cit. 2022-05-18]. Dostupné z: <https://numpy.org/>
- [29] Matplotlib: Visualization with Python [online]. c2021 [cit. 2022-05-18]. Dostupné z: <https://matplotlib.org/>
- [30] Sklearn.preprocessing.StandardScaler. Scikit-learn [online]. c2007-2022 [cit. 2022-05-13]. Dostupné z: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>
- [31] Sklearn.model_selection.train_test_split. Scikit-learn [online]. c2007-2022 [cit. 2022-05-14]. Dostupné z: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html#sklearn.model_selection.train_test_split
- [32] Sklearn.neighbors.KNeighborsClassifier. Scikit-learn: Machine Learning in Python [online]. c2007-2022 [cit. 2022-05-11]. Dostupné z: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html?highlight=k%20neighbors%20classifier#sklearn.neighbors.KNeighborsClassifier>
- [33] Tf.keras.optimizers.Adam. TensorFlow [online]. [cit. 2022-05-16]. Dostupné z: https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam
- [34] Tf.keras.losses.CategoricalCrossentropy. TensorFlow [online]. [cit. 2022-05-16]. Dostupné z: https://www.tensorflow.org/api_docs/python/tf/keras/losses/CategoricalCrossentropy
- [35] Joblib: running Python functions as pipeline jobs [online]. c2008-2021 [cit. 2022-05-19]. Dostupné z: <https://joblib.readthedocs.io/en/latest/>

10. Seznam použitých zkratek a symbolů

FS	frekvenční spektrum
FLZ	frekvence ložiskových závad
Dataset A	Dataset Paderborn univerzity
Dataset B	Dataset Universidade Federal do Rio de Janeiro
f_r	rotorová frekvence
RPM	
OK	ložisko bez poškození
OR	poškození vnějšího kroužku
IR	poškození vnitřního kroužku
KB	kombinované poškození
normal	ložisko bez poškození
cage	poškození klece ložiska
outer	poškození vnějšího kroužku
ball	poškození kuličky
s	standardní směrodatná odchylka výběru
RMS	Root Mean Square - efektivní hodnota
CF	crest faktor
FT	Fourierova transformace
DFT	Diskrétní Fourierova transformace
FFT	Fast Fourier transform
RFFT	Real Fast Fourier transform
SVC	Support Vector Classifier - klasifikace metodou podpůrných vektorů
KNN	K-Nearest Neighbors Classifier - klasifikace k-nejbližšími sousedy
RFC	Random Forest Classifier - klasifikace náhodným lesem
SVM	Support Vector Machines - metody podpůrných vektorů
RBF	Radial Basis Function - funkce radiální báze
CNN	Convolutional Neural Network - konvoluční neuronová síť
ReLU	Rectified linear unit

11. Seznam příloh

A. Příloha - seznam a popis skriptů

B. Příloha

A. Příloha - seznam a popis skriptů

Seznam příložených skriptů

CNN.py
CNN_1.py
CNN_2.py
classifiers.py
classification1.py
classification2.py

Seznam příložených modelů

SVC_model1.joblib
KNN_model1.joblib
SVC_model2.joblib
KNN_model2.joblib

Potřebné knihovny: `scikit-learn`, `NumPy`, `Keras`, `TensorFlow`, `csv`, `cv2`, `pathlib`, `(joblib)`

`classifiers.py`

Skript pro natrénování klasifikátorů.

Je potřeba zadat:

- cestu k souboru cílových hodnot
- cestu k souboru s prediktory

Cílové hodnoty musí být v souboru `.csv`, kde na každém řádku je cílová hodnota daného měření.

Prediktory musí být v souboru `.csv`, kde na každém řádku jsou jednotlivé prediktory pro jedno měření odděleny mezerou (nebo je potřeba nastavit jiný oddělující znak `delimiter=' '`).

Dále je možné nastavit procentuální rozdělení trénovacích a testovacích dat. V zakomentované části je možné otestovat zadané parametry pro určenou klasifikační metodu.

Výsledkem je výpis skóre modelu a confusion matrix pro metody SVC, KNN a RFC.

`classification1.py`

Pro klasifikaci v tomto skriptu je potřeba mít prediktory v souboru `.csv` dle obrázku 8.8, kde na každém řádku jsou jednotlivé prediktory pro jedno měření odděleny mezerou.

Dále je potřeba nahrát modely klasifikátorů ('SVC_model1.joblib', 'KNN_model1.joblib') za pomoci knihovny `joblib` [35] (model 'RFC_model1.joblib' není z důvodu jeho velikosti přiložen).

Po zadání cesty k souboru prediktorů (`predictors file path`) se:

- do proměnné `svc_predictions` uloží predikce modelu SVC
- do proměnné `knn_predictions` uloží predikce modelu KNN

classification2.py

Pro klasifikaci v tomto skriptu je potřeba mít prediktory dle obrázku 8.28 v souboru `.csv`, kde na každém řádku jsou jednotlivé prediktory pro jedno měření odděleny mezerou.

Dále je potřeba nahrát modely klasifikátorů ('`SVC_model2.joblib`', '`KNN_model2.joblib`') za pomoci knihovny `joblib` [35] (model '`RF_model2.joblib`' není z důvodu jeho velikosti přiložen).

Po zadání cesty k souboru prediktorů (predictors file path) se:

- do proměnné `svc_predictions` uloží predikce modelu SVC
- do proměnné `knn_predictions` uloží predikce modelu KNN

CNN.py

Skript pro natrénování sítě podle vlastních dat.

Je potřeba zadat:

- počet klasifikačních tříd (`num_classes`)
- cestu k obrázkům (`input_dir`)
- příponu obrázků (`files`)
- cestu k souboru cílových hodnot (`file_y`)

Cílové hodnoty musí být v souboru `.csv`, kde na každém řádku je číslo klasifikační třídy ($0 - n$), kdy n je počet klasifikačních tříd. Dále je možné nastavit procentuální rozdělení trénovacích a testovacích dat, nastavit si architekturu sítě, kompilaci nebo trénování modelu.

Provede se zhodnocení modelu na testovacích datech. Do proměnné `predictions` se uloží predikce modelu v podobě hodnot $0 - n$ a do proměnné `predictions_percent` se uloží pravděpodobnost predikce (v podobě hodnoty na dané pozici $0 - n$). Dále je možné model uložit (tato možnost je defaultně zakomentovaná).

CNN_1.py

Pro klasifikaci pomocí konvoluční neuronové sítě skriptu `CNN_1.py` je potřeba mít klasifikovaná data v podobě obrázku velikosti (240x240), které budou mít nahoře graf vibrací v časové oblasti, dole frekvenční oblast (viz obrázek 8.12). Dále je potřeba nahrát model natrénované sítě (model není z důvodu jeho velikosti přiložen) za pomoci knihovny `keras`. Po zadání cesty k obrázkům (image path) a jejich příponě (image type) se:

- do proměnné `predictions` uloží predikce modelu v podobě hodnot 0-3 (0 je ložisko bez poškození '`OK`', 1 je poškození vnějšího kroužku '`OR`', 2 je kombinované poškození '`KB`' a 3 je poškození vnitřního kroužku '`IR`')
- do proměnné `predictions_percent` se uloží pravděpodobnost predikce (v podobě hodnoty na dané pozici 0-3)
- do proměnné `predictions_word` se uloží názvy stavu ložiska

CNN_2.py

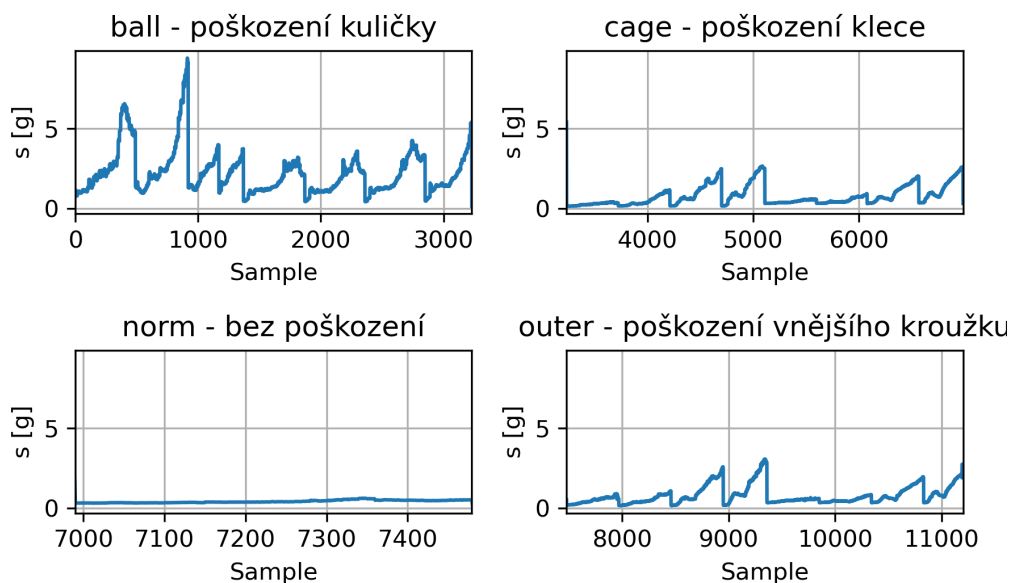
Pro klasifikaci pomocí konvoluční neuronové sítě skriptu `CNN_2.py` je potřeba mít klasifikovaná data v podobě obrázku velikosti (315x315), které budou mít stejnou strukturu jako je na obrázku 8.32. Dále je potřeba nahrát model natrénované sítě (model není z důvodu jeho velikosti přiložen) za pomoci knihovny `keras`.

