

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STROJNÍ

ÚSTAV MECHANIKY, BIOMECHANIKY A MECHATRONIKY



System se strojovým učením pro detekci a analýzu anomálií
dynamického systému

Machine Learning System for Anomaly Detection and Analysis of
Dynamical System

Diplomová práce

Vypracoval: Bc. Vladislav Erdman

Obor: Průmysl 4.0

Vedoucí práce: doc. Ing. Ivo Bukovský, Ph.D.

Rok: 2022



MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Erdman Vladislav** Personal ID number: **503822**
Faculty / Institute: **Faculty of Mechanical Engineering**
Department / Institute: **Department of Mechanics, Biomechanics and Mechatronics**
Study program: **Industry 4.0**
Branch of study: **No Special Fields of Study**

II. Master's thesis details

Master's thesis title in English:

Machine Learning System for Anomaly Detection and Analysis of Dynamical System

Master's thesis title in Czech:

Systém se strojovým učením pro detekci a analýzu anomálií dynamického systému

Guidelines:

- 1) Review machine learning methods for anomaly detection and analysis in dynamic systems - take into account the principle of statistical methods in comparison with machine learning methods.
- 2) Analyze a dynamic system with respect to its dynamic properties and possible nature of anomalies (if possible for both the simulated system and for the real system)
- 3) Select suitable methods for detecting anomalies and their possible aggregation and test on a dynamic system (simulated and real), properly document the algorithms.
- 4) Design a method of supervised, non-supervised, or, if necessary, combined classification of detected anomalies, properly document the algorithms.
- 5) Program the user interface of the entire system and document the solution in the minimum necessary form.
- 6) Test the functionality of the programmed system on the analyzed dynamic systems

Bibliography / sources:

Například:

- [1] MARKOU, Markos a Sameer SINGH. Novelty detection: a review—part 1: statistical approaches. Signal Processing [online]. 2003, 83(12), 2481–2497. ISSN 0165-1684. Dostupné z: doi:10.1016/j.sigpro.2003.07.018
- [2] MARKOU, Markos a Sameer SINGH. Novelty detection: a review—part 2: neural network based approaches. Signal Processing [online]. 2003, 83(12), 2499–2521. ISSN 0165-1684. Dostupné z: doi:10.1016/j.sigpro.2003.07.019
- [3] BUKOVSKY, Ivo, Witold KINSNER a Noriyasu HOMMA. Learning Entropy as a Learning-Based Information Concept. Entropy [online]. 2019, 21(2), 166. ISSN 1099-4300. Dostupné z: doi:10.3390/e21020166
- [4] HEISTRACHER, Clemens, Anahid JALALI, Indu STROBL, Axel SUENDERMANN, Sebastian MEIXNER, Stephanie HOLLY, Daniel SCHALL, Bernhard HASLHOFER a Jana KEMNITZ. Transfer Learning Strategies for Anomaly Detection in IoT Vibration Data. In: IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society: IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society [online]. 2021, s. 1–6. ISSN 2577-1647. Dostupné z: doi:10.1109/IECON48115.2021.9589185
- [5] ZHAN, Jun, Chengkun WU, Xiandong MA, Canqun YANG, Qiucheng MIAO a Shilin WANG. Abnormal vibration detection of wind turbine based on temporal convolution network and multivariate coefficient of variation. Mechanical Systems and Signal Processing [online]. 2022, 174, 109082. ISSN 0888-3270. Dostupné z: doi:10.1016/j.ymssp.2022.109082

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a použil pouze podklady uvedené v příloženém seznamu literatury.

V Praze dne _____

Vladislav Erdman

Poděkování

Když jsem přijal téma této práce neměl jsem ponětí o datovém inženýrství, práci s AI ani programování v python. Rád bych proto předal poděkování doc. Ivo Bukovskému, Ph.D., který mi při vypracování práce poskytl odborné rady a potřebné vzdělání. Zároveň bych chtěl poděkovat týmu lidí, se kterým mě Ivo propojil. Vítu Pawlikovi za pomoc se strojem při testování algoritmu, Kateřině Marzové za konzultaci problematiky predikce údržby a Ondřeji Budíkovi za konzultaci v oblasti kódového provedení.

Abstrakt

Tato práce se zabývá návrhem systému pro analýzu a detekci anomálií v průmyslovém dynamickém systému. V teoretické části je popsán charakter anomálií v signálu, existující přístupy detekce anomálie, nároky kladené na detekční algoritmus v praxi a příznaky, kterými je možné charakterizovat signál. V praktické části je zvolen a popsán stroj, na kterém je provedeno ověření navrženého systému, proveden výběr příznaků na základě testování a z vybraných příznaků sestaven systém schopný detekce a analýzy anomálie. Je následně ověřena přesnost tohoto systému a navrženo uživatelské rozhraní.

Klíčová slova:

learning entropy, detekce anomálií, charakter anomálií, volba příznaků, supervizované strojové učení

Abstract

This thesis designs a system capable of analyzing and detecting anomalies in an industrial dynamic system. The theoretical part describes the nature of anomalies in the signal, existing approaches for anomaly detection, the demands placed on the detection algorithm in practice and the features that can be used to analyze the signal. In the practical part, a testing machine on which the verification of the proposed system will be performed is selected and described, features are selected based on testing, and a system capable of anomaly detection and analysis is built from the selected features. The accuracy of this system is then verified, and the user interface is designed.

Key words:

learning entropy, anomaly detection, anomaly characterization, feature selection, supervised machine learning

Obsah

1	Úvod	3
2	Teoretická východiska	4
2.1	Charakter anomálií	4
2.2	Detekční algoritmy.....	5
2.2.1	Statistický přístup	6
2.2.2	Přístup strojového učení	8
2.3	Prediktivní model.....	10
2.4	Klasifikátory	10
2.4.1	Logistická regrese	11
2.4.2	Logistická regrese pro více tříd.....	12
2.4.3	Klasifikační přesnost	13
2.5	Příznaky.....	13
2.5.1	Příznaky pro predikci dynamiky	13
2.5.2	Příznaky pro klasifikátor	14
3	Praktická část.....	17
3.1	Stanovení základních požadavků	17
3.2	Zvolený stroj	19
3.2.1	Popis stroje	19
3.2.2	Popis dat.....	21
3.3	Volba příznaků	22
3.3.1	Testované anomálie	23
3.3.2	Popis simulovaných modelů.....	25
3.3.3	Výběr a testování příznaků	29
3.4	Kompletní popis návrhu	38
3.4.1	Přijetí signálu	38
3.4.2	Získání příznaků	39
3.5	Ověření navrženého systému	45
3.5.1	Learning Entropy příznaky.....	47
3.5.2	FFT příznaky.....	47
3.5.3	Statistické příznaky.....	49
3.5.4	Drsnostní příznaky.....	50
3.5.5	Celkový test	51
3.6	Uživatelské rozhraní	53
3.6.1	Knihovna python	53

3.6.2	Návrh front-end.....	54
3.6.3	Manažer.....	56
3.7	Celkové zhodnocení.....	57
4	Závěr.....	59
5	Bibliografie	60

1 Úvod

Se současným trendem průmyslu 4.0 roste důraz na predikování údržby. Ve strategii údržby strojů je snaha předcházet neočekávaným selháním součástí stroje, které by mohly vést k ohrožení životů a majetku a zároveň snaha hlídat, že stroj stále operuje v požadované přesnosti. Údržba se tak musí konat v dostatečně častých intervalech, což zvyšuje náklady výroby. Proto společnosti hledají způsoby optimalizace intervalů údržby tak, aby byla nastavena nejmenší nutná frekvence, ale zachováno bezpečí lidí, strojů a kvalita výroby. V oblasti prediktivní údržby a detekce anomálií je velký rozsah prací, avšak malé množství těchto prací se věnuje praktické aplikaci těchto systémů.

Hlavním cílem této práce je navržení holistického systému, který bude schopen detekovat a analyzovat anomálie (poruchy) na zvoleném stroji. Systém bude navržen tak, aby byl aplikovatelný v praxi a bylo možné jej přizpůsobit i rozličným průmyslovým strojům. Výstupy tohoto systému lze použít k prediktivní údržbě.

V první části práce se věnuji popisu anomálií a existujícím přístupům detekce anomálií, ze kterých budu volit přístupy, které použiji pro vytvoření detekčního systému. V druhé části práce se věnuji detailnímu popisu navrženého systému, testování jeho částí a analýzy celkové funkčnosti, přesnosti a aplikovatelnosti v praxi.

2 Teoretická východiska

2.1 Charakter anomálií

Anomálie jsou velmi obecným pojmem, který lze v rámci této práce definovat jako odchylku pozorované veličiny od její vypočítané hodnoty. Zdroj anomálie tak závisí na pozorovaném stroji, jevu a na použité technice pozorování veličiny.

Zatímco zdroj anomálie je individuální daným okolnostem, charakter anomálie lze do určité míry zobecnit. Pochopení charakteru anomálií je důležité pro důsledné otestování detekčního algoritmu.

I když nebyla nalezena literatura shrnující možné projevy anomálie pro účely detekce, je možné se inspirovat metodikami z průmyslové metrologie. Vzhledem k tomu, že primárním záměrem práce je analýza anomálií průmyslového stroje, je možné se odrazit od charakteru anomálií na povrchu obrobku, které se projevují změnou vnitřního stavu stroje [1]. Zkoumané anomálie pro účely této práce tak lze rozdělit do skupin:

1. **Změna struktury šumu** (pravděpodobnostního rozložení hodnot)

Šum (drsnost) je v metrologii anomálie, která nejvíce závisí na změně vnitřní dynamiky stroje, na rozdíl od trendu, který je dán konstrukcí stroje a vlnitostí, která bývá způsobená externími jevy. Zpravidla se jedná o změnu tuhosti stroje nebo jeho části, která může být ovlivněna změnou teploty, únavou, povolením nebo dalšími jevy [1].

2. **Vznik nové frekvence**

Další detailně zkoumaná oblast v metrologii. Označuje se také jako vlnitost a je způsobená periodickými cykly externích nebo interních vlivů na stroj. Může se jednat o špatné upevnění součásti, cyklický vznik tepla v důsledku zvýšení tření součásti nebo činností vedlejšího pracovního procesu [1].

3. **Změna frekvence**

Na rozdíl od vzniku nové frekvence, který je zpravidla důsledkem externích vlivů, změna frekvence ve stroji může být důsledkem vnitřních vlivů stroje. Například změna tuhosti montážního celku a tím i vlastní frekvence.

4. **Změna parametru součásti**

Lineární změna může být způsobená například změnou koeficientů pružnosti a tlumení v důsledku creepu, cyklické únavy nebo poškození. Nelineární změna může být způsobená komplikovanými anomáliemi ve stroji v důsledku kterých by vznikla dříve neexistující závislost mezi více vnitřními procesy (například povolení tlumení nebo těsnění nějaké součásti, popřípadě změna vlastní frekvence).

5. **Skoková porucha**

Může být důsledkem technických i elektronických problémů. V případě technických problémů se může jednat o externí vlivy jako náraz do stroje, náraz pracovní hlavy do obrobku, prasknutí součásti ve stroji apod. V případě elektronických problémů se může jednat o výpadek nebo chybu senzorů.

2.2 Detekční algoritmy

Pro detekci anomálií existuje široká škála algoritmů, které jsou přizpůsobeny rozdílným problémům a tím i požadavkům kladeným na výkon algoritmu. Detekce anomálie v chování bankovních aktivit má jiné optimální řešení a klade jiné požadavky na detekční systém než detekce anomálie v signálu z měření mechanického systému. Rozdíl mezi požadavky těchto dvou případů lze hodnotit dle 7 kritérií, se kterými přichází Markou [2]. Ve své práci shrnuje 7 nejdůležitějších požadavků, které se na detekční algoritmus kladou v praxi:

1. **Robustnost a kompromis**

Detekční algoritmus musí maximalizovat rozpoznání dat zatížených anomálií a minimalizovat chybné rozpoznání normálních dat jako anomálních, tj. musí maximalizovat přesnost predikce a minimalizovat chybu prvního druhu. Tento kompromis mezi přesností a chybou musí být do určité míry pod kontrolou a stabilní.

2. **Jednotné škálování dat**

Data vstupující do detekčního algoritmu by měla po normalizaci spadat do stejného rozsahu hodnot. To zlepšuje robustnost a stabilitu detekčního algoritmu.

3. **Minimalizace parametrů**

Detekční algoritmus by měl mít co nejmenší nutné množství uživatelem nastavitelných parametrů. Parametry algoritmu by mělo být možné automatizovat, aby byla zredukována lidská chyba a komplikovanost nasazení algoritmu v praxi.

4. **Generalizace**

Detekční algoritmus bude dávat smysluplné výsledky i při zaznamenání dosud neviděných dat. Tento problém se vyskytuje nejčastěji u klasifikátorů, které nová data řadí do existujících kategorií.

5. **Nezávislost**

Detekce anomálie algoritmem by neměla být závislá na množství tříd a příznaků. Algoritmus by měl dávat smysluplné výsledky i při menším množství dat nebo se zvýšeným šumem v datech.

6. **Adaptace**

Detekční algoritmus by mělo být možné jednoduše přeučit na jiný systém při zachování jeho schopnosti identifikovat anomálie.

7. **Výpočetní komplexnost**

Velké množství aplikací pro detekci anomálií požadují nasazení v reálném čase. Proto by měl být výpočet detekce co nejméně komplexní.

Algoritmy pro detekci anomálií se rozdělují na dvě velké skupiny, které se odlišují principem a postupem detekce. Jsou to přístupy statistické a přístupy s neuronovým učením [2], [3].

2.2.1 Statistický přístup

Statistické přístupy se odlišují od přístupů s neuronovým učením prací s pravděpodobností. Jsou z velké části založeny na modelování dat na základě statistických charakteristik signálu (např. průměr a směrodatná odchylka). Tyto charakteristiky jsou použity k rozhodnutí, zda nový vzorek dat má stejné pravděpodobnostní rozložení jako tréninková data [2].

Definice statistického přístupu umožňuje vytvoření široké škály algoritmů, které budou zapadat do této kategorie. Nejjednodušší metodou by mohlo být určení pravděpodobnostního rozložení normálních dat a následná volba prahu pravděpodobnosti, která pokud bude překročena, bude nový datový vzorek považován za anomálii. Takový přístup by vyžadoval velké množství dat pro určení pravděpodobnostního rozložení, mohl by mít problémy s robustností a nebyl by schopen detekovat anomálie, které se vyskytují pod zvolenou hladinou pravděpodobnosti. Proto se pracuje na více sofistikovaných statistických metodách, které lze rozdělit na parametrické a neparametrické.

2.2.1.1 Parametrický přístup

Parametrický přístup se zakládá na Gaussovo distribuci a předpokládá se, že zkoumaný signál má pravděpodobnostní distribuci odpovídající normálnímu rozdělení. Základem je stanovení hladiny pravděpodobnosti, při jejíž překročení dojde k odmítnutí a označení nového vzorku dat jako anomálie. To vede i k odmítnutí určitého množství odlehlých hodnot normálních dat [2].

Základním postupem je výpočet základních parametrů Gaussovo křivky (průměr a směrodatná odchylka) a stanovení prahu, jehož překročení odděluje anomálii a normální signál. Z toho vyplývají dvě kritická místa parametrického přístupu – dostatečné množství dat pro stanovení přesného Gaussovo rozdělení (kritérium nezávislosti) a optimální stanovení prahu (kritérium adaptace).

Optimální stanovení prahu parametrického přístupu řeší Chow [4]. Přistupuje k řešení minimalizací chyby prvního druhu, protože považuje eliminaci falešně pozitivních výsledky za základ pro nasazení algoritmu v praxi. Z jeho práce vyplývá, že optimální práh pro stanovení anomálie Gaussovo přístupu je nad maximální hodnotou normálních dat.

Tax a Dunin [5] přistupuje ke stanovení prahu 3 směrodatných odchylek od průměru natrénovaného modelu, čímž za předpokladu Gaussového rozložení pravděpodobnosti se spokojí s 0.3% šancí falešné identifikace normálního signálu jako anomálie.

Z pohledu [7 kritérií hodnocení](#) detekčních algoritmů je problematickým místem parametrického přístupu robustnost. Tj. detekce komplexních anomálií, které se neprojevují významnou změnou pravděpodobnostního rozložení. Parametrický přístup se proto někdy využívá v kombinaci s jinými metodami. Terassenko ve své práci [6] využívá parametrického přístupu společně s metodou k-means. Metoda k-means rozděluje data do daného počtu shluků (běžně na 2D rovině), které jsou definovány svým středem a datovými body, které pod daný shluk spadají. Samotná metoda se používá především ke klasifikaci dat, ne k detekci anomálií [7]. Pro detekci anomálie využívá Terassenko právě parametrického přístupu. Pokud je nový datový bod vzdálen od centra přiřazeného shluku o nastavený práh daný pravděpodobností, pak je považován za anomálii.

2.2.1.2 *Neparametrický přístup*

Neparametrický přístup na rozdíl od parametrického nezávisí na předpokladech o skutečné distribuci, ale stále využívá pro detekci anomálií pravděpodobnost, čímž se řadí do statistických přístupů [2]. Příklady neparametrického jsou k-NN, Parzen okno, nebo porovnávání řetězců.

2.2.1.2.1 *k-NN*

Metoda k-NN (z angl. k – nearest neighbors, česky k – nejbližších sousedů) je v základu klasifikačním algoritmem, ale využívá se i jako neparametrický detekční algoritmus. Spočívá v rozložení datových bodů na n -dimenzionální rovině, kde n je počet příznaků dat. Vektor příznaků nových dat vytváří na rovině nový datový bod. Okolo nového bodu se stanoví okruh, ve kterém se nachází přesně k - nejbližších bodů normálních dat. Kategorie nového bodu je daná kategorií s největším zastoupením bodů ve stanoveném okruhu.

Pro detekci anomálie se využívá několik přístupů. Jedním z nich je stanovení hranice, kterou nesmí poloměr kružnice pevně daného počtu nejbližších sousedů překročit. S dalším přístupem přichází Hellman. Ve své práci [8] úspěšně používá pro detekci anomálie 2 nejbližší sousedy. Pokud 2 nejbližší sousedi nového bodu jsou z různých kategorií, pak je tento nový bod označen za anomálii.

Největším záporem této metody je výpočetní komplexnost v případě, že chceme kategorizovat velké množství dat, kvůli nutnosti počítat vzdálenosti ke každému bodu [2], [9].

2.2.1.2.2 *Parzenovo okno*

Parzenovo okno je metoda podobná k-NN, na rozdíl od které se místo stálého počtu bodů využívá stálý poloměr s vyhlazovací funkcí. Vyhlazovací funkcí je myšleno, že body daleko od středu kružnice mají na kategorizaci menší vliv než body blíž ke středu kružnice [10]. Pro detekci anomálie se používá práh zvolené vyhlazovací funkce. Pokud se ve stanoveném okruhu nového bodu nenachází žádný bod normálních dat, pak je tento bod považován za anomálii [2].

Stejně jako u metody k-NN je největším záporem Parzenova okna výpočetní komplexnost. Výpočet vzdáleností od bodů normálních dat je stále potřebný pro zjištění, která data spadají do zvoleného okruhu.

2.2.1.2.3 *Porovnání řetězců*

S novou metodou neparametrického přístupu přichází Forrest et al. [11], [12]. Pro vytvoření detekčního algoritmu se autoři inspirovali imunologií. Ve své práci [12] přirovnávají detekci anomálie algoritmem k detekci cizorodé látky lidským organismem.

Principem je vytvoření matice příznaků S o rozměru k, m z normálního signálu, kde m je počet příznaků a k je počet příznakových vektorů charakterizujících signál. Matice S reprezentuje vlastnosti normálních dat. Poté je vygenerovaná matice R s náhodnými hodnotami o rozměru d, m , kde d je zvolený počet náhodných vektorů. Každý náhodný vektor z R je porovnáván s každým vektorem S . Pokud mezi porovnávanými vektory dochází ke 75% shodě, pak je náhodný vektor z matice R odstraněn. Po ukončení procesu eliminace

funguje matice R jako filtr. Pokud se vektor příznaků nových data shoduje s některým z vektorů v R alespoň ze 75%, pak je identifikován jako anomálie [12].

Z pohledu 7 kritérií se algoritmus na teoretických datech prokázal jako robustní. Forrest et al. [11], [12] požívá k vytváření příznakových vektorů binárního kódování, kterým zajišťuje jednotné škálování dat. Algoritmus nebyl testován v reálném čase a potenciální nevýhodou tohoto algoritmu může být kritérium adaptace. Pro přeučení algoritmu na nová data je nutné algoritmus naučit od znova.

2.2.2 Přístup strojového učení

Základem přístupu strojového učení je využití neuronu nebo neurálních sítí, které predikují budoucí výstupy [13]. Z popisu tohoto přístupu vyplývá, že obdobně jako statistický přístup shrnuje modely, které pracují s pravděpodobnostmi, přístup strojového učení shrnuje modely, které pracují s predikcí.

Podle složitosti modelů lze rozdělovat algoritmy založené na strojovém učení na samotné neurony a neuronové sítě.

2.2.2.1 *Techniky založené na neurálních sítích*

Velké množství neurálních sítí ve své podstatě nedokáže detekovat anomálii, protože primárním účelem neurálních sítí bývá klasifikace [2]. To znamená, že rozřídí nový signál do známých kategorií, navzdory tomu, že nezapadá do žádné z nich. Jedním ze způsobů predikce je využití pravděpodobnosti zařazení signálu do dané kategorie. Pokud je pravděpodobnost, že signál patří do kterékoliv kategorie pod určitou hranicí, pak se považuje za anomálii [3].

Mezi příklady technik založených na neurálních sítích spadá vícevrstvá dopředná neuronová síť, SOMS (z angl. self organizing maps, v čj. samoorganizační mapy), síť radiálních bázových funkcí, metoda podpůrných vektorů, síť založené na logistické regresi a další [3].

Z pohledu 7 kritérií je zásadní nevýhodou těchto přístupů požadavek velkého množství anomálních dat pro natrénování modelu. I přes navržené metody, které redukuje množství trénovacích dat, jsou tyto techniky stále více náročné na množství dat než samostatné neurony. Z tohoto důvodu nebudou neurální sítě v této práci dále rozebírány jako detekční algoritmy. Neurální síť bude v této práci použita pouze jako klasifikační algoritmus, viz [kapitola 2.4](#).

2.2.2.2 *Techniky založené na samostatných neuronech*

Princip detekční metody na základě samostatných neuronů se zakládá na pohybu koeficientů, které reprezentují změnu vnitřní dynamiky systému, tzv. vah. Jestli se vztah mezi vstupy a výstupy systému změní, pak se změní i váhy systému. Detailnější popis funkce neuronu je rozebrán v [kapitole 2.3](#).

Protože se detekce zakládá na pohybu vah, je zásadní důkladné počáteční natrénování neuronu. Váhy neuronu musí konvergovat k ustáleným hodnotám a zkoumaný pohyb vah na zkoumaném signálu tak nebyl vyvolán nedostatečným natrénováním, ale anomálií.