Po zadání cesty k obrázkům (`image path`) a jejich příponě (`image type`) se:

- do proměnné `predictions` uloží predikce modelu v podobě hodnot 0-3 (0 je poškození kuličky *'ball'*, 1 je poškození klece *'cage'*, 2 je bez poškození ložiska *'norm'* a 3 je poškození vnějšího kroužku *'outer'*)
- do proměnné `predictions_percent` se uloží pravděpodobnost predikce (v podobě hodnoty na dané pozici 0-3)
- do proměnné `predictions_word` se uloží názvy stavu ložiska

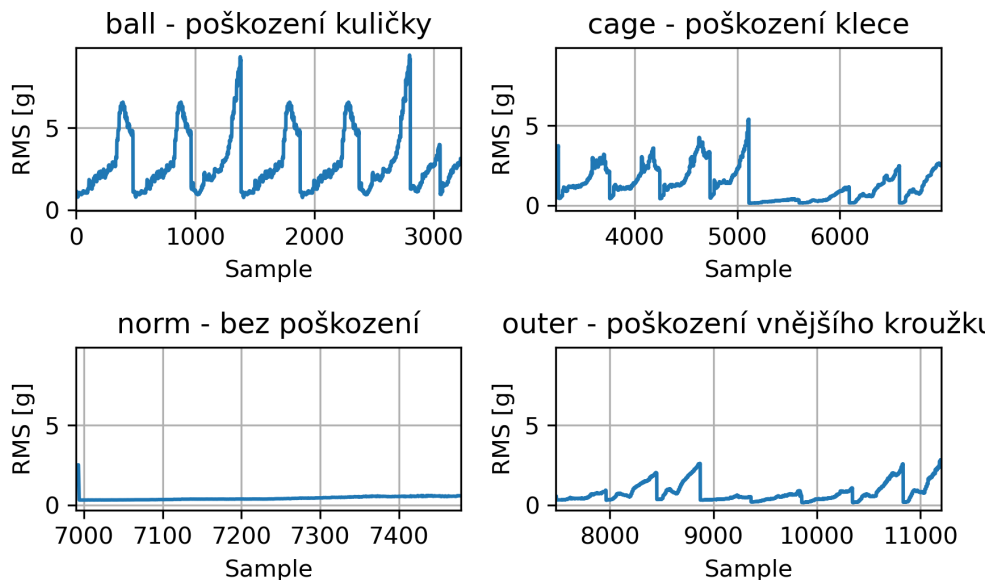
B. Příloha

Standardní směrodatná odchylka výběru



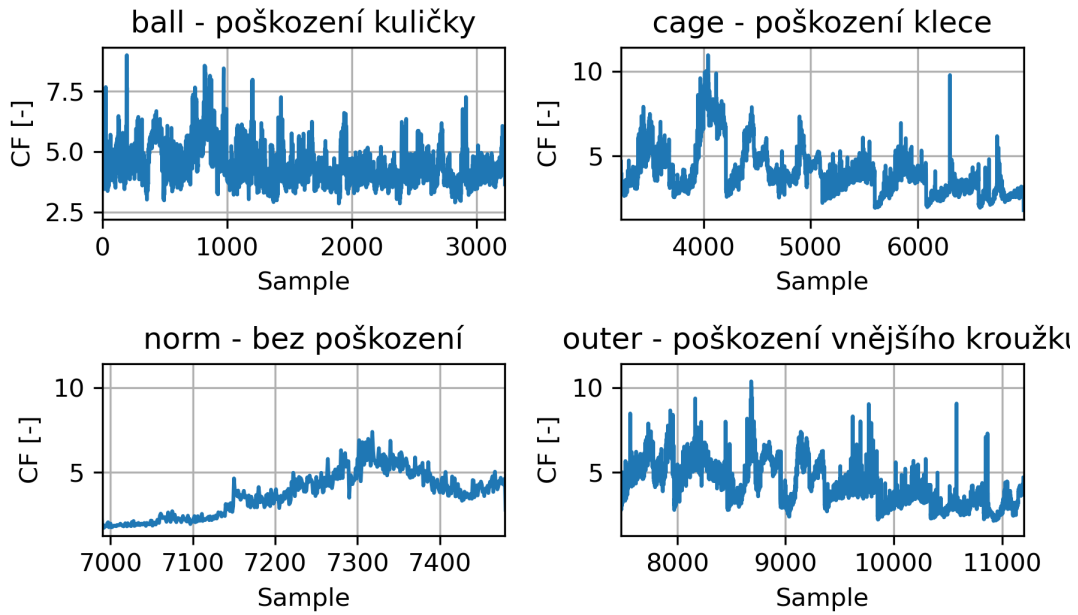
Obrázek B.1: Standardní směrodatná odchylka výběru s pro axiální směr v pozici underhang datasetu B.

RMS - efektivní hodnota



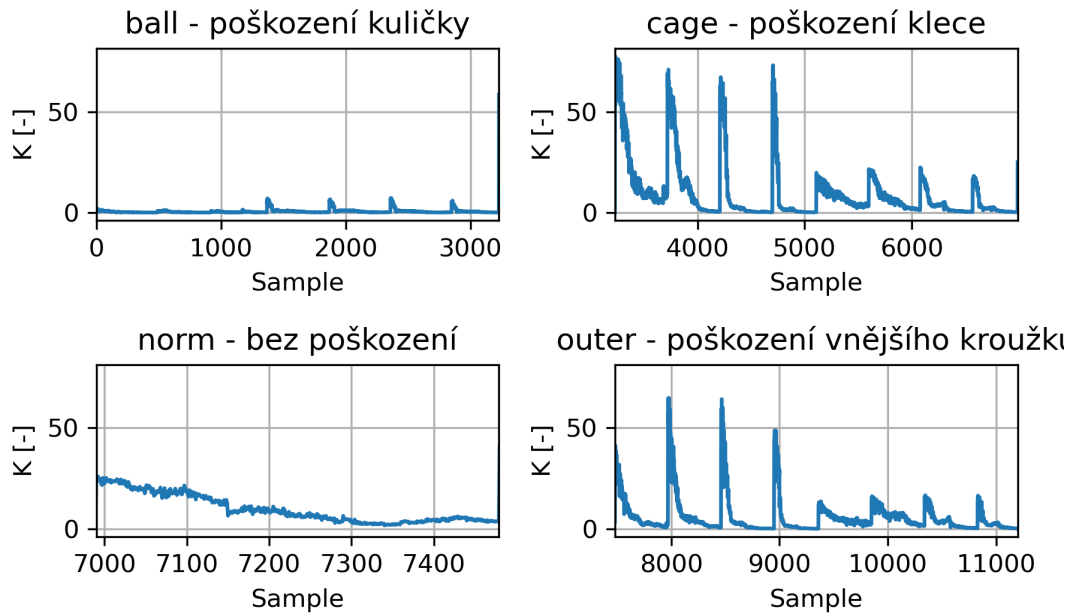
Obrázek B.2: Efektivní hodnota RMS pro axiální směr v pozici underhang datasetu B.

Crest faktor - činitel výkmitu



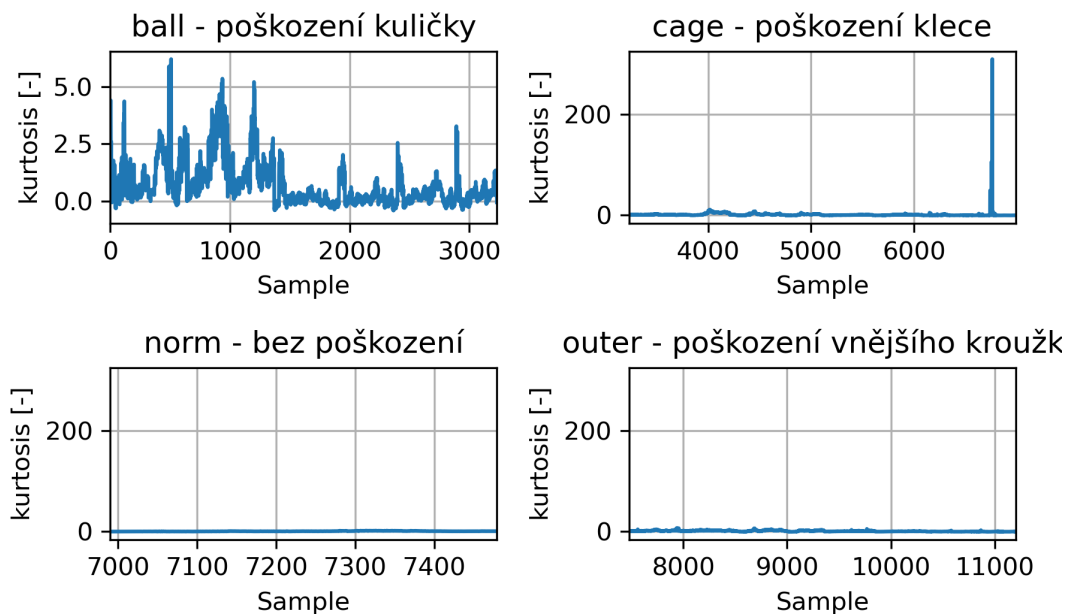
Obrázek B.3: Činitel výkmitu *crestfaktor* pro axiální směr v pozici underhang datasetu B.

K metoda



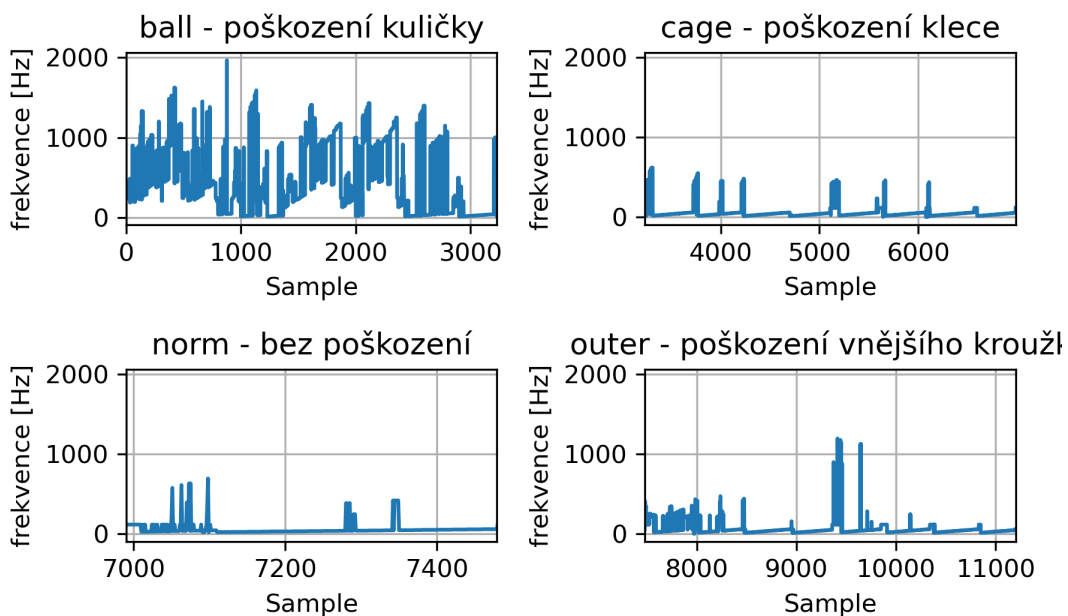
Obrázek B.4: Parametr $K(t)$ pro axiální směr v pozici underhang datasetu B.

Kurtosis faktor



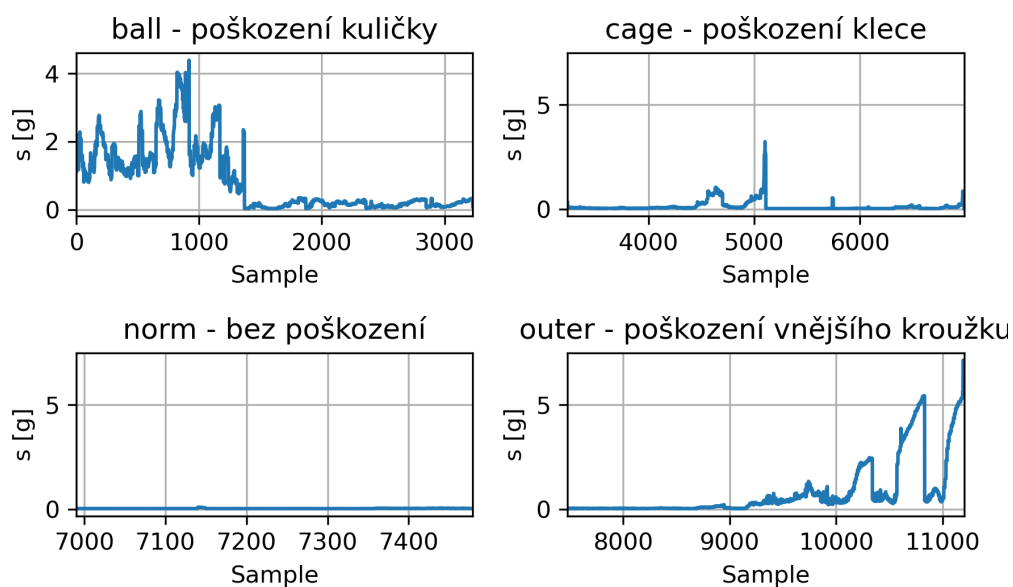
Obrázek B.5: Kurtosis faktor pro axiální směr v pozici underhang datasetu B.

Frekvence maximální hodnoty z FT



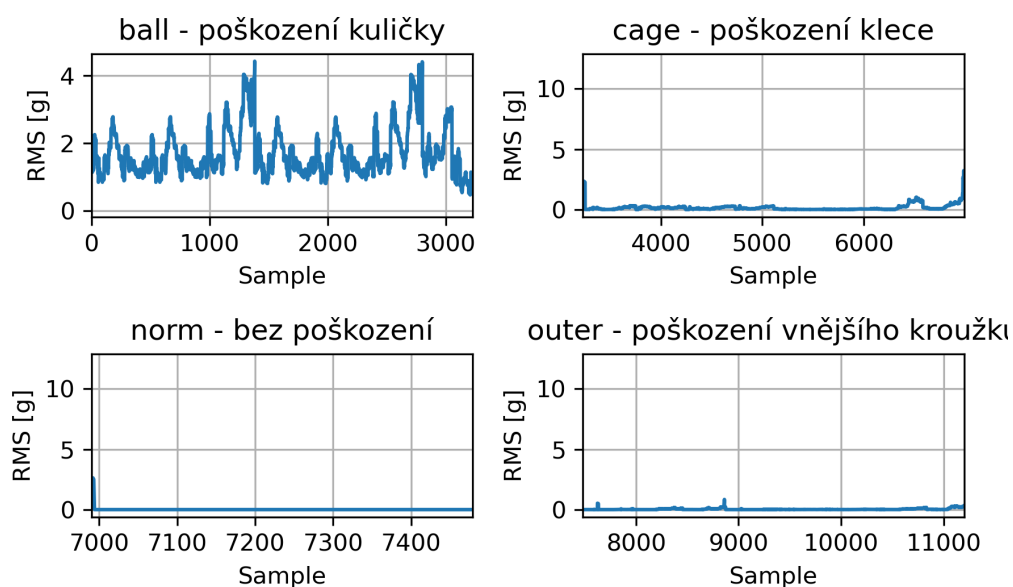
Obrázek B.6: Frekvence s největší amplitudou pro axiální směr v pozici underhang datasetu B.

Standardní směrodatná odchylka výběru



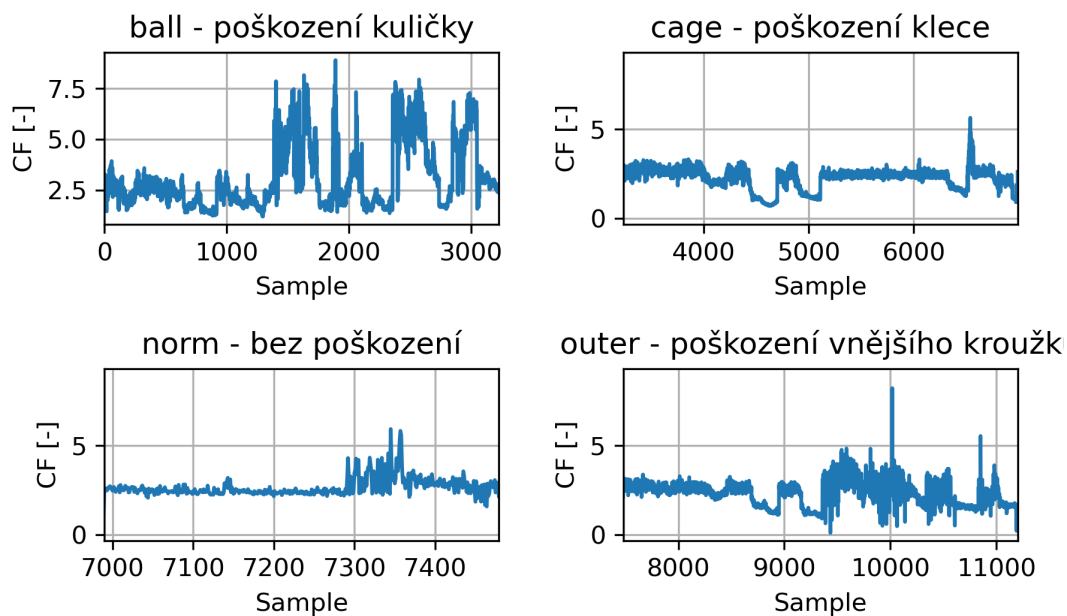
Obrázek B.7: Standardní směrodatná odchylka výběru s pro axiální směr v pozici overhang datasetu B.