Pro detekci anomálie je možné kombinovat průběh vah se statistickými technikami. S originální technikou detekci anomálie na samostatných neuronech přichází Bukovský et al.

Ve své práci [14] přichází s konceptem learning entropy, který využívá pro detekci novosti (anomálií) na neuronu s gradientním učením.

2.2.2.2.1 Learning Entropy

Learning entropy přistupuje k neuronovému učení jako k termodynamickému systému. Vnesení anomálie (nové informace) do systému způsobí zvýšenou aktivitu neuronů, které se ustálí na nové hladině, podobně jako se termodynamický systém přizpůsobí vnesenému teplu.

Základním principem fungování je, že pokud je aktivita neuronů neobvykle vysoká, pak do systému vstupují neznámá data, a to detekuje learning entropy zvýšenými hodnotami. Když se systém na tato data adaptuje pak hodnota learning entropy klesá k nule.

Pro výpočet learning entropy E_A lze využít přímého algoritmu, který popisuje Bukovský v práci [15]:

$$E_{ADirect}(k) = \frac{1}{\eta_\beta \cdot \eta_\omega} \sum_{j=1}^{\eta_\beta} \sum_{i=1}^{\eta_\omega} h(z(|\Delta w_i(k)|) > \beta_j), \quad (1.1)$$

$E_{ADirect}(k) \in \langle 0,1 \rangle$ je learning entropy vypočítaná přímým algoritmem. Hodnota 0 znamená, že žádný z přírůstků vah není neobvykle velký a 1, že všechny přírůstky vah jsou neobvykle velké.

β_j je j -ta prahová hodnota z celkového počtu prahových hodnot η_β .

$\Delta w_i(k)$ je i -ty přírůstek váhy v kroku k z celkového počtu vah η_ω .

$h(x)$ je funkce booleovské transformace na binární hodnotu $h(false) = 0$ a $h(true) = 1$.

$z(x)$ je normalizační funkce z-skóre, která se počítá dle následujícího vzorce:

$$z(x(k)) = \frac{x(k) - \mu_x}{\sigma_x} \quad (1.2)$$

kde μ_x je průměrná hodnota vektoru x a σ_x je směrodatná odchylka.

Postupem je získání snahy učení $|\Delta w_i(k)|$ pro krok k a váhu i , která se vypočítá absolutní hodnotou přírůstků dané váhy v kroku k . Následně dochází k porovnání této snahy učení se sérií prahů ve vektoru β . Pokud je daná snaha učení větší než daný práh, pak se k learning entropy v tomto kroku přičte hodnota 1. To se opakuje pro všechny prahy a všechny váhy ve zvoleném kroku.

Ve svých pracích [14], [16] Bukovsky et al. ukazují schopnost použití learning entropy algoritmu v reálném čase pomocí metody gradientního učení. Poukazují přitom na problematiku detekce neobvykle nízkých hodnot. Použití learning entropy jak je uvedeno

v rovnici (1.1) je schopné detekovat anomálie, které zvyšují aktivitu učení, ale ne ty, které ji snižují (např. pokud by v důsledku poruchy stroje došlo k nečinnosti vedlejšího pohonu chlazení vnášejícího šum do systému).

2.3 Prediktivní model

Část této práce se zakládá na detekčním algoritmu založeným na strojovém učení, proto je důležité pochopit dostupné možnosti prediktivních modelů a princip jejich sestavování. Sestavování prediktivních modelů lze rozdělit podle principu výpočtu koeficientů na iterativní a dávkový, nebo podle principu rovnic, ke kterým jsou počítány koeficienty na lineární, polynomiální, goniometrický, kombinovaný apod. (princip hledání vah je stále lineární).

Iterativní výpočet se provádí v každém datovém kroku a k výpočtu přírůstku vah potřebuje váhy stanovené v posledním kroku a aktuální hodnotu výstupů. Má vyšší pravděpodobnost zapadnutí do lokálního minima [17], ale dokáže identifikovat změny vnitřní dynamiky v každém kroku signálu. Příkladem iterativního výpočtu je gradientní učení. Dávkový výpočet se provádí pomocí celého okna signálu, běžně pomocí Jacobi matic. Má menší pravděpodobnost zapadnutí do lokálního minima, ale klade vyšší nároky na výpočet. Příkladem takového výpočtu je algoritmus Levenberg–Marquardt. V praxi se také využívá kombinace obou metod, přičemž na počátku se více používá dávkový výpočet, který umožňuje rychlejší přiblížení se globálnímu minimu na úkor komplexnějšího výpočtu a následně se použije iterativní výpočet pro rychlejší konvergenci vah ke konkrétním hodnotám [17].

Architektura prediktivního modelu je supervizovaná. To znamená, že při volbě rovnic, ke kterým jsou počítány koeficienty, vycházíme z předpokladů a dedukcí o tom, jak se skutečný systém chová. Volba těchto rovnic tak závisí na poznání mechanismu modelovaného stroje, nebo sledováním chyby predikce a volbou rovnic iterativním postupem. Další možností je vycházet z toho, že velké množství strojů je kvůli říditelnosti a jednoznačnosti z velké míry lineární, a proto by mohla být základní lineární rovnice dostatečná. Linearitu stroje lze ověřit pomocí koeficientu determinace R^2 , který se v tomto případě počítá jako druhá mocnina koeficientu korelace mezi vstupy a výstupy [18].

$$R^2 = r_{xy}^2 = \left(\frac{1}{n-1} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right) \right)^2, \quad (1.3)$$

kde r_{xy} je koeficient korelace, n je množství dat, x_i, y_i jsou hodnoty dat v obou osách na kartézské rovině (KR), \bar{x}, \bar{y} jsou průměry dat v obou osách na KR a s_x, s_y jsou směrodatné odchylky dat v obou osách na KR.

R^2 se pohybuje mezi hodnoty 0 a 1 a v daném případě popisuje do jaké míry je variabilita systému popsána lineárním vztahem. Posouzení významnosti hodnoty se liší dle oboru i účelu [19]. Pro potřeby datového inženýrství lze považovat za dostatečné $R^2 > 0.7$ [20].

2.4 Klasifikátory

Klasifikátory jsou algoritmy, které využívají technik strojového učení pro predikci kategorie vzorku dat. Podle řešeného problému je lze rozdělit na binární klasifikátory pro řešení dvou

tříd (např. detekce anomálie, rozpoznávání spamu), klasifikátory pro více tříd (např. rozpoznávání písma nebo textu), klasifikátory pro více značení (např. detekce vícero objektů na jedné fotografii) a nevybalancované klasifikátory, ve kterých množství dat jedné třídy značně převyšuje množství dat z jiných tříd [21]. Stejně jako u strojového učení lze klasifikátory dělit na supervizované a nesupervizované [22]. Účinnost těchto modelů je zpravidla měřena metrikou klasifikační přesnosti [21], která je popsána dále v [kapitole 2.4.3](#).

Pro široké spektrum řešených problémů je vytvořeno široké spektrum používaných algoritmů. Logistická regrese, k-NN, rozhodovací stromy, metoda podpůrných vektorů, Naive Bayes jsou algoritmy používané pro binární klasifikaci i klasifikace více tříd. Náhodný les je používán pro klasifikaci více tříd a je přizpůsoben rozhodování na základě booleovských tříd. Pro nesupervizovanou klasifikaci lze použít SOMS, k-means, ISODATA [22].

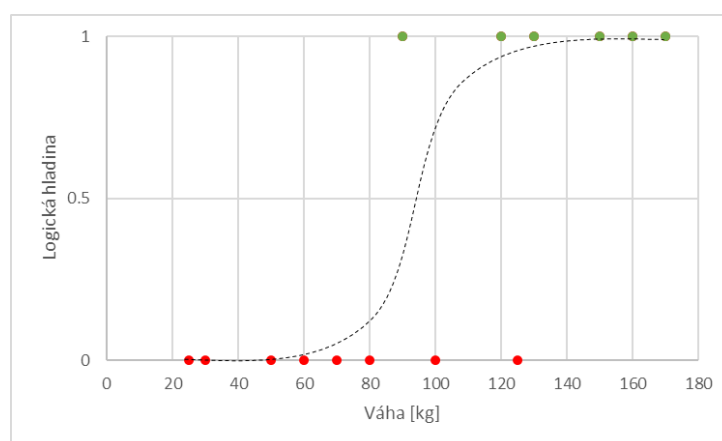
Účelem klasifikátoru v této práci je zjistit, jestli jsou zvolené příznaky dostatečné pro klasifikaci anomálií, proto je dále popsán jednoduchý lineární model logistické regrese, který umožňuje detailnější analýzu vlivu příznaků na klasifikaci konkrétní třídy.

2.4.1 Logistická regrese

Logistická regrese je lineární model, který predikuje pravděpodobnost, zda daný jev nastal nebo nenastal. Jedná se o binární klasifikátor, který rozhoduje mezi stavem 0 a 1 (neboli false a true) [23].

Vstupem do algoritmu je matice vektorů příznaků X o rozměrech m, n , kde m je počet příznaků a n množství trénovacích vektorů. Dalším vstupem je kategorieční vektor y o rozměru n , který nese binární hodnoty kategorie pro každý vektor příznaků v X .

Použití algoritmu lze ukázat na jednoduchém ilustrativním příkladu detekce obezity s jediným příznakem – váhou bytosti. Ilustrativní data jsou zobrazena na obr. č. 1.



Obr. 1 – Ilustrativní ukázka dat obezity v závislosti na váze s proloženou logistickou regresí (čárkovaně). Barevně je rozlišena kategorie dat: červeně – neobézní, zeleně – obézní.

Cílem je proložit data logistickou funkcí (zpravidla sigmoid) tak, abychom maximalizovali celkovou pravděpodobnost správného rozřazení dat. Metoda této optimalizace je nazývána „metoda maximální věrohodnosti“ [24]. Provádí se projektováním datových bodů na

navrhovanou křivku a následným součinem pravděpodobnosti, že je daný bod zařazen do správné kategorie. Například pravděpodobnost, že krajní levý zelený bod na obr. č. 1 je zařazen do své kategorie je přibližně 0.4. Pravděpodobnost, že krajní pravý červený bod je zařazen do té své je rovna $1 - 0.95$, kde 0.95 je přibližná hodnota daného bodu na křivce. Matematicky je kritická funkce optimalizace rovna:

$$CF = \sum_{i=1}^n \text{sign}(f(x) - 0.5) * g(x) - \text{sign}(f(x) - 1) \quad (1.4)$$

pro $i = 1, 2, \dots, n$,

kde $f(x)$ je funkce odpovídající binární hodnotě daného bodu (1/0), $g(x)$ je funkce navrhované křivky.

Optimalizovanými parametry logistické regrese jsou stejně jako u lineární regrese parametry přímky. U logistické regrese tato přímka vzniká převodem *pravděpodobnosti* jevu do logaritmických souřadnic *šance*, že daný jev nastane. Převod pravděpodobnosti na šanci a zlogaritmování je provedeno dle následující rovnice [25]:

$$y_{new} = \log\left(\frac{y}{1-y}\right), \quad (1.5)$$

kde y_{new} je bod na nové přímce a y bod na logistické křivce.

Hodnoty 0, 1 z logistické přímky jsou po transformaci na přímku rovny $-\infty, \infty$. Hodnota 0.5 na logistické přímce tvoří počátek nové y osy, tj. je rovna 0 na přímce. Po transformaci jsou optimalizovanými parametry m, b z rovnice přímky:

$$y = m \cdot x + b \quad (1.6)$$

Kvůli datům v nekonečnu není možné tuto přímku optimalizovat metodou nejmenších čtverců často používanou u lineární regrese. Proto se pro optimalizaci používá zmíněná metoda maximální věrohodnosti.

Po optimalizaci přímky je zpětná transformace provedena pomocí rovnice:

$$y = \frac{e^{\log(y_{new})}}{1 + e^{\log(y_{new})}} \quad (1.7)$$

2.4.2 Logistická regrese pro více tříd

Základní algoritmus logistické regrese řeší binární problém, ale při využití metody one-vs-rest je možné jej použít i pro klasifikaci s více kategoriemi [21].

Pokud při logistické regresi obsahuje kategorizační vektor y vícero tříd, pak se provede binární regrese pro každou ze tříd zvlášť. Při binární regresi je u zvolené třídy v y nastavena

logická hodnota 1 a u zbytku dat logická hodnota 0. Pro každý vektor příznaků v X je zjištěna pravděpodobnost každé třídy v y a je zvolena kategorie s nejvyšší pravděpodobností [26].

Největší nevýhodou tohoto přístupu je komplexní výpočet, který způsobí problémy při klasifikaci velkého množství dat s velkým počtem tříd [26].

2.4.3 Klasifikační přesnost

Nejběžněji používanou metrikou klasifikátorů (ale i prediktorů) je klasifikační přesnost, která je počítaná na testovacích datech jako počet správných predikcí dělený celkovým počtem predikcí [21]. Největší nevýhodou této metriky je selhání u nevybalancovaných dat, kde je větší množství vektorů jedné kategorie oproti dalším kategoriím [27]. Tento problém lze řídit rovnoměrným výběrem dat.

2.5 Příznaky

V popisu detekčních algoritmů i klasifikátorů se často pracuje s pojmem *příznaky dat*. Jedná se o měřitelnou vlastnost nebo charakteristiku signálu, která do určité míry popisuje jeho varianci. Jedná se o široký pojem a spadá do něj velké množství nástrojů, kterými lze různá data popsat. Příznaky jsou vstupy do prediktivních modelů a jejich volbou se rozumí rozhodnutí o tom, které charakteristiky signálu budou vstupovat do modelu a které ne, na základě toho, jak moc jsou přínosné pro detekci daného problému [28].

Může se jednat o volbu vstupů a výstupů mechanického systému, které budeme měřit, nebo o jejich modifikaci. Vstupní signály mohou být mezi sebou násobené (tzv. polynomiální neuron), zpožděné, popsané jinou charakteristikou (průměr, směrodatná odchylka atp.) a mnohé další [20].

V této práci se pracuje se dvěma modely strojového učení a každý bude vyžadovat jiné příznaky. Jeden model je pro predikci výstupů stroje, jehož cílem je napodobit dynamiku systému, která poslouží k odhalení anomálie. Druhým modelem je klasifikátor, jehož cílem je stanovit jakou chybu nalezená anomálie reprezentuje.

Základním způsobem volby příznaků je lidská dedukce podpořená technickým ověřením (zpravidla korelací, nebo kontrolou kvality predikce modelu), další možností je použití genetických nebo jiných optimalizačních algoritmů pro testování, které příznaky mají největší a které nejmenší vliv na výsledek predikce [28].

Dále jsou popsány všechny příznaky, které byly zváženy a otestovány v rámci této práce. Počáteční volba příznaků byly provedena na základě rešerše technik popisů signálu a inženýrské dedukce.

2.5.1 Příznaky pro predikci dynamiky

V případě příznaků prediktivního modelu popsaného v [kapitole 2.3](#) se zpravidla jedná o rekurenci, vytváření polynomů, zpoždění vstupů a přidávání na vstup minulé hodnoty výstupu.

2.5.1.1 Zpoždění vstupů

Zpožďování vstupů spočívá v rozšíření matice vstupních vektorů o vektory se zpožděním o několik časových kroků. V důsledku měření nebo zpožděné reakce některé jevy mohou být

vůči sobě fázově posunuty, v takových případech může být zpoždění vstupů účinným příznakem [20].

2.5.1.2 Přidání výstupů

Přidávání vektorů výstupu z minulých časových korekčů na vstupy je další účinná metoda, jak zlepšit kvalitu predikce [20]. Zakládá se to na principu setrvačnosti určitých jevů. Např. pokud by bylo vstupem do predikce zmáčknutí pedálu plynu v autě a predikovaným výstupem rychlost auta, potřebovali bychom pro predikci navíc předchozí hodnoty rychlosti, protože tu nelze zjistit ze zrychlení samotného.

2.5.1.3 Rekurence

Rekurence spočívá v rozšíření vstupní matice o přechodí hodnoty predikovaných výstupů místo měřených výstupů. Při malé hodnotě chyby predikce tento příznak umožňuje predikování hodnot po omezenou dobu bez měření výstupů. Po překročení určité periody by došlo ke značnému zvětšení chyby predikce.

2.5.1.4 Polynomiální

Vytváření polynomů rozšiřuje vstupní matici o násobky vstupů mezi sebou. Nejčastěji se jedná o kvadratické a kubické násobky nebo jejich kombinaci [20].

2.5.2 Příznaky pro klasifikátor

2.5.2.1 PCA

PCA (v angl. Principle Component Analysis, v čj. analýza hlavních komponent) je algoritmus, který se ve strojovém učení používá pro mnoho účelů, např. snižování dimenzí, shluková analýza, detekce anomálií, filtrace, výběr příznaků apod. Analýza hlavních komponent se provádí singulárním rozkladem kovariační matice, ze kterého jsou získány vlastní vektory a vlastní čísla matice.

Vlastní čísla s nejvyššími hodnotami označují vlastní vektory, které popisují nejvíce variance dat. Rekonstrukce signálu na základě vlastních vektorů s nejvyššími vlastními čísly tak umožňuje filtraci dat od anomálních hodnot.

Příznakem do klasifikátoru mohou být vlastní čísla kovariační matice, která se budou měnit v případě, že dojde ke změně závislostí ve váhách systému.

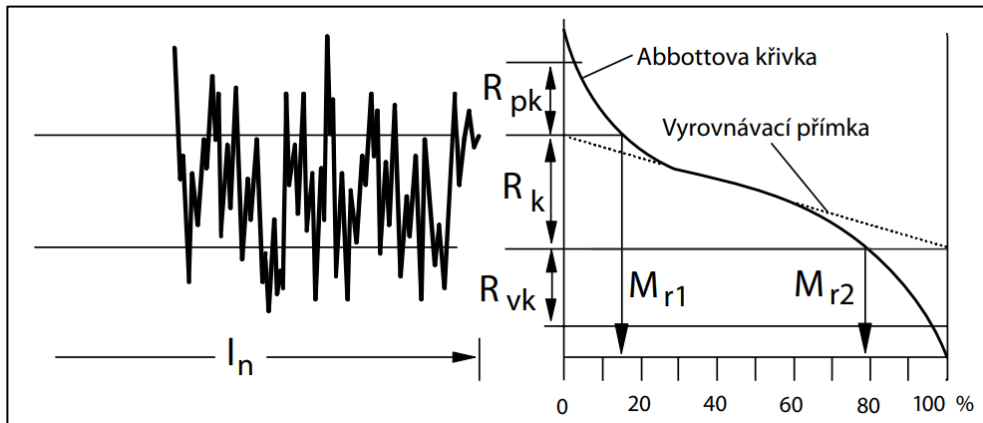
2.5.2.2 Korelace

Koeficient korelace vyjadřuje lineární vztah mezi dvěma zvolenými průběhy a jeho hodnoty se pohybují v rozmezí $\langle -1, 1 \rangle$. Hraniční hodnoty vyjadřují, že je mezi dvěma průběhy silný lineární vztah, zatímco 0 označuje, že mezi dvěma průběhy neexistuje lineární závislost [29].

Jako příznak lze použít korelační koeficient mezi oknem aktuálních vah a referenčním signálem, který reprezentuje určitý průběh (lineární nebo polynomiální). To umožní sledovat vznikající trend v pohybu vah a identifikovat, že váhy konvergují k novým hodnotám nebo že se v nich objevují nelinearity.

2.5.2.3 Abbottova–Firestoneova křivka

Abbottova–Firestoneova křivka je nástroj používaný v metrologii pro popis nerovnosti a drsnosti povrchu a stanovení nosného podílu. Principiálně se jedná o kumulativní distribuční funkci profilu nerovností nebo šumu. Tento nástroj poskytuje 5 příznaků, které lze použít k limitované charakterizaci harmonického průběhu, ale hlavně k charakterizaci šumu [1], viz obr. č. 2.



Obr. 2 – Ilustrativní obrázek Abbottovo-Firestoneovo křivky na pravé straně pro šum na levé straně [1]

Princip vzniku křivky zobrazuje obr. č. 2, kde šum po levé straně je převeden na Abbottovo-Firestoneovo křivku po pravé straně a jsou zobrazeny měřené parametry (příznaky):

R_k je hloubka jádra drsnosti

R_{pk} je redukovaná výška výstupků nad jádrem profilu drsnosti

R_{vk} je redukovaná hloubka prohlubní pod jádrem profilu drsnosti

M_{r1} je nejmenší materiálový podíl jádrového profilu drsnosti

M_{r2} je největší materiálový podíl jádrového profilu drsnosti

2.5.2.4 Rychlá Fourierova transformace (FFT)

Rychlá Fourierova transformace umožňuje získat příznaky, které souvisí s frekvencemi vyskytujícími se v signálu. Konkrétním příznakem mohou být nejvíce významné frekvence, které překračují vypočtený práh.

2.5.2.5 Základní statistický popis

Základními statistickými parametry okna dat je možné obdržet další sérii příznaků. Jedná o aritmetický průměr, směrodatnou odchylku a základní kvantily (0.2, 0.5, 0.75). Stabilita těchto parametrů bude růst se zvětšujícím se oknem dat.

2.5.2.6 Parametry drsnosti

Průmyslová metrologie dále poskytuje přes 15 parametrů drsnosti, které lze použít pro detailní analýzu šumu signálu. Tato oblast příznaků se prolíná s příznaky, které lze získat ze statistického popisu. To, co metrologický přístup přináší navíc je nápad oddělení šumu od

vlnění a trendů a zaměření se na detailní analýzu čistého šumu. Parametry drsnosti tak poskytují příznaky pro charakterizaci změn v šumu.

Nejběžnějším parametrem je R_a (arithmetic), který reprezentuje průměrnou aritmetickou hodnotu absolutního průběhu profilu šumu. Výpočet [1]:

$$R_a = \frac{\sum_{i=1}^n |y_i|}{n} \quad (1.8)$$

Dalším parametrem je R_q (quadratic). Průměrná kvadratická úchylka profilu, která se svým principem podobá R_a . Výpočet [1]:

$$R_q = \sqrt{\frac{\sum_{i=1}^n y_i^2}{n}} \quad (1.9)$$

R_{sk} (skewness) popisuje šikmost profilu drsnosti. Tento parametr je velmi citlivý na odlehle hodnoty a je více vhodný na periodické signály. Výpočet [30]:

$$R_{sk} = 3 \cdot \frac{\bar{y} - \tilde{y}}{s_y}, \quad (1.10)$$

kde \bar{y} je aritmetický průměr šumu, \tilde{y} je medián šumu a s_y je směrodatná odchylka.

R_{ku} (kurtosis) popisuje špičatost profilu drsnosti a stejně jako R_{sk} je velmi citlivý na odlehle hodnoty. Jedná se o čtvrtý centrální moment vydělený kvadrátem rozptylu. Výpočet [31]:

$$R_{ku} = \frac{\sum_{i=1}^n \bar{y}^4}{n \cdot s_y^4} \quad (1.11)$$

3 Praktická část

Cílem praktické části této práce je navrzení, sestavení a otestování algoritmu schopného detekce anomálií na skutečném stroji a to tak, aby byl použitelný v praxi.