RMS - efektivní hodnota

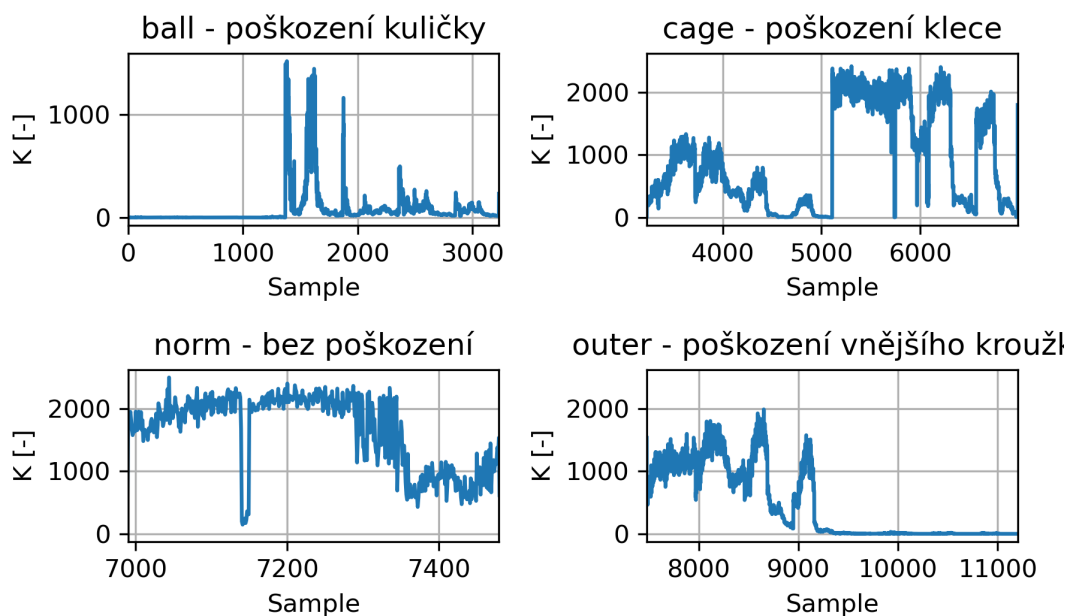


Obrázek B.8: Efektivní hodnota RMS pro axiální směr v pozici overhang datasetu B.

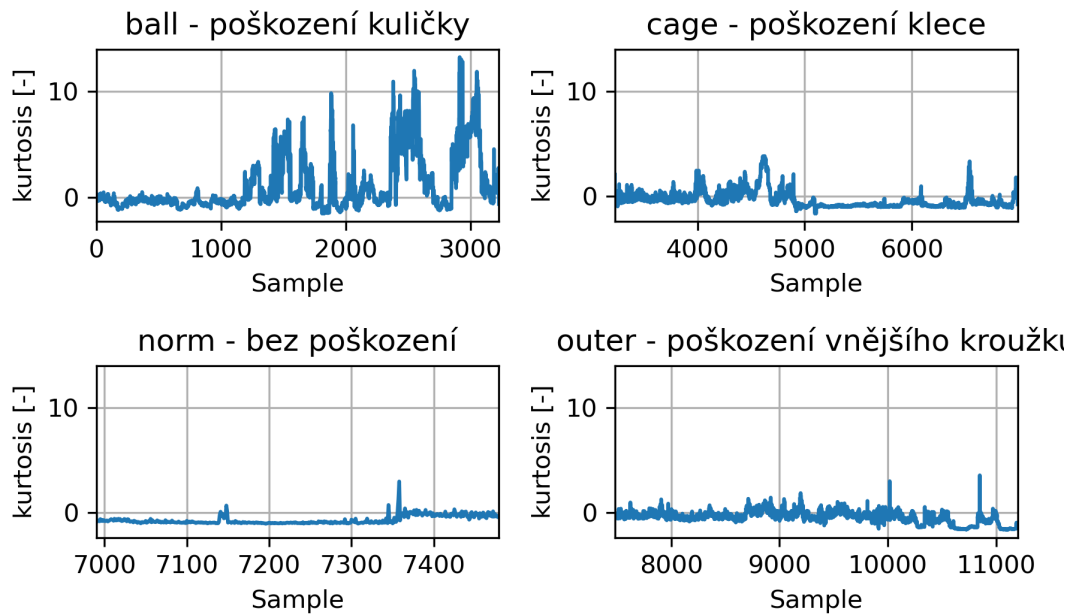
Crest faktor - činitel výkmitu

Obrázek B.9: Činitel výkmitu *crestfaktor* pro axiální směr v pozici overhang datasetu B.

K metoda

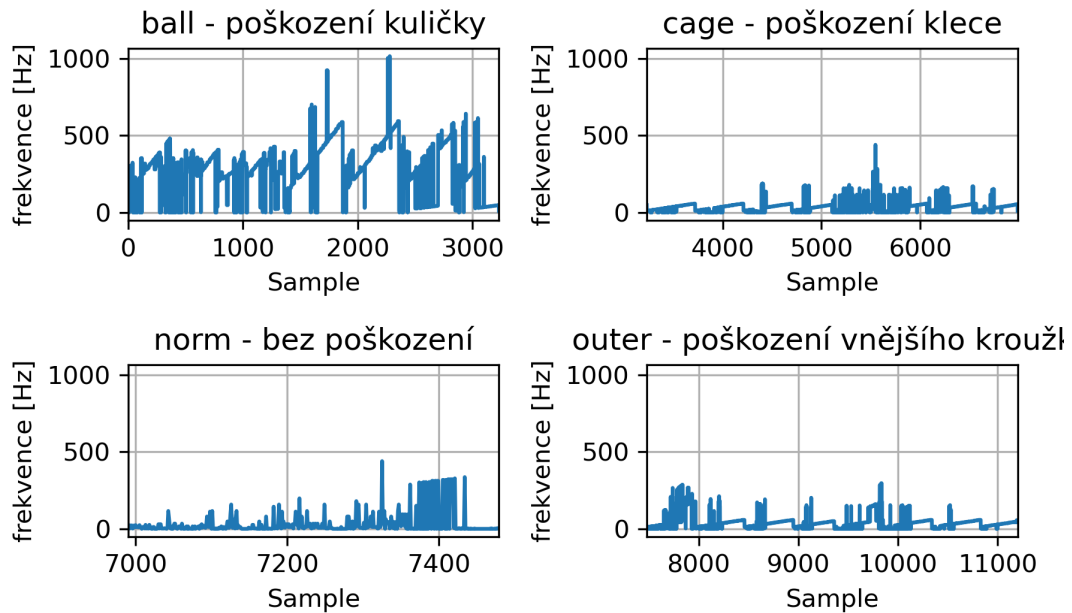
Obrázek B.10: Parametr $K(t)$ pro axiální směr v pozici overhang datasetu B.

Kurtosis faktor



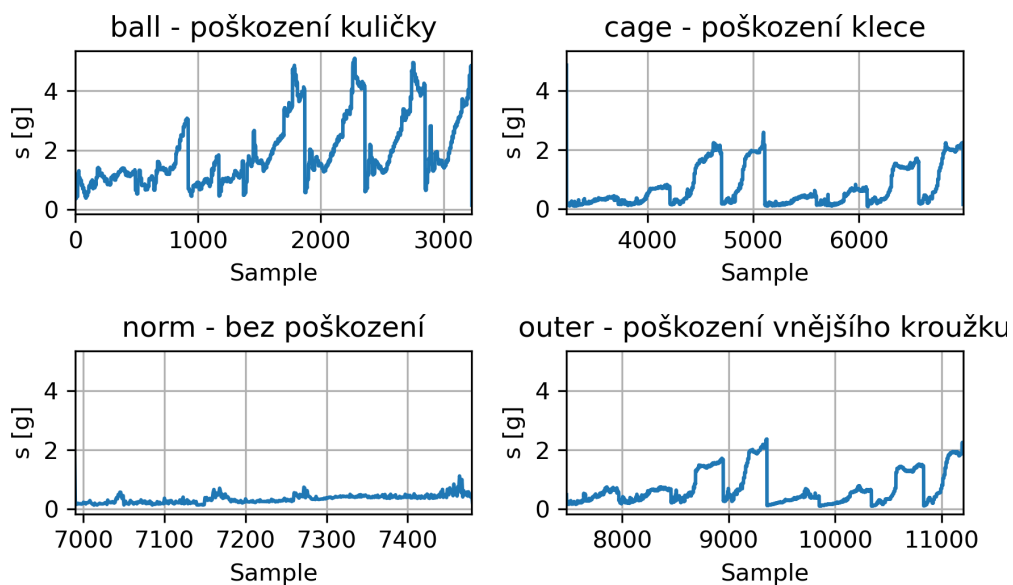
Obrázek B.11: Kurtosis faktor pro axiální směr v pozici overhang datasetu B.

Frekvence maximální hodnoty z FT



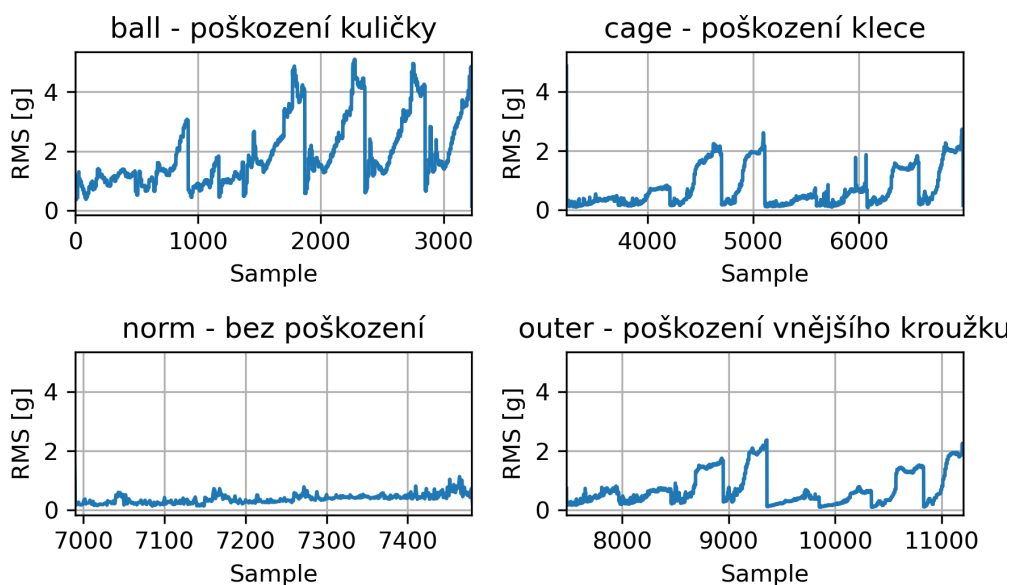
Obrázek B.12: Frekvence s největší amplitudou pro axiální směr v pozici overhang datasetu B.

Standardní směrodatná odchylka výběru



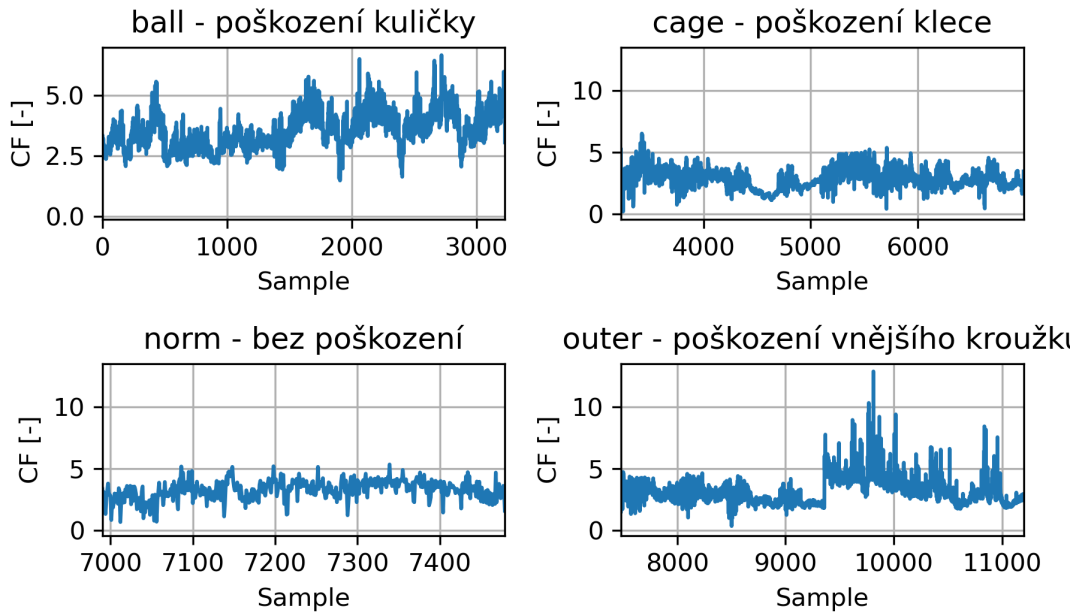
Obrázek B.13: Standardní směrodatná odchylka výběru s pro radiální směr v pozici overhang datasetu B.

RMS - efektivní hodnota



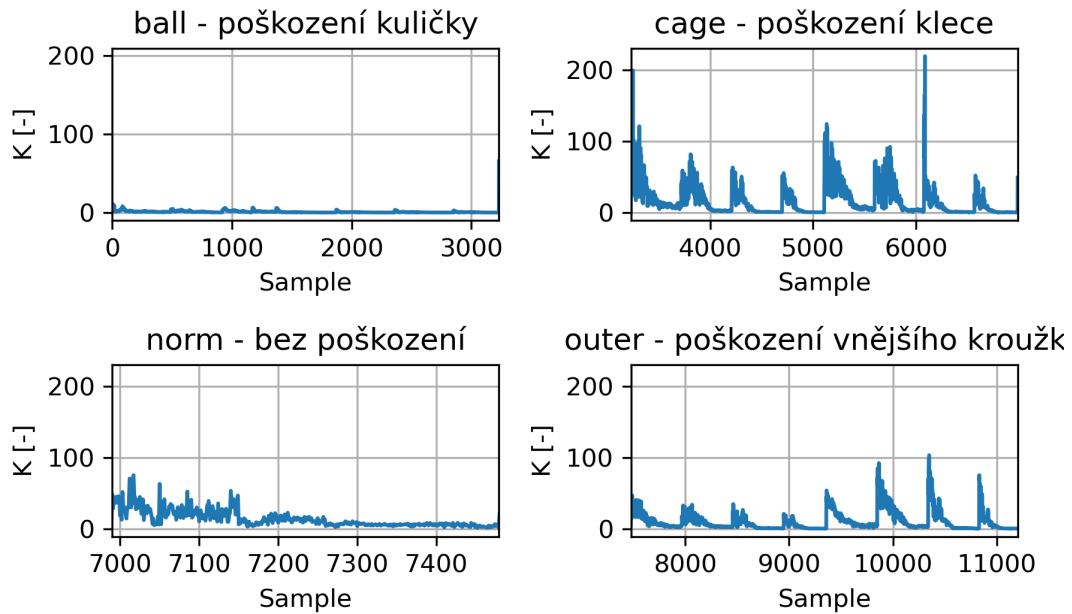
Obrázek B.14: Efektivní hodnota RMS pro radiální směr v pozici overhang datasetu B.

Crest faktor - činitel výkmitu

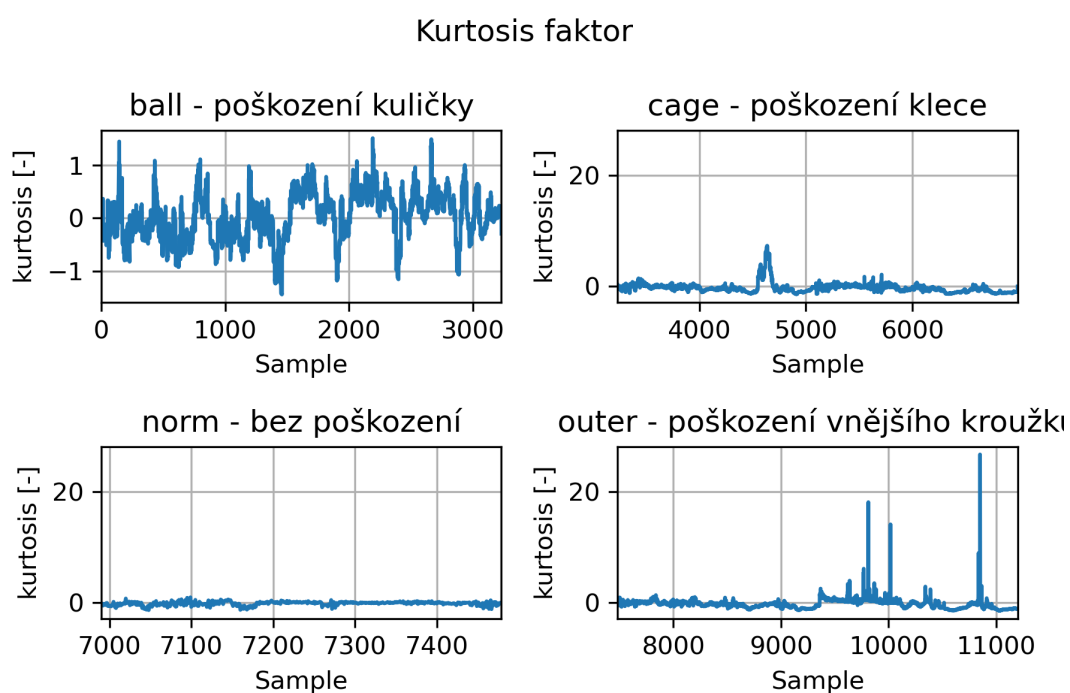


Obrázek B.15: Činitel výkmitu *crestfaktor* pro radiální směr v pozici overhang datasetu B.

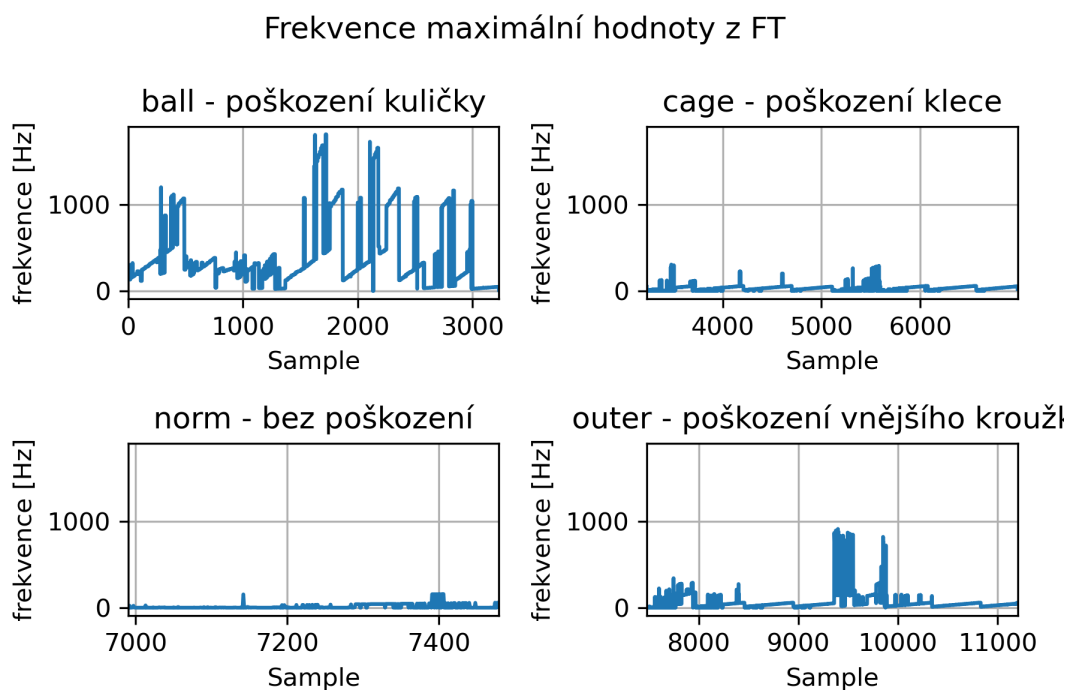
K metoda



Obrázek B.16: Parametr $K(t)$ pro radiální směr v pozici overhang datasetu B.