Nejdříve jsem stanovil základní požadavky kladené na algoritmus, ze kterých jsem vycházel v pozdějším návrhu. Dále byl zvolen stroj, na kterém bude navržený algoritmus testovaný. Dřívější volba stroje umožní pochopení konkrétních praktických výzev, které nejsou patrné v teorii. Dalším krokem je výběr a testování příznaků a detekčního algoritmu, které budou použity ve výsledném detekčním systému. Následuje kompletní kódové sestavení celkového systému včetně zpracování dat, interpretace výsledků a zabalení algoritmu do rozhraní, se kterým může pracovat uživatel a pomoci, kterého bude možné algoritmus aplikovat na průmyslový stroj.

3.1 Stanovení základních požadavků

V této kapitole bude popsáno nejen stanovení základních požadavků, ale i jakým způsobem se zajistí jejich dosažení a jakým způsobem se budou kontrolovat na finálním systému. Základní požadavky na detekční stroj vychází ze 7 nejčastějších kritérií kladených v praxi popsanych v [kapitole 2s.2](#).

- **Robustnost a kompromis**

Zajištění: Bude zajištěno vytvořením odolných metod detekce anomálie, možností nastavení kompromisu (prahu detekce) a důkladným testováním těchto metod na různých variacích možných anomálií popsanych v [kapitole 2.1](#).

Kontrola: Metrika predikční přesnosti popsané v [kapitole 2.4.3](#).

- **Jednotné škálování dat**

Zajištění: V detekčním algoritmu bude škálování dat kompenzováno pomocí metod, které nebudou závislé na jednotném škálování dat vzhledem k neustálému porovnávání s referenční hodnotou, jejíž škálování se bude měnit společně s novým vzorkem dat.

Kontrola: Prohlédnutí kódu a ujistění se, že kód pro škálování dat je zajištěn dle plánu.

- **Minimalizace parametrů**

Zajištění: Průzkum charakteru optimálních parametrů algoritmu a navrzení automatizace jejich volby.

Kontrola: Soupis všech nastavitelných parametrů a zohlednění možnosti jejich automatizace.

- **Generalizace**

Zajištění: Volba vhodných detekčních algoritmů, jejichž detekce anomálie bude záviset pouze na důkladném poznání normálního signálu.

Kontrola: Kontrola proběhne posláním na vstupy do systému vektor náhodných dat místo aktuálního signálu a následným zkontrolováním, jestli výsledky dávají smysl.

- **Nezávislost**
Zajištění: Zajištěna vhodným navržením algoritmu a jeho průběžným simulováním při různém množství vah, dat a šumu.
Kontrola: Vyzkoušení navrženého systému na jiném množství vah a jiné délce okna.
- **Adaptace**
Zajištění: Zajištěna snížením nároků na množství dat a vytvořením kalibračního algoritmu na přizpůsobení detekčního prahu na základě nových normálních dat.
Kontrola: Změření množství dat potřebných smysluplnému natrénování systému a otestování trénování systému pomocí navrženého kalibračního algoritmu.
- **Výpočetní komplexnost**
Zajištění: Výpočetní komplexnost závisí nejen na správné volbě detekčních algoritmů, ale i na naprogramování a vyhodnocovaném množství dat. Pro snížení výpočetní komplexnosti bude kladen důraz na volbu nástrojů s potenciálem práce v živém čase.
Kontrola: Změření času na vypracování jediného vzorku dat.

Těchto 7 kritérií vychází z obecných požadavků kladených v různých praktických aplikacích, musíme proto také zohlednit požadavky, které vychází z účelu navrhovaného systému a z cíle práce.

- Systém musí být **dostatečně citlivý** na rozpoznání změn v dynamice systému, ještě před závažným selháním, které by vedlo k neplánované údržbě.
Zajištění: Vytváření systému schopného detekovat chyby pod hladinou šumu modelu.
Kontrola: Metrikou predikční přesnosti dle [kapitoly 2.4.3](#) na anomáliích pod hladinou šumu.
- Systém musí být z **programátorského hlediska** lehce přizpůsobitelný rozličným strojům.
Zajištění: Psaní kódu systému do většího množství nahraditelných modulů a normalizace dat, které se mezi moduly předávají.
Kontrola: Vizuální kontrola grafického znázornění struktury kódu
- Systém musí mít zredukované nebo nulové **chyby prvního druhu**, které by mohly způsobit falešný poplach a tím pádem falešnou neplánovanou údržbu.
Zajištění: Volba detekčního algoritmu na základě jeho výsledků v metrice chyby prvního druhu.
Kontrola: Metrika falešně pozitivních detekcí při testování algoritmů.

Řešený problém popsany v [úvodní kapitole](#), tj. vytvoření systému, který zpřesní plánování údržby, neklade takový důraz na práci systému v živém čase. Větší důraz se klade na práci algoritmu v určitých intervalech v závislosti na množství vyrobených kusů. Tím se snižují nároky na výpočetní komplexnost.

3.2 Zvolený stroj

Pro otestování navrženého algoritmu jsem zvolil stroj Shaker Tira TV 51140. Jedná se o stroj, který je používán pro vibrační zkoušky při různém vysokofrekvenčním průběhu signálu. Volbu uvedeného stroje opodstatnily tři charakteristiky: jednoduchá konstrukce, vysoké dynamické zatížení a vysoký stupeň linearity. Tyto charakteristiky usnadňují práci na vedlejších algoritmech a umožňují se soustředit na hlavní náplň práce – detekci anomálií.

Jednoduchá konstrukce spočívá v relativně nízkém počtu součástí, které ovlivňují výsledný pohyb a zároveň se jedná o SISO (v angl. single-input-single-output, v čj. jeden-vstup-jeden-výstup) systém. Jednoduchá konstrukce umožňuje pochopit jaký vliv bude mít simulovaná chyba na výstupní hodnoty a také pochopení možných zdrojů chyb. SISO snižuje nároky vedlejší algoritmy jako je zpracování dat a prediktivní algoritmus, čímž se zrychluje doba simulace.

Vysoké dynamické zatížení poskytuje motivaci pro aplikaci vytvořeného systému na tomto stroji. Je způsobeno účelem vibrační zkoušky, která je používána pro vysokofrekvenční testování součástí [32]. Všechny vodící a elektronické prvky mechanismu jsou tak vystaveny vysokému počtu cyklů. Při hodinovém vysokofrekvenčním testování pomocí bílého šumu (frekvence 3200 Hz) dojde k přes 10 miliónům změn pohybu [32].

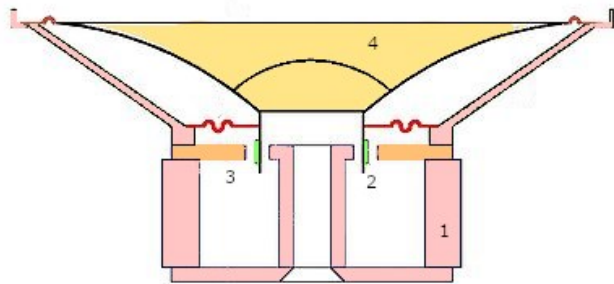
Vysoký stupeň linearity je očekáván s ohledem na princip fungování stroje, protože hodnota proudu přímo zodpovídá za sílu působící na pracovní plošinu, jak je popsáno v [kapitole 3.3.2.2](#). Vypočtený koeficient determinace po zpracování dat pro daný stroj je $R^2 = 0.7$, což poukazuje na vysoký stupeň linearity, viz [kapitola 2.3](#). Hlavním zdrojem nelinearity jsou pružné vodící prvky, vyrovnávací polohové mechanismy a pružení mezi pracovní plošinou a cívkou armatury, na kterou působí síla (viz popis stroje dále v této kapitole). V důsledku vysokého stupně linearity dojde k usnadnění práce s prediktivním modelem.

3.2.1 Popis stroje

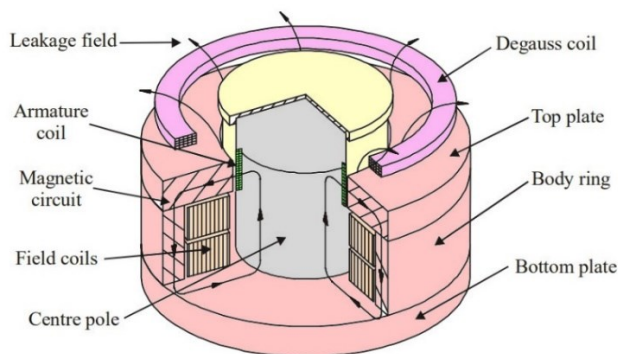
Ve své podstatě Shaker funguje jako reproduktor. Skládá se z těla (červeně na obr. č. 3, 4, 5) a armatury, která je uprostřed těla (žlutě na obr. č. 3, 4, 5). Tělo obsahuje pólové cívky (oranžově na obr. č. 3, 4, 5), do kterých vstupuje stejnosměrný proud a generuje statické magnetické pole. Armatura obsahuje cylindrickou cívku (zeleně na obr. č. 3, 4, 5), kterou prochází řízený střídavý proud a tím dochází k pohybu armatury ve statickém magnetickém poli pólové cívky.

Ideální kmitání armatury probíhá, když se nachází okolo výchozího tzv. centrálního bodu (ve vertikální ose pohybu). Když je na armaturu upevněné závaží je bod kmitání sesunut dolů. Pro kompenzaci nežádoucího posunu bodu kmitání je pod armaturou zařízení regulující tlak vzduchu, který armaturu vrací zpět do počáteční polohy.

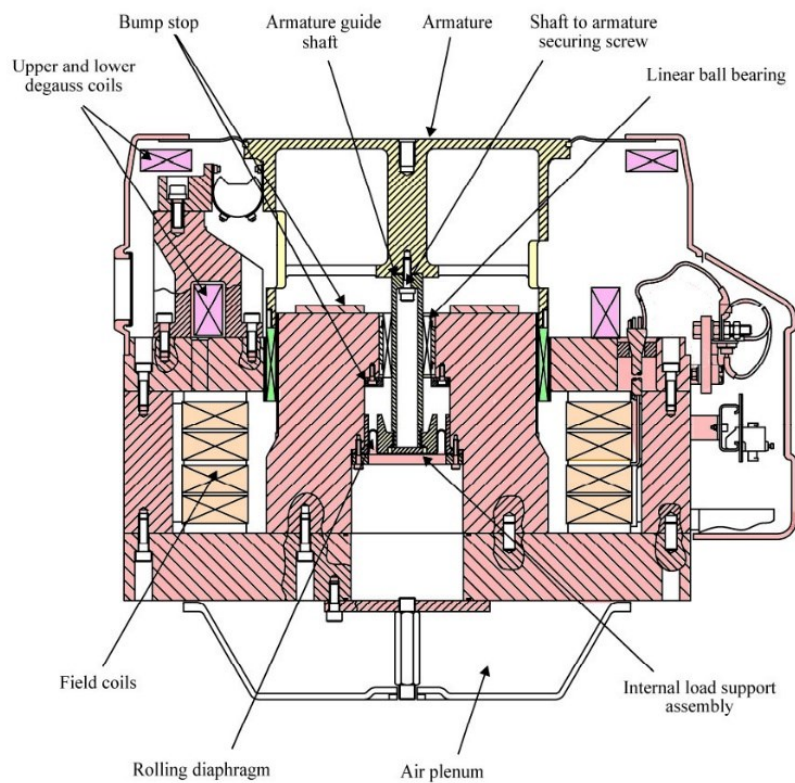
Stroj navíc obsahuje Degaussovo cívku (růžově na obr. č. 4, 5), která je za účelem eliminace šumových magnetických proudů, které by jinak vířily do okolí.



Obr. 3 – Schématické zobrazení reproduktoru



Obr. 4 – Schématické zobrazení modelu Shakeru [33]



Obr. 5 – Schéma skutečného Shakeru V830 od LDS [33]

Z popisu stroje vyplývá, že nejvíce dynamicky zatěžované součásti (tj. ty, které se aktivně účastní kmitavého pohybu) jsou horizontální a vertikální vodiče, pružná stěna armatury, těsnění mechanismu na vyrovnávání polohy a cylindrická cívka.

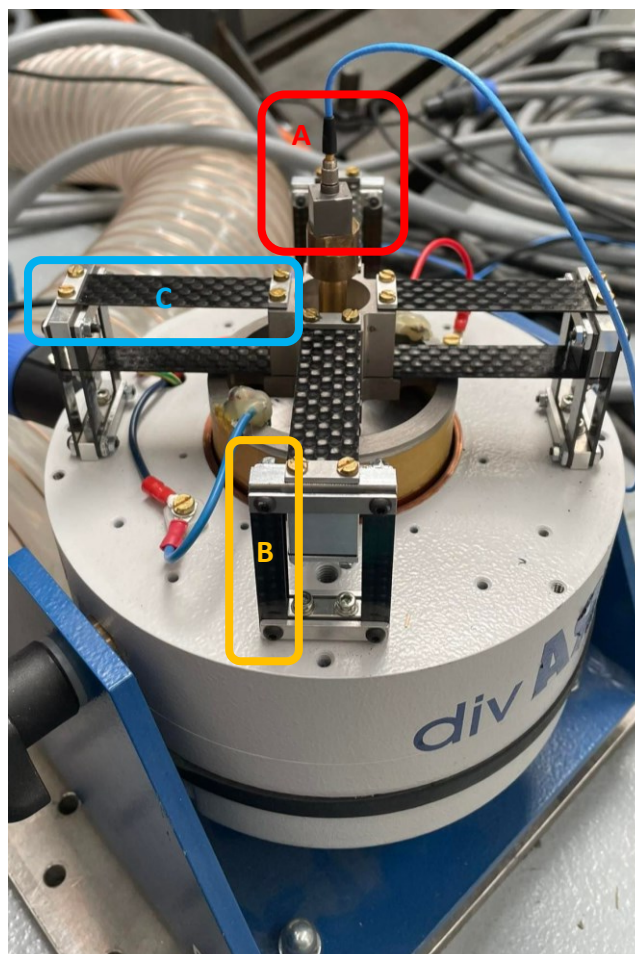
3.2.2 Popis dat

Je důležité si uvědomit, že detekční algoritmus nebude detekovat anomálie samostatného stroje, ale celého řetězce zařízení od získání vstupního signálu po získání výstupního. Proto je důležité si uvést všechna zařízení, která v tomto procesu účinkovala.

Výstupním signálem je průběh napětí korespondující s vertikálním pohyb armatury, který byl získán piezoelektrickým akcelometrem TRIAXIAL ICP® ACCELEROMETE připevněným ke středu pracovní plochy armatury, viz červený obdélník A na obr. č. 6.

Vstupním signálem je zesílený průběh napětí vstupující do cylindrické cívky stroje, který byl získán počítačem přímo ze 16-kanálového zesilovače CCLD typu 2694 od společnosti Brüel & Kjær.

Celým řetězcem zařízení, ve kterých mohlo dojít k nalezení anomálie, byl zesilovač, shaker a akcelometr. S ohledem na cíl měření nebyla měřící zařízení kalibrovaná, zásadní podmínkou je, že nastavení těchto zařízení zůstanou v průběhu měření neměnná.



Obr. 6 – Fotografie Shaker Tira TV 51140 s otevřeným krytováním

Pro simulaci jsem využil dvou přístupných nedestruktivních způsobů zavedení anomálie do systému – povolení vertikálního a horizontálního vedení armatury, viz žlutý (B) a modrý (C) obdélník na obr. č. 6.

Nejprve bylo naměřeno 30 vzorků signálu bez zavedení anomálie v periodě 30 vteřin s intervalem měření 10 vteřin. Poté byl stroj zastaven a došlo k úmyslnému povolení jediného šroubku na zvoleném vedení. Šroubek byl povolen tak, aby vůle vedení nebyla zřetelná lidskému oku a aby nebyl zaznamenán slyšitelný rozdíl ve zvuku vedení před a po povolení. Následně bylo provedeno měření o stejném počtu vzorků, intervalu a periodě. Na závěr bylo po upevnění armatury znovu provedeno měření bez anomálie.

3.3 Volba příznaků

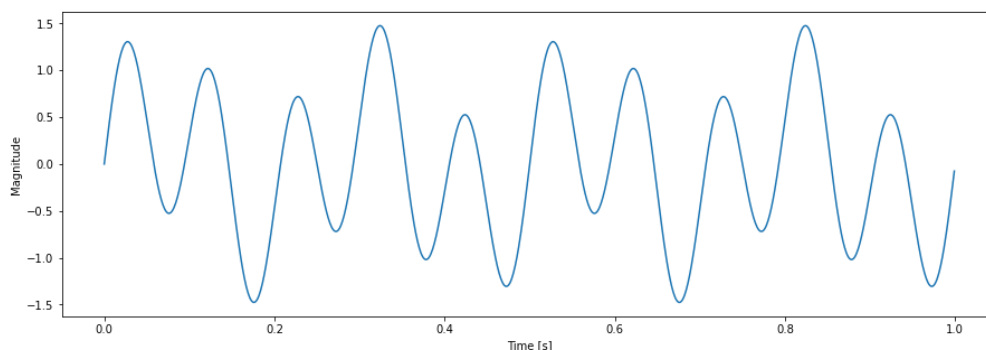
Volbu příznaků jsem provedl na dvou simulovaných modelech a s následným otestováním na datech naměřených ze stroje. Pro ohodnocení funkcionality příznaku jsem se zaměřil na 3 druhy anomálií: změna charakteristiky šumu, zavedení nové frekvence, lineární a kvadratická změna parametrů. Zbylé 2 anomálie uvedené v [kapitole 2.1](#) jsem netestoval kvůli obdobnému principu detekce. Detekce změny frekvence funguje na podobném principu jako detekce objevení nové frekvence a skoková porucha se při detekci projeví obdobně jako změna koeficientu na určitou hodnotu.

Anomálie jsou do modelu zaváděny postupně nárůstem amplitudy pomocí funkce sigmoid (viz obr. č. 1) od 0 až na vybraný násobek šumu, aby bylo možné testovat i detekci náběhu anomálie. Příznaky jsou poté testovány individuálně jak je uvedeno v [kapitole 3.3.3](#).

Řídicím signálem u simulovaného modelu je dvojitá sinusoida o nejvyšší frekvenci 10 Hz a maximální amplitudě 1.5 (viz obr. č. 7). Je popsána rovnicí:

$$x = \sin(10.2.\pi.t) + 0.5*\sin(4.2.\pi.t) \quad (1.12)$$

Do simulačních rovnic je zanesen Gaussovo šum se směrodatnou odchylkou 0.01 (0.6% maximální amplitudy signálu).

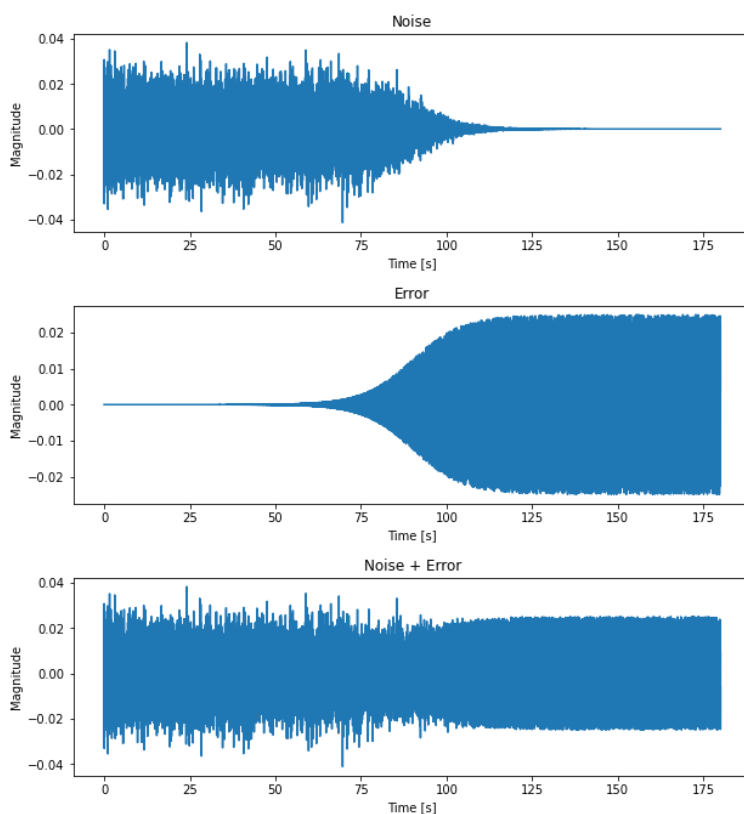


Obr. 7 – Obrázek průběhu vstupního signálu do simulovaných modelů

3.3.1 Testované anomálie

3.3.1.1 Změna charakteristiky šumu

Změna charakteristiky šumu byla provedena změnou amplitudy šumu nebo změnou hustotního rozložení šumu. Pro otestování příznaků byla provedena změna z normální distribuce šumu na uniformní (viz obr. č. 8), Poissonovou a beta distribuci. Změna distribuce byla uskutečněna pomocí funkce sigmoid, aby došlo k plynulému přechodu. Na obr. č. 8 je zobrazeno použití sigmoidu pro útlum původního šumu, náběh nového šumu a jejich kombinaci.



Obr. 8 – Přechod z normálního šumu na uniformní

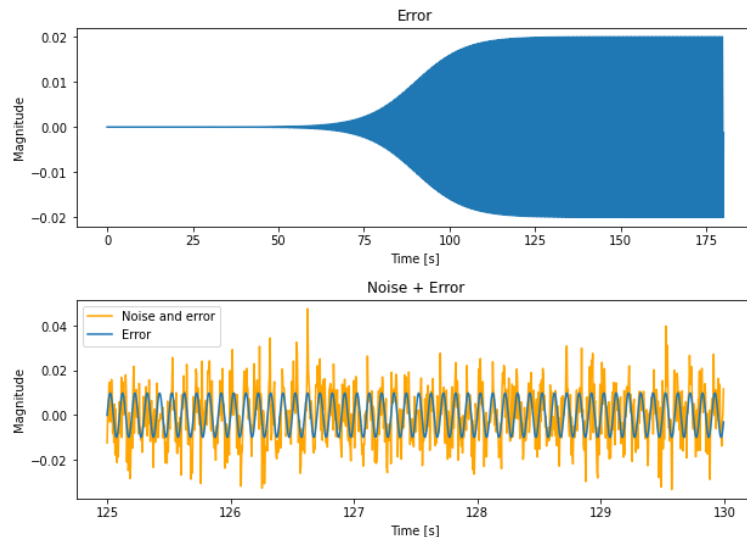
Dále v této práci mluvím o amplitudách anomálií v násobcích šumu. Kde hodnoty blízké 0 znamenají, že jsou anomálie téměř nepatrné. Jak lze vidět z obr. č. 8 změna amplitudy u nového struktury šumu blízcí se 0 by naopak způsobila značné změny v systému. Z tohoto důvodu je nově zaváděný (anomální) šum šum_{anomal}^R složen z originální distribuce šumu šum_{orig} a nové distribuce šumu šum_{anomal} , dle následující rovnice:

$$\text{šum}_{anomal}^R = a \cdot \text{šum}_{anomal} + \min((1-a), 1) \cdot \text{šum}_{orig}, \quad (1.13)$$

kde a je amplituda anomálie. Pokud $a = 0$, pak je nová struktura šumu identická s šumem původním; pokud $(a > 0) \vee (a < 1)$, pak je nová struktura šumu kombinací obou šumů; pokud $a \geq 1$ potom nová struktura šumu zcela odpovídá šumu zaváděné anomálie s amplitudou rovné a .

3.3.1.2 Výskyt nové frekvence

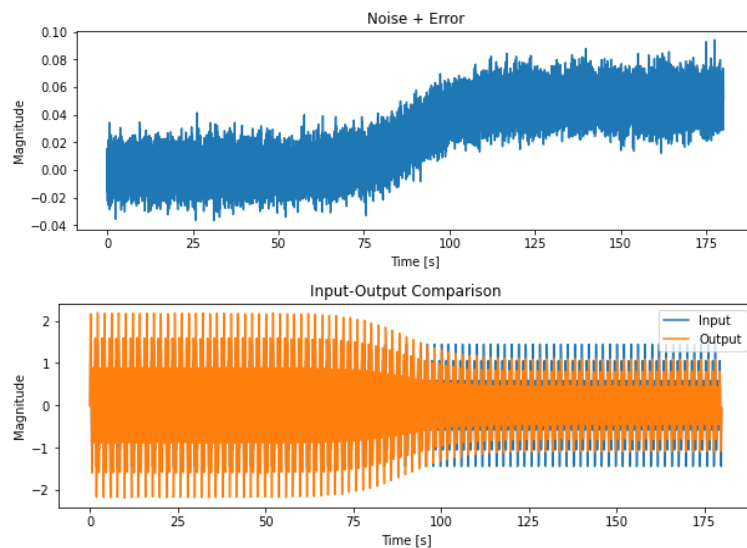
Nová frekvence je do systému zaváděna pomocí sinusoidy o frekvenci 20 Hz. Frekvenci jsem zvolil tak, aby nebylo nutné zvyšovat vzorkování simulace, čímž by se zpomalil výpočetní čas simulace, aby frekvence byla jasně viditelná na grafu Fourierovo transformace (tj. byla daleko od nuly) a aby nedošlo k záměně s frekvencemi řídicího signálu. Průběh zavádění nové frekvence je zobrazen na obr. č. 9.



Obr. 9 – Zavedení nové anomální frekvence do signálu

3.3.1.3 Změna koeficientů

V případě simulačních modelů mechanického systému se jedná o degradaci vybraného koeficientu, který reprezentuje součást mechanického systému. V jednodušším lineárním modelu je změnou koeficientu myšlena změna přírůstku b , nebo změna zesílení výstupu m , dle rovnice (1.14) v následující kapitole. Testovaná změna koeficientů probíhá lineárně, kvadraticky a skokově. Příklad průběhu simulace anomálie změnou koeficientů a její dopad na výstupní hodnoty je zobrazen na obr. č. 10.



Obr. 10 – Zavedení anomálie změny koeficientů dle funkce sigmoid

3.3.2 Popis simulovaných modelů

3.3.2.1 Jednoduchý lineární model

Pro rychlé otestování velkého množství příznaků jsem použil základní lineární model rozšířený o šum:

$$y = m \cdot x + b + \text{šum} , \quad (1.14)$$

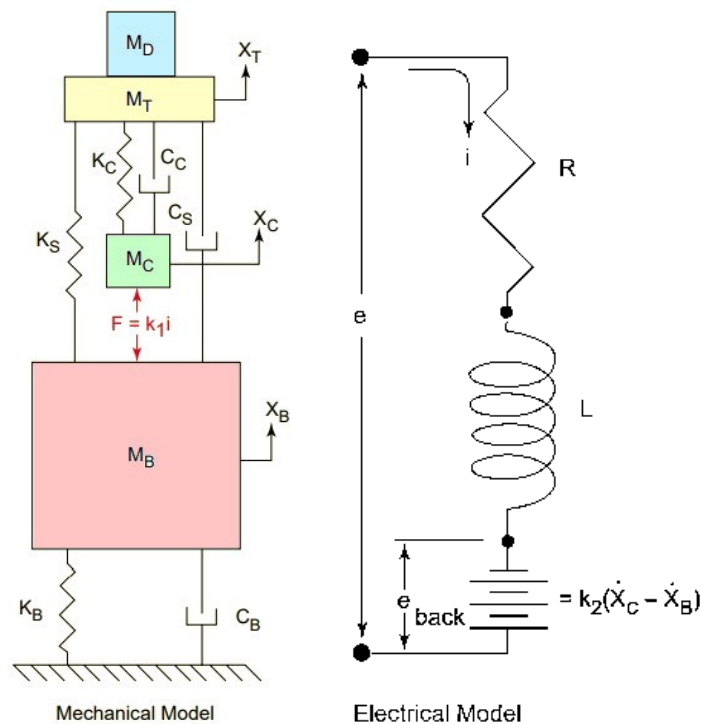
kde $m = 1.5$ a $b = 0$.

Chyby uvedené v [kapitole 2.1](#), se vnáší do systému úpravou šumu, koeficientů m, b nebo úpravou signálu y_{skut} , kde

$$y_{skut} = y + \text{chyba} \quad (1.15)$$

3.3.2.2 Model skutečného dynamického systému

Pro důkladnější kontrolu zvolených algoritmů a jejich schopnosti popsat změny v systému jsem sestavil simulační model zvoleného stroje. S ohledem na strukturu stroje popsanou v [kapitole 3.2.1](#), lze jeho mechanický model sestavit tak jak je zobrazeno na obr. č. 11.



Obr. 11 – Mechanický a elektrický model shakeru

Tělo shakeru (červeně na obrázku) je pomocí upevňovací konstrukce připevněno k podlaze a pomocí vodících prvků drží armaturu (žlutě na obrázku) v ose. K armatuře je připevněná

cylindrická cívka (zeleně na obrázku) skrz tenkou pružnou stěnu. Cylindrická cívka se nachází ve stacionárním magnetickém poli a v závislosti na velikosti a směru proudu uvnitř cívky na ní působí síla přímo úměrná proudu.

Lang [34] rozšiřuje popis modelu o elektrickou část. Zmiňuje problematiku zpětné elektromotorické síly, která vzniká v důsledku pohybu cívky v magnetickém poli. Při pohybu vzniká v cívce proud přímo úměrný rychlosti cívky v poli. Tento jev zahrnuje do elektrické části popisu systému, jak je uvedeno na obr. č. 11, kde rychlost cívky vůči elektrostatickému poli je definovaná jako $\dot{x}_C - \dot{x}_B$, kde \dot{x}_C je rychlost cívky vůči počátku souřadnicového systému a \dot{x}_B je rychlost těla shakeru vůči počátku souřadnicového systému.

Maticový zápis pohybových rovnic modelu vypadá následovně:

$$\begin{aligned} & \begin{bmatrix} M_C & 0 & 0 & 0 \\ 0 & M_T + M_D & 0 & 0 \\ 0 & 0 & M_B & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{x}_C \\ \ddot{x}_D \\ \ddot{x}_B \\ 0 \end{bmatrix} + \begin{bmatrix} C_C & -C_C & 0 & 0 \\ -C_C & C_C + C_S & -C_S & 0 \\ 0 & -C_S & C_B + C_S & 0 \\ k_2 & 0 & -k_2 & L \end{bmatrix} \begin{bmatrix} \dot{x}_C \\ \dot{x}_T \\ \dot{x}_B \\ di/dt \end{bmatrix} + \\ & + \begin{bmatrix} K_C & -K_C & 0 & -k_1 \\ -K_C & K_S + K_C & -K_S & 0 \\ 0 & -K_S & K_B + K_S & k_1 \\ 0 & 0 & 0 & R \end{bmatrix} \begin{bmatrix} x_C \\ x_T \\ x_B \\ i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ e \end{bmatrix} \end{aligned} \quad (1.16)$$

Pro simulaci modelu pomocí knihovny python je třeba přepsat maticový zápis na stavový model:

$$\dot{x} = Ax + Bu \quad (1.17)$$

$$y = Cx, \quad (1.18)$$

kde stavové matice jsou:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{K_C}{M_C} & -\frac{C_C}{M_C} & \frac{K_C}{M_C} & \frac{C_C}{M_C} & 0 & 0 & \frac{k_1}{M_C} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{K_C}{M_T + M_D} & \frac{C_C}{M_T + M_D} & -\frac{K_C + K_S}{M_T + M_D} & -\frac{C_C + C_S}{M_T + M_D} & \frac{K_S}{M_T + M_D} & \frac{C_S}{M_T + M_D} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{K_S}{M_B} & \frac{C_S}{M_B} & -\frac{K_B + K_S}{M_B} & -\frac{C_B + C_S}{M_B} & -\frac{k_1}{M_B} \\ 0 & -\frac{k_2}{L} & 0 & 0 & 0 & \frac{k_2}{L} & -\frac{R}{L} \end{bmatrix} \quad (1.19)$$

$$B = \left[0, 0, 0, 0, 0, 0, \frac{1}{L} \right]^T \quad (1.20)$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{k_C}{m_T + m_D} & \frac{c_C}{m_T + m_D} & -\frac{k_S + k_C}{m_T + m_D} & -\frac{c_S + c_C}{m_T + m_D} & \frac{k_S}{m_T + m_D} & \frac{c_S}{m_T + m_D} & 0 \end{bmatrix} \quad (1.21)$$

stavový vektor:

$$x = [x_C, \dot{x}_C, x_T, \dot{x}_T, x_B, \dot{x}_B, i]^T \quad (1.22)$$

a vstup:

$$u = [i] \quad (1.23)$$

Matice vstupního vektoru C je zvolena tak, aby odpovídala veličinám získaným z akcelometru – poloha a rychlost.

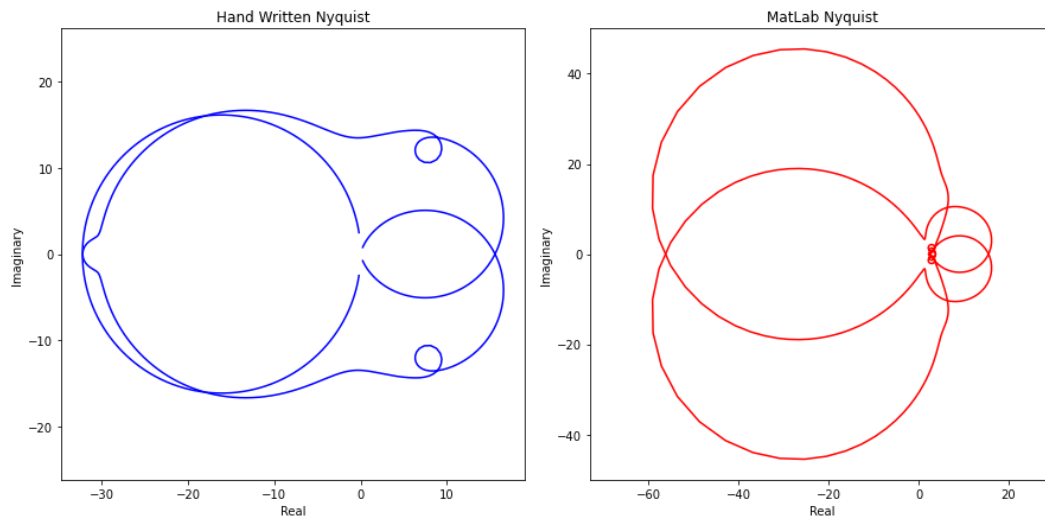
Simulace je provedena pomocí skriptu `signal.StateSpace` z knihovny `scipy`. Tento skript umožňuje na základě stavových matic, vstupů, počátečních podmínek a daného časového vektoru provést simulaci systému. Problematické je zavádění anomálie. Zavedení anomálie do takového systému vyžaduje změnu koeficientů stavových matic v průběhu simulace, což simulační nástroj `scipy` neumožňuje. Zavedení změny koeficientů pomocí vstupního signálu, by vedlo k přepsání stavových rovnic na komplikované nelineární stavové rovnice, které by se ve výsledku musely linearizovat a zavedené změny by se tím eliminovaly. Potenciálním dalším řešením je sepsání vlastního řešiče stavových rovnic, což by bylo časově a programátorský náročné.

Dal jsem proto prioritu přístupu kosimulace. Zakládá se na mikro a makro simulaci. V mikro simulaci je vytvořen `scipy` model na krátký časový interval. Po mikro simulaci jsou stavové matice A , B , C změněny podle strategie zavádění anomálie pro daný časový okamžik dle [kapitoly 3.3.1](#). Poté je vytvořen nový `scipy` model na základě změněných matic a s počátečními podmínkami odpovídajícími stavům v posledním kroku předešlé mikro simulace. Proces provedení simulace a výpočet nových hodnot se nazývá makro simulace. Kódové provedení simulátoru je dostupné v příloze č. 1.

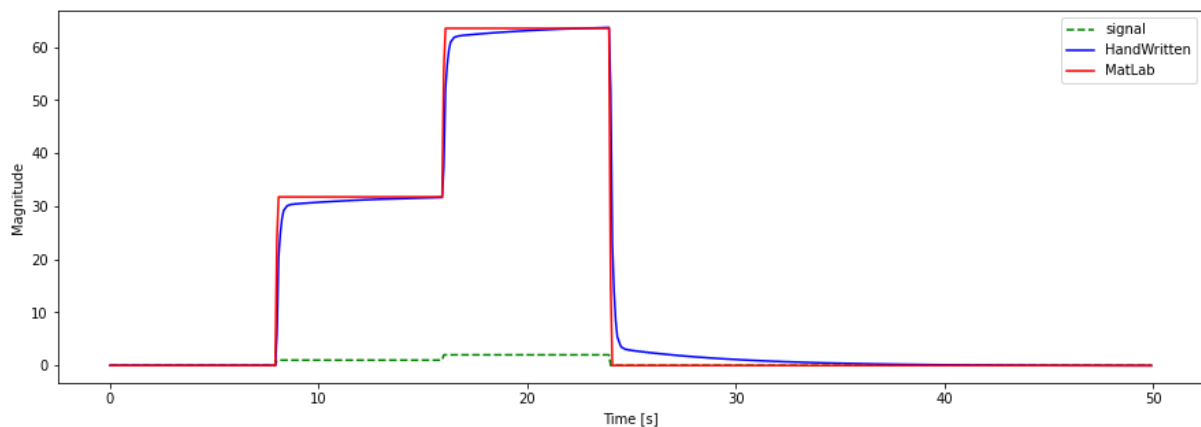
Mikro simulace je nastavena na nejmenší možný časový interval – jeden krok vstupního signálu. Celá simulace probíhá pro simulovaný interval 30 s.

Stanovení koeficientů navrženého modelu bylo provedeno ručně při porovnávání výsledků s modelem navrženým Pawlíkem [32]. V jeho práci vypracovává identifikaci stejného modelu shakeru s úspěšnou shodou výsledků. Odhad koeficientů byl zvolen volbou realistických hodnot modelu a jejich následnou úpravou tak, aby se průběh výstupů z modelu podobal

průběhu výstupu identifikovaného modelu. Z Nyquistovy křivky na obr. č. 12 vyplývá, že v důsledku ručního stanovování koeficientů se navržený model značně liší od skutečného průběhu. Jak lze pozorovat na průběhu vstupů a výstupů u navrženého modelu a identifikovaného modelu na obr. č. 13, je nastavení pro účely detekce anomálie dostatečné.



Obr. 12 – Nyquistova křivka navrženého modelu (modře) a identifikovaného modelu (červeně)



Obr. 13 – Průběh výstupů navrženého modelu (modře) a identifikovaného modelu (červeně) při krokovém průběhu vstupu (zeleně čárkovaně)

3.3.3 Výběr a testování příznaků

Cílem je vybrat příznaky, které budou použity pro detekci a popis anomálie. S ohledem na sedmé kritérium principů detekčních algoritmů – komplexnost výpočtu – je důležité zvolit takové příznaky, které nebudou závažně zatěžovat dobu výpočtu.

Výběr a testování se skládá ze tří částí:

1. Průzkum – Výběr velkého množství algoritmů/příznaků na základě jejich popisu. Výstupem průzkumu je výpis příznaků a algoritmů s potenciální schopností detekovat a klasifikovat anomálie.
2. Test – Zredukování vybraných algoritmů/příznaků a popřípadě přidání nových na základě testu na jednodušších rovnicích. Výstupem testu je výpis nástrojů, které se použijí pro vytvoření detekčního systému.
3. Due Diligence – Zredukování a úprava algoritmů na základě důkladného testu celého systému a jeho jednotlivých částí na všech simulovaných modelech a skutečných datech, viz [kapitola 3.5](#).

3.3.3.1 Průzkum

Na základě průzkumu byl zvolen jeden detekční algoritmus a 7 příznakových nástrojů popsanych v [kapitole 2.5.2](#). Z dostupných detekčních algoritmů byl zvolen algoritmus learning entropy, především kvůli možnosti online nasazení, vysoké citlivosti na neočekávanou změnu vah a také kvůli nízkým nárokům na přesnost modelu. Neschopnost detekovat anomálie, které snižují snahu učení je vykompenzovaná, jak je uvedeno [kapitole 3.4.2.2](#) popisující programové řešení learning entropy.

Zvolené příznaky pro další testování jsou singulární hodnoty z PCA, koeficient korelace mezi referenčním a aktuálním signálem, parametry z Abbottovy-Firestonovy křivky, základní statistické koeficienty, nejvýznamnější frekvence Fourierovo transformace a základní parametry drsnosti. Tento soubor příznaků byl zvolen pro jejich potenciální schopnost popsat velkou škálu anomálií a budou více rozebrány v následující kapitole testování.

3.3.3.2 Testování

Vytřídění základních příznaků pomocí testování bylo provedeno na [jednoduchém lineárním modelu](#). Příznaky byly hodnoceny na základě schopnosti detekovat anomálie uvedené v [kapitole 3.3.1](#). Cílem je vytvořit nejmenší možnou sadu příznaků, které jsou schopné charakterizovat všechny druhy anomálií. Schopnost charakterizovat anomálii hodnotím jako schopnost odlišit normální signál od signálu zatíženého anomálií na základě daného příznaku.

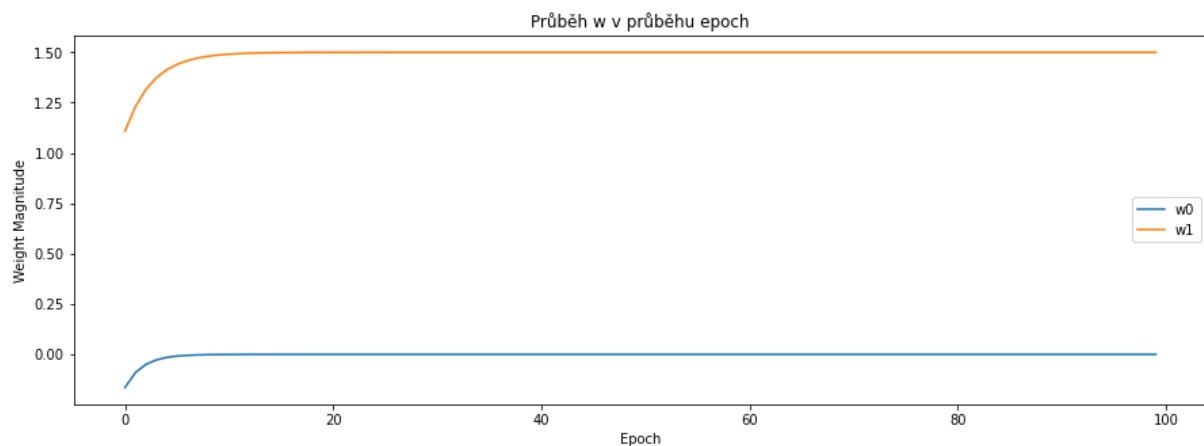
Pro testování je vygenerován signál o intervalu 3 min s frekvencí vzorkování 200 Hz. Tímto je splněna podmínka prevence aliasingu dle Shannon-Kotělnikova teoremu: „Aby nedocházelo k aliasingu, musí být vzorkovací frekvence větší než dvojnásobek nejvyšší frekvence harmonických složek obsažených ve vzorkovaném signálu.“ [35]. Řídicí signál je generován dle rovnice (1.12), který vstupuje do lineárního modelu popsaneho rovnicí (1.14).

Každý příznak je testován na 5 různých amplitudách anomálie. Amplitudy jsem stanovil jako násobky šumu pro pochopení citlivosti příznaku. Navíc byl každý příznak testován na

normálním signálu bez anomálie pro stanovení běžných odchylek. Jsou otestovány amplitudy na 5 hladinách šumu: $0x$, $0.25x$, $0.5x$, $1x$, $2x$.

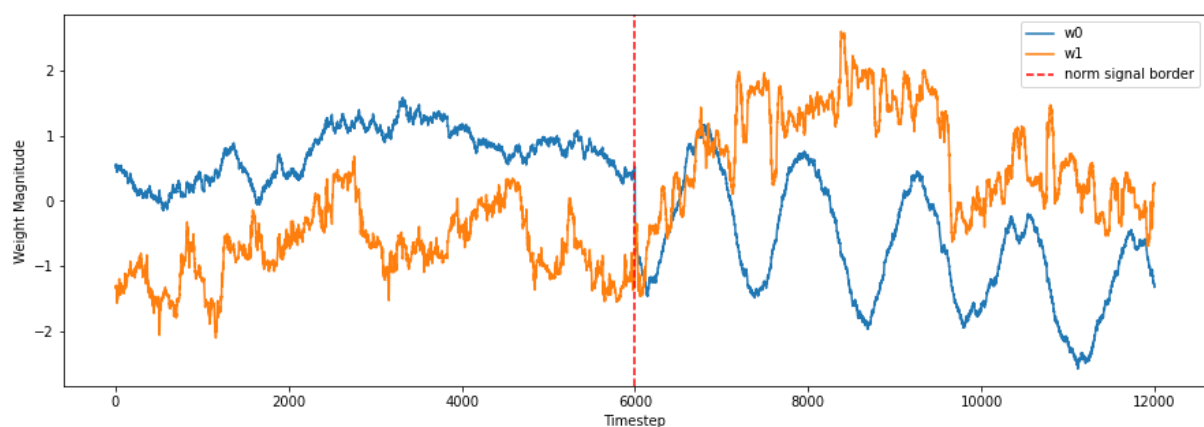
Pro danou chybu a amplitudu je vygenerován signál s chybou a bez chyby s normálním (Gaussovo) šumem o směrodatné odchylce 0.01 . V důsledku vysoké frekvence a nízké amplitudy anomálií a šumu je každá anomálie na celé délce signálu graficky nepatrná. Výsledný signál vypadá velmi podobně průběhu vstupů na obr. č. 7.