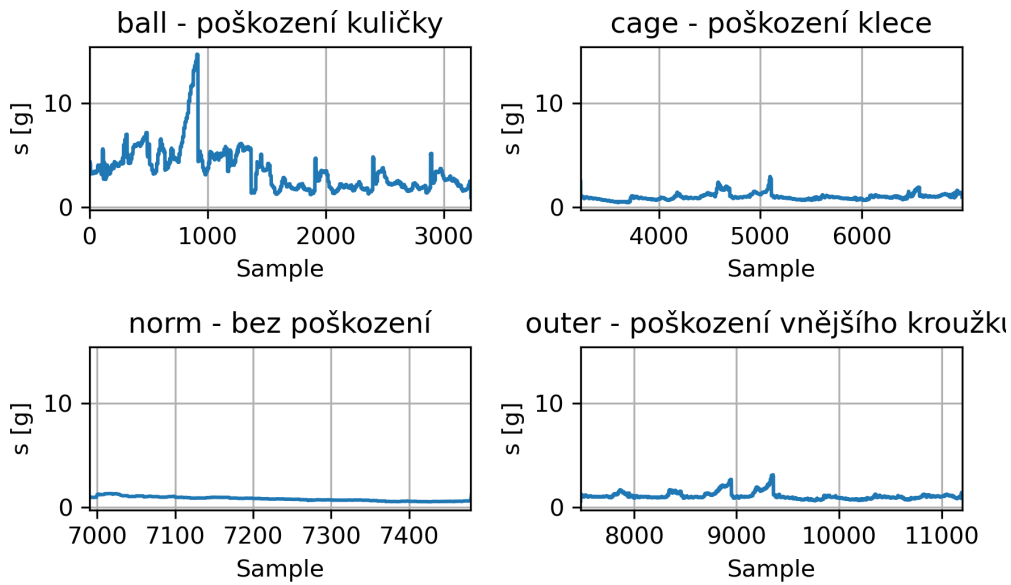


Obrázek B.17: Kurtosis faktor pro radiální směr v pozici overhang datasetu B.



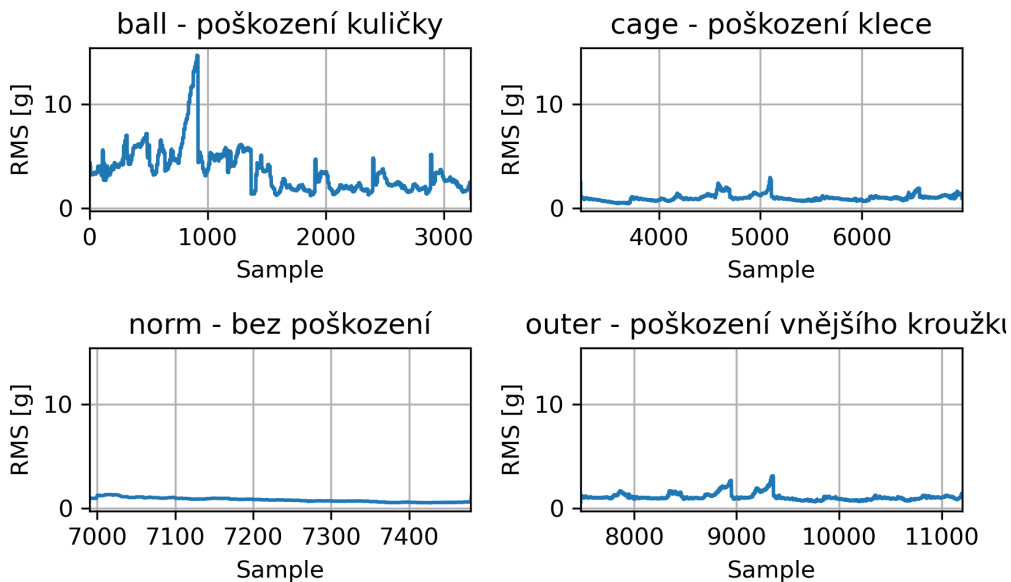
Obrázek B.18: Frekvence s největší amplitudou pro radiální směr v pozici overhang datasetu B.

Standardní směrodatná odchylka výběru



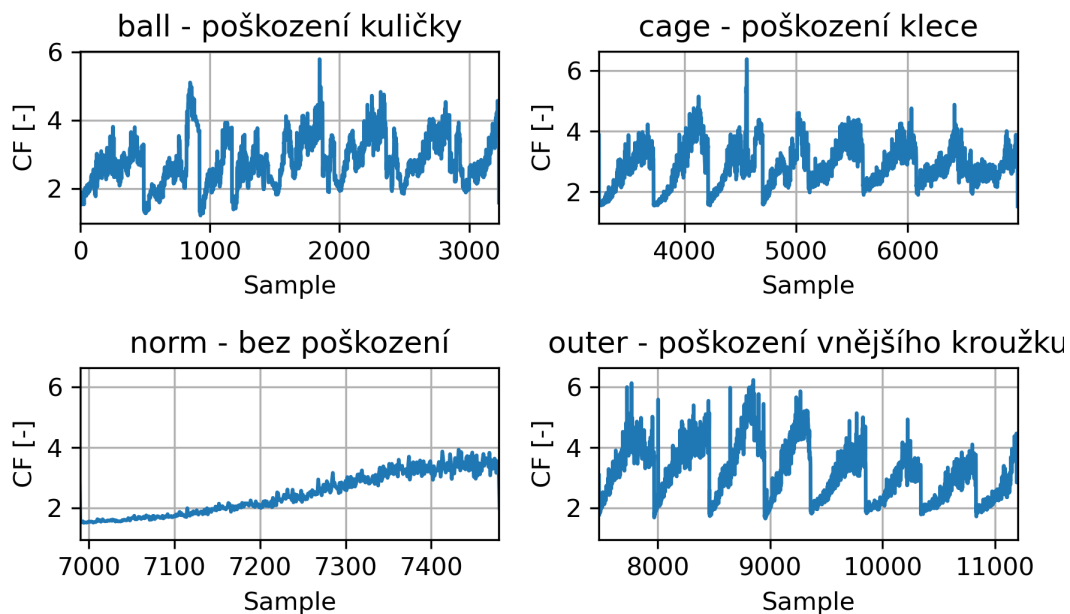
Obrázek B.19: Standardní směrodatná odchylka výběru s pro tangenciální směr v pozici underhang datasetu B.

RMS - efektivní hodnota



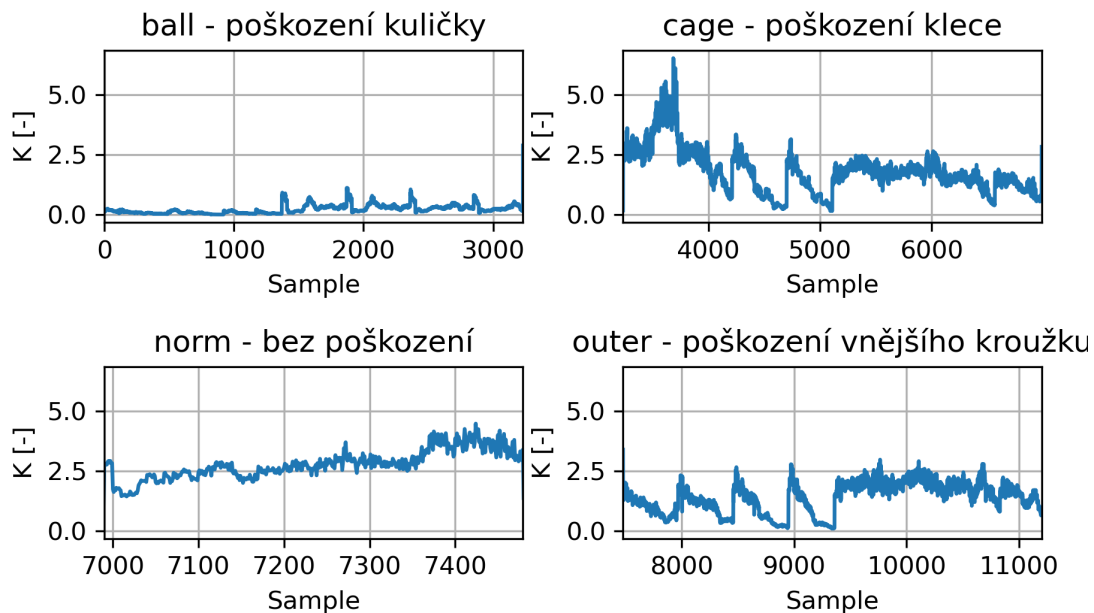
Obrázek B.20: Efektivní hodnota RMS pro tangenciální směr v pozici underhang datasetu B.