Na signálu bez chyby je naučen lineární prediktivní model s gradientním učením. Pro SISO systém tím vznikají dvojce váhy, které reprezentují koeficienty lineární rovnice. Učení je provedeno pomocí 100 epoch s mírou učení $1e-4$, což jsou parametry zajišťující konvergenci vah pro daný případ, viz obr. č. 14.

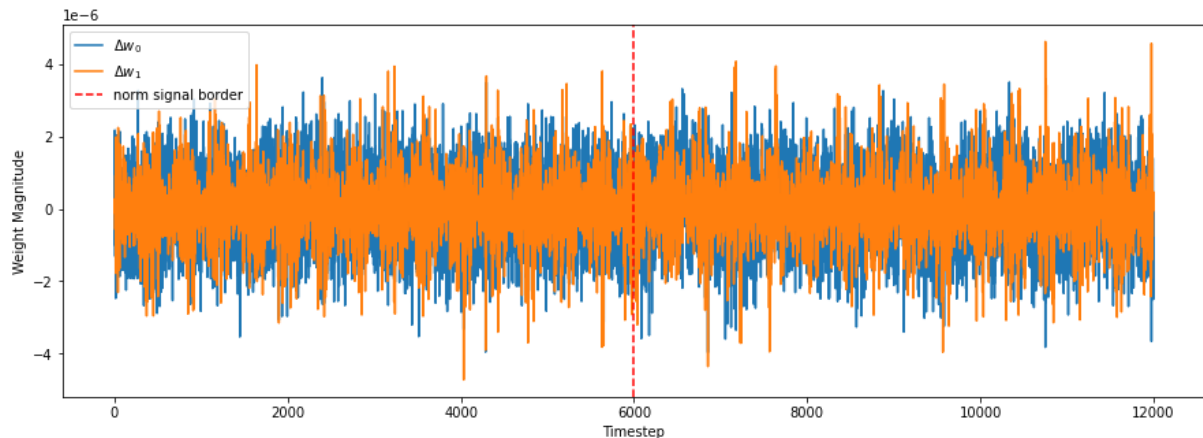


Obr. 14 – Konvergence vah jednoduchého modelu v průběhu epoch

Následně je model aktualizován 1 epochou na signálu s chybou. Předmětem průzkumu jsou průběh vah v této aktualizované epoše, viz obr. č. 15 a přírůstek vah, viz obr. č. 16. Z jednoho z těchto průběhů (v závislosti na zvolené technice) je poté získán testovaný příznak. Pro danou chybu a amplitudu se tento proces opakuje 100x a sleduje se průměr a směrodatná odchylka průběhu testovaného příznaku.

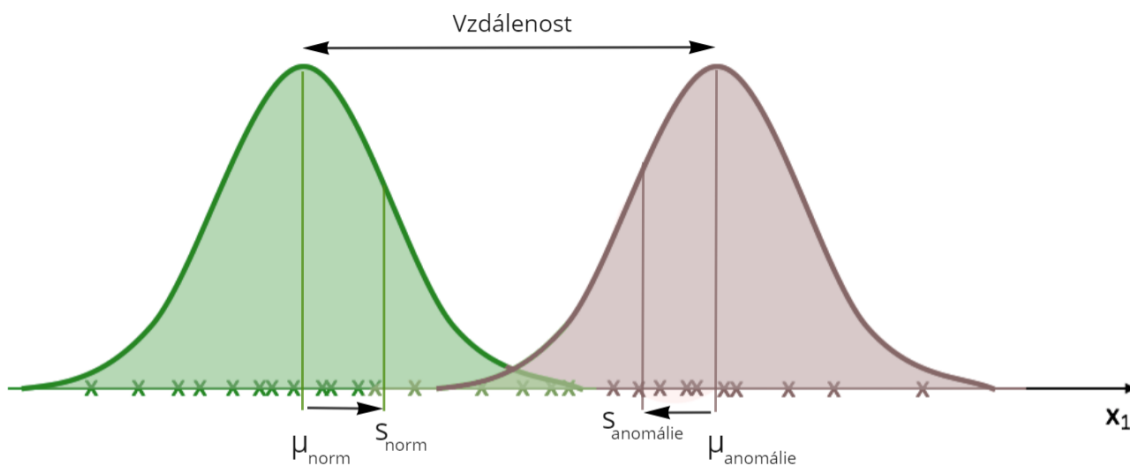


Obr. 15 – Průběh vah normálního signálu v poslední epoše (před červenou linií) a průběh vah signálu s anomálií (po červené linii) po z-skóre. Pozn.: z-skóre provedeno pro zviditelnění rozdílů



Obr. 16 – Průběh přírůstků vah normálního signálu v poslední epoše (před červenou linií) a průběh přírůstků vah signálu s anomálií (po červené linii)

Výstupem testu je ohodnocení významnosti příznaku na základě toho, jestli hodnoty příznaku jsou značně vyšší u signálu s anomálií než u normálního signálu. To je posouzeno porovnáním pravděpodobnostních průběhů hodnot, viz ilustrativní obr. č. 17. Hodnotí vzdálenost průměru příznaku dat s anomálií od průměru příznaků normálních dat a významnost rozdílu těchto rozložení.



Obr. 17 – Ilustrativní obrázek zkoumaných kritérií příznaků

Kritérium vzdálenosti mezi dvěma průměry charakterizuje průměrnou schopnost daného příznaku odlišit anomálii od normálních dat a vypočítá se v násobcích směrodatné odchylky normálních dat:

$$\text{vzdálenost} = \frac{\mu_{\text{anomálie}} - \mu_{\text{norm}}}{s_{\text{norm}}}, \quad (1.24)$$

kde $\mu_{anomalie}$ je průměr příznaků dat s anomálií, μ_{norm} je průměr příznaků normálních dat a s_{norm} je směrodatná odchylka normálních dat. Čím vyšší je hodnota vzdálenosti tím větší je šance, že příznak dokáže kategorizovat danou anomálii.

Kritérium odlišnosti popisuje významnost příznaku pro charakteristiku signálu, a i pro schopnost jeho detekce bez chyb prvního druhu. Významnost se určuje pomocí Z-statistiky:

$$Z = \frac{\mu_{anomalie} - \mu_{norm}}{\text{sqr}(s_{anomalie}^2 + s_{norm}^2)}, \quad (1.25)$$

kde $s_{anomalie}$ je směrodatná odchylka dat s anomálií.

V praxi se považuje signál za stejný, pokud je hodnota Z-statistiky nižší než 2, ale v případě této práce, nebude větší odlišení možné. Pravděpodobnostní křivky obou průběhů budou velmi blízké. Vychází ze stejné distribuce s velmi malými změnami v signálu, které se na váhách modelu projeví nepatrně a tím pádem nedojde u příznaků k významnému statistickému odlišení. Proto subjektem pozorování není absolutní hodnota Z-statistiky a kritéria vzdálenosti, ale jejich porovnání mezi všemi příznaky.

Základní podmínkou je, že obě statistiky prokazují korelaci s rostoucí amplitudou anomálie. Kromě porovnávání příznaků mezi sebou se ještě sleduje, který typ anomálie daný příznak detekuje lépe.

Mezi kritériem Z-statistiky a vzdálenosti platí vztah, že čím větší je rozdíl mezi hodnotami kritéria Z a kritéria vzdáleností, tím větší je šance chyby druhého druhu, tj. že chybný signál bude identifikován jako normální.

3.3.3.2.1 PCA

V PCA jsou objektem průzkumu vlastní čísla kovariační matice vytvořené ze signálu s anomálií. V porovnání s ostatními příznaky se v tab. č. 1 neukazuje schopnost popsat šum a výsledky v popisu anomálie frekvence a anomálie koeficientu jsou průměrné.

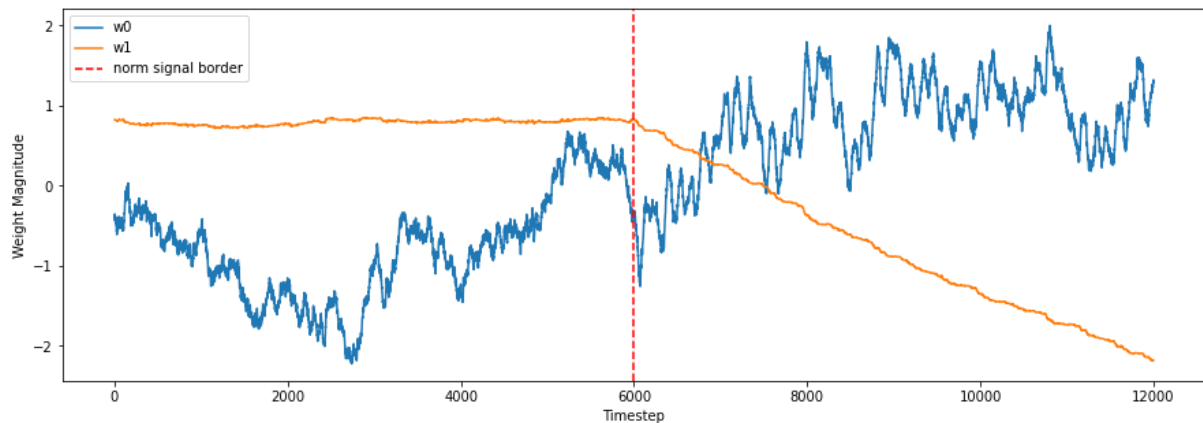
Největším problémem je výpočetní doba kovariační matice, která roste s množstvím vah, velikostí okna a vzorkovací frekvencí. Z tohoto důvodu nejsou příznaky PCA zvoleny do dalšího kola.

Tab. 1 – Skóre testu příznaku PCA

Z	Vzdálenost	Amplituda	Strategie
0.131118	0.182574	0.25	šum
0.231368	0.31429	0.5	šum
0.092343	0.11592	1	šum
0.386835	1.561602	2	šum
0.395806	1.947415	0.25	frekvence
0.563723	5.27409	0.5	frekvence
0.629019	10.967778	1	frekvence
0.63893	21.064787	2	frekvence
0.68188	21.335278	0.25	koeficient
0.770759	23.717764	0.5	koeficient
0.835891	29.707967	1	koeficient
0.766206	42.157275	2	koeficient

3.3.3.2.2 Příznaky korelace

Příznak korelace byl zvolen pro potenciál detekovat trendy na váhách modelu zjištěním jejich koeficientu korelace s jednou z referenčních funkcí, která reprezentuje hledaný trend. Referenčními funkcemi jsou přímka svírající s oběma rovinami počátku 45° a druhá mocnina této přímky. Ukázka případu vhodného korelace je zobrazena na obr. č. 18, kde dochází k lineárnímu poklesu váhy modelu w_1 v důsledku nečekané změny koeficientu. Jedná se o případ chyby popsany v [kapitole 3.3.1.3](#). Korelační koeficient této váhy s referenční rostoucí přímkou je roven -1.



Obr. 18 – Lineární pokles váhy modelu v důsledku změny hodnoty koeficientu lineární rovnice. Červenou linií je indikovaný vznik anomálie

Dle předpokladu, korelace ukázali v testu lepší vlastnosti při detekci skokových změn, u kterých se neuron přizpůsobuje náběhem bez výkyvů, jak je zobrazeno na obr. č. 18. Skokovou anomálií je změna koeficientu, na kterém příznak ukazuje v tab.č. 2 vyšší skóre než u ostatních anomálií. V porovnání s ostatními příznaky je ale skóre podprůměrné. Příznak korelace neukazuje schopnost detekce šumu a schopnost detekce frekvencí je také v porovnání s dalšími příznaky podprůměrná. Z těchto důvodů není příznak vybrán do dalšího kola testování.

Tab. 2 – Skóre testu příznaku korelace

Z	Vzdálenost	Amplituda	Strategie
0.063243	0.090336	0.25	šum
0.05786	0.081334	0.5	šum
0.051145	0.07173	1	šum
0.068281	0.096572	2	šum
0.221021	0.325296	0.25	frekvence
0.349008	0.52116	0.5	frekvence
0.447652	0.668059	1	frekvence
0.528012	0.782313	2	frekvence
0.595756	0.873717	0.25	koeficient
0.654203	0.948502	0.5	koeficient
0.705485	1.010822	1	koeficient
0.751059	1.063555	2	koeficient

3.3.3.2.3 Příznaky Abbottovo–Firestoneovo křivky

Pro získání Abbottovy-Firestonovy křivky neexistuje v knihovně python žádný algoritmus, proto k obdržení této křivky je sestaven vlastní algoritmus přiložený v příloze č. 3. Z této křivky se získává 5 koeficientů z průběhu přírůstků vah, jak je uvedeno v [kapitole 2.5.2.3](#) a do tabulky se přidává koeficient s nejlepší hodnotou Z-statistiky a vzdálenost této hodnoty.

Abbottova-Firestoneova křivka v tab. č. 3 ukazuje lepší výsledky v detekci anomálie šumu než v detekci ostatních anomálií a zároveň ukazuje v detekci šumu lepší výsledky než ostatní příznaky s výjimkou příznaků drsnosti. V porovnání s drsností má lepší hodnoty z-statistiky, ale menší vzdálenost. Největším záporem tohoto příznaku je velké množství nastavitelných parametrů. Kvůli podobným výsledkům s příznaky drsnosti, menší vzdálenosti a narušení podmínky minimalizace parametrů tento příznak nepostoupil do dalšího kola testování.

Tab. 3 – Skóre testu příznaku Abbottovy-Firestonovy křivky

Z	Vzdálenost	Amplituda	Strategie
0.533029	0.892183	0.25	šum
0.893918	2.496407	0.5	šum
0.94183	3.159482	1	šum
1.0131	6.360481	2	šum
0.805478	5.163372	0.25	frekvence
0.641504	4.1778	0.5	frekvence
0.682074	4.149182	1	frekvence
0.747527	7.336649	2	frekvence
0.769815	6.849251	0.25	koeficient
0.864231	7.076207	0.5	koeficient
0.920825	8.732164	1	koeficient
0.97934	12.91392	2	koeficient

3.3.3.2.4 FFT příznaky

Zvolený princip příznaku rychlé Fourierovo transformace je volba tří nejvýznamnějších frekvencí v anomálním i normálním průběhu přírůstků vah a jejich vzájemné porovnání. Pokud se frekvence obou signálů neshodují pak příznak vypisuje tyto nové frekvence, jinak je nulový. Významné frekvence jsou stanoveny šestinásobkem směrodatné odchylky ve frekvenčním grafu.

Kvůli principu příznaku FFT nelze příznak objektivně testovat hodnotami Z-statistiky a vzdálenosti. Normální data vykazují absenci nových frekvencí, a tak jsou průměr i směrodatná odchylka normálních dat nulové.

Posoudil jsem příznaky FFT na základě hrubých dat přiložených v příloze č. 2. Navržené řešení FFT neukázalo schopnost přesně detekovat anomální frekvenci, ale prokázalo odolnost vůči chybám prvního druhu a variabilitu hodnot v různých anomáliích. Proto je tento příznak vybrán pro testování v dalším kole.

3.3.3.2.5 Statistické příznaky

Statistickými příznaky jsou průměr, směrodatná odchylka a 75., 50. a 25. kvantil, které se zkoumají na průběhu vah. Statistické příznaky ukázaly nejlepší schopnost v detekci anomálie koeficientu s největší slabinou v detekci šumu.

Nesrovnalost mezi vzdáleností a z-statistikou v tab. č. 4 poukazuje na vysoký rozptyl hodnot příznaků při detekci anomálie. To může být způsobeno větším rozptylem vah při učení s velkým počtem odlehlých hodnot, které významně ovlivňují výsledek. Podobný rozptyl lze pozorovat u průběhu váhy w_0 na obr. č. 18.

Rozdíl mezi Z-statistikou a vzdáleností neovlivňuje první kolo výběru příznaků, protože tyto dvě statistiky nejsou dostatečné pro posouzení přínosu příznaku při klasifikaci. Jsou dostatečné na posouzení, jestli příznak popisuje anomálii.

V porovnání s ostatními algoritmy má nejlepší schopnost detekovat anomálii koeficientu, je jednoduše proveditelný s minimální výpočetní komplexností a nemá žádné nastavitelné parametry. Proto je sada statistických příznaků zvolena do dalšího kola.

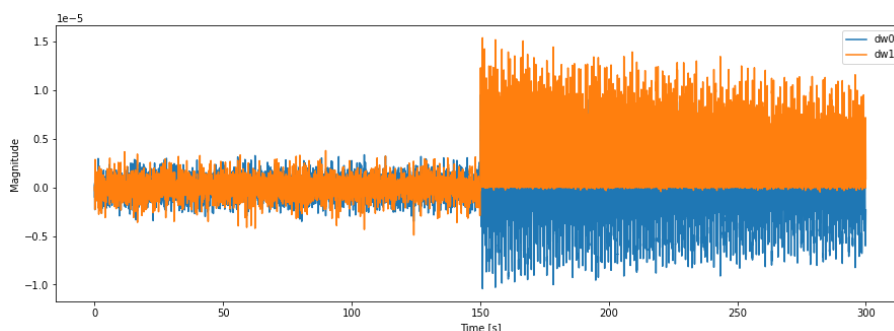
Tab. 4 – Skóre testu statistických příznaků

Z	Vzdálenost	Amplituda	Strategie
0.087466	0.124394	0.25	šum
0.167713	0.237009	0.5	šum
0.090014	0.121546	1	šum
0.337888	0.682865	2	šum
0.486831	7.543862	0.25	frekvence
0.604418	18.786928	0.5	frekvence
0.646516	38.243407	1	frekvence
0.647883	73.017951	2	frekvence
0.690684	73.769306	0.25	koeficient
0.779525	81.918135	0.5	koeficient
0.843677	102.488784	1	koeficient
1.071294	145.419092	2	koeficient

3.3.3.2.6 Příznaky drsnosti

Koeficienty drsnosti jsou další sérií příznaků orientovaných na detekci šumu. Pro popis přírůstku vah jsou zvoleny příznaky drsnosti R_a , R_q , R_dq , šikmost a špičatost, které jsou detailně popsány v [kapitole 2.5.2.6](#). Zvolené příznaky ukazují relativně dobrou schopnost detekce všech anomálií, ale především šumu.

Velký rozdíl mezi Z-statistikou a vzdáleností v tab. č. 5 způsobují obdobné důvody jako u statistických příznaků, kdy v případě přírůstku vah lze také pozorovat velký rozptyl s velkým množstvím odlehlých hodnot na obr. č. 19.



Obr. 19– Velký rozptyl přírůstku vah způsobený prudkou změnou koeficientu

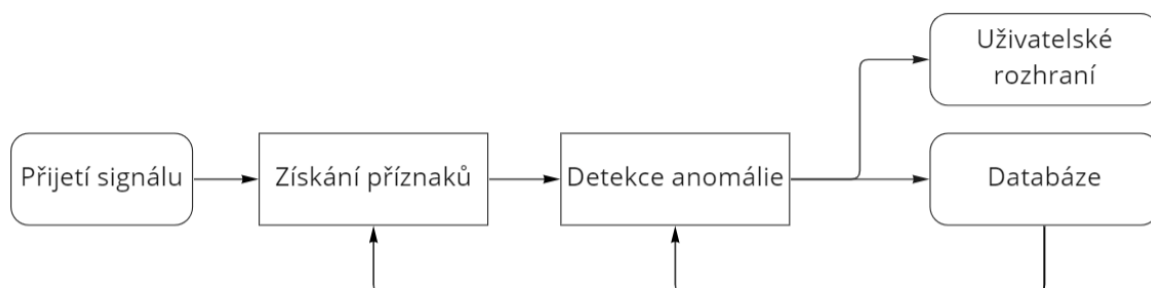
V porovnání s ostatními příznaky projevují drsnostní příznaky nejlepší skóre v rozpoznávání šumu (podobně jako Abbottovo-Firestonovo křivka) a nabývají dobrého skóre i u ostatních anomálií. Jejich kódové provedení jednoduché, výpočetně rychlé a nemá žádné nastavitelné parametry. Pro tyto důvody jsou drsnostní příznaky zvoleny do dalšího kola.

Tab. 5 – Skóre testu příznaků drsnosti

Z	Vzdálenost	Amplituda	Strategie
0.901471	2.838273	0.25	šum
1.049713	7.568527	0.5	šum
1.508438	17.75332	1	šum
1.55765	25.083739	2	šum
0.436641	21.718739	0.25	frekvence
0.469303	21.21327	0.5	frekvence
0.60537	26.112685	1	frekvence
0.679479	42.118714	2	frekvence
0.682701	39.105845	0.25	koeficient
0.761903	39.388803	0.5	koeficient
0.806328	46.563715	1	koeficient
0.731479	67.334962	2	koeficient

3.4 Kompletní popis návrhu

Na základě zvolených příznaků jsem navrhl systém se strojovým učením pro detekci a analýzu anomálií dynamického systému (zkráceně STUDNA), který se skládá z 5 kódových celků, viz obr. č. 20. V této kapitole bude popsán rozbor každého z těchto celků, jejich provedení a aplikace na zvoleném stroji.



Obr. 20– Datové schéma STUDNA (úroveň 0)

Přijetí signálu je část kódu, která se stará o získání signálu ze senzorů vybraného stroje a úpravu tohoto signálu do normalizované podoby.

Získání příznaků a detekce anomálie je soubor kódů, který normalizuje a analyzuje obdržený signál a detekuje, jestli ve stroji nedošlo k poruše. Jedná se o hlavní zaměření této práce a jsou subjektem dalšího kola testování. Detekce anomálie je následně popsána v [kapitole 3.5](#).

Databáze je kód, který ukládá a nahrává vytvořené prediktivní modely. Je podkladem pro klasifikační algoritmy a dlouhodobou predikci poruch.

Uživatelské rozhraní je kód, který umožňuje uživateli (obsluha stroje, manažer, administrátor, programátor) aplikovat daný algoritmus na zvolený stroj, provádět úpravy v nastavení detekce, sledovat stav stroje a sledovat predikci a analýzu poruch daného stroje.

3.4.1 Přijetí signálu

Kód pro přijetí signálu ze senzorů a jeho zpracování bude vždy individuální pro zvolený stroj. Proto v této práci kód pro přijetí signálu detailně nerozebírám. Přijetí dat ze zvoleného stroje je provedeno pomocí existujícího kódu v rozhraní MatLab použitého Vítem Pawlíkem v jeho práci [32]. Obdržená data jsou uložena ve formátu .csv a následně podávána do algoritmu STUDNA v offline režimu.

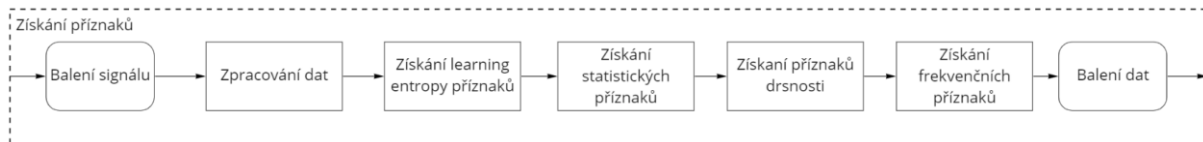
S ohledem na dobu a princip výpočtu je přijímání signálu dávkové v intervalech, které jsou při použití jednoho procesoru v rozestupu alespoň 18 vteřin. Dávkovým přijímáním je myšleno, že do algoritmu vstupuje nejen aktuální hodnota signálu, ale i 10 vteřin předešlých hodnot s konstantní periodou vzorkování.