Crest faktor - činitel výkmitu



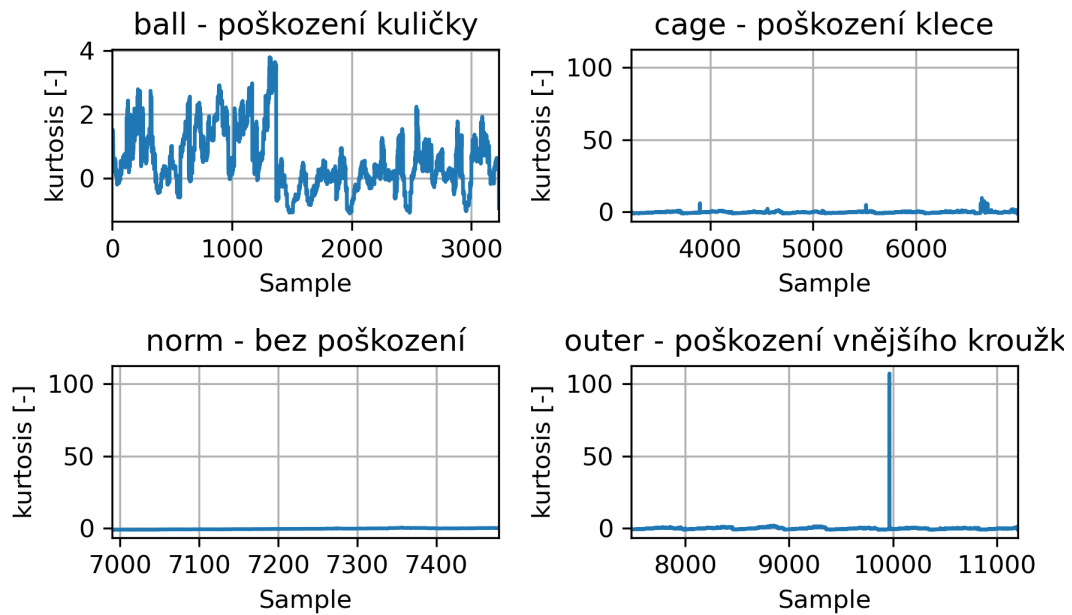
Obrázek B.21: Činitel výkmitu *crestfaktor* pro tangenciální směr v pozici underhang datasetu B.

K metoda



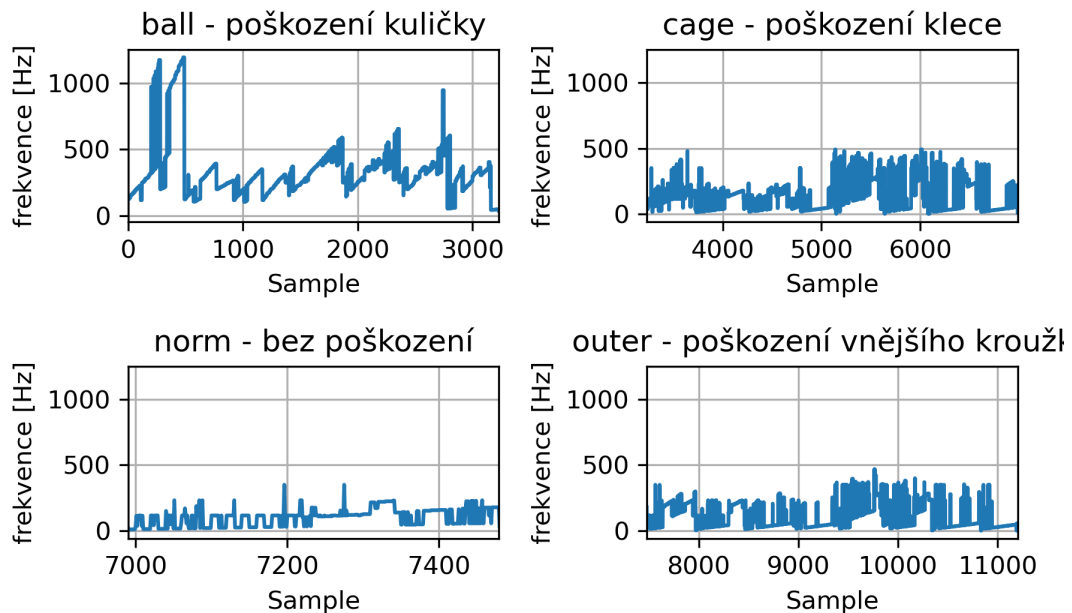
Obrázek B.22: Parametr $K(t)$ pro tangenciální směr v pozici underhang datasetu B.

Kurtosis faktor



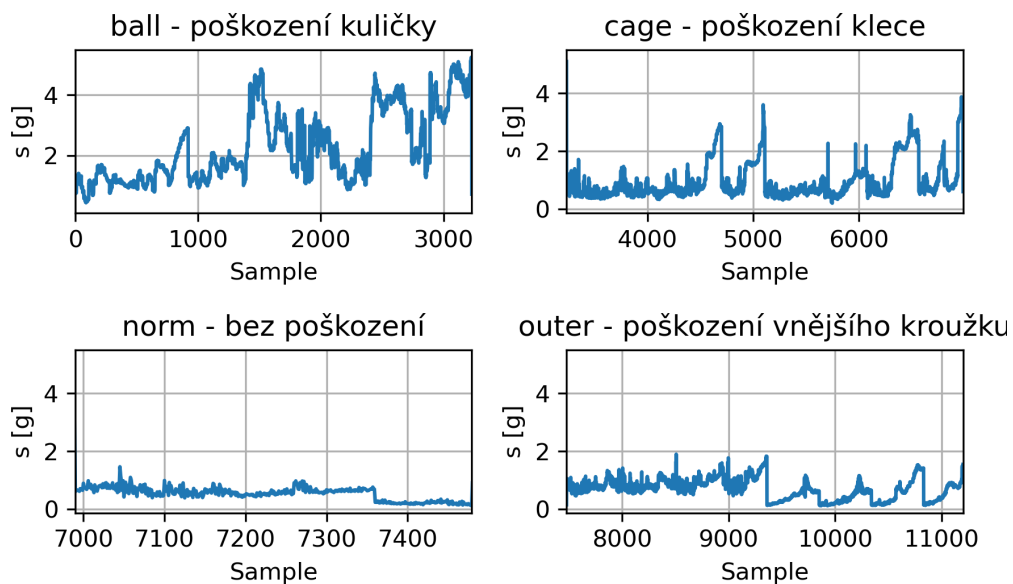
Obrázek B.23: Kurtosis faktor pro tangenciální směr v pozici underhang datasetu B.

Frekvence maximální hodnoty z FT



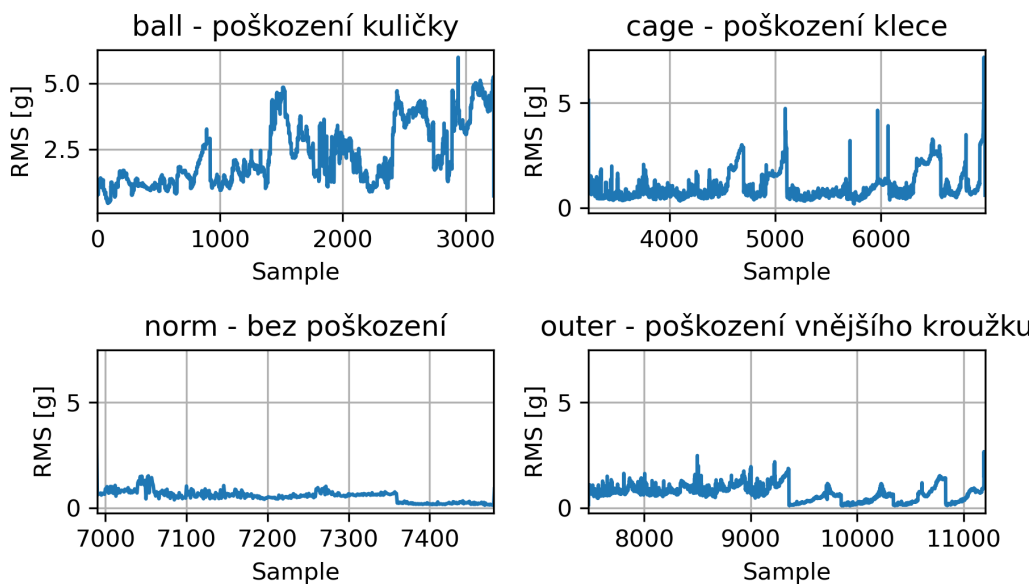
Obrázek B.24: Frekvence s největší amplitudou pro tangenciální směr v pozici underhang datasetu B.