Délka dávkového okna a frekvence vzorkování ovlivňují především dobu výpočtu a natrénování modelu. Délka dávkového okna by měla být dostatečně dlouhá na to, aby zachytila chování systému i v přítomnosti šumu. Frekvence vzorkování by měla být dostatečná na to, aby nedocházelo k aliasingu.

V rámci zpracování přijatého signálu musí dojít k odstranění NaN hodnot, odstranění duplikátů, převzorkování a případným dalším úpravám individuálním konkrétnímu stroji [20].

3.4.2 Získání příznaků

Získávané příznaky jsou zvoleny na základě testu v [kapitole 3.3.3](#) a jejich princip je detailně popsán v [kapitole 2.5](#). Dále je popsáno praktické provedení získání příznaků ze skutečného signálu. Jedná se o nejvíce rozsáhlý a klíčový kód této práce. Skládá se ze 7 částí, viz obr. č. 21: balení signálu, zpracování dat, získání příznaků pomocí learning entropy, statistických koeficientů, drsnostních koeficientů a frekvenčního rozboru a balení dat.

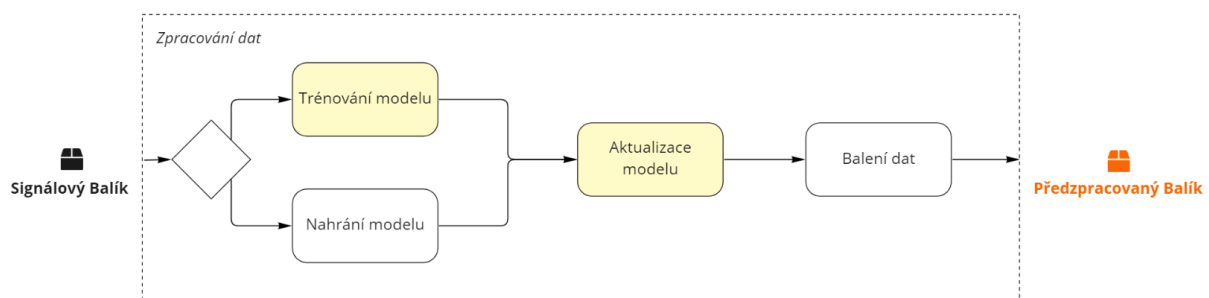


Obr. 21– Datové schéma získání příznaků (úroveň 1)

Balením signálu a balením dat je myšlena úprava datových vstupů nebo výstupů do normalizované podoby, která je použitelná dalším algoritmem na stejné úrovni datového schématu a zároveň srozumitelná pro uživatele. Výsledkem balení je python dictionary, která obsahuje klíče s názvem dat, které daný klíč nese. Balení signálu na konci algoritmu je součástí každého kódového celku v kódu získání příznaků.

3.4.2.1 Zpracování dat

Při zpracování dat dochází k natrénování vah neuronového modelu na normálních datech a poté k aktualizaci těchto vah na aktuálních datech, viz obr. č. 22. Pokud natrénovaný model již existuje v databázi, bude načten.



Obr. 22– Datové schéma zpracování dat (úroveň 2)

Při důkladném natrénování iterativního prediktivního modelu s dostatečným množstvím epoch se váhy systému stabilizují a omezeně pohybují okolo konkrétních hodnot. Pohyb je způsoben neschopností systému se naučit na šum v datech a zpravidla i neschopností konvergovat k lokálnímu minimum v důsledku zafixované hodnoty míry učení. To znamená, že přírůstek vah je při dalším trénování shora a spodu omezený a nezmenšuje se.

Aktualizací vah je myšlena úprava vah neuronu pomocí nových dat. Dochází tím k přetrénování modelu, při kterém jsou váhy z poslední epochy učení výchozími hodnotami. Při aktualizaci vah na novém signálu identického systému bude mít průběh přírůstku vah stejnou statistiku, jako v poslední generaci jeho trénování (za předpokladu, že jsou váhy stabilizované). Pokud ale v sledovaném systému došlo ke změně v jeho dynamice, pak se to při aktualizaci projeví na přírůstku vah dw , tj. bude platit, že

$$dw_{akt} > \max(|dw_{norm}|), \quad (1.26)$$

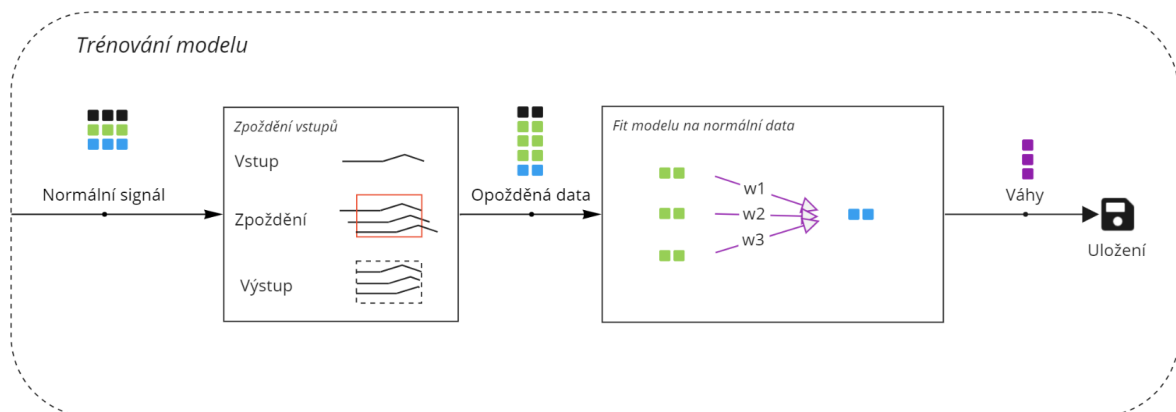
kde dw_{akt} jsou přírůstky vah aktuálního signálu a dw_{norm} normálního signálu.

Vstupem do algoritmu je vstupní signál stroje, výstupní signál stroje, čas, metadata a uživatelská data pro nastavení učícího algoritmu. Výstupem je python dictionary obsahující průběh snahy učení (absolutní hodnota přírůstků vah), průběh vah, průběh přírůstků vah a zkrácené vstupy, výstupy a čas v důsledku procesu tvorby příznaků pro prediktivní model popsaných v [kapitole 2.5.1](#) a více rozebírány dále. Kódové provedení zpracování dat je v příloze č. 4 s názvem kódu `batch_preprocessing`.

3.4.2.1.1 Trénování modelu

Volba struktury prediktivního modelu je individuální na základě konkrétního problému, viz [kapitola 2.3](#). Model v této práci je sestaven jako rekurentní neuron metodou gradientového učení s lineární strukturou.

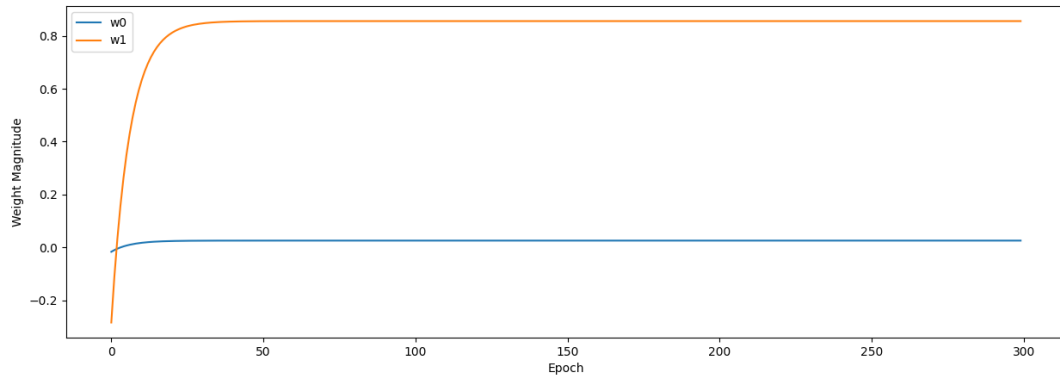
Před natrénováním modelu je možné zvolit dvě úpravy vstupních dat (viz. obr. č. 23), popřípadě zvolit pokračování bez úpravy. Úpravou dat je možnost zpoždění a rekurence, v důsledku kterých je nutné zmenšit velikost vstupního okna. Tyto úpravy se zpravidla používají pro zlepšení kvality predikce, ale ve STUDNA je hlavním účelem zlepšení detekce anomálie.



Obr. 23 – Datové schéma trénování modelu (úroveň 3)

Na skutečných datech ze Shakeru se ukázalo dostačující trénovat model i bez zpoždění vstupů, což značně zrychluje výpočetní rychlost v případě testování na velkém množství dat.

Pro predikci bylo zvoleno 300 epoch s mírou učení 1. Při tomto nastavení došlo ke konvergenci vah, viz obr. č. 24 a chyby predikce pod 10 % maximální hodnoty signálu. Toto nastavení lze automatizovat nastavením neomezeného počtu epoch a sledováním hodnoty přírůstku vah. Pokud je změna přírůstku vah zanedbatelná, pak je možné učení zastavit.



Obr. 24– Průběh vah prediktivního modelu v závislosti na množství učících epoch

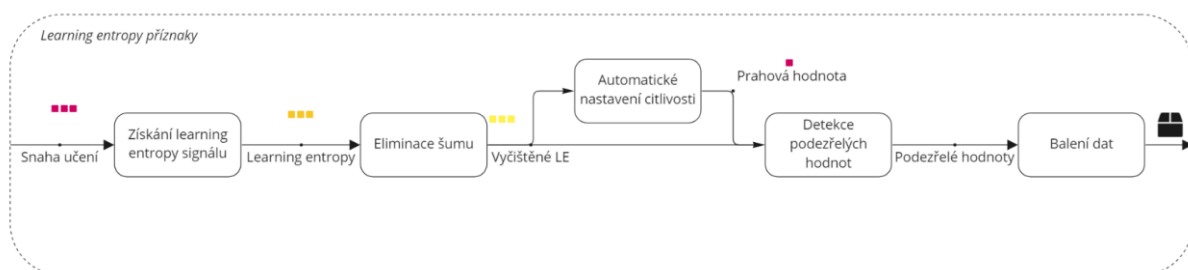
3.4.2.1.2 Aktualizace modelu

Schéma aktualizace modelu je obdobné jako na obr. č. 23 s rozdílem výstupu. Na výstupu z aktualizace jsou poskytnutá data uvedená v popisu zpracování dat a nové váhy modelu nejsou uloženy, tj. prediktivní model je po aktualizaci vrácen do stavu před aktualizací. Aktualizace v takovém případě slouží pouze pro získání výstupů. Na rozdíl od trénování modelu není cílem konvergence vah, ale naopak sledování první epochy, ve které jsou přírůstky vah největší.

Nastavením počtu epoch v aktualizaci vah je uživatel schopen ovlivnit citlivost modelu. Aktualizace modelu v této práci byla testovaná pouze na první epoše, proto je to také doporučené nastavení.

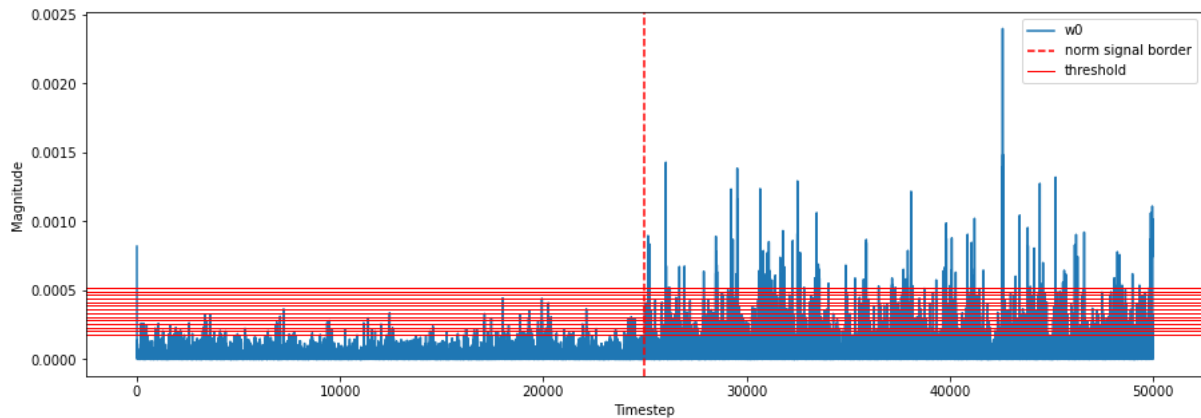
3.4.2.2 Získání learning entropy příznaků

Algoritmus získání příznaků learning entropy funguje na principu uvedeném v kapitole 2.2.2.2 s rozdílem úprav, které umožňují použití algoritmu i při detekci anomálií snižujících aktivitu neuronů na úkor přesné predikce časového okamžiku vzniku anomálie. Celkový postup je schématicky zobrazen na obr. č. 25.



Obr. 25 – Datové schéma získání learning entropy příznaků (úroveň 3)

Prvním krokem je obdržení snahy učení z poslední epochy trénování modelu na normálních datech a z poslední epochy aktualizace modelu na aktuálních datech. Oba průběhy jsou spojeny do navazujícího průběhu snahy učení individuální pro každou váhu, viz obr. č. 26.

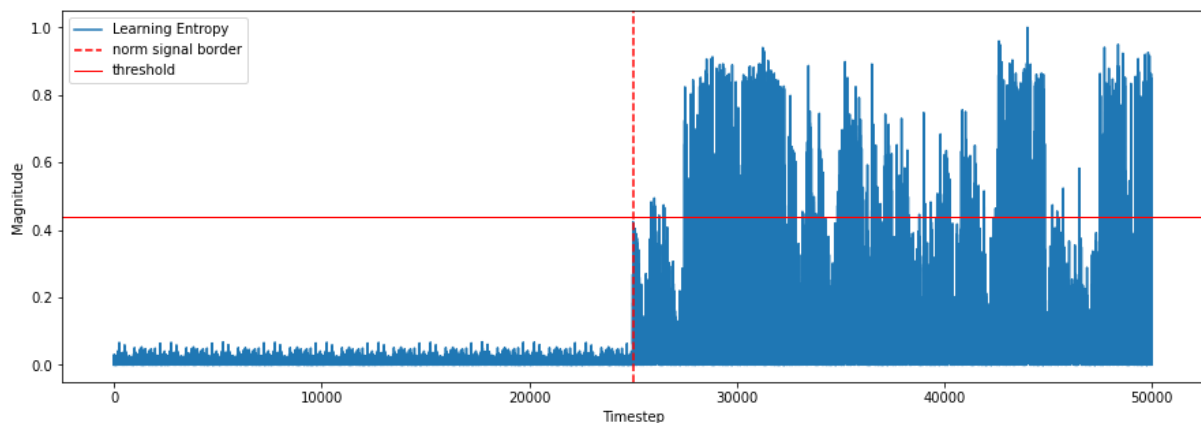


Obr. 26 – Snaha učení váhy w_0 pro natrénovaný prediktivní model na datech ze shakeru se zobrazenými optimalizovanými prahovými hodnotami (horizontálně červeně). Data bez anomálie před červenou čárkovanou linií, data s anomálií po červené čárkované linii

Poté je proveden výpočet learning entropy, dle vzorce (1.1), viz obr. č. 27. Výpočet se provádí na spojeném průběhu snahy učení normálního signálu a aktuálního signálu. Koeficient β je vypočten dle vzorce:

$$\beta_j = \mu_{|\Delta w_j|} + \alpha_j \cdot s_{|\Delta w_j|}, \quad (1.27)$$

kde $\mu_{|\Delta w_j|}$ je aritmetický průměr i -té snahy učení, α_j je prahový koeficient z rozsahu fixně stanovených koeficientů $\underline{\alpha}$ a $s_{|\Delta w_j|}$ je směrodatná odchylka pro i -tou snahu učení.



Obr. 27 – Learning entropy pro natrénovaný prediktivní model na datech ze shakeru se zobrazenou optimalizovanou prahovou hodnotou (horizontálně červeně). Data bez anomálie před červenou čárkovanou linií, data s anomálií po červené čárkované linii

Následně je vypočtena detekční prahová hodnota TFD (z angl. threshold for detection), pomocí které jsou detekovány podezřelé výkyvy v learning entropy. Je vypočtena na obdobným způsobu jako prahové hodnoty β :

$$TFD = \mu_{LE} + c \cdot s_{LE}, \quad (1.28)$$

kde μ_{LE} je aritmetický průměr LE, c je fixně nastavený prahový koeficient a s_{LE} je směrodatná odchylka LE.

Prahové koeficienty jsem nastavil ve výchozí formě na $\alpha = \langle 2, 5 \rangle$ s krokem 0.5 a $c = 3$. Toto nastavení ukázalo dobré výsledky při detekci na simulovaných datech, ale ne při testu na skutečných datech. Pro skutečný mechanismus jsem provedl genetickou optimalizaci těchto koeficientů a jednou z optimalizovaných možností (viz příloha č. 5) jsou hodnoty $\alpha = \langle 1.4, 4.4 \rangle$ s krokem 0.1 a $c = 3$, které značně zlepšili výsledky predikce, více v [kapitole 3.5.1](#). Prahy vytvořené z optimalizovaných hodnot jsou zobrazeny na obr. č. 26, 27.

Na rozdíl od postupu uvedeného v teorii, po detekci podezřelých hodnot LE dochází k sečtení celkového počtu podezřelých hodnot, které se objevily v oblasti normálního signálu a odděleně, které se objevily v oblasti aktuálního signálu.

Množství chybných detekcí pomocí learning entropy bude statistický stejný v normálním i aktuálním signálu za předpokladu, že oba signály vychází z identického dynamického systému. Proto je možné nastavit prahovou hodnotu počtu podezřelých hodnot, které se nepovažují za anomálie a tím řídit citlivost learning entropy a zmírnit důraz na přesnost zvolených koeficientů α, c .

V případě, že v důsledku anomálie dojde k snížení snahy učení, pak bude obsahovat aktuální signál menší množství podezřelých hodnot než normální signál a bude možné detekovat i snížení snahy učení.

3.4.2.3 Získání statistických příznaků

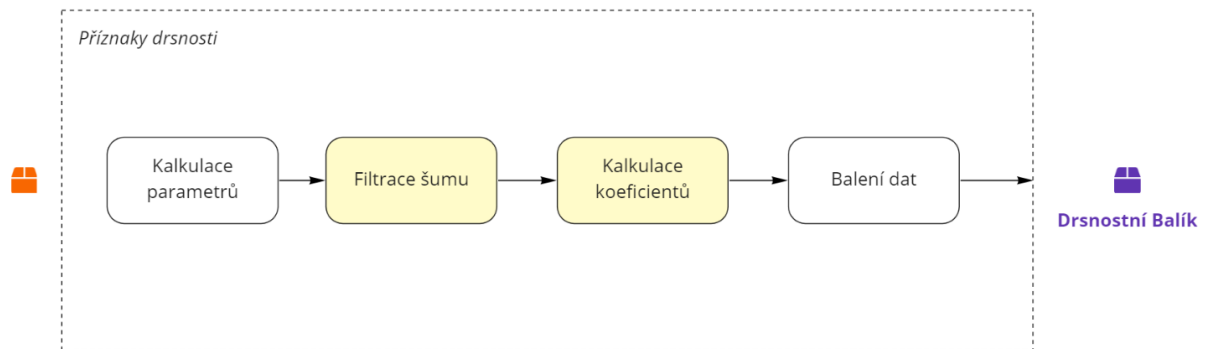
Rychlý a jednoduchý algoritmus, který slouží především pro rozšíření popisu signálu pro klasifikaci. Vstupem do kalkulace statistik jsou váhy z poslední epochy prediktivního modelu pro normální a aktuální signál. Výstupem je průměr, směrodatná odchylka a 75., 50. a 25. kvantil.

3.4.2.4 Získání příznaků drsnosti

Na rozdíl od statistických příznaků je pro získání příznaků drsnosti potřeba odfiltrovat z vah šum a následně jej analyzovat, viz schéma na obr. č. 28. Provádí se výpočet směrodatné odchylky šumu, R_a , R_q , R_dq , šikmost a špičatost, detailněji popsané v [kapitole 2.5.2.6](#).

Pro filtraci šumu je použit PCA filtr, který jsem sestavil na základě metodiky popsané v [kapitole 2.5.2.1](#). Prvním krokem je transformace vektoru signálu na matici. Algoritmus jsem nastavil tak, aby nová matice byla složena s nejvíce rovnoměrným poměrem stran. Poté dojde k singulárnímu rozkladu matice a znovusložení z nastaveného počtu základních komponent.

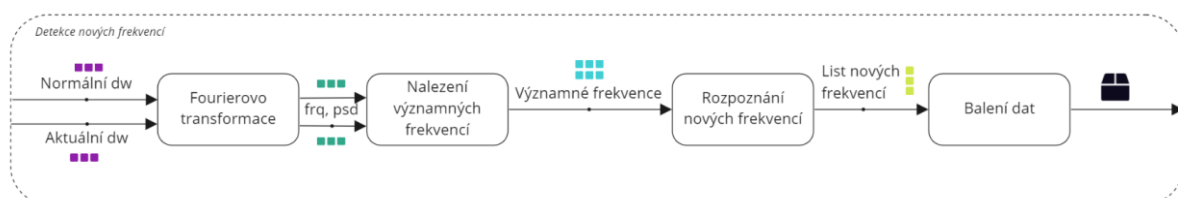
Počet základních komponent je nastaven na 30 % strany matice. V poslední fázi je signál znovu složen z vlastních vektorů s nejvyšší hodnotou vlastních čísel. Pro provedení PCA transformace je využit skript PCA z knihovny sklearn.decomposition. Kód filtru je k nalezení v příloze č. 6 pod názvem clear_noise_rough.



Obr. 28 – Datové schéma získání příznaků drsnosti (úroveň 2)

3.4.2.5 Získání frekvenčních příznaků

Příznaky jsou získávány z přírůstků vah. Postup jejich získání je schématicky znázorněn na obr. č. 29. Nejdříve se provede rychlá Fourierova transformace zvlášť na normálních datech a zvlášť na aktuálních datech. Následně jsou pomocí prahové hodnoty vybrány významné frekvence v daném signálu (s nejvyššími hodnoty PSD*). V poslední části kódu jsou tyto významné frekvence porovnány s frekvencemi v původním signálu a výstupem je vektor o omezené velikosti s novými frekvencemi.



Obr. 29 – Datové schéma detekce nových frekvencí (úroveň 3)

*PSD – Power Spectral Density (v čj. síla spektrální hustoty). Odpovídá energetické distribuci pro danou frekvenci. Čím vyšší je hodnota, tím významnější je frekvence.

3.5 Ověření navrženého systému

Ověření systému STUNDA je provedeno pomocí metrik definovaných v základních požadavcích v [kapitole 3.1](#) s cílem zjistit, jak navržený systém splňuje podmínky robustnosti a kompromisu, dostatečné citlivosti a chyby prvního druhu. Tyto metriky jsou zkoumané na [jednoduchém lineárním modelu](#) (model A), [modelu skutečného dynamického systému](#) (model B) a skutečných datech (model C) zvláště pro celkový systém i pro jednotlivé příznaky systému.

Na každém příznaku bude ohodnocena jeho detekční přesnost, velikosti chyb a klasifikační přesnost. Detekční přesnost se stanoví jako počet správně označených dat vůči celkovému počtu dat. Velikostmi chyb se měří chyba prvního a druhého druhu, tj. jaké procento normálních dat bylo při detekci označeno jako anomální (falešně pozitivní), a naopak jaké procento anomálních dat bylo označeno jako normální (falešně negativní). Klasifikační přesnost se stanovuje jako množství správně klasifikovaných dat vůči celkovému počtu dat.

Pro jednoduchý model a model skutečného systému jsou vygenerovány vzorky 30s signálu s vzorkovací frekvencí 2 kHz. Ve vytvořených vzorcích převažují vzorky normálního signálu s menším zastoupením anomálního signálu. Příloha č. 8 poskytuje detailní popis množství dat a jejich zastoupení u klasifikace a detekce. Data jsou rozdělena na 60 % trénovacích dat a 40 % testovacích tak, aby měly přesně stanovené procentuální zastoupení anomálií a normálních hodnot s náhodným vzorkem dat.

Anomálie je vnášena do jednoduchého lineárního modelu způsobem, který je detailně popsán v [kapitole 3.3.3.2](#). U modelu skutečného systému jsou pro vnesení anomálie zvoleny koeficienty L, R, ks, cs, kc, cc dle obr. č. 11. Anomálie, které jsou na tyto koeficienty aplikované byly zvoleny náhodně v amplitudě 30% původní hodnoty koeficientu. Popis parametrů, zvolené strategie anomálie a vliv změny těchto parametrů na výsledný signál je v příloze č. 7. Na skutečném modelu jsou časování měření a vnesení anomálie popsány v [kapitole 4.2.2](#).

Detekce signálu je prováděna dvěma metodikami:

Prahová predikce. Jedná se o nastavení horního a spodního prahu příznaků normálního signálu a anomálie je detekovaná, pokud alespoň jeden z příznaků překračuje nebo podkračuje nastavený práh. Práh se nastavuje dle poznatků Chowa, viz [kapitola 2.2.1.1](#) ze zaznamenaného množství dat normálního signálu. Pro stanovení prahu jsou použity 4 směrodatné odchylky daného příznaku a algoritmicky zajištěno zvýšení této hladiny, pokud by došlo k objevu falešně pozitivních hodnot na trénovaném signálu.

Logistická predikce je predikce dat pomocí klasifikátoru popsaného v [kapitole 2.4](#). Principem je naučení klasifikátoru na anomálních a normálních datech a kategorizování dat do dvou tříd: normální, s anomálií. U takového algoritmu se očekává vyšší přesnost, ale hlavní nevýhodou při použití v praxi je potřeba velkého množství dat s anomálií.

Klasifikace je prováděna pomocí logistické regrese rozšířenou o metodu one-vs-all umožňující klasifikaci pro více tříd, viz [kapitola 2.4.2](#).

Pro kalibraci prahové predikce jsou trénovací data pouze normálním signálem bez anomálií. Pro logistický model je součástí trénovacích dat i data s anomáliemi, jak je zobrazeno v příloze č. 8.

V tabulkách vyhodnocení příznaků jsou dále uvedeny úpravy signálu. Pro všechny testy byly testovány dva typy příznaků. Jedním je čistá hodnota příznaků daného signálu bez žádného dalšího zpracování. Druhým je poměr aktuálních příznaků a příznaků referenčních normálních dat, na kterých byl naučen prediktivní model (levá strana dat na obr. č. 27). Kvůli zamezení chyb nekonečna jsou nulové hodnoty příznaků nastaveny na velmi nízké číslo, které se v příznacích nevyskytuje (v daném případě $1e-5$). To umožňuje předejít inf hodnotám a poskytnout klasifikátoru data k hodnocení.

Uvedené hodnoty v tabulkách vyhodnocení mají směrodatnou odchylku pod 5% pokud není uvedeno jinak.

3.5.1 Learning Entropy příznaky

V testu ukázal příznak learning entropy vysokou odolnost vůči falešně pozitivním chybám při zachování vysoké přesnosti predikce. V prahové predikci na jednoduchém modelu (A) v tab. č. 6 dosahuje v porovnání s ostatními příznaky nejlepších výsledků přesnosti. Největší problém se projevuje v rozpoznávání anomálií 50% změny struktury šumu. V logistické regresi se dosahuje podprůměrných výsledků, ale pouze na základě learning entropy je možné klasifikovat až okolo 60 % dat a predikovat až 97.8 % chyb.

Z testu na reálných datech a datech simulovaného modelu je přesnost prahové predikce modelu pod 77%. Z detailnějšího průzkumu dat simulovaného modelu jsem zjistil, že model nebyl schopen detekovat chyby koeficientů L, R, kc a ks, jejichž společnou vlastností je nízký projev anomálie na výsledném průběhu (viz příloha č. 7). Z důvodu podezření, že snížení výkonu je důsledkem nastavení parametrů algoritmu jsem provedl optimalizaci genetickým algoritmem pomocí 9tí datových vzorků skutečného signálu (1 normální a 8 pro anomálii svislého a horizontálního vodič) a individuální snížení prahu na 2 násobek směrodatné odchylky. Po optimalizaci ukazuje algoritmus výrazně vyšší přesnost u skutečných dat (nejvyšší ze všech příznaků), při snížení falešně pozitivních výsledků a snížení směrodatné odchylky výsledků z 20.1% na 6%.

Oproti drsnostním a statistickým příznakům vykazuje v testech tento příznak horší schopnost detekovat nízké amplitudy chyb. To lze kompenzovat optimalizací nastavení parametrů, přičemž poté tento příznak vykazuje vysokou přesnost při zachování minimálního množství falešně pozitivních chyb. Zásadní nevýhodou tohoto algoritmu je množství nastavitelných parametrů.

Tab. 6 – Přesnost predikce příznaků learning entropy

Model	Test	Zpracování	Přesnost	Falešně pozitivní	Falešně negativní
A	Prahová predikce	Bez	87.2 %	0 %	12.8 %
		Poměr	94.5 %	0 %	5.5 %
	Logistická predikce	Bez	88.7%	7 %	4.3 %
		Poměr	83.4%	4.8 %	11.8 %
	Klasifikace	Bez	60.3 %	-	-
		Poměr	60.6 %	-	-
B	Prahová predikce	Bez	52.4 %	0.0 %	47.6 %
		Poměr	52.4 %	0.0 %	47.6 %
	Logistická predikce	Bez	90.5 %	7.7 %	1.8 %
		Poměr	91.6 %	8.3 %	0.0 %
	Klasifikace	Bez	60.7 %	-	-
		Poměr	65.1 %	-	-
C	Prahová predikce	Bez	77.7 %*	1.0 %	21.3 %
		Poměr	74.6 %*	1.6 %	22.2 %
		Optimalizace	90.1 %	0.8 %	8.3 %
	Logistická predikce	Bez	97.8 %	1.1 %	1.1 %
		Poměr	76.6 %	1.5 %	21.9 %
	Klasifikace	Bez	83.0 %	-	-
		Poměr	64.2 %	-	-

*velmi vysoký rozptyl hodnot (+- 20.1 %)

3.5.2 FFT příznaky

Příznaky rychlé Fourierovy transformace dosahují oproti ostatním příznakům relativně nízkého množství falešně pozitivních hodnot, a to především při prahové predikci. Z hlediska klasifikace tento algoritmus ukázal potenciál samostatně klasifikovat změny v signálu, ale především při klasifikaci s ostatními příznaky zlepšuje celkovou přesnost.

Nízká přesnost prahové predikce u modelu B v tab. č. 6 signalizuje absenci frekvenčně založené anomálie v simulovaném modelu.

Na FFT nebyla testována metoda poměru kvůli výchozím nulovým hodnotám.

Tab. 7 – Přesnost predikce příznaků FFT

Model	Test	Zpracování	Přesnost	Falešně pozitivní	Falešně negativní
A	Prahová predikce	Bez	78.8 %	0.9 %	19.3 %
		Poměr	-	-	-
	Logistická predikce	Bez	92.0 %	4.5 %	3.5 %
		Poměr	-	-	-
	Klasifikace	Bez	74.1 %	-	-
		Poměr	-	-	-
B	Prahová predikce	Bez	28.6 %	0.0 %	71.4 %
		Poměr	-	-	-
	Logistická predikce	Bez	91.7 %	8.3 %	0 %
		Poměr	-	-	-
	Klasifikace	Bez	50 %	-	-
		Poměr	-	-	-
C	Prahová predikce	Bez	82.3 %	0.4 %	17.3 %
		Poměr	-	-	-
	Logistická predikce	Bez	97.5 %	1.7 %	0.8 %
		Poměr	-	-	-
	Klasifikace	Bez	86.6 %	-	-
		Poměr	-	-	-

3.5.3 Statistické příznaky

Statistické příznaky dosahují relativně vysoké přesnosti při nízkém počtu falešně pozitivních hodnot, především v prahové predikci. V logistické predikci mají druhé nejlepší místo po drsnostních příznacích.

Dosažení nízkých výsledků u prahové predikce skutečného modelu (C) je způsobeno stejnou příčinou jako u learning entropy, kde by výsledky výrazně zlepšuje individuální prahová hodnota, viz model C, prahová predikce se zpracováním změny prahu v tab. č. 7.

Tab. 8 – Přesnost predikce statistických příznaků

Model	Test	Zpracování	Přesnost	Falešně pozitivní	Falešně negativní
A	Prahová predikce	Bez	83.5 %	6.4 %	10.1 %
		Poměr	83.6 %	6.3 %	10.1 %
	Logistická predikce	Bez	92.8 %	5.2 %	2.0 %
		Poměr	92.9 %	5.5 %	1.6 %
	Klasifikace	Bez	59.2 %	-	-
		Poměr	66.4 %	-	-
B	Prahová predikce	Bez	88.1 %	0.0 %	11.9 %
		Poměr	88.1 %	0.0 %	11.9 %
	Logistická predikce	Bez	100 %	0.0 %	0.0 %
		Poměr	100 %	0.0 %	0.0 %
	Klasifikace	Bez	83.3 %	-	-
		Poměr	83.3 %	-	-
C	Prahová predikce	Bez	50.4 %	0.0 %	49.6 %
		Poměr	50.5 %	0.0 %	49.5 %
		Změna prahu	69.0 %	3.3 %	27.7 %
	Logistická predikce	Bez	82.8 %	8.7 %	8.5 %
		Poměr	81.6 %	8.5 %	9.9 %
	Klasifikace	Bez	74.6 %	-	-
		Poměr	73.9 %	-	-

3.5.4 Drsnostní příznaky

Drsnostní příznaky se ukázali jako zásadní pro logistickou regresi a klasifikaci signálu. Při prahové predikci mají vysoké množství falešně pozitivních hodnot při průměrné přesnosti. Pouze na základě příznaků drsnosti je možné klasifikovat až 92% anomálií a jedná se tak o nejvýznamnější z příznaků pro klasifikaci.

Z tab. č. 9 je také patrné zlepšení poměrovým zpracováním u prahové predikce. Zatímco žádný jiný příznak neprokázal korelaci se změnou zpracování, příznaky drsnosti tuto korelaci ukazuje a jsou důvodem proč test celkového systému dosahuje v prahové predikci lepších výsledků při poměrovém zpracování.

Usoudil jsem, že to může být způsobeno změnou prahové hodnoty v důsledku zvýšení množství odlehlých hodnot v důsledku dělení hodnotou blížíící se nule, ale při iterativním zvyšování a snižování prahové hodnoty na signálu bez zpracování se přesnost prahové predikce pouze zhoršila. Při detailnějším průzkumu jsem zjistil, že při zpracování příznaku drsnosti byl nedopatřením použit jiný princip zpracování dat:

$$p_{zprac} = \frac{p_{act} - p_{ref}}{\min(p_{act}, p_{ref})}, \quad (1.29)$$

kde p_{zprac} je zpracovaný příznak, p_{act} je příznak aktuálního signálu a p_{ref} příznak normálního referenčního signálu.

Tab. 9 – Přesnost predikce příznaků drsnosti

Model	Test	Zpracování	Přesnost	Falešně pozitivní	Falešně negativní
A	Prahová predikce	Bez	86.2 %	6.4 %	7.3 %
		Poměr	94.5 %	2.8 %	2.8 %
	Logistická predikce	Bez	94.7 %	3.4 %	1.9 %
		Poměr	93.8 %	3.0 %	3.2 %
	Klasifikace	Bez	79.9 %	-	-
		Poměr	74.0 %	-	-
B	Prahová predikce	Bez	71.4 %	28.6 %	0.0 %
		Poměr	86.7 %	4.3 %	9.0 %
	Logistická predikce	Bez	100.0 %	0.0 %	0.0 %
		Poměr	100.0 %	0.0 %	0.0 %
	Klasifikace	Bez	91.4 %	-	-
		Poměr	91.1 %	-	-
C	Prahová predikce	Bez	80.6 %	19.4 %	0.0 %
		Poměr	88.3 %	5.3 %	6.4 %
	Logistická predikce	Bez	98.0 %	0.6 %	1.4 %
		Poměr	98.8 %	0.5 %	0.7 %
	Klasifikace	Bez	91.0 %	-	-
		Poměr	92.1 %	-	-

3.5.5 Celkový test

Celkový test při využití všech příznaků dokáže predikovat chybu na skutečném signálu (model C v tab. č. 10) pomocí prahové predikce s přesností 92 % \pm 0.1 %, ale s vysokým poměrem falešně pozitivních hodnot, až 6.3 %. Schopnost kategorizace anomálie je dosažena s přesností 95.7 %.

Pro zlepšení přesnosti predikce i klasifikace doporučuji rozdělit příznaky, které budou použity pro detekci a které budou použity pro klasifikaci. Například pro snížení falešně pozitivních hodnot prahové predikce při zachování přesnosti lze využít samotný příznak learning entropy nebo především odebrat příznak drsnosti, který značně přidává na množství falešně pozitivních hodnot.

Další možností zlepšení výsledků prahové predikce je individualizování nastavení prahových hodnot. Při testu je prahová hodnota nastavená jako 4 násobek směrodatné odchylky daného příznaku. Při testu bylo zjištěno, že zatímco snížení násobku směrodatné odchylky zhoršuje výsledky většiny příznaků tak zlepšuje výsledky learning entropy. Takový přístup by zvýšil množství uživatelem nastavitelných hodnot a v případě automatizace zvýšil komplexnost kalkulace.

Dalším případným zlepšením přesnosti je využití optimalizačních algoritmů na příznaky s nastavitelnými parametry – learning entropy a FFT.

Vliv zpracování na výslednou přesnost algoritmu je dan pouze příznakem drsnosti.

Tab. 10 – Přesnost predikce systému STUDNA

Model	Test	Zpracování	Přesnost	Falešně pozitivní	Falešně negativní
A	Prahová predikce	Bez	89.0 %	7.3 %	3.7 %
		Poměr	96.3 %	2.8 %	0.9 %
	Logistická predikce	Bez	94.4 %	3.4 %	2.2 %
		Poměr	96.1 %	2.0 %	1.9 %
	Klasifikace	Bez	85.0 %	-	-
		Poměr	84.1 %	-	-
B	Prahová predikce	Bez	71.4 %	28.6 %	0.0 %
		Poměr	86.6 %	3.8 %	9.5 %
	Logistická predikce	Bez	100.0 %	0.0 %	0.0 %
		Poměr	100.0 %	0.0 %	0.0 %
	Klasifikace	Bez	91.3 %	-	-
		Poměr	91.3 %	-	-
C	Prahová predikce	Bez	80.6 %	19.4 %	0.0 %
		Poměr	92.0 %	6.3 %	1.6 %
	Logistická predikce	Bez	99.9 %	0.1 %	0.0 %
		Poměr	99.5 %	0.2 %	0.3 %
	Klasifikace	Bez	95.7 %	-	-
		Poměr	94.6 %	-	-

Pro porovnání účinnosti jednotlivých příznaků vůči sobě je třeba stanovit základní kvalitativní podmínku, kterou všechny příznaky musí splnit a poté porovnat výsledky jejich přesnosti. V tab. č. 11 se nachází přesnosti jednotlivých příznaků s individuálním nastavením prahové hodnoty, tak aby jejich množství falešně pozitivních detekcí bylo pod 0.1 %.

Názvy sloupců reprezentují příznaky learning entropy, FFT, statistické příznaky, drsnostní příznaky a také modifikované drsnostní příznaky. Při práci s drsnostními příznaky jsem si všiml, že šikmost a špičatost vykazují relativně vysoký rozptyl, proto jsem v rámci vyzkoušení odebral z drsnostních příznaků. Důsledkem toho bylo navýšení přesnosti prahové predikce o 49 % při konst. množství falešně pozitivních hodnot, ale snížení přesnosti klasifikace o 2.6 %.

Výsledky LE jsou oproti ostatním příznakům dominující při požadavku falešně pozitivních hodnot < 0.1 %. Přestože se zdá, že modifikované drsnostní příznaky jsou přesností velmi blízké příznakům LE, nebyly otestované i na jiných typech anomálií, které nemusí být schopné detekovat.

Tab. 11 – Porovnání účinnosti příznaků při chybě prvního druhu < 0.1 %

	LE	FFT	Statistické	Drsnostní	Drsnostní*
Nastavený koeficient prahu	2.5	5.2	3	21	3.8
Přesnost prahové predikce	72.8 %	51.5 %	58.5 %	23.5 %	72.5 %
Přesnost klasifikace	83.0 %	86.6 %	74.6 %	92.1 %	89.5 %

3.6 Uživatelské rozhraní

3.6.1 Knihovna python

Základní vytvořené uživatelské rozhraní je skrz knihovnu python a umožňuje programátorům aplikovat algoritmus na stroj, rozšiřovat a měnit jednotlivé moduly. Moduly jsou jednotlivé sekce kódu se specifickým účelem, kterých je v systému 8: `batched_tools`, `mtools`, `ptools`, `calibration`, `data_logger`, `detection_tools`, `mlearning`, `shaker_modul`, viz obr. č. 30.

Využité knihovny: `numpy`, `warnings`, `pickle`, `pandas`, `datetime`, `matplotlib`, `os`, `csv`, `sklearn`, `itertools`, `scipy`

STUDNA			
batched_tools	ptools	calibration	shaker_modul
<code>def copanda</code> <code>def batch_preprocessing</code> <code>def batch_suspint</code> <code>def batch_suspfreq</code> <code>def batch_suspmat</code> <code>def batch_susprough</code>	<code>def change_order</code> <code>def learning_entropy_1</code> <code>def learning_entropy_2</code> <code>def buffer</code> <code>def debuffer</code> <code>def clear_noise_rough</code> <code>def fourier_transformation</code> <code>def find_fft_peaks</code> <code>def new_fft_peaks</code> <code>def return_roughness</code> <code>def calculate_Ra</code> <code>def calculate_Rq</code> <code>def calculate_Rdq</code> <code>def calculate_Skew</code> <code>def calculate_Kurt</code>	<code>def calibrate_whole_system</code> <code>def apply_calibration</code> <code>def calibrate_threshold</code> data_logger <code>def data_logger</code> detection_tools <code>def original_method</code> <code>def ratio_method</code> mlearning <code>class LNU</code> <code>def untouched_inputs</code> <code>def delayed_inputs</code> <code>def delayed_inout</code>	<code>def prepare_data</code> <code>def matdf_to_datetime</code> <code>def check_timesample</code> <code>def filter_surrounding</code> <code>def take_care_of_duplicates</code> <code>def resample</code>
mtools			
<code>def suspicious_intervals</code> <code>def suspicious_frequencies</code>			

Obr. 30 – Modulové složení systému STUDNA a jejich funkcionální obsah

3.6.1.1 `batched_tools`

Modul zodpovědný za práci s normalizovaným přenosem dat. Pomoci funkce `copanda` umožňuje spustit systém kompletní pro získání příznaků z dat, nebo pomoci jednotlivých funkcí `batch_` umožňuje otestovat velké celky kódu. Jedná se o kód, který zodpovídá za 1. úroveň datového schématu zobrazeného na obr. č. 21.

Kompletní dokumentace této části kódu je v příloze č. 4.

3.6.1.2 `mtools`

Jedná se o nápomocný modul k `batched_tools`, který usnadňuje komunikaci mezi vybranými funkcemi v `batched_tools` a hlavními detekčními funkcemi v `ptools` a organizuje kód.

3.6.1.3 `ptools`

Je soubor dílčích funkcí pro získávání příznaků a dalších nápomocných funkcí. Přímé využití těchto funkcí se signálem získaným ze stroje je obtížné kvůli nutnosti zpracovat signál do

vhodné podoby. Proto pro komunikaci s funkcemi z ptools se využívá modulu `batched_tools` a `mtools`.

3.6.1.4 *calibration*

Modul zodpovědný za kalibraci dat pro prahovou predikci, jak je uvedeno v [kapitole 3.5](#). Kódové provedení je dostupné v příloze č. 11.

3.6.1.5 *data_logger*

Část kódu umožňující jednoduše ukládat tabulkové datové soubory. Slouží především pro testovací účely – ukládání signálu, prahových hodnot apod. Kódové provedení je dostupné v příloze č. 12.

3.6.1.6 *detection_tools*

Modul zodpovědný za techniky vylepšení detekce. Aktuálně zde obsahuje pouze dvě úpravy signálu, které byly zmíněny v [kapitole 3.5](#) – nezměněný signál (pouze vhodná úprava matice pro test) a metoda poměru. Kódové provedení je dostupné v příloze č. 13.

3.6.1.7 *mlearning*

Modul zodpovědný za iterační prediktivní modely použité k obdržení vah a nápomocné funkce pro vytvoření příznaků dle [kapitoly 2.5.1](#). Kódové provedení je dostupné v příloze č. 14.

3.6.1.8 *shaker_modul*

Modul, který se bude měnit dle stroje, na kterém bude systém použit. Kód v tomto modulu je zodpovědný za zpracování dat získaných z shakeru uvedeného v popisu zvoleného stroje. Kódové provedení je dostupné v příloze č. 15.