Standardní směrodatná odchylka výběru



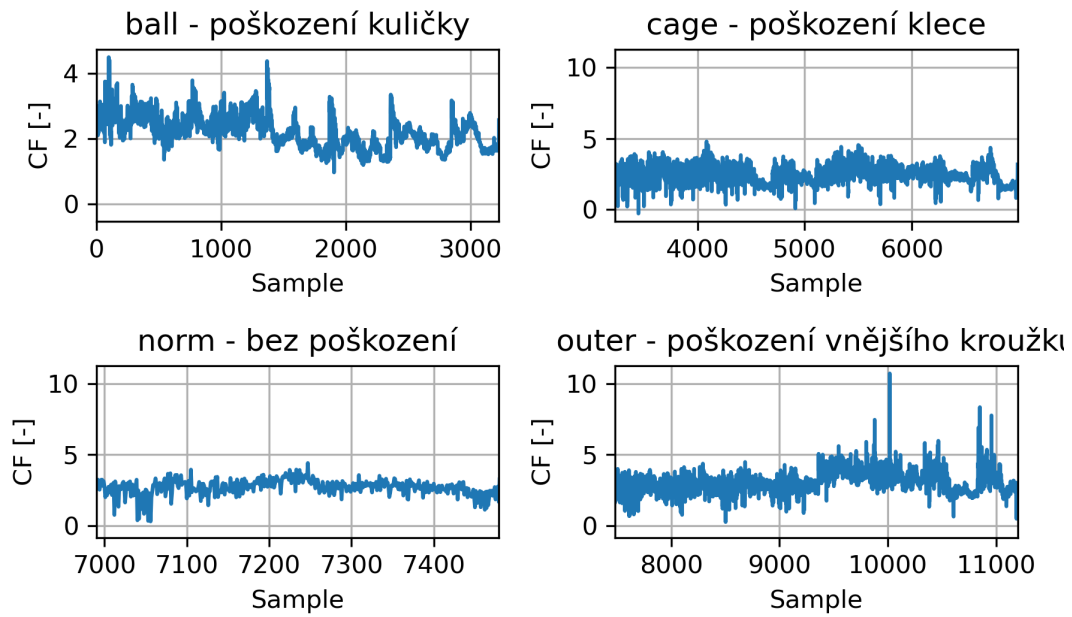
Obrázek B.25: Standardní směrodatná odchylka výběru s pro tangenciální směr v pozici overhang datasetu B.

RMS - efektivní hodnota



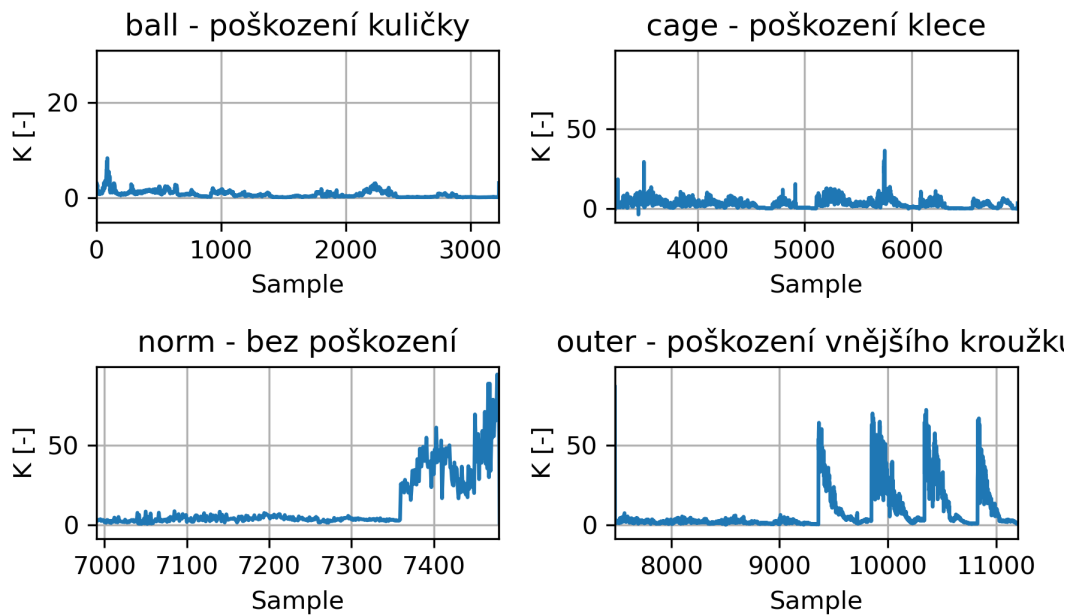
Obrázek B.26: Efektivní hodnota RMS pro tangenciální směr v pozici overhang datasetu B.

Crest faktor - činitel výkmitu



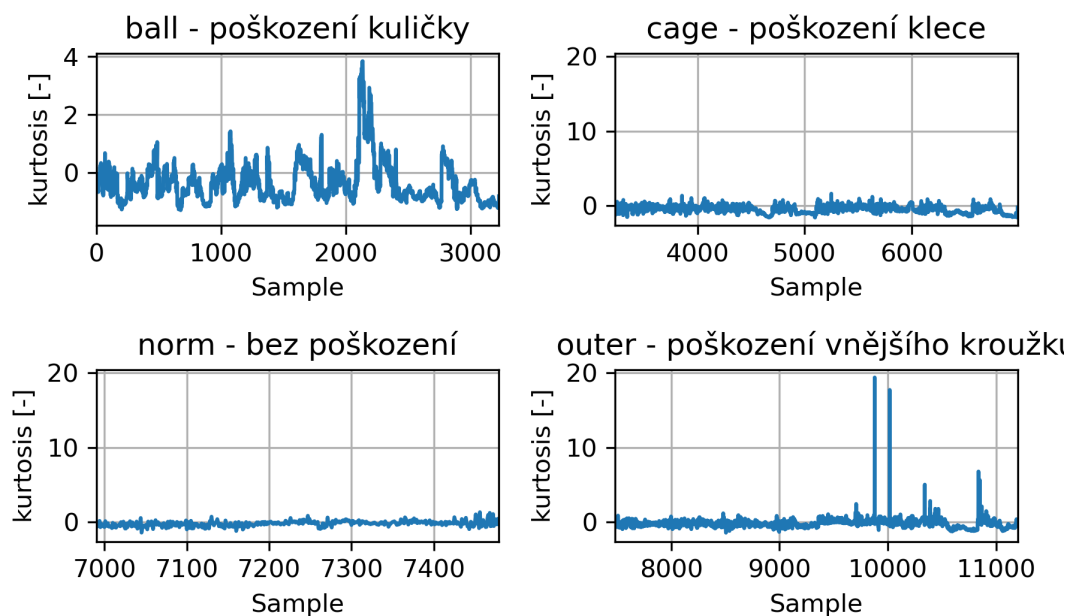
Obrázek B.27: Činitel výkmitu *crestfaktor* pro tangenciální směr v pozici overhang datasetu B.

K metoda



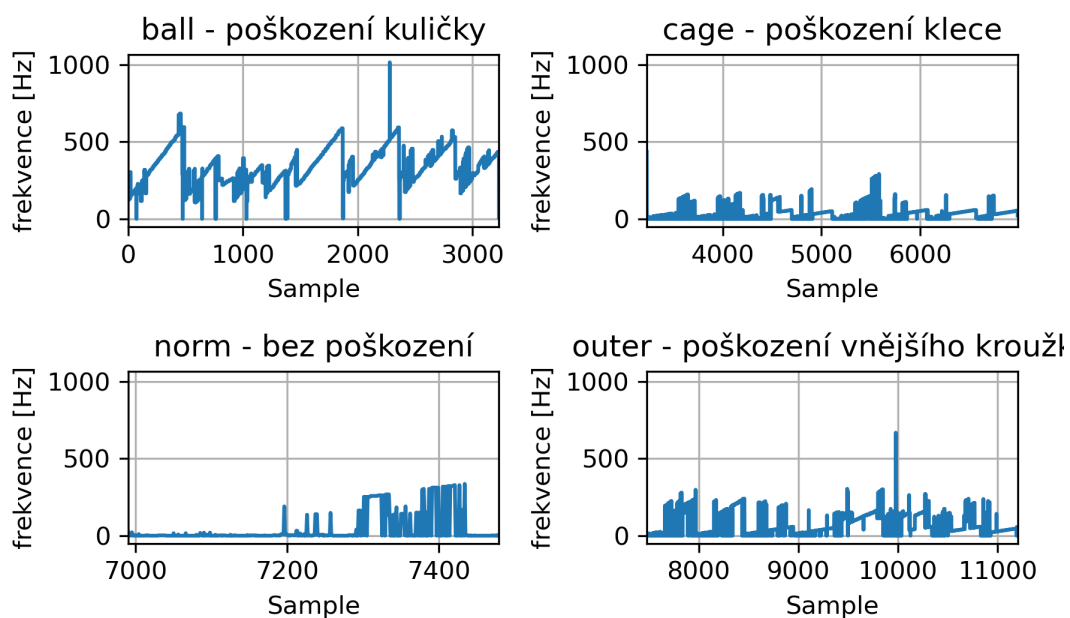
Obrázek B.28: Parametr $K(t)$ pro tangenciální směr v pozici overhang datasetu B.

Kurtosis faktor



Obrázek B.29: Kurtosis faktor pro tangenciální směr v pozici overhang datasetu B.

Frekvence maximální hodnoty z FT



Obrázek B.30: Frekvence s největší amplitudou pro tangenciální směr v pozici overhang datasetu B.