3.6.2 Návrh front-end

S cílem vytvořit systém použitelný pro společnosti jsem navrhnul uživatelské rozhraní, které by bylo mohlo být využíváno ve praktickém nasazení. Základními uživatelskými rolemi tohoto rozhraní jsou:

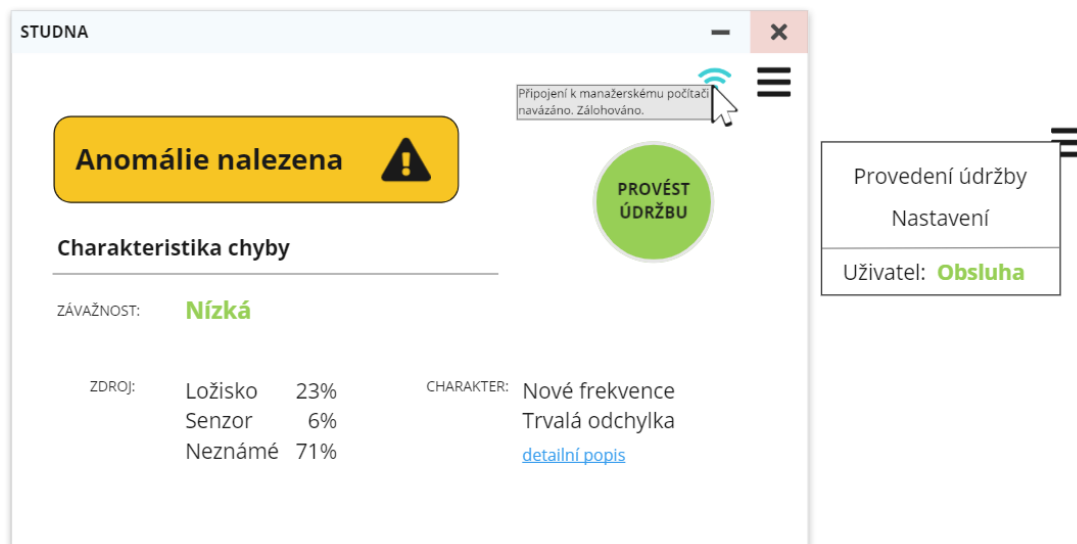
Manažér. Má přístup k historickým údajům odstávek a druhu poruch. S daty interaguje v interaktivním prostředí rozšířeným o interaktivní grafy. Ve svém rozhraní není schopen provádět změny, které by ovlivnily funkcionalitu detekce chyby. V případě detekce poruchy stroje je uživatel upozorněn.

Obsluha. Má přístup k základnímu nastavení numerických filtrů a parametrů pro detekci chyby. Uživatel vidí aktuální data a je upozorněn v případě detekce poruchy s detailním popisem vlastností poruchy (vzniklé frekvence, vznik trvalé regulační odchylky atp.)

Admin má přístup k úpravě front end kódu a změně nastavení, které zásadně ovlivňují princip detekce chyby.

3.6.2.1 Operátor + Admin

Výchozí domovská stránka na obr. č. 31 zobrazuje stav stroje. V případě objevení anomálie podává popis této anomálie, její závažnost dle vlivu na výstupy stroje nebo dle historických údajů (pokud došlo ke shodě s již existující chybou), pravděpodobné určení zdroje a charakter chyby. Nachází se zde tlačítko „Provést údržbu“, které otevře okno pro zadání názvu objevené chyby poté, co byla údržba provedena po naučení klasifikátoru a kalibrace algoritmu prahové predikce na nové výstupy po údržbě.



Obr. 31 – Domovská stránka STUDNA se zobrazením rozbaleného navigačního menu

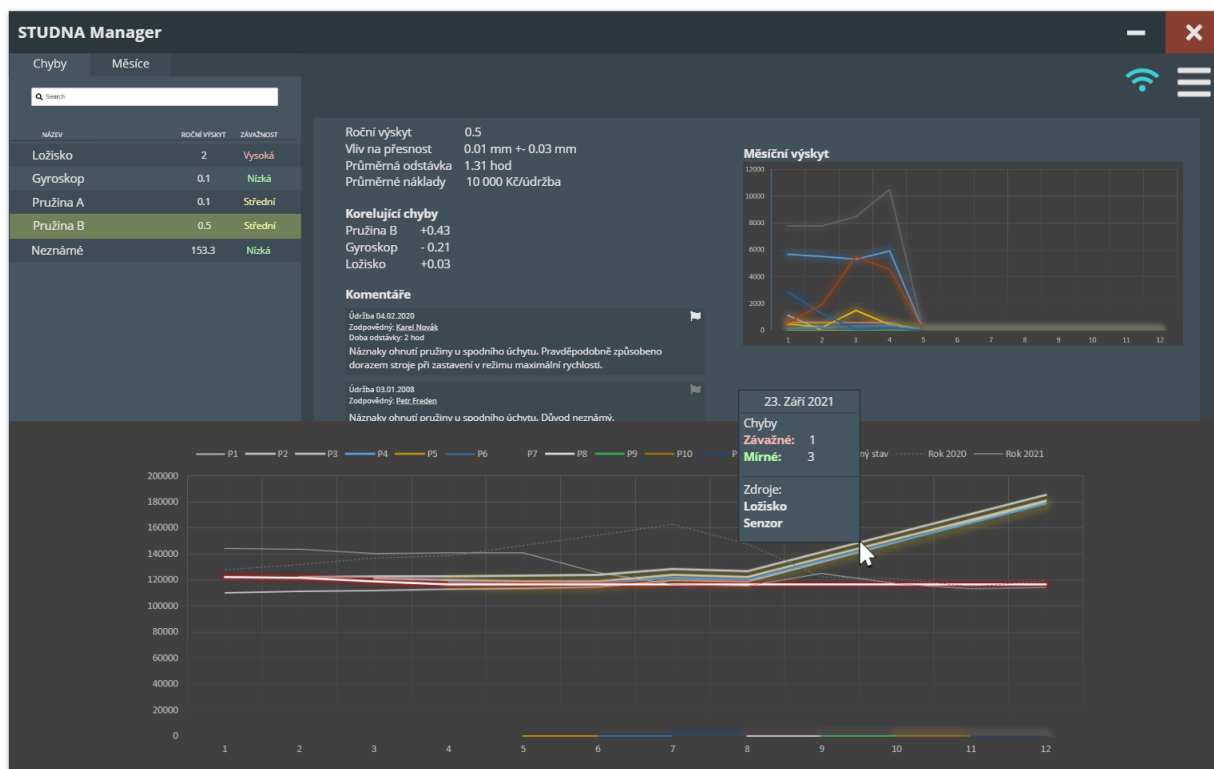
V okně *nastavení* je možnost nastavit všechny parametry STUDNA (viz obr. č. 32). Některá nastavení jsou povolena pouze uživateli Admin, ale stále viditelná pro ostatní uživatele. Po provedení změn se nabídne možnost uložit nebo zahodit změny. Změny se v systému projeví až po jejich uložení.



Obr. 32 – Ilustrativní stránka nastavení detekčního algoritmu systému STUDNA

3.6.3 Manažer

Okno STUDNA Manager na obr. č. 33 poskytuje náhled do detailních informací o chybě pro strategické rozhodování. Umožňuje detailní rozbor se zaměřením na konkrétní chyby nebo na konkrétní měsíce. K dispozici jsou základní informace jako je výskyt chyby, vliv chyby na přesnost stroje, průměrná doba odstávky, průměrné náklady na údržbu, ale i korelace s jinými chybami. Ke každé chybě je možné sledovat komentáře údržbářů a také si všechna data vizualizovat v grafech, které zobrazují náklady na údržbu v jednotlivých měsících a ročních obdobích.



Obr. 33 – Okno STUDNA Manager pro manažery

3.7 Celkové zhodnocení

Navržený algoritmus je schopen detekce anomálií s přesností 92 % \pm 0.1 % s 6.3 % falešně pozitivních hodnot. V rámci testu je navrženo řešení, které může zaručeně změnit přesnost na 90.1 % \pm 0.1 % s 0.8 % falešně pozitivních hodnot. Zároveň jsou poskytnuty další netestované návrhy na zlepšení přesnosti a snížení chyby prvního druhu. Tímto je splněné kritérium **robustnosti a kompromisu**.

Kritérium **jednotného škálování** je opatřeno snížením závislostí detekčního algoritmu na škálování. Pro detekci jsou porovnávány normální a aktuální signál, které jsou vždy škálovány společně s novým vstupem aktuálního signálu a poté vzájemně porovnávány. Všechny detekční prahy v algoritmu jsou závislé na směrodatné odchylce signálu a nejsou fixní.

Problematickým je množství nastavitelných parametrů detekčního systému, kterých je celkem 9. V následujícím seznamu parametrů systému jsou zeleně vyznačeny lehce automatizovatelné parametry.

Učící algoritmus

1. **Počet epoch**. Lze automatizovat za pomoci sledování ustálení vah.
2. **Počet updatu epoch**. Prakticky se využívá pouze 1 epochy updatu.
3. **Míra učení**. Lze automatizovat postupným zmenšováním gradientu až do překročení stanovené hranice chyby, nebo hodnoty změny vah.
4. **Technika učení**. V této práci byl použit lineární neuron metodou gradientního učení. Je možné využít i jiných metod, které dokážou krokově popisovat dynamiku systému. Zvolenou techniku učení není nutné měnit v závislosti na stroji.
5. Zpracování vstupních dat. Během testování práce bylo objeveno, že větší množství vah zlepšuje schopnost algoritmu detekovat chybu, ale dále tento jev nebyl zkoumán. Je možné tak zvýšit počet vstupů korelacemi, zpomalováním, rekurenci jak je popsáno v [kapitole 2.5.1](#). Nastavení zpracování vstupních dat je individuální každému stroji.

Learning entropy algoritmus

6. Prahový násobič. Stanovuje práh, při kterém je learning entropy podezřele vysoké,
7. Rozložení koeficientu bety. Klíčový prvek detekce pomocí learning entropy, který může značně ovlivnit kvalitu detekčních a klasifikačních výsledků na daném stroji. Nerovnoměrné rozložení prahových hodnot by mohlo značně zlepšit detekční schopnosti algoritmu, ale napasovat algoritmus danému problému a zvýšit komplexnost nastavení parametrů.

Frekvenční parametry

8. Počet frekvencí. Rozšířené množství frekvencí má potenciál zlepšit přesnost klasifikace algoritmu (založeno na odborném odhadu).

Detekční algoritmus

9. Prahové hodnoty. Prahové hodnoty lze do určité míry automatizovat v závislosti na jiné hodnotě jako je šum signálu nebo směrodatná odchylka při učení. Testování algoritmu poukázalo na důležitost individualizovat nastavování prahových hodnot každému příznaku pro zvýšení přesnosti predikce. To by mohlo značně zvýšit počet proměnných, ale dále z testu vyplývá potřeba omezit počet použitých příznaků. Proto by detekční algoritmus mohl po úpravách vyžadovat okolo 4 nastavitelných parametrů.

Další sérii parametrů přinese zavedení klasifikačního algoritmu do systému. Zpravidla je nastavení těchto algoritmů dáno doporučeními a tabulkami, případně je možnost využít samo organizačních klasifikátorů.

Z pohledu **generalizace** princip detekce anomálií ve STUDNA umožňuje odlišení všech dat, která nejsou součástí normálního signálu. Problém generalizace se týká především klasifikátorů, které nejsou schopné detekovat anomálii, na kterou nejsou naučeny.

Vliv změny počtu vah systému nebyl ověřen na přesnosti detektoru při stejném nastavení parametru. Je očekáváno, že se změnou počtu vah systému bude potřebné změnit parametry learning entropy a prahové detekce. Délka okna dat nemá na predikci značný vliv. Plné pochopení splnění kritéria **nezávislosti** systému vyžaduje další průzkum.

Z ověření přesnosti navrženého systému vyplynulo, že k naučení systému pro získání příznaku stačí 10 vteřinový záznam signálu. Pro následnou detekci na skutečném stroji je využito 17 minut signálu. Z hlediska množství dat lze navržený systém relativně rychle **adaptovat** novému signálu. Problematické z hlediska adaptace je správné nastavení všech parametrů a případná optimalizace.

Výpočetní doba všech příznaků a následná detekce chyby trvá okolo 18 vteřin, při výpočetním zařízení s 8 jádry je možná detekce anomálií přibližně každých 2.25 vteřin. V průběhu práce jsou poskytnuto několik návrhů na zrychlení výpočetní doby, a to především omezením příznaků detekce pouze na learning entropy. Další možností, která zásadně zrychlí výpočet je ukládání posledního okna dat a vypočtených vah. Díky využití jediné epochy pro aktualizaci prediktivního modelu to umožní model i příznaky aktualizovat na posledním kroku místo jejich úplnému přepočítání. V aktuálním provedení systém není schopen živé detekce anomálií, ale obsahuje oblasti, které lze upravit pro dosažení živé detekce. Vzhledem k účelu algoritmu – průběžná kontrola stroje pro prediktivní údržbu – není kontinuální detekce anomálie nutná ani doporučená. Detekční algoritmus může být používán v závislosti na počtu odpracovaných cyklů.

Zásadním problémem algoritmu je potřeba kalibrovat a detekovat anomálii na stroji za stejných podmínek nebo při úplném poznání vstupů. V případě shakeru pro detekci anomálií za provozu je důležité zahrnout do vstupů STUDNA váhu testovaného objektu a případně i polohu posunutí těžiště tělesa od osy armatury. Neúplná znalost vstupů při testování v provozu by zvýšila množství falešně pozitivních chyb nebo značně snížila přesnost detekce.

4 Závěr

V této práci byla nejdříve provedená rešerše problematiky detekce anomálií, na základě které byly sepsány dostupné detekční algoritmy, charakteristika možných anomálií a soupis požadavků, které se kladou na detekční algoritmus v praxi. Následně byla provedena rešerše možných příznaků, kterými lze signál charakterizovat a pomocí kterých lze detekovat anomálie. Na základě průzkumu byl zvolen hlavní detekční algoritmus – learning entropy a 7 příznaků.

V praktické části práce došlo nejdříve ke stanovení základních požadavků a očekávání od navrhovaného systému, návrhu strategie pro naplnění těchto požadavků a způsob jejich měření. Byl zvolen a popsán stroj, na kterém budou provedeny testy navrhovaného algoritmu a sestaven simulační model tohoto stroje. Následně došlo k otestování navržených příznaků a výběr nejmenší nutné sady příznaků, která projevila schopnost rozpoznat specifické anomálie a charakterizovat je. Tyto příznaky byly použity k návržení kompletního systému STUDNA (akronym pro systém se strojovým učením pro detekci a analýzu anomálií), který je schopen zpracovat signál přijatý ze stroje, charakterizovat jej pomocí příznaků a detekovat anomálie. Navržený systém byl poté důkladně otestován. Subjektem testování byla především kontrola přesnosti predikce, klasifikace a počet chyb prvního a druhého druhu. Systém byl navržen tak, aby byl uživatelsky přijatelný pro programátora a celé uživatelské rozhraní je koncipováno formou knihoven. Pro vyjádření představy použití tohoto kódu v praxi byly navrženy dvě uživatelská rozhraní, která by mohlo být možné použít obsluhou stroje, administrátorem a zvláště manažerem společnosti.

V závěru praktické části byl navržený systém STUDNA kompletně zhodnocen a posouzen na základně nastavených požadavků. Systém splnil všechny kladené požadavky s ohledem na cíl této práce a většinu kritérií zjištěných z rešerše. Problematická kritéria bylo kritérium nasazení v živém čase a kritérium množství parametrů. Pro splnění těchto kritérií byla navržena možná zlepšení algoritmu, která umožní splnění obou kritérií.

Výsledkem této práce je prototyp produktu pro predikci anomálii ve stroji. Práce poskytuje mnoho návrhů pro zlepšení přesnosti algoritmu, snížení chyb prvního druhu, zrychlení výpočtu a zautomatizování parametrů. Kód je napsán v přehledné podobě, kterou lze jednoduše upravit a rozšířit a je náležitě zdokumentován. Dalším prostorem pro rozvoj této práce je testování systému na různém množství vah, vyzkoušení jiného principu prediktivního modelu a klasifikátoru, pokračování v testech systému na různých strojích a zkoumání jeho limit, vytvoření navrhovaného uživatelského rozhraní a nalezení partnera, kterému bude toto řešení poskytnuto. To umožní plné pochopení potenciálu algoritmu, sběr zpětné vazby a vytvoření plně funkčního produktu.

5 Bibliografie

- [1] L. Beránek, „Průmyslová metrologie, 9. přednáška“, ČVUT, Praha, 2022.
- [2] M. Markou a S. Singh, „Novelty detection: a review—part 1: statistical approaches“, *Signal Process.*, roč. 83, č. 12, s. 2481–2497, pro. 2003, doi: 10.1016/j.sigpro.2003.07.018.
- [3] M. Markou a S. Singh, „Novelty detection: a review—part 2: neural network based approaches“, *Signal Process.*, roč. 83, č. 12, s. 2499–2521, pro. 2003, doi: 10.1016/j.sigpro.2003.07.019.
- [4] C. K. Chow, „On Optimum Recognition Error and Reject Tradeoff“, dub. 1969, Viděno: 12. červenec 2022. [Online]. Dostupné z: <https://dspace.mit.edu/handle/1721.1/6177>
- [5] D. Tax a R. Duin, „Outlier Detection Using Classifier Instability“, *Adv. Pattern Recognit.*, roč. 1451, čer. 1999, doi: 10.1007/BFb0033283.
- [6] L. Tarassenko, A. Nairac, N. Townsend, a P. Cowley, „Novelty detection in jet engines“, in *IEE Colloquium on Condition Monitoring: Machinery, External Structures and Health (Ref. No. 1999/034)*, dub. 1999, s. 4/1-4/5. doi: 10.1049/ic:19990187.
- [7] I. Dabbura, „K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks“, *Medium*, 10. srpen 2020. <https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a> (viděno 28. červenec 2022).
- [8] M. E. Hellman, „The Nearest Neighbor Classification Rule with a Reject Option“, *IEEE Trans. Syst. Sci. Cybern.*, roč. 6, č. 3, s. 179–185, čvc. 1970, doi: 10.1109/TSSC.1970.300339.
- [9] O. Harrison, „Machine Learning Basics with the K-Nearest Neighbors Algorithm“, *Medium*, 14. červenec 2019. <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761> (viděno 18. červenec 2022).
- [10] *kNN.12 Parzen windows, kernels and SVM*, (15. září 2015). Viděno: 18. červenec 2022. [Online Video]. Dostupné z: https://www.youtube.com/watch?v=e_TKljo62b8
- [11] S. Forrest, A. S. Perelson, L. Allen, a R. Cherukuri, „Self-nonsel self discrimination in a computer“, in *Proceedings of 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, kvě. 1994, s. 202–212. doi: 10.1109/RISP.1994.296580.
- [12] D. Dasgupta a S. Forrest, „Novelty detection in time series data using ideas from immunology“, *undefined*, 1996, Viděno: 18. červenec 2022. [Online]. Dostupné z: <https://www.semanticscholar.org/paper/Novelty-detection-in-time-series-data-using-ideas-Dasgupta-Forrest/26d15311d3384b2032d24aa5a28bdaaf406bb6de>
- [13] „What is Machine Learning?“, 6. červenec 2022. <https://www.ibm.com/cloud/learn/machine-learning> (viděno 28. červenec 2022).
- [14] I. Bukovsky, N. Homma, M. Cejnek, a K. Ichiji, „Study of Learning Entropy for Novelty Detection in lung tumor motion prediction for target tracking radiation therapy“, čvc. 2014. doi: 10.1109/IJCNN.2014.6889834.
- [15] I. Bukovsky, G. Dohnal, P. Steinbauer, O. Budik, K. Ichiji, a N. Homma, „Learning Entropy of Adaptive Filters via Clustering Techniques“, zář. 2020, s. 1–5. doi: 10.1109/SSPD47486.2020.9272138.
- [16] I. Bukovsky, W. Kinsner, a N. Homma, „Learning Entropy as a Learning-Based Information Concept“, *Entropy*, roč. 21, s. 166, úno. 2019, doi: 10.3390/e21020166.
- [17] Z. Šika, „Syntéza a optimalizace mechanických systémů“. Přednášky ČVUT, 2022.

- [18] „Koeficient determinace", *Wikipedie*. 27. leden 2022. Viděno: 9. srpen 2022. [Online]. Dostupné z:
https://cs.wikipedia.org/w/index.php?title=Koeficient_determinace&oldid=20878613
- [19] „Correlation". <https://condor.depaul.edu/sjost/it223/documents/correlation.htm> (viděno 9. srpen 2022).
- [20] I. Bukovsky a O. Budík, „Znalostní a datové inženýrství". Přednášky ČVUT, 2021.
- [21] J. Brownlee, „4 Types of Classification Tasks in Machine Learning", *Machine Learning Mastery*, 7. duben 2020. <https://machinelearningmastery.com/types-of-classification-in-machine-learning/> (viděno 27. červenec 2022).
- [22] T. Hastie, R. Tibshirani, a J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, 2nd edition. New York, NY: Springer, 2016.
- [23] „Logistická regrese", *Wikipedie*. 25. srpen 2021. Viděno: 26. červenec 2022. [Online]. Dostupné z:
https://cs.wikipedia.org/w/index.php?title=Logistická_regrese&oldid=20416696
- [24] *StatQuest: Logistic Regression*, (5. březen 2018). Viděno: 22. červenec 2022. [Online Video]. Dostupné z: <https://www.youtube.com/watch?v=yIYKR4sgzI8>
- [25] „The Difference Between ‚Probability‘ and ‚Odds‘". https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_confidence_intervals/BS704_Confidence_Intervals10.html (viděno 26. červenec 2022).
- [26] J. Brownlee, „One-vs-Rest and One-vs-One for Multi-Class Classification", *Machine Learning Mastery*, 12. duben 2020. <https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/> (viděno 26. červenec 2022).
- [27] J. Brownlee, „Failure of Classification Accuracy for Imbalanced Class Distributions", *Machine Learning Mastery*, 31. prosinec 2019. <https://machinelearningmastery.com/failure-of-accuracy-for-imbalanced-class-distributions/> (viděno 9. srpen 2022).
- [28] *Feature Selection In Machine Learning | Feature Selection Techniques With Examples | Simplilearn*, (25. listopad 2020). Viděno: 1. srpen 2022. [Online Video]. Dostupné z: https://www.youtube.com/watch?v=5bHpPQ6_OU4
- [29] *R-squared or coefficient of determination | Regression | Probability and Statistics | Khan Academy*, (5. listopad 2010). Viděno: 1. srpen 2022. [Online Video]. Dostupné z: <https://www.youtube.com/watch?v=Ing4ZgConCM>
- [30] „Skewness: Definition, Formula, Calculate by Hand, Excel", *Statistics How To*. <https://www.statisticshowto.com/probability-and-statistics/descriptive-statistics/skewness/> (viděno 9. srpen 2022).
- [31] Admin, „Examples of Kurtosis Calculation (Step by Step)", *All Things Statistics*, 8. leden 2022. <https://allthingsstatistics.com/descriptive-statistics/kurtosis-calculation-example/> (viděno 9. srpen 2022).
- [32] V. Pawlik, „Pokročilé řízení vibrační zkoušky", s. 79.
- [33] LDS, „V830 Vibration Test Systems", roč. Manual 805341, č. 28.
- [34] „Understanding the physics of electrodynamic shaker performance by g f lang and d snyder by Manel Montesinos - Issuu". <https://issuu.com/manelmontesinos/docs/understanding-the-physics-of-electr> (viděno 2. srpen 2022).
- [35] P. Novák, „Průmyslové řídicí systémy", *Vysoká Šk. Baňská*, s. 17, 2013